МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

# РАДИОТЕХНИКА

## Всеукраинский межведомственный научно-технический сборник

### ТЕМАТИЧЕСКИЙ ВЫПУСК «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

Основан в 1965 г.

ВЫПУСК 126



Харківський національний
університет радіоелектроніки

2002

УДК 621.3

*Сборник включен в список специальных изданий ВАК Украины по физико-математическим и техническим наукам*

*Для научных работников, специалистов, преподавателей вузов, аспирантов*

Настоящий сборник является тематическим и посвящен решению ряда задач классификации, анализа и синтеза в области информационной безопасности различных информационных технологий, автоматизированных систем управления, компьютерных систем и сетей. В нем обсуждаются возникшие на современном этапе противоречия по обеспечению информационной безопасности, анализируется их сущность, определяются и предлагаются методы, алгоритмы и средства их разрешения. Ряд статей носит методологический характер, в них сделаны попытки определить направления развития этой динамичной области, анализа и синтеза систем и средств защиты информации.

В сборнике представлен, по мнению кафедры «Безопасности информационных технологий» ХНУРЭ и АО «Институт информационных технологий» (Харьков), ряд результатов практической направленности в области криптографической защиты информации. Прежде всего - результаты анализа и разработки рекомендаций по использованию существующей и перспективной методологической и нормативной базы. Другие связаны с симметричными и асимметричными (несимметричными) криптографическими преобразованиями, криптографическими протоколами, а также криптоанализом. Ряд задач и результатов носят системно-технический характер.

Ответственность за содержание статей несут авторы.



Ответственный за выпуск д-р техн. наук, проф. *И.Д.Горбенко*

Рекомендовано Ученым советом Харьковского национального университета радиоэлектроники, протокол № 40 от 25.03.2002.

**Адрес редакционной коллегии:** Украина, 61166 Харьков, просп. Ленина, 14,

Харьковский национальный университет радиоэлектроники (ХНУРЭ), тел. 40-93-97.



ЗБІРНИК НАУКОВИХ ПРАЦЬ
**РАДІОТЕХНІКА**
Випуск 126

СБОРНИК НАУЧНЫХ ТРУДОВ
**РАДИОТЕХНИКА**
Выпуск 126

Редактор Л. Денисова
Підп. до друку 16.05.2002. Формат 60x90/8.
Папір офсетний. Друк офсетний. Обл. вид. арк. 14,2.
Тираж 300 прим. Зам. №12. Ціна договірна.
Харківський національний університет радіоелектроніки (ХНУРЕ)
Україна, 61166 Харків, просп. Леніна, 14.
Оригінал-макет підготовлено і збірник надруковано у ПФ "Колегіум", тел. (0572) 28-53-74

## *Уважаемые читатели, коллеги!*

Тематический выпуск посвящен решению актуальных задач в области теории и практики информационной безопасности. Публикуемые статьи отражают, по мнению специалистов ХНУРЭ и АО «Институт информационных технологий» (Харьков), ряд противоречий в области информационной безопасности и направлены на их разрешение. Значительная часть статей посвящена решению актуальных задач криптографической защиты информации, а ряд статей – решению задач синтеза и анализа криптографических протоколов управления ключами, установления ключей, разделения секрета и аутентификации. Это объясняется тем, что в настоящее время реальная криптостойкость и криптоживучесть и, как следствие, качество решения задач обеспечения конфиденциальности, целостности, наблюдаемости (включая причастность) и доступности определяются в значительной мере протоколами-примитивами и прикладными криптографическими протоколами.

Несмотря на большое внимание со стороны технологически развитых государств к проблеме обеспечения информационной безопасности, значительные вложения финансовых и материально-технических средств, привлечение больших интеллектуальных сил, число возникающих противоречий и сложно разрешаемых проблем только возрастает. Это объясняется тем, что в начале XXI века возникли неуправляемые или слабо управляемые процессы информатизации, которые называют компьютеризацией. Примером тому являются системы Internet, Extranet, операционные системы Windows-2000, UNIX и др., с использованием которых, по сути, уже создано единое информационное пространство.

Реально также и то, что в этом информационном пространстве непрерывно ведутся информационные войны, создается, применяется и совершенствуется информационное оружие. Предварительные оценки и грубое моделирование этих неуправляемых процессов позволяет сделать выводы об их особой опасности в смысле национальной безопасности государств, отдельных содружеств и союзов и Земной цивилизации в целом. Разрешение названных противоречий и проблем может быть осуществлено за счет создания и применения в информационных технологиях, системах информационных технологий, автоматизированных системах управления, компьютерных системах и сетях комплексных систем обеспечения информационной безопасности. Определяющими составляющими комплексных систем являются

криптографические системы и [redacted] протоколы, а также криптоаналитические системы [redacted] исследований их криптографической стойкости

В сборнике представлены результаты исследований и разработок специалистов Украины в области информ[redacted]асности. Это издание позволит, на наш взгляд, довести до спе[redacted]стов и интересующихся проблемами криптографической защиты информации ряд интересных результатов, обсудить состояние вопроса и противоречия, продолжить уже начавшиеся в Украине дискуссии.

Безусловно, статьи отражают прежде всего мнения и взгляды авторов. В то же время все статьи прошли обсуждение на кафедре «Безопасность информационных технологий» ХНУРЭ и рекомендованы к опубликованию Ученым советом университета. Считаем, что представленные материалы могут быть использованы также в учебном процессе, прежде всего по специальностям направления «Информационная безопасность».

***С уважением и благодарностью к специалистам, коллегам и читателям, которые остаются неравнодушными к проблемам информационной безопасности***

***М. Ф. Бондаренко,***
***ректор ХНУРЭ, профессор***

***И. Д. Горбенко,***
***зав. кафедрой БИТ, профессор***

УДК 681.322

*М.Ф.БОНДАРЕНКО., д-р техн. наук, И.Д. ГОРБЕНКО, д-р техн. наук, С.П. ЧЕРНЫХ, А.В. ПОТИЙ, канд. техн. наук*

# ИНФРАСТРУКТУРА ОТКРЫТЫХ КЛЮЧЕЙ КАК ОСНОВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ НАЦИОНАЛЬНЫХ, ВЕДОМСТВЕННЫХ И КОММЕРЧЕСКИХ СИСТЕМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Введение.**

В настоящее время на международном уровне создана нормативная база и сформировались устойчивые взгляды на решения проблем обеспечения информационной безопаности различных систем информационных технологий (ИТ-систем). ИТ-система представляет собой организационно-техническую систему, которая включает в себя:

– совокупность технических средств передачи и обработки информации (ИТ-продуктов), объединенных в функционально полный комплекс;

– совокупность методов и алгоритмов обработки информации в виде соответствующего программного и математического обеспечения;

– информационные и иные ресурсы;

– персонал и пользователей, объединенных по организационно-структурному, тематическому, технологическому, функциональному и другим принципам для осуществления автоматизированной обработки информации.

Подробно взгляды на обеспечение инофрмационной безопасности в национальных системах нами рассмотрены в работах [8-10]. Общим выводом является то, что безопасность систем информационных технологий (ИТ-систем) достигается путем решения взаимосвязанной совокупности задач защиты. Основными задачами защиты является обеспечение конфиденциальности, доступности, целостности и наблюдаемости. Для решения этих задач в рамках ИТ-системы создается комплексная система обеспечения безопасности, которая объединяет административные, технические и криптографические средства защиты (аппаратные, программные, программно-аппаратные), алгоритмическое, математическое, программное, информационное и иное обеспечение и персонал, отвечающий за реализацию в ИТ-системе политики безопасности. В современных ИТ-системой основой решения перечисленных выше задач являются методы криптографии.

В данной статье авторы рассматривают возможность создания в Украине полноценной инфраструктуры открытых ключей, целью наиболее полного удовлетворения потребностей собственников ИТ-систем в обеспечении информационной безопасности и предоставления полного спектра услуг безопасности.

## 1. Инфраструктура безопасности

В функциональном плане задачи защиты решаются путем реализации функций (услуг) безопасности. На рисунке 1 представлена модель обеспечения безопасности информационных технологий, которая рекомендована Национальным институтом стандартизации и технологий США (NIST) в качестве базовой технической модели [7,9].

Данная модель существенно расширяет существующие модели безопасности и соответсвует концепции обеспечения безопасности ИТ-систем, заложенной в таких документах как ISO/IEC 15408 «Общие критерии оценки ИТ-безопасности», ISO/IEC 17779 «Управление безопасностью», CEM 97/017 и CEM99/045 «Методология оценки ИТ-безопасности».

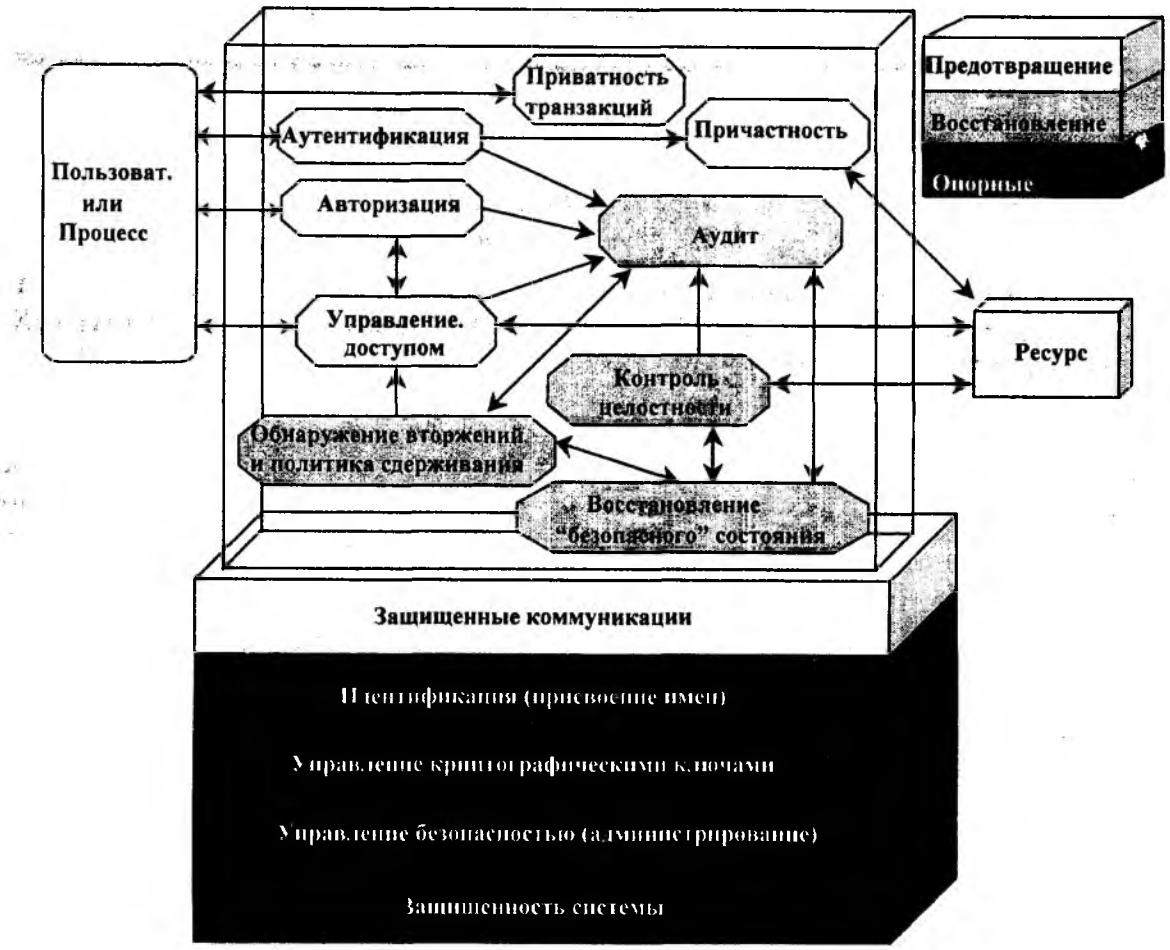



Рис.1

Базовая модель определяет три класса услуг безопасности:

- услуги предотвращения нарушений безопасности;
- услуги обнаружения и восстановления безопасности;
- опорные услуги безопасности.

Рассматриваемые услуги безопасности позволяют решить широкий спектр различных задач обеспечения безопасности, которые присутствуют в любой ИТ-системе. К таким задачам относятся идентификация пользователей, обеспечения конфиденциальности сообщений, управление доступом к различным документам, соблюдение конфиденциальности личной информации (приватности), обеспечение причастности и многие другие. Взаимодействующие объекты и субъекты при установке контакта, при обмене сообщениями (как в on-line, так и в off-line режимах) должны быть твердо «уверены» в личности абонента или подлинности ресурса.

Сейчас можно с уверенность сказать, что основным технологическим инструментом решения этих и других задач в ИТ-системах являются криптографические технологии и особенно методы открытой криптографии. Задачи обеспечения целостности, конфиденциальности, аутентификации, доступности, причастности решаются путем реализации механизмов шифрования, генерации кодов аутентификации сообщений, цифровой подписи, хеширования в основании которых лежит использование одного или нескольких криптографических примитивов [8]. Для решения задач защиты пользователи должны использовать несколько классов криптографических механизмов безопасности. Корпоративные сети отличаются распределенностью (как временной, так и пространственной) информационных и иных ресурсов.

Поэтому одной из важных задач является распределение криптографических ключей и иной служебной информации, необходимой для эффективного управления безопасностью. Основой надежной и эффективной работы криптографических механизмов является управление ключами, которое является опорной услугой безопасности. В международной и национальной практике стандартизации модели, технологии и методы управления ключами рассматриваются в отдельных нормативных документах, что только подчеркивает базисную роль управления ключами.

Согласно международному стандарту ISO/IEC 11770 под *управлением ключами* понимают совокупность методов и процедур, используемых для установления и управления ключевыми взаимоотношениями между авторизованными объектами.

Стандарт определяет одиннадцать функций управления ключами: генерация, регистрация, сертификация, распределение, инсталляция, хранение, формирование производной или разворачивание, архивирование, отмена (аннулирование), дерегистрация и уничтожение ключа. Жизненный цикл управления ключами в общем виде представлен на рис.2. [11]

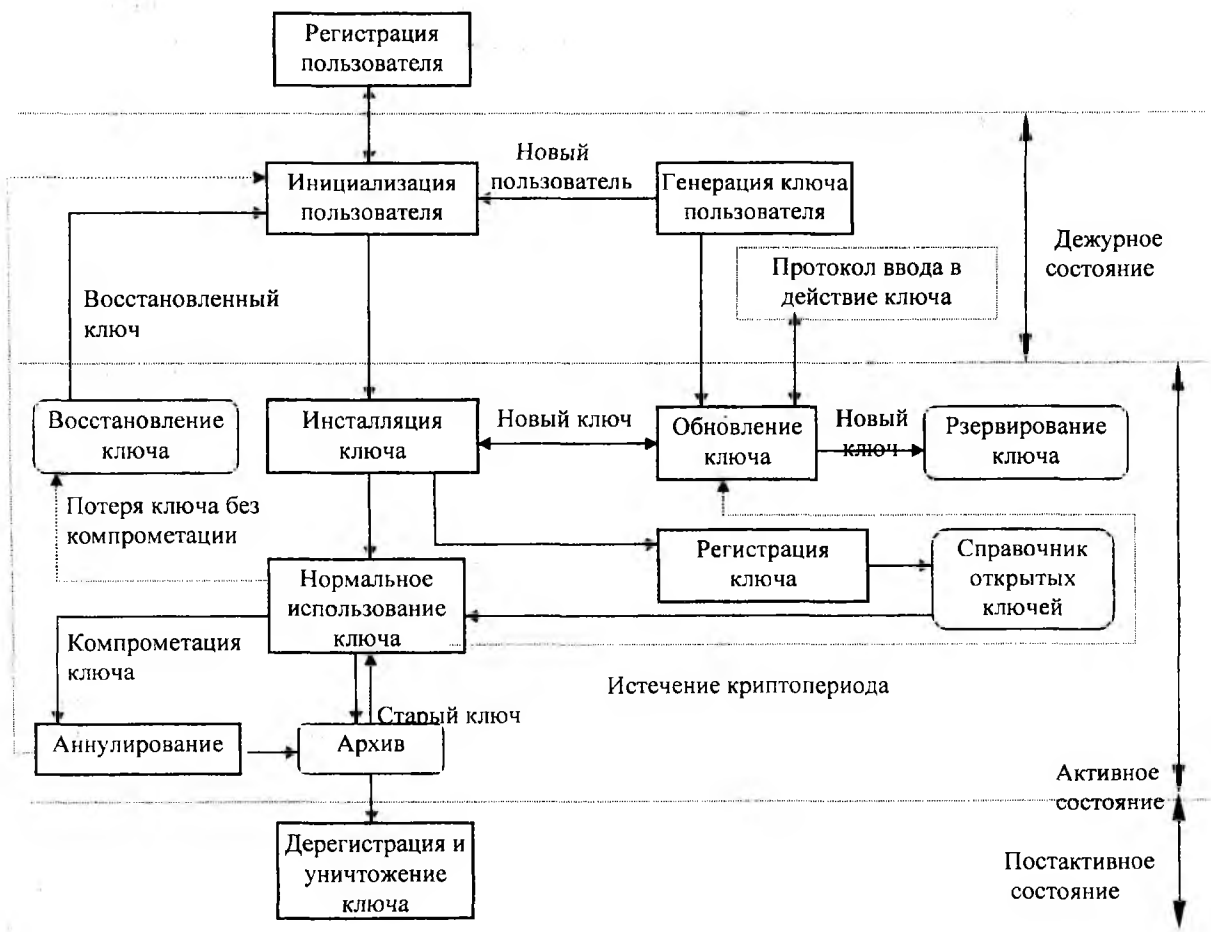



Рис. 2

Для достижения необходимого уровня конфиденциальности, нужно распределить между абонентами симметричные ключи шифрования. Данная задача может быть решена тремя способами:

1) непосредственный обмен ключами между сторонами с использованием механизмов симметричного шифрования;
2) обмен ключами с использование механизмов симметричного шифрования и привлечением доверительной третьей стороны (например, центр распределения ключей или центр передачи ключей);

3) обмен ключами с использование методов несимметричной криптографии с привлечением доверительной третьей стороны (ДТС).

Первый способ эффективен только в малых закрытых системах.

Второй способ может быть распространен и на большие системы. Однако использование механизмов симметричной криптографии не позволяет реализовать услугу причастности при ключевых взаимоотношениях. В связи с этим данный метод нашел широкое распространение лишь в военных системах, и неприемлем в ИТ-системах, которые принадлежат другим ведомствам (здравоохранение, почта, связь, банковская система и т.п.).

Наиболее приемлемым на сегодня способом распределения ключей в ИТ-системах различной принадлежности (государственные и коммерческие) является третий способ, основа которого есть создание доверительной третьей стороной цифрового сертификата открытого ключа. Если ДТС свяжет открытый ключ с пользователем или системой, то есть проверит подлинность стороны владеющей личным ключом, то можно осуществить полный спектр услуг безопасности. Пользователи получают в своё распоряжение услуги целостности, аутентификации и причастности через реализацию механизмов цифровой подписи. Симметричные ключи могут быть распределены путем использования либо транспортных протоколов (протоколов передачи ключей), либо протоколов согласования или установления ключей. А это, в свою очередь, позволит обеспечить необходимый уровень конфиденциальности.

**Сертификат ключа** – это цифровой документ, подписанный органом или администратором сертификации (Certificate Authority, AC) и подтверждающий однозначное соответствие между открытым ключом и идентификационной информацией пользователя-владельца ключа.

Однако одного сертификата недостаточно для решения всех задач защиты. Для обеспечения безопасного взаимодействия ИТ-систем, принадлежащих различным ведомствам, организациям и предприятиям необходимо создать сеть взаимодействующих администраторов сертификации. Такая сеть реально взаимодействующих АС формирует инфраструктуру безопасности ведомственного, регионального и, наконец, национального масштаба, в рамках которой пользователи могут получить качественные услуги безопасности. Инфраструктура безопасности, которая обеспечивает распределение сертификатов открытых ключей, поддерживает защищенный обмен сообщениями, надежную идентификацию и аутентификацию, электронную коммерцию получила название **инфраструктуры открытых ключей (Public Key Infrastructure, PKI).**

## 2. Инфраструктура открытых ключей

В настоящее время в отечественной практике нет какого-либо законодательно или нормативно закрепленного определения понятия инфраструктуры открытых ключей. Нами предлагается определение, которое является синтезированным из ряда документов Национального института стандартизации и технологий США (NIST) [3-6].

*Инфраструктура открытых ключей* представляет собой комплексную организационно-техническую систему, которая обеспечивает необходимые услуги для использования криптографических технологий с открытыми ключами. Основная цель PKI – обеспечение, путем применения цифровых сертификатов, надежной связи (ассоциирования) открытых ключей с объектами, что позволяет другим объектам проверить эту связь и получить необходимые услуги для осуществления управления ключами в распределенных системах. PKI интегрирует цифровые сертификаты, криптографию с открытыми ключами и органы сертификации в единую архитектуру безопасности корпоративной сети. Обычно инфраструктура открытых ключей создается с целью решения следующих задач:

- осуществление выпуска в обращение сертификатов для отдельных пользователей;
- регистрации программного обеспечения оконечных пользователей;
- объединения сертификатов в специальные справочники сертификатов и обеспечение их обслуживания;

– реализации механизмов управления, обновления, восстановления и аннулирования сертификатов;
– предоставления дополнительных услуг поддержки управления сертификатами.

Международный стандарт ISO/IEC 11770 определяет основные компоненты PKI. На рисунке 3 представлена модель организации сертификации, которая закреплена в стандарте. Данная модель сертификации характеризует взаимосвязь между основными логическими объектами, которые принимают участие в формировании и управлении сертификатами.


Рис.3

*Орган сертификации* или *администратор сертификации* (АС) это доверенный объект, который несет ответственность за организацию и осуществление процесса сертификации открытых ключей пользователей, а также ручается за подлинность открытых ключей. Это включает выполнение таких действий как назначение открытым ключам уникальных имен посредством подписанных сертификатов, управление назначением порядковых номеров сертификатам, аннулирование сертификатов и других действий. Орган сертификации является основным компонентом PKI. Он включает в себя аппаратное и программное обеспечение, иное оборудование, персонал и как минимум обладает двумя атрибутами: собственным именем и собственной ключевой парой. АС выполняет четыре основные функции PKI [3]:

– выпуск сертификатов (т.е. создает и подписывает их);
– поддержку информации о статусе сертификатов (информация состояния сертификатов) и ведет список аннулированных сертификатов (CRL);
– публикацию текущих (имеющие силу ) сертификатов и последней версии CRL, для того, чтобы пользователи могли получить самую свежую информацию о состоянии сертификатов и возможности использования услуг безопасности;
– поддержку архивов информации состояния сертификатов, у которых истек срок действия.

Орган сертификации выпускает для каждого объекта *цифровой сертификат* или *сертификат открытого ключа* (public key certificate). Это список данных связанных с конкретным пользователем, включающий открытый ключ (или ключи) этого пользователя подписанный органом сертификации. Таким образом, сертификат состоит из двух полей – поля данных и поля подписи. Поле данных содержит, как минимум, открытый ключ пользователя и иден-

тификационную информацию пользователя (например, идентификатор пользователя). Поле подписи содержит подпись органа сертификации, которая является поручительством за аутентичность открытого ключа пользователя. Сертификат также может содержать и другую дополнительную информацию, например, указатели, каким образом может быть использован тот или иной ключ. В настоящее время формат сертификата определяется стандартом ITU-T X.509. Каждый пользователь должен быть приписан к конкретному органу сертификации, и обладать доверенной копией ключа проверки подписи этого органа. Орган сертификации может выпускать сертификаты как для отдельных пользователей, так и для других органов сертификации. АС вставляет свое имя в каждый сертификат (и CRL), которые он формирует, и подписывает их на своем личном ключе. Таким образом, обеспечивается доверие пользователей к АС (непосредственно, либо через сертификационный путь). Для обеспечения надежной работы АС должен использовать сертифицированные криптографические модули.

*Сервер имен* (name server) несет ответственность за управление пространством имен пользователей, с целью обеспечения каждого пользователя уникальным неповторяющимся именем.

*Справочник сертификатов* или *депозитарий* ( certificate directory or repository) это база активных цифровых сертификатов (т.е. действующих), которая обеспечивает поддержку сертификатов в on-line режиме, т.е. в состоянии полной готовности к использованию их пользователями. Обычно депозитарием является база данных или сервер, доступные для пользователей в режиме только для чтения. Пополнение и поддержку справочника осуществляет орган сертификации. Пользователи также могут иметь свои справочники сертификатов. В этом случае за их поддержку отвечает сам пользователь. Все приложения PKI достаточно сильно зависят от эффективности реализации услуги справочника сертификатов, поскольку именно через депозитарий осуществляется распределение всей информации о состоянии сертификатов и собственно самих сертификатов. Услуга справочника реализуется в соответствии с требованиями стандарта ITU-T X.500.

*Центр генерации ключей* осуществляет генерацию пар открытый/закрытый ключ, а также генерацию симметричных ключей и паролей. Центр генерации ключей может быть частью оборудования пользователя, если пользователь самостоятельно генерируют себе ключи, либо частью органа сертификации, либо вообще может являться отдельной доверительной системой

*Орган регистрации* или *администратор регистрации* (registration authority) несет ответственность за авторизацию объектов, отличающихся уникальными именами, в качестве члена домена безопасности или доверительного домена. Основная задача органа регистрации – проверка содержания информации в сертификате. Сертификат может содержать информацию предоставленную объектами, регистрирующих сертификаты, например номер лицензий или сведения о последнем платеже. Сертификат может содержать информацию третьей доверительной стороны и другую информацию. Чаще всего орган регистрации представляет собой сервер с соответствующим программным обеспечением, который управляется одним администратором. Орган сертификации может взаимодействовать с несколькими аккредитованными органами сертификации.

Орган сертификации создает и поддерживает *список аннулированных сертификатов* (Certificate Revocation List (CRL)). CRL является списком порядковых номеров или других идентификаторов сертификатов, которые были аннулированы определенным органом сертификации и служит для оповещения в on-line режиме об аннулировании сертификатов. На сегодняшний день CRL является самым распространенным способом оповещения пользователей об аннулировании сертификатов

*Архив* представляет собой базу данных, содержащую информацию о сертификатах, у которых истек срок действия. Архив используется для долговременного хранения информации состояния сертификатов. Используя архив, можно убедиться, что конкретные выведенные из действия сертификаты, действительно были выпущены данным органом сертификации. При

возникновении споров относительно старых документов, которые были подписаны на выведенных из действия ключах, также будет полезна архивная информации, предоставленная органом сертификации.

*Пользователями PKI* могут быть как организации, так и отдельные пользователи (физические лица). Эта категория не имеет права выпуска сертификатов. Они доверят другим компонентам PKI. На основе этого доверия они получают и осуществляют проверку сертификатов других пользователей. Различают две категории пользователей PKI. К первой категории относятся *пользователи сертификата* (пользователи, доверяющие сертификату) – это пользователи, которые используют сертификат, с целью определения принадлежности открытого ключа другому объекту. Другой категорией являются *держатели сертификатов* – пользователи, которые могут подписывать документы и совершать иные действия с помощью сертификатов.

В зависимости от масштабов организации, степени зависимости функционирования организации от информационных технологий, основных функций организации состав компонентов PKI и их функции могут варьироваться. Совсем не обязательно, что в конкретной реализации эти объекты будут раздельными. В отдельных случаях некоторые из них вообще могут не существовать.

### 3. Услуги PKI

Основными услугами PKI, являются услуги управления сертификатами. Услуги управления сертификатами – это услуги, образующие ядро инфраструктуры с открытыми ключами. К основним услугам относятся следующие услуги [1]:

- **выпуск сертификата** для пользователей и администраторов сертификации;
- **аннулирование сертификата** в случае компрометации секретных ключей пользователя или в других ситуация, определенных в политике сертификации;
- **приостановление действия сертификата** с последующим автоматическим восстановлением или аннулированием сертификата;
- **публикация сертификатов** через каталог сертификатов ( в соответствии с требованиями X.500 или иных нормативных документов и стандартов), с целью обеспечения доступа к спискам действующих сертификатов заинтересованных лиц;
- **хранение сертификата** с возможностью восстановления сертификата;
- **архивирование сертификатов вышедших из употребления** с целью обеспечения возможности проверки электронных документов, сделок и других операций, выполненных с использованием данных сертификатов.

Кроме основных услуг, в PKI могут поддерживаться и дополнительные услуги:

- **регистрация.** Услуги регистрации обеспечивают регистрацию и контроль индивидуальной информации объектов процессов сертификации;
- **хранение информации в архиве.** И Услуга предназначена для долговременного хранения и управления цифровыми документами и другой информации;
- **нотариальная сертификация,** которая включает аутентификацию отправителя, подтверждения целостности и юридической силы цифровых документов;
- **создание резервных копий и восстановление ключей;**
- **поддержка каталога.** Данная услуга обеспечивает всесторонне управление и обеспечение информацией относительно пользователей или атрибутивной информацией;
- **поддержка услуг причастности;**
- **услуги корректировки и управления историей ключей.**

Реализация всех услуг и взаимодействие объектов процесса сертификации осуществляется в рамках единой политики сертификации. Под **политикой сертификации** понимают совокупность правил, отражающих и регулирующих порядок применения сертификатов в конкретной совокупности приложений с общими требованиями по обеспечения информационной безопасности. Политика сертификации является составной частью политики безопас-

ности и применяется пользователями сертификата с целью принятия решения о достаточной степени доверия к сертификату для его применения [2, 12].

### 4. Архитектура PKI

Держатели сертификатов, в зависимости от принадлежности к организации или сообществу, могут получать сертификаты от различных органов (администраторов) сертификации. Поэтому PKI обычно является совокупностью нескольких администраторов сертификации, связанных доверенными путями.

В организации (например, в банке) разворачивается своя PKI. В настоящее время выделяется два основных типа архитектуры PKI – **иерархическая** и **сетевая.** С целью обеспечения возможности взаимодействия пользователей, которые принадлежат различным ведомствам, ведомственные PKI объединяются через **узловые органы сертификации.**

Остановимся на данных типах архитектур более подробно.

В PKI, построенной на принципах иерархии все администраторы объединяются по принципу иерархического соподчинения (рис. 4). Корневой (центральный, главный) АС выпускает сертификаты для подчиненных АС, а те в свою очередь для АС следующего уровня иерархии или своим пользователям. В иерархических PKI любые связывающиеся стороны знают открытый ключ главного администратора. Любой сертификат может быть верифицирован через проверку сертификационного пути от главного АС. Обычно такая инфраструктура строится в ведомственных корпоративных системах, в которых существует необходимость и возможность полностью контролировать размер и конфигурацию сети.

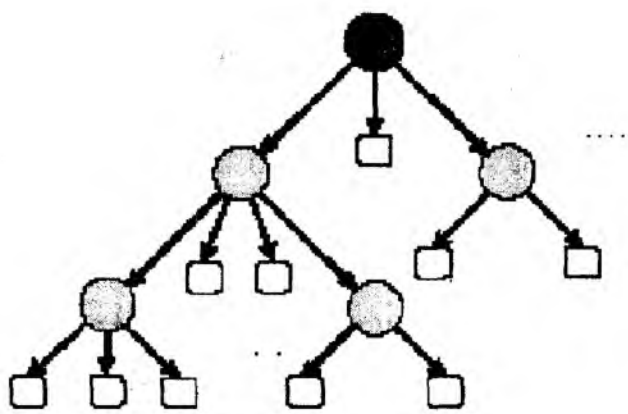


Рис. 4

В сетевой архитектуре PKI (рис.5) все администраторы являются равными или одноранговыми, т.е. не размещаются на различных уровнях иерархии. В сети доверенные отношения между равными АС поддерживаются независимой взаимной кросс-сертификацией администраторов.

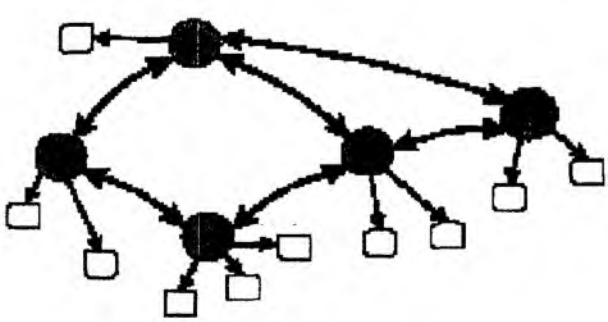


Рис. 5

Связывающиеся стороны знают открытый ключ «ближайшего» администратора, который и выпускает для них сертификаты. Верификация сертификатов происходит путем верификации сертификационного пути сертификатов, который проходит от данного доверенного администратора. Под *сертификационным* путем понимают упорядоченную последовательность сертификатов, которые вместе с открытым ключом исходного объекта в пути, могут быть обработаны для получения оконечным объектом пути [12]. Администраторы сертификации выпускают кросс-сертификаты, т.е. выпускают сертификаты друг для друга, а затем объединяют их в *пару кросс-сертификатов*. Данная архитектра применяется в открытых сетях, с неконтролируемым подсоединением (удалением) абонентов. Особое распространение она получила в системах электронной коммерции через Internet.

Архитектура узловых администраторов была разработана для объединения ведомственных PKI. Архитектуры объединяются путем введения нового администратора, который называется узловым. Узловой АС не выпускает сертификаты для пользователей PKI. Все пользователи, независимо от архитектуры PKI, рассматривют узловой АС промежуточным узлом. Узловой АС обеспечивает равноправные отношения между ведомственными PKI. При соединении иерархических архитектур, узловой АС обеспечивает взаимодействие главных АС, при соединении сетевых – с одним из АС каждой сети. При соединении различных архитектур узловой АС устанавливает отношение главного АС иерархической PKI с одним из АС сетевой PKI.

## 5. Архитектура Национальной PKI

В настоящее время можно говорить о реальной интеграции ведомственных, коммерческих и общенациональных информационных сетей. Пользователи имеют потребности в использовании ресурсов сетей различной принадлежности для решений своих задач. Требуемый уровень безопасности взаимодействия сетей при сохранении требуемого уровня доступности ресурсов сегодня может быть достигнут путем построения Национальной архитектуры PKI. Но для этого в рамках государства необходимо на законодательном уровне установить правовые взаимоотношения между государственными и коммерческими органами сертификации. Данная проблема усложняется еще и тем, что в Украине на данные момент получили лишь PKI, принадлежащие кредитно-финансовым учреждениям (банковские платежные системы) и системам электронной коммерции. В других государственных учреждениях данные технологии пока еще к сожалению не нашли широкого применения.

Основной проблемой Национальной PKI является создание сертификационных путей между различными ведомствами и организациями, которые будут обеспечивать высокий уровень доверительности. Национальная PKI должна объединять государственный и негосударственный сектора инфораструктуры. Одним из путей решения этой проблемы применение администраторами сертификации проверенных механизмов, специфицированных в стандартах и удовлетворение требования, предъявляемых к кросс-сертификатам.

На рисунке 6 представлена Национальная архитектура PKI, рекомендуемая NIST [3,6].

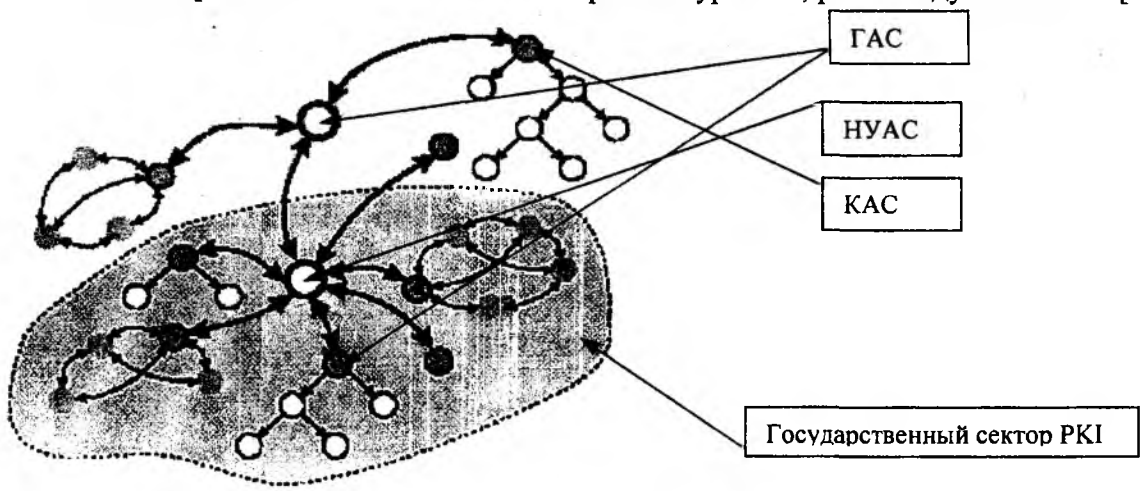



Рис. 6

Основными архитектурными компонентами данной архитектуры являются:

1. **Орган управления Национальной политикой сертификации**. Данный орган устанавливает общую политику в Национальной PKI, и утверждает правила и процедуры доверительных доменов в Национальной PKI. Данный орган осуществляет оправление Национальным узловым органом сертификации и Национальным депозитарием.
2. **Доверительные домены.** В национальном масштабе, доверительный домен являются частью Национальной PKI, который функционирует под управление единой органа управления политики сертификации. В домене может существовать один или более администраторов сертификации. Каждый доверительный домен имеет одного главного администратора сертификации (principal CA) и доменный депозитарий.
3. **Орган управления политикой сертификации домена** утверждает практические действия администраторов сертификации домена и осуществляет наблюдение (отслеживает) эти действия. Орган организует работу или осуществляет надзор за депозитарием домена.
4. **Администраторы сертификации:**
   - Узловой администратор сертификации. Это Национальный узловой администратор сертификации (НУАС), функционирующий под управленеим органа управления Национальной политики сертификации. Цель НУАС – обеспечить узел доверия, посредством которого будут построены доверительные пути между различными доверительными доменами Национальной PKI, а также между ведомственными (государственными) и коммерческими PKI. Орган управления Национальной политикой сертификации утверждает главных администраторов доверительных доменов, которые имеют право выпускать кросс-сертификаты с Национальным узловым АС. Отметим, что НУАС не является корневым АС, поскольку он не является началом сертификационных путей.
   - Главный АС (ГАС) это АС, который внутри доверительного домена отвечает за выпуск кросс-сертификатов с НУАС. Любой доверительный домен может иметь одного ГАСа. В домене с иерархической архитектурой ГАСом является корневой АС. В домене сетевой архитектуры – любой из АС, обычно назначаемый НУАСом.
   - Одноранговый АС (ОАС) – администратор в доверительном домене сетевой архитектуры. ОАС обладает собственным сертификатом, который распределяется среди держателей сертификатов и используется ими для инициализации сертификаци-

онных путей. ОАС также выпускает кросс-сертификаты с другими ОАС своего доверительного домена.

➢ Корневой (центральный) АС (КАС) – администратор, который в доверительном домене иерархической архитектуры является началом всех сертификационных путей. Держатели сертификатов и связывающиеся стороны получают сертификат КАС каким-либо надежным способом (например, при личной встрече уполномоченных лиц) и все доверительные пути начинаются с этой точки. Для иерархических архитектур КАС одновременно является и ГАСом данного домена.

➢ Подчиненные АС (ПАС) – администраторы в домене иерархической архитектуры, не являющиеся начальными точками доверительных путей. ПАС получают сертификаты от своих АС, находящихся на высшем уровне иерархии, и в свою очередь выпускают сертификаты для своих подчиненных АС.

5. **Депозитарии** являются on-line средствами, которые поддерживают в актуальном состоянии базу сертификатов и информацию о статусе сертификатов. Депозитарии в Национальной PKI предоставляют информацию посредством протокола LDAP (Lightweight Directory Access Protocol), а также другими средствами. За поддержку депозитария и CRL сертификатов АС отвечает Орган управления Национальной политики сертификации.

6. **Депозитарий НУАС.** Общепринятым решением является открытый доступ к депозитарию НУАС через Internet. Депозитарий содержит следующую информацию:

- все сертификаты, выпущенные НУАСом;
- все сертификаты, удерживаемые НУАСом;
- все пары кросс-сертификатов, содержащие удерживаемый и выпущенные НАУСом сертификаты;
- текущую версию CRL для всех сертификатов, выпущенных НУАСом;
- большинство или все сертификаты, выпущенные администраторами Национальной PKI, с целью поддержки в поиске сертификационных путей;
- большинства или все пары кросс-сертификатов между администраторами сертификации Национальной PKI;
- другие сертификаты и CRL, определенные органом управления Национальной политикой сертификации.

Национальный узловой администратор сертификации является объединяющим элементом для объединения на общих методологических принципах ведомственный администраторов сертификации в единую Национальную PKI. НУАС, как уже отмечалось выше, не является корневым АС, однако он играет важную ***системную*** роль. Он соединяет доверительные домены посредством пар кросс-сертификатов уполномоченных главных администраторов сертификации и является *узлом доверия.* Необходимо отметить, что в теории распределения ключей *модель доверия* и *доказательство доверия*, являются весьма важными и принципиальными вопросами. В Украине в настоящее время теоретические вопросы моделирования доверия и доверительных отношений в гетерогенных сетях пока изучены недостаточно. Орган управления Национальной политики сертификации осуществляет надзор за деятельностью НАУС и определяет требования для осуществления процессов кросс-сертификации с НАУСом. Доверительные домены, осуществляющие кросс-сертификацию с НАУСом могут находится как в государственной, так и в коммерческой области.

Государственные и негосударственные АС, функционирующие в доверительных доменах, обязаны удовлетворять требованиям, определенным Органом управления Национальной политики сертификации. С целью определения возможности осуществления деятельности субъектов хозяйственной деятельности в области распределения сертификатами, необходимо разработать и законодательно закрепить лицензионные осуществления соответствующей деятельности и процедуры аттестации и аккредитации органов сертификации. К процессам кросс-сертификации могут быть допущены только аккредитованные АС. Для таких админи-

страторов НУАС обеспечит надежное соединение с общей доверенной сетью Национальной PKI.

Однако для обеспечения реально гибкости в ведомственных взаимоотношениях, необходимо избегать монополии НУАС в определении политики сертификации. Для это НАУС должен быть ограничен относительно вмешательства в деятельность ведомственных АС, а именно:

- ведомства не могут быть ограничены условиями полной адаптации к политике сертификации НАУС. Более, того необходимо создать такие условия на рынке данных услуг, при которых ведомства могут использовать другие правила сертификации, определенные либо их собственными органами управления политикой сертификации, либо коммерческими провайдерами услуг сертификации;
- ведомства не могут быть ограничены в праве взаимодействия с другими ведомственными и коммерческими организациями исключительно через НАУС. Как альтернатива, необходимо предоставить возможность непосредственного взаимодействия организация на договорных основах.

Эти и другие вопросы, которые несомненной появятся при более детальном изучении проблем построения Национально PKI, должный учитываться при разработке соответствующих нормативно-правовых документов всех уровней государственного управления.

**Заключение**

Для всесторонней проработки вопросов построения Национальной PKI, видится целесообразным создать под эгидой Департамента специальных телекоммуникационных систем и защиты информации, Госстандарта Украины, Министерства образования и науки рабочий комитет, который возглавит разработку и создание Национально PKI. В частности рабочие группы комитета могут решать следующие задачи:

- обеспечение руководства и оказание методической, теоретической поддержки в разработке PKI, которая опирается на использование коммерческих продуктов, удовлетворяющих требованиям национальных и международных стандартов;
- определение и нормативно закрепление требования ведомственным PKI;
- выработка рекомендаций по определению политики, процедур и выбору стандартов, которые направлены на поддержку PKI;
- осуществление надзора за деятельностью объектов PKI, в пилотных проектах по реализации Национальной PKI;
- осуществление руководства и надзора за технологиями управления ключами;
- определение технологии, необходимые для эффективной реализации Национальной PKI;
- поддержка связи с заинтересованными государственными, промышленными, научными и общественными организациями;
- разработка требования по взаимодействию, совместимости и безопасности продуктов и протоколв, связанных с реализацитей Национальной PKI;
- выработка рекомендация относительно создания, демонстрации и функционирования Национальной PKI и др.

На наш взгляд проведение работ по созданию Национально PKI в Украине являются крайне актуальными и весьма своевременными, особенной учитывая тот факт, что в Верховной раде уже лежит проект Закона о цифровой подписи. Его реальная работа может быть построена только при условии создания в Украине Национальной инфраструктуры открытых ключей.

**Список литературы:** 1. *В. Горбатов, О. Полянская.* Доверенные центры как звено обеспечения безопасности корпоративных ресурсов. // JetInfo №11 (78). – 1999. – С. 13-20. 2. *ISO/IEC 9549-8:1993| ITU-T Recommendation X.509.* Information Technology – OSI – The Directory: Authentication Framework. 3. *D.R. Kuhn, V.C. Hu, W.T. Polk, S.J. Chang.* Introduction to Public Key Technology and the Federal PKI Infrastructure. NIST SP 800-32. – 2000. 4. *ISO/IEC 11770:1996.* Information Technology – Security techniques – Key management. 5. *G. Stoneburner.* Underlying Technical Models for Information Technology Security. NIST SP 800-44 – 2001. 6. *Anabelle Lee.* Guideline for Implementing Cryptography in the Federal Government. NIST SP 800-21 – 1999. 7. *ISO 7498-2:1989* – Open System Interconnection Reference Model – Security Architecture. 8. *Потий А.В.* Криптография в защите информации. // Служба безопасности. - №4-5. – 2001. – С.7-9. 9. *Горбенко И.Д., Бондаренко М.Ф., Скрипник Л.В., Потий А.В.* Перспективы применения международного стандарта ISO/IEC 15408 в Украине. В сб. Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – №3. – 2001. 10. *Бондаренко М.Ф., Черных С.П., Горбенко И.Д., Замула А.А., Ткач А.А.* Методологические основы концепции и политики безопасности информационных технологий. // Радиотехника: Всеукр. Межвед. Научн.-техн. сб. 2001. Вып.19. – С.5-16. 11. *A. Menezes, P. van Oorschot, S. Vanstone.* Handbook of Applied Cryptography. – CRC Prerss, Inc. – 1997. 12. *S.Chokhani, W. Ford.* Internet X.509 Public Key Infrastructure Certificate Policy and Certifications Framework. – RFC 2527, 1999.

*Харьковский национальный университет радиоэлектроники. Служба безопасности Украины.*

*Поступила в редколлегию 09.04.2002.*

УДК 681.3.06:519.248.681

*Л. В. СКРИПНИК, , д-р техн. наук, М. Ф. БОНДАРЕНКО, д-р техн. наук,*
*И. Д. ГОРБЕНКО, д-р техн. наук, , А. А. ТКАЧ, А.В. ПОТИЙ, канд.техн. наук*

# МЕТОДОЛОГИЧЕСКИЕ АСПЕКТЫ ГЕРМАНСКОГО СТАНДАРТА «РУКОВОДСТВО ПО БАЗОВОЙ ЗАЩИТЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

## Введение

Одной из актуальнейших задач обеспечения безопасности информационных систем различного класса и назначения является формирование отечественной нормативной и методологической базы. В настоящее время в Украине идет разработка отечественной нормативной базы по защите информации в различных информационных технологиях (ИТ), автоматизированных системах управления (АСУ), компьютерных системах и сетях (КСС) и др. Обзору существующих современных международных стандартов была посвящена статья [1]. Определенным прорывом в области проектирования, изготовления, оценки (сертификации) и эксплуатации защищенных информационных технологий стало принятие международного стандарта ISO/IEC 15408 «Общие критерии оценки безопасности информационных технологий» [2]. Ряд государств мира принимают или рассматривают его с целью принятия в качестве национального методом «обложки». Среди технически развитых государств, которые разработали национальные стандарты обеспечения информационной безопасности, необходимо, прежде всего, выделить Германию, в частности «Руководство по базовой защите информационных технологий» [3], принятое в качестве стандарта Германии.

Цель настоящей статьи является рассмотрение основных методологических положений германского стандарта «Руководство по базовой защите информационных технологий» (далее – Руководство), который разработан Германским информационным агентством безопасности (BSI).

Центральная идея разработчиков Руководства заключается в оказании помощи пользователям и владельцам информационных технологий в оперативном решении общих задач информационной безопасности, повышении уровня защиты ИТ и ИТ-систем , а также упрощения процессов создания политики и концепции безопасности ИТ. Это достигается посредством предоставления определенных наборов стандартных защитных мероприятий безопасности, рекомендаций и средств по реализации для типичных систем ИТ.

Под понятием «базовой защиты информационной технологии» в данном документе понимается некоторый обоснованный в пределах разумного уровень защиты для ИТ, адекватный нормальным требованиям защиты, который в тоже время может служить базисом для ИТ-систем и ИТ-приложений, требующих более высокую степень защиты. Это обеспечивается через соответствующее применение организационных, кадровых, инфраструктурных и технических стандартных защитных мер безопасности. Следует попутно отметить, что понятия «разумный уровень защиты» и «нормальные требования защиты» в Руководстве не определены и скорее интуитивно понимается пользователями, точнее разработчиками документа, т.к. последний регулярно модифицируется и расширяется.

Под стандартными защитными мерами безопасности в Руководстве понимается некоторая совокупность идентифицированных мероприятий безопасности, являющихся общими для групп «типичных» ИТ-систем.

Под группами «типичных» ИТ-систем понимаются специфические группы активов (assets) информационных технологий, которые в большинстве неспециализированных ИТ можно выделить как «типичные», то есть являющихся широко распространенными (не индивидуальными)

решениями и характеризующиеся обычными требованиями защиты в отношении конфиденциальности, целостности и доступности.

Активы (assets) информационной технологии включают не только ресурсы самой компьютерной системы, но и организационную, кадровую и инженерно-строительную инфраструктуры, обеспечивающие и поддерживающие её функционирование.

**1. Структура документа**

Концептуальный подход Руководства заключается в том, что обеспечение безопасности информационной технологии должно носить комплексный характер, т.е. необходимый уровень ИТ-безопасности должен достигаться не только реализацией стандартных защитных мер непосредственно в технических компонентах (отдельных продуктах информационной технологии), но также и реализацией мероприятий, охватывающих аспекты организационной, кадровой и инженерно-строительной инфраструктур.

Чтобы облегчить структурирование и обработку высоко гетерогенных областей ИТ, включая операционные среды, Руководство базовой защиты ИТ построено по модульному принципу. Отдельные модули отражают типичные области использования ИТ-активов, например сети клиент/сервер, здания, связь, прикладные компоненты и т.д.

Руководство базовой защиты ИТ состоит из следующих разделов.

Раздел «Введение и процедура» (главы 1 и 2) определяет концепцию базовой защиты ИТ, содержит указания по использованию Руководства и описывает процедуру составления концепции безопасности, которая позволяет представить базовую защиту ИТ.

Второй раздел Руководства (главы 3- 9) содержит отдельные (унифицированные) структурные модули для различных ИТ-компонентов, процедур и ИТ-систем. Модули сгруппированы в следующие главы:

Глава 3: Базовая защита универсальных ИТ-компонентов;
Глава 4: Инфраструктура;
Глава 5: Системы с несетевой структурой;
Глава 6: Системы с сетевой структурой;
Глава 7: Системы передачи данных;
Глава 8: Передача данных;
Глава 9: Другие ИТ-компоненты.

Раздел «Каталоги угроз» содержит детальные описания угроз, которые включены в сценарии угрозы для индивидуальных модулей. Угрозы сгруппированы в пять каталогов:

T1: Форс-мажор;
T2: Организационные недостатки;
T3: Ошибка оператора;
T4: Технический отказ;
T5: Преднамеренные действия.

Раздел «Каталоги защитных мер» содержит детальные описания стандартных защитных мер безопасности ИТ, упомянутых в различных модулях Руководства. Мероприятия сгруппированы в шесть каталогов мер безопасности:

S 1: Меры защиты инфраструктуры;
S 2: Организационные меры защиты;
S 3: Меры безопасности персонала, в смысле безопасности персонала для активов;
S 4: Меры защиты аппаратных средств и программного обеспечения;
S 5: Меры защиты связи;
S 6: Планирование мер безопасности для нештатных ситуаций (непредвиденных обстоятельств).

Последний раздел «Приложения» Руководства содержит вспомогательные средства, формы, краткие описания инструментальных средств, охватывающих все положения базовой защиты ИТ и списка зарегистрированных пользователей справочника.

В свою очередь каждый отдельный (унифицированный) структурный модуль включает сценарии угроз для типичных областей использования ИТ-активов и рекомендуемый набор стандартных мер защиты. «Сценарий угрозы» обеспечивает базис для формирования определенного набора стандартных мер защиты, путем из соответствующих каталогов S 1, S 2, S 3, S 4, S 5, S 6 Каталога защитных мер. Они разработаны с точки зрения пользователя ИТ и приведены для лучшего понимания и в дальнейшем не требуются пользователю для создания концепции безопасности,

Для каждого отдельного (унифицированного) структурного модуля связь между угрозами и рекомендуемыми защитными мерами безопасности показывается в таблице «Угрозы – защитные меры безопасности». Как пример, ниже в табл. 1 приведена выборка из Руководства таблицы «Угрозы – защитные меры безопасности» для модуля «Обмен данными средств информации».

**Таблица 1**

| | Приоритет | Т 1. 7 | Т 1. 8 | Т 1. 9 | Т 2. 3 | Т 2. 10 | Т 2. 17 | Т 2. 18 | Т 2. 19 | Т 3. 1 | Т 3. 3 | Т 3. 12 | Т 3. 13 | Т 4. 7 | Т 5. 1 | Т 5. 2 | Т 5. 4 | Т 5. 9 | Т 5. 23 | Т 5. 29 | Т 5. 43 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S 1.36 | 2* | X | X | | | | | | | X | | | | | X | X | X | X | | X | |
| S 2.3 | 2 | | | | X | X | X | | | | | | | X | X | X | X | X | X | X | X |
| S 2.42 | 2 | | | | | | | X | | X | | | | | | | | | | | |
| S 2.43 | 1 | | | | | X | X | X | | | | X | | | | | | | | | |
| S 2.44 | 1 | X | X | X | | | | | | | | X | | X | X | X | | X | | X | |

Заголовки столбцы показывают угрозы, перечисленные в модуле. Крайний левый столбец показывает номера защитных мер безопасности. Столбец 2 показывает приоритет, назначенный для данной защитной меры безопасности. Если этот столбец содержит звездочку, то соответствующая мера безопасности рассматривается как "необязательная" в этом модуле. Другие столбцы показывают зависимость между защитными мерами безопасности и угрозами. Символ "X" в данной клетке показывает, что соответствующая защитная мера безопасности эффективна против соответствующей угрозы. Эффект меры безопасности может быть или профилактической природы иначе нацелен на смягчение потери или повреждения. Отсутствие символа "X" в каком-либо столбце говорит о том, что против данной угрозы отсутствует защитные меры безопасности.

## 2. Применяемые критерии определения и оценки уровня безопасности ИТ

В отличие от традиционного подхода к анализу риска, подход, принятый в Руководстве базовой защиты ИТ, требует только, чтобы было выполнено целевое сравнение, путем реального сопоставления, между рекомендуемыми и уже осуществленными мероприятиями. Для определения и оценки уровня безопасности ИТ используется модель базовой защиты активов ИТ, которая составлена из различных унифицированных модулей глав 3 – 9 Руководства и отображает существенные аспекты безопасности активов ИТ на определенные модули и наоборот. Критерием является степень соответствия совокупности реализованных мер защиты активов ИТ стандартным защитным мероприятиям из соответствующих унифицированных модулей модели базовой защиты активов ИТ, которая используется как испытательный план. Недостатки безопасности, которые должны быть устранены посредством принятия рекомендуемого мероприятия, определены в терминах того идентифицированного мероприятия безопасности, которое отсут-

ствует или еще не осуществлено. Только в случаях, когда требование защиты ИТ значительно выше, необходимо также выполнить дополнительный анализ безопасности, взвешивая рентабельность осуществления дополнительного мероприятия.

Руководство базовой защиты ИТ определяет качественные формулировки при назначении требований защиты по трем категориям.

- базовая умеренная - воздействие любой грозы или ущерба ограничено;
- высокая - воздействие любой угрозы или ущерба может быть значительно;
- очень высокая - воздействие любого ущерба может достигать катастрофических размеров, которые могли угрожать самому выживанию агентства/компании.

Также Руководство содержит следующие рекомендации относительно категорий требований защиты, обеспечиваемых стандартными защитными мерами безопасности (табл. 2).

**Таблица 2**

| Категория требований защиты | Защитный эффект стандартных защитных мер безопасности, нацеленных на достижение базовой защиты ИТ |
|---|---|
| Базовая умеренная | Стандартные защитные меры безопасности, нацеленные на базовую защиту ИТ, в целом адекватны и разумны. |
| Высокая | Стандартные защитные меры безопасности, нацеленные на базовую защиту ИТ, предоставляют базовый уровень защиты, но могут быть недостаточными. Дополнительные защитные меры могут быть установлены, выполняя дополнительный анализ безопасности. |
| Очень высокая | Стандартные защитные меры безопасности, нацеленные на базовую защиту ИТ, предоставляют базовый уровень защиты, но в общем недостаточны. Необходимые дополнительные защитные меры безопасности должны быть установлены на разовом базисе на основе дополнительного анализа безопасности. |

Оценка требований защиты проводится последовательно для приложений, систем, линий связи и помещений ИТ. Цель оценки требований защиты состоит в определении требуемой степени защиты в терминах конфиденциальности, целостности и доступности.

### 3. Рекомендации по методологии и практических средствах реализации

Модульная структура Руководства предоставляет пользователям возможность просто и экономно формировать и осуществлять концепции безопасности ИТ в терминах требуемых ресурсов. Для достижения среднего уровня защиты достаточно идентифицировать модули, существенные для рассматриваемой системы или активов ИТ, и осуществить все защитные меры, рекомендуемые в тех модулях непротиворечивым способом. Руководство определяет, что эффективная реализация защитных мер безопасности ИТ требует хорошо обдуманного и управляемого процесса безопасности ИТ. Для обеспечения соответствующего уровня безопасности ИТ функциональное управление защитой ИТ в обязательном порядке должно быть организовано в начале процесса безопасности ИТ и интегрировано в существующие структуры организации. Для обеспечения непрерывного и эффективного процесса безопасности ИТ Руководство базовой защиты ИТ содержит рекомендации по методологии и практических средствах реализации. Оно также содержит возможные решения к различным задачам, касающимся безопасности ИТ, типа составления концепции безопасности ИТ, контроля средств защиты и сертификации. Различные варианты использования Руководства базовой защиты ИТ зависят от решаемых задач. Руководство определяет следующий план действий, которые являются необходимыми для поддержания непрерывного процесса безопасности ИТ:

- разработка политики безопасности ИТ;
- выбор и создание соответствующей организационной структуры для управления безопасностью ИТ;
- составление концепции безопасности ИТ;
- реализация защитных мер безопасности ИТ;
- организация обучения персонала и пользователей;
- текущее поддержание безопасности ИТ.

Процесс безопасности ИТ начинается с определения целей защиты ИТ и установления управления защитой ИТ. В состав функций управления защитой ИТ включены функции по составлению и осуществлению концепции безопасности ИТ. Этот подход иллюстрирован схематично в рис. 1.


Рис. 1

В главе 3.0 Руководства приведено описание процесса управления защитой ИТ и детальное объяснение индивидуальных действий в форме рекомендуемых стандартных защитных мер безопасности ИТ. Руководство определяет, что первичной функцией управления защитой ИТ является разработка концепции безопасности ИТ, которая является необходимой для реализации необходимых защитных мер безопасности ИТ. В частях 2.1 – 2.6 главы 2 приведено описание создания концепции безопасности ИТ, с использованием Руководства базовой защиты ИТ. Общая процедура создания концепции безопасности ИТ схематически приведена на рис. 2.

Методология анализа структуры ИТ приведена в части 2.1 главы 2 и включает следующие подзадачи:

- подготовка плана сети ИТ;
- сокращение сложности для идентификации групп подобных активов
- сбор информации о системах ИТ
- фиксация информации о приложении ИТ и связанной информации

Отправной точкой для анализа структуры ИТ служит топологический план сети ИТ, который является графическим представлением рассматриваемых компонентов, используемых в ИТ, области связи и способов их объединения в сеть. В процессе анализа структуры ИТ должны

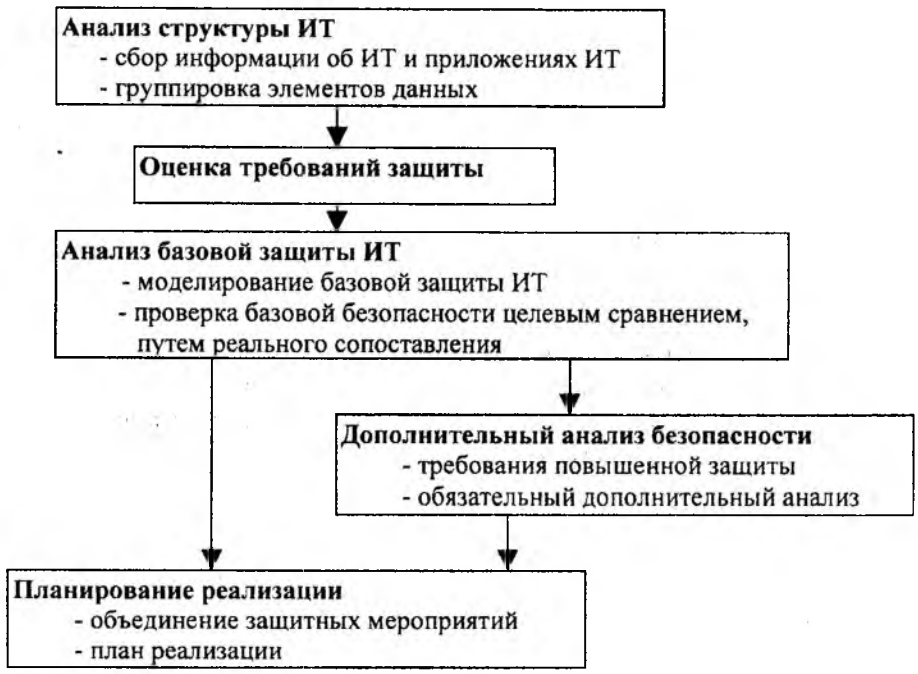



Рис. 2

быть рассмотрены:

- ·существующая инфраструктура;
- ·поддерживающее организационное и кадровое окружение активов ИТ;
- ·используемые системы ИТ (сетевые и несетевые);
- ·линии коммуникаций между системами ИТ и внешним миром;
- ·приложения ИТ, исполняемые на активах ИТ.

Руководство определяет, что план должен представлять следующие объекты:

- системы ИТ, то есть клиентские и сервер компьютеры, активные сетевые компоненты (типа центров, переключателей, маршрутизаторы), сетевые принтеры и т.д.;
- сетевые подключения между этими системами, т. е. подключения LAN (например локальная сеть на основе протокола CSMA-CD, эстафетное кольцо), базовые технологии (например. FDDI, ATM), и т.д.;
- подключения между рассматриваемой областью ИТ и внешним миром, т. е. удаленный доступ по телефону или модему, подключения Internet, использующие ISDN, модем или маршрутизаторы, радио связь или арендованные каналы и т.д.

Кроме того, для каждого из объектов, представленных в плане, должна быть записана следующая информация для каждой системы ИТ:

- уникальное название (например, полное имя хоста или номер идентификации);
- тип и функция (например, сервер базы данных для приложения X);
- поддерживающая платформа (то есть аппаратная платформа и операционная система);
- расположение (например, здание и номер комнаты);
- название ответственного администратора;
- тип сетевого подключения и сетевого адреса.

Помимо этого, не только для систем ИТ, но также и для сетевых подключений между системами и подключений к внешнему миру, должно быть приведены:

- тип соединения (например, волокно оптический кабель);
- максимальная объем передачи данных (например, 10 Mbps),
- сетевые протоколы, используемые на более низких уровнях (например, локальная сеть на основе протокола CSMA-CD, TCP/IP),
- для внешних подключений, детали внешней сети (например, Internet, название средства доступа).

Поскольку структура ИТ вообще адаптируется к определенным требованиям организации, Руководство указывает на необходимость своевременного обновления сетевого плана, для отображения текущей ситуации.

На следующем шаге анализа структуры ИТ Руководство рекомендует упростить сложность плана для идентификации групп подобных активов путем удаления любой информации, которая не нужна для решения последующего множества задач, и объединения любых идентичных компонентов в одну соответствующую группу, которая будет представлена в сетевом плане отдельным объектом. Компоненты могут быть объединены в одну и ту же группу если компоненты

- имеют тот же самый тип;
- имеют идентичные или почти идентичные конфигурации;
- присоединены к сети в тем же самым или почти тем же самым способом;
- подчинены тем же самым базовым условиям и
- используют те же самые приложения.

После того, как процесс группировки закончен, компоненты, сгруппированные вместе, показываются на сетевом плане как отдельный объект. Тип и число компонентов, представленных в каждой группе, должны быть задокументированы в виде таблицы, как показано для примера на табл. 3

**Таблица 3**

| Номер | Описание | Платформа | Номер | Инсталляционный сайт | Состояние | Пользователь (и) / Администрация. |
|---|---|---|---|---|---|---|
| S1 | Сервер для людских ресурсов | Windows NT Сервер | 1 | Бонн, R 1.01 | Операционный | Людские ресурсы |
| S2 | Первичный контроллер домена | Windows NT Сервер | 1 | Бонн, R 3.10 | Операционный | Все пользователи ИТ |
| C1 | Группа клиентов в HR обработке данных | Windows NT Рабочая станция | 5 | Бонн, R 1.02 - R 1.06 | Операционный | Людские ресурсы |
| C2 | Группа клиентов в отделе администрации | Windows NT Рабочая станция | 10 | Бонн, R 1.07 - R 1.16 | Операционный | Отдел Администрации |

Для уменьшения затрат по фиксации информации о приложениях ИТ и обрабатываемой информации Руководство рекомендует рассматривать только наиболее важные выполняющиеся или планируемые приложения ИТ, для группирования которых определены три категории:

- приложения, относительно которых необходимо, чтобы их данные/ информация и программы остались конфиденциальными (т. е. максимальное требование конфиденциальности);
- приложения, относительно которой необходимо, чтобы их данные/ информация и программы были правильными и неизменными (целостность);

– приложения, для которой только минимальное количество времени простоя может допускаться (т. е. максимальные требования доступности).

Результат этого рассмотрения - резюме какие большие приложения ИТ обрабатываются, и на каких системах ИТ. Рекомендуется, чтобы результаты были задокументированы в табличной форме.

После того, как был выполнен анализ структуры ИТ, должна быть проведена оценка требований защиты (см. рис. 2). Цель оценки требований защиты состоит в том, чтобы установить то, что защита является адекватной и разумной для информации и использованного актива ИТ. Она проводится в соответствии с методическими указаниями части 2.2. главы 2 Руководства, согласно которым оценка требования защиты зафиксированной структуры ИТ осуществляется последовательным выполнением четырех отдельных шагов.

Прежде всего, должны быть определены категории требований защиты для различных приложений ИТ. Для определения требований защиты для различных приложений ИТ рекомендуется использовать типичные сценарии ущерба. По результатам определения требований защиты для различных приложений ИТ определяются требования защиты для систем ИТ. Они в свою очередь используются, чтобы определить требования защиты для маршрутов передачи и для помещений, в которых расположены активы ИТ. Оценка требований защиты для приложений ИТ проводится для каждого приложения, включая содержащиеся или используемые в нем данные. Цель оценки требований защиты состоит в определении требуемой степени защиты в терминах конфиденциальности, целостности и доступности. Категории требований защиты для приложений ИТ определяются в Руководстве через следующие сценарии ущерба:

– нарушение законов, устава или контрактов,
– ухудшение информационного самоопределения,
– физический ущерб,
– уменьшенная эффективность режимов работы,
– отрицательные эффекты на внешние отношения и
– финансовые последствия.

Часто отдельный случай потери или ущерба может включать несколько категорий ущерба. Например, отказ приложения ИТ мог мешать выполнению основной работы, заканчивающийся прямой финансовой потерей и в то же самое время потерей имиджа.

Для получения четких границ между категориями требования защиты «базовая умеренная», «высокая» и «очень высокая», верхние и нижние пределы рекомендуется определять для индивидуальных сценариев ущерба. Чтобы получить грубое представление относительно того, какое требование защиты является соответствующим данному уровню потенциального ущерба и его воздействия, можно воспользоваться соответствующими таблицами из части 2.1. Руководство допускает в отдельных конкретных ИТ наличие сценариев ущерба отличных от вышеперечисленных. В таких случаях соответствующие таблицы из части 2.1. должны быть дополнены. Каждый из сценариев ущерба рассматривается с позиции потери конфиденциальности, целостности и доступности на основе вопроса: «Что если ...?». Чтобы упростить определение возможного ущерба, в Руководстве приведен, как справочный, набор вопросов для каждого из упомянутых сценариев ущерба, как инструмент разработки сценариев ущерба. Рекомендуемый набор вопросов на является исчерпывающим, и в каждом случае необходимо рассмотреть ситуацию в определенной организации, и включить свои вопросы в дополнение к приведенным в этом Руководстве. Пример определения категории требований защиты приложений ИТ через сценарии ущерба приведен в табл. 4.

**Таблица 4**

| Категория требования Защиты " Базовая умеренная " | |
|---|---|
| 1. Нарушение законов, устава или контрактов | • Нарушения устава и законов с незначительными последствиями<br>• Незначительные нарушения контракта, которые приводят к небольшим договорным штрафамв |
| 2. Ухудшение права на информационное самоопределение | • Ухудшение права на информационное самоопределение было бы оценено как терпимое.<br>• Возможное неправильное употребление персональных данных имеют минимальные эффекты на социальное или финансовое положение, тех кого касались. |
| 3. Физический ущерб | • Не кажется возможным. |
| 4. уменьшенная эффективность режимов работы | • Ухудшение было бы оценено как терпимое.<br>• Максимальное приемлемое время простоя большее чем 24 часа. |
| 5. Отрицательные эффекты на внешние отношения | • Минимальное ухудшение репутации / доверие, ограниченное в пределах агентства/предприятия. |
| 6. Финансовые последствия | • Финансовая потеря приемлема для агентства/компании. |

Оцененные таким образом требования защиты для различных приложений ИТ рекомендуется зарегистрировать в таблице, которая будет использована в последующей оценке требований защиты систем ИТ. Ниже, в табл. 5, приведен пример заполнения такой таблицы, который показывает главные приложения ИТ, их требования защиты и обоснование (объяснение) после назначения категорий требований защиты.

**Таблица 5**

| Приложение ИТ | | | Оценка требований защиты | | |
|---|---|---|---|---|---|
| Номер | Имя | Личные данные | Основной параметр | Требование Защиты | Объяснение |
| A1 | Обработка HR данных | X | Конфиденциальность | Высокая | HR данные составляет особенно чувствительные персональные данные, раскрытие которых может значительно вредить человеку. |
| | | | Целостность | Умеренный | Требование защиты только "умеренно", так как ошибки могут быть обнаружены быстро и исправлены. |
| | | | Доступность | Умеренный | Время простоя может быть до недели. |
| A2 | Обработка дохода | X | Конфиденциальность | Высокая | Данные дохода включают персональные данные, который имеет особенно высокое требование защиты. Раскрытие этих данных могло быть очень вредно для персонала. |

После рассмотрения приложений определяются требования защиты для систем ИТ. Оценка требований защиты для каждой системы ИТ осуществляется на основе рассмотрения всех приложений ИТ, которые имеют прямую ассоциацию с данной системой ИТ. При этом рекомендуется руководствоваться следующими принципами.

Требования защиты конкретной системы ИТ определяются ущербом или суммой наиболее серьезных случаев ущерба соответствующих приложений (**максимальный принцип**)

При изучении возможного ущерба и его значений, необходимо учитывать взаимосвязь приложений ИТ в системе ИТ (одно приложение может использовать результаты других приложений) **(отношения зависимости)**.

В случаях обработки несколько приложений ИТ или наборов информации в одной системе ИТ, необходимо определить возможность проявления совокупного (кумулятивного) эффекта как увеличения ущерба **(совокупный эффект)**, так и уменьшения (**дистрибутивный эффект**).

Результаты оценки требований защиты каждой системы ИТ должны быть задокументированы в табличной форме.

Ниже приведен пример заполнения такой таблицы из Руководства, который показывает системы ИТ, оценки требований защиты для каждой системы в терминах конфиденциальности, целостности и доступности и обоснование (объяснение) после назначения категорий требований защиты (табл.6).

**Таблица 6**

| Система ИТ | | Оценка требований защиты | | |
|---|---|---|---|---|
| Номер | Описание | Основной Параметр | Требование | Объяснение |
| S1 | Сервер для Че- | Конфиденциальность | Высокая | Максимальный принцип |
| | | Целостность | Умеренная | Максимальный принцип |
| | | Доступность | Умеренная | Максимальный принцип |
| S2 | Первичный | Конфиденциальность | Умеренная | Максимальный принцип |
| | | Целостность | Высокая | Максимальный принцип |

Аналогичным образом проводится оценка требований защиты для линий связи и для помещений, в которых установлены активы ИТ. Оценки проводятся на основе результатов оценки требований защиты ИТ-системи сетевого плана, подготовленного в разделе 2.1 при обследовании активов ИТ. Результаты, полученные на подэтапе «Оценка требований защиты», служат исходными данными для последующих действий по составлению концепции безопасности ИТ.

После завершения подэтапа «Оценка технических требований» следующим этапом создания концепции безопасности ИТ (см. рис.2) является «Анализ базовой защиты ИТ», который включает подэтап «Моделирование базовой защиты ИТ» и подэтап «Проверка базовой безопасности ИТ». Методология проведения данных работ описана соответственно в частях 2.3 и 2.4 главы 2 Руководства.

Моделирование активов ИТ заключается в сравнительном анализе и установлении соответствия каждому компоненту (или группе однотипных компонентов) из плана сети рассматриваемой ИТ определенного унифицированного модуля из Руководства, который содержит набор типовых угроз и защитных мероприятий безопасности. При этом, один и тот же модуль может быть использован несколько раз для сходных организационных компонент активов ИТ.

Для упрощения отображения сложного многообразия активов ИТ на унифицированные модули глав 3 – 9 и исключения дублирования Руководство рекомендует 5-ти уровневую модель, со следующим распределением аспектов безопасности по отдельным уровням.

Уровень 1 «Высший порядок аспектов безопасности ИТ» охватывает все общие аспекты безопасности ИТ, которые универсально применимы к каждому отдельному моделируемому компоненту активов ИТ. Первичными рассматриваемыми элементами являются политики, концепции и полученные из них процедуры. На уровне 1 используются модули 3.1 –3.8 и 9.1.

Уровень 2 «Безопасность инфраструктуры» связан со строительными и структурными условиями функционирования компонентов активов ИТ. Моделируется при помощи унифицированных модулей главы 4 Руководства, применяемых к каждому зданию, помещению, соединению (или группе этих компонентов)

Уровень 3 «Безопасность систем ИТ» рассматривает аспекты безопасности отдельных систем ИТ. И охватывается унифицированными модулями из глав 5, 6, 8 и 9 Руководства, которые применяются как к отдельным системам ИТ так и к выборка из групп.

Уровень 4 «Безопасность в сети» рассматривает аспекты безопасности при организации сети систем ИТ, которые не могут быть изолированы в отдельных системах ИТ и относятся к сетевым подключениям и связи между системами ИТ. Для упрощения рассмотрения Руководство рекомендует разделять большие сети на отдельные законченные подсети в соответствии с двумя критериями:

- оценка требований защиты идентифицировала подключения, по которым некоторые данные ни при каких условиях не должны передаваться. Такие подключения должны рассматриваться как «интерфейсы» между подсетями. И наоборот, подключения, по которым передаются данные, имеющие «высокие» или «очень высокие» требования защиты, не должны пересекать через никакие подсетевые границы;
- компоненты, связанные между собой по удаленному подключению, не должны быть назначены в одну и туже подсеть.

На уровне 4 используются унифицированные модули из глав 6, 7 и 8 Руководства.

Уровень 5 «Безопасность в приложениях» является самым нижним уровнем и рассматривает аспекты безопасности реальных приложений ИТ, которые используются в активах ИТ. Для моделирования должны применяться унифицированные модули из глав 7, 8 и 9 Руководства для каждого приложения.

Модель базовой защиты ИТ, т.е. назначение унифицированных модулей к целевым объектам, должна быть задокументирована в форме таблицы, пример которой приведен в Руководстве представлен ниже в табл. 7

**Таблица 7**

| Номер | Название модуля | Объект-цель / целевая группа | Произведена выборка | Посредник | Примечания |
|---|---|---|---|---|---|
| 3.1 | Организация | Боннский сайт | | | Модуль Организации должен работать отдельно для Боннских и Берлинских сайтов,. |
| 3.1 | Организация | Берлинский сайт | | | |
| 3.2 | Персонал | Полный BOV | | | Отдел людских ресурсов BOV'S расположен центрально в Бонне. |
| 4.3.3 | Архив Носителей данных | R U.02 (Бонн) | | | резервные средства данных сохраняются в этой комнате |
| 5.3 | PC Портативной ЭВМ | C5 | 1 в R 1.06 (Бонн) | | выборка будет выбрана из всех портативных ЭВМ, и в Бонне и Берлине. |
| 5.3 | PC Портативной ЭВМ | C6 | 1 в R 2.01 (Берлин) | | |
| 7.5 | Сервер WWW | S5 | | | Функции S5 как сервер для Intranet. |
| 9.2 | Базы данных | S5 | | | База данных используется на сервере S5. |

Результатом моделирования активов ИТ – модель базовой защиты активов ИТ, которая составлена из различных унифицированных модулей глав 3 – 9 Руководства и отражает существенные аспекты безопасности активов ИТ на определенные модули и наоборот. В зависимости от состояния рассматриваемых активов ИТ (находятся в эксплуатации либо планируются) мо-

ель может использоваться в форме тест – плана (в первом случае) либо концептуального про-кта (во - втором случае). В случаях модернизации действующих активов ИТ результирующая модель базовой защиты ИТ будет одновременно и тест – планом и концептуальным проектом, т. к. будет содержать комбинацию уже реализованных защитных мер безопасности и находящихся в стадии планирования.

По завершению моделирования базовой защиты ИТ должна быть выполнена проверка базовой безопасности ИТ. Методология проведения данных работ описана в части 2.4 главы 2 Руководства.

Унифицированный модуль из состава модели базовой защиты ИТ теперь используется как испытательный (тест) план чтобы установить, используя целевое сравнение, путем реального сопоставления, какие стандартные защитные меры безопасности были соответственно осуществлены и какие были не осуществлены либо осуществлены неудовлетворительно. Часть 2.4 главы 2 Руководства описывает, как исполнить проверку базовой безопасности в контексте центральной задачи составления концепции безопасности ИТ и определяет следующие три стадии проведения проверки. Первый стадия - предварительная организация работы и, в частности, выбор соответствующих посредников для целевого сравнения, путем реального сопоставления. На стадии 2 выполняется целевое сравнение, путем реального сопоставления, используя интервью и производя выборочную проверку. На конечной стадии, результаты целевого сравнения, путем реального сопоставления, задокументированы, вместе с обоснованием.

После окончания проверки базовой безопасности должен быть разработан и исполнен план реализации защитных мероприятий безопасности ИТ в соответствии с рекомендациями, изложенными в части 2.6 главы 2 Руководства

**Заключение**

В заключение данного краткого обзора Руководства по базовой защите ИТ следует отметить следующее.

С позиции пользователей нацеленность Руководства на оказании помощи пользователям в оперативном решении общих задач и упрощении процессов обеспечения безопасности информационных технологий безусловно является положительным фактором.

Вместе с тем, поскольку информационные технологии постоянно развиваются, BSI каждые шесть месяцев модифицирует и расширяет Руководство путем переделки (модификации) существующих либо добавлением новых компонент (структурных модулей, процедур, угроз и т. д.). Таким образом, границы понятий «разумный и достаточный уровень защиты ИТ» и «уровень базовой защиты ИТ» зыбкие и неопределенные. Аналогично обстоит дело и с понятиями «совокупность идентифицированных мероприятий безопасности» и «типичные» системы ИТ. Кроме того, в случае наличия ИТ-системс высокими либо очень высокими требованиями защиты Руководство в части 2.5 главы 2 методологически дополнительный анализ безопасности возлагает на пользователей. Сформированные пользователями по результатам анализа защитные меры безопасности будут отсутствовать в текущей редакции Руководства, и, при определенных условиях, будут включены BSI в последующие редакции. Вследствие этого, при сравнении двух редакций возникает вопрос что понимать под «уровнем базовой защиты ИТ».

Данные обстоятельства, на наш взгляд, не позволяют в принципе ввести четкую иерархию классов защищенности (безопасности) ИТ, в отличие от таких нормативных документов, как международный стандарт ISO 15408 и отечественные НД ТЗИ 2.5-004-99 Критерии оценки защищенности информации в компьютерных системах от несанкционированного доступа и НД ТЗИ 2.5-005 –99 Классификация автоматизированных систем и стандартные функциональные профили защищенности обрабатываемой информации от несанкционированного доступа. По этой причине возможность применение данного документа в целях сертификации ИТ проблема-

тична, хотя эти вопросы находится в состоянии обсуждения.

По мнению авторов, Руководство целесообразно применять в дополнение указанным стандартам в целях облегчения процессов определения перечня угроз и совокупности защитных мер безопасности ИТ.

**Список литературы** 1 *И. Д. Горбенко, д-р техн. наук, А. В. Потий, канд. техн. наук,П. И. Терещенко.* Критерии и методология оценки безопасности информационных технологий// Радиотехника. Всеукр. межвед. науч.-техн сб. 2000. Вып. 114, С 25-38. 2. *ISO/IEC 15408* Information technology - Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model, Part 2: Security functional requirements, Part 3: Security assurance requirements. 3. *German Information Security Agency.* IT Baseline Protection Manual - Standard security safeguards. *2000.*

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 11.04.2002.*

УДК 681.3.06

*А.Н. НОВИКОВ, А.А. ТИМОШЕНКО*

# ОЦЕНКА ЭФФЕКТИВНОСТИ ДЕЙСТВИЙ ЗЛОУМЫШЛЕННИКА ПРИ РЕАЛИЗАЦИИ УГРОЗ ИНФОРМАЦИИ В РАСПРЕДЕЛЕННЫХ КОМПЬЮТЕРНЫХ СИСТЕМАХ С ОТКРЫТОЙ АРХИТЕКТУРОЙ

## Введение

Согласно [1], решения, какими услугами безопасности обеспечить защиту информации, обрабатываемой в компьютерной системе (КС), где реализовать механизмы защиты и какими должны быть эти механизмы, принимаются по результатам проведения анализа угроз информации, обрабатываемой в КС, оценки вероятности реализации данных угроз и величины возможного ущерба, а также оценки рисков, связанных с их реализацией как функции вероятности и величины возможного ущерба. В [2] в качестве характеристики защищенности обрабатываемой в КС информации предложено использовать вероятность сохранения защищенности как функцию множества потенциальных угроз и множества реализованных в системе защиты информации КС механизмов защиты $P(E,M)$. Там же получено следующее ее выражение для КС с открытой архитектурой [3] и многоуровневым стеком протоколов:

$$P(E,M)=\prod_{i=1}^{L}\left(\prod_{j=1}^{N}(1-E_{ij})+\sum_{j=1}^{N}\left[E_{ij}\prod_{k=1}^{j-1}(1-E_{ik})\cdot\left(\sum_{k=1}^{j}M_{ik}\cdot\prod_{l=k+1}^{j}(1-M_{il})\right)\right]\right), \qquad (1)$$

где $L$ - количество потенциальных угроз информации, обрабатываемой в КС;

$N$ - количество уровней стека протоколов КС;

$M_{ij}$ – целочисленная переменная, значение которой определяет факт наличия/отсутствия механизма защиты от $i$-й угрозы на протоколе $j$-го уровня;

$E_{ij}$ - показатель эффективности реализации $i$-й угрозы на протоколе $j$-го уровня.

Показатель эффективности реализации угрозы информации $E_{ij}$ определен при этом как показатель оценки риска, связанного с реализацией данной угрозы:

$$E_{ij}=Q_{ij}R_{ij} \qquad (2)$$

где $Q_{ij}$ - показатель, характеризующий относительный вклад, вносимый $i$-й угрозой информации, реализованной на протоколе $j$-го уровня КС, в суммарный эффект действий злоумышленника;

$R_{ij}$ - показатель, характеризующий статистическую вероятность реализации $i$-й угрозы информации, реализованной на протоколе $j$-го уровня КС.

Целью данной статьи является разработка инженерных методик оценки значений показателей $Q_{ij}$ и $R_{ij}$ и, таким образом, определения эффективности реализации угроз информации, обрабатываемой в КС с открытой архитектурой.

## Методика оценки относительного вклада различных угроз информации в суммарный эффект действий злоумышленника

Относительный вклад различных угроз информации в суммарный эффект действий злоумышленника определяется относительной величиной возможных потерь, вызванных реализациями тех или иных угроз информации. К сожалению, оценка величины возможных потерь, вызванных реализациями различных угроз, трудно поддается прямому расчету,

поскольку, как правило, неизвестна стоимостная оценка той или иной информации. Однако, данную задачу можно решить при помощи методов теории принятия решений, например, при помощи метода анализа иерархий (МАИ), суть которого заключается в декомпозиции проблемы на все более простые составные части и дальнейшей обработке последовательности суждений лиц, принимающих решения (ЛПР) по парным сравнениям [4]. Система парных сведений приводит к результату, который может быть представлен в виде обратно симметричной матрицы (матрицы приоритетов), элементами которой $W_{ij}$ являются интенсивности проявления элемента иерархии $i$ относительно элемента иерархии $j$, оцениваемая по шкале интенсивности от 1 до 9, где оценки имеют следующих смысл:

1 - равная важность;
3 - умеренное превосходство одного над другим;
5 - существенное превосходство одного над другим;
7 - значительное превосходство одного над другим;
9 - очень сильное превосходство одного над другим;
2, 4, 6, 8 - соответствующие промежуточные значения.

При заполнении матрицы приоритетов руководствуются правилом: если при сравнении элемента $i$ с элементом $j$ получено $W_{ij} = b$, то $W_{ji} = 1/b$.

После построения матрицы приоритетов приоритет каждого отдельного объекта в иерархии определяется оценкой соответствующего ему элемента нормализованного главного собственного вектора данной матрицы. После этого по определенным правилам может быть выражена относительная степень (интенсивность) взаимодействия элементов в системе, представляющая, в нашем случае, значение показателя $Q_{ij}$. Точное определение главного собственного вектора матрицы приоритетов достаточно сложно, на практике [5] предлагается пользоваться одним из следующих способов:

1. Суммировать элементы каждой строки и нормализовать делением каждой суммы на сумму всех элементов матрицы. Первый элемент результирующего вектора будет приоритетом первого объекта, второй- второго объекта и т.д.
2. Суммировать элементы каждого столбца и получить обратные величины этих сумм. Нормализовать их так, чтобы сумма равнялась единице, разделив каждую обратную величину на сумму всех обратных величин.
3. Разделить элементы каждого столбца на сумму элементов данного столбца (нормализовать столбец), затем сложить элементы каждой полученной строки и разделить эту сумму на число элементов строки.
4. Вычислить среднее геометрическое каждой строки и нормализовать полученные числа.
5. Возводить матрицу в произвольно большие степени, вычислять сумму элементов строк и нормализовывать полученные суммы.

Хотя наиболее точным является последний способ, в [5] рекомендуется использовать четвертый способ, то есть вычислять компоненты вектора приоритетов $V_i$ следующим образом:

$$V_i = \frac{\sqrt[N]{\prod_{j=1}^{N} W_{ij}}}{\sum_{k=1}^{N} \sqrt[N]{\prod_{j=1}^{N} W_{kj}}} \qquad (3)$$

где $N$ - размерность матрицы приоритетов;
$W_{ij}$ - элемент матрицы приоритетов, отражающий результат сравнения $i$ - го элемента с $j$ -м.

Основной задачей, которую необходимо решить для успешного применения МАИ, является задача декомпозиции сложной проблемы- определение ее компонентов и отношений между ними, то есть построение модели реальной системы в виде иерархии. В нашем случае для построения такой иерархии необходимо выделить включаемые в нее характеристики угроз информации. Для выбора набора таких характеристик проанализируем набор классификационных признаков известных [6-9] систем классификации угроз информации, обрабатываемой в КС:

- область поражения (угрозы для информационной среды, ее подсистем и элементов; угрозы для предметных областей информационного обеспечения - субъектов и объектов пользования; угрозы для всей социальной системы, исходящие от информационной среды);
- связь с информационной средой определенной социальной системы (внешние; внутренние - идущие от социальной системы и ее элементов; внутрисистемные - исходящие от самой информационной среды);
- сила воздействия на область поражения (разрушительные; дестабилизирующие; парализующие; стимулирующие);
- организационная форма выражения и степень социальной опасности (коллизии; конфликты; проступки; преступления; аварии; катастрофы);
- цель реализации угрозы (нарушение конфиденциальности информации; нарушение целостности информации; нарушение доступности информации);
- принцип воздействия на КС (с использованием доступа субъекта КС к объекту; с использованием скрытых каналов);
- характер воздействия на КС (активное воздействие; пассивное воздействие);
- причина появления используемой уязвимости в системе защиты (неадекватность реализуемой политики безопасности реальной КС; ошибки административного управления; ошибки, допущенные на этапе проектирования; ошибки, допущенные на этапе реализации);
- способ воздействия на объект атаки (с непосредственным воздействием на объект атаки; с воздействием на систему разрешений; с опосредованным воздействием);
- способ воздействия на КС (в интерактивном режиме; в пакетном режиме);
- объект атаки (с воздействием на КС в целом; с воздействием на объекты КС; с воздействием на субъекты КС; с воздействием на канал передачи данных);
- используемые средства атаки (стандартное программное обеспечение; специально разработанное программное обеспечение);
- состояние объекта атаки (объект атаки, находящийся в состоянии хранения; обьект атаки, находящийся в состоянии передачи; обьект атаки, находящийся в состоянии обработки);
- условие начала осуществления воздействия (по запросу от атакуемого объекта; по наступлению ожидаемого события на атакуемом объекте; безусловное);
- наличие обратной связи с атакуемым объектом (с обратной связью; без обратной связи);
- расположение субъекта атаки в вычислительной сети относительно атакуемого объекта (внутрисегментное; межсегментное);
- уровень стека протоколов эталонной модели взаимодействия открытых систем (ВОС) ISO, на котором осуществляется воздействие (физический; канальный; сетевой; транспортный; сеансовый; представления данных; прикладной).

Из указанных классификационных признаков в наибольшей степени величину потерь, вызванных реализацией угроз информации, характеризуют: цель реализации угрозы, объект атаки, область поражения (компонент КС) и уровень стека протоколов, на котором осуществляется воздействие. Именно данные характеристики и предлагается использовать для оценки относительного вклада различных угроз обрабатываемой в КС информации в

суммарный эффект действий злоумышленника. В качестве иерархии приоритетов данных характеристик угроз, используемой при оценке, предлагается использовать следующую (рис.1):

**1-й уровень-** приоритет цели угрозы (нарушение конфиденциальности, целостности, доступности), определяемый типом угрозы и классом автоматизированной системы (АС), основой которой является рассматриваемая КС, согласно [10], то есть тем, к обеспечению каких свойств информации (конфиденциальности, целостности или доступности) в данной АС предъявляются повышенные требования;

**2-й уровень-** приоритет информационного объекта, по отношению к которому реализуется угроза;

**3-й уровень-** приоритет компонента КС, в котором реализуется угроза, определяемый, в первую очередь, количеством потенциально возможных каналов взаимодействия с другими компонентами КС;

**4-й уровень-** приоритет уровня стека протокола, на котором реализуется угроза, более низкие уровни стека протоколов имеют более высокий приоритет, поскольку на протоколах более низкого уровня обрабатываются большие, по сравнению с протоколами более высокого уровня, объемы информации.

На рисунке 1 представлена иерархия приоритетов характеристик угроз информации.

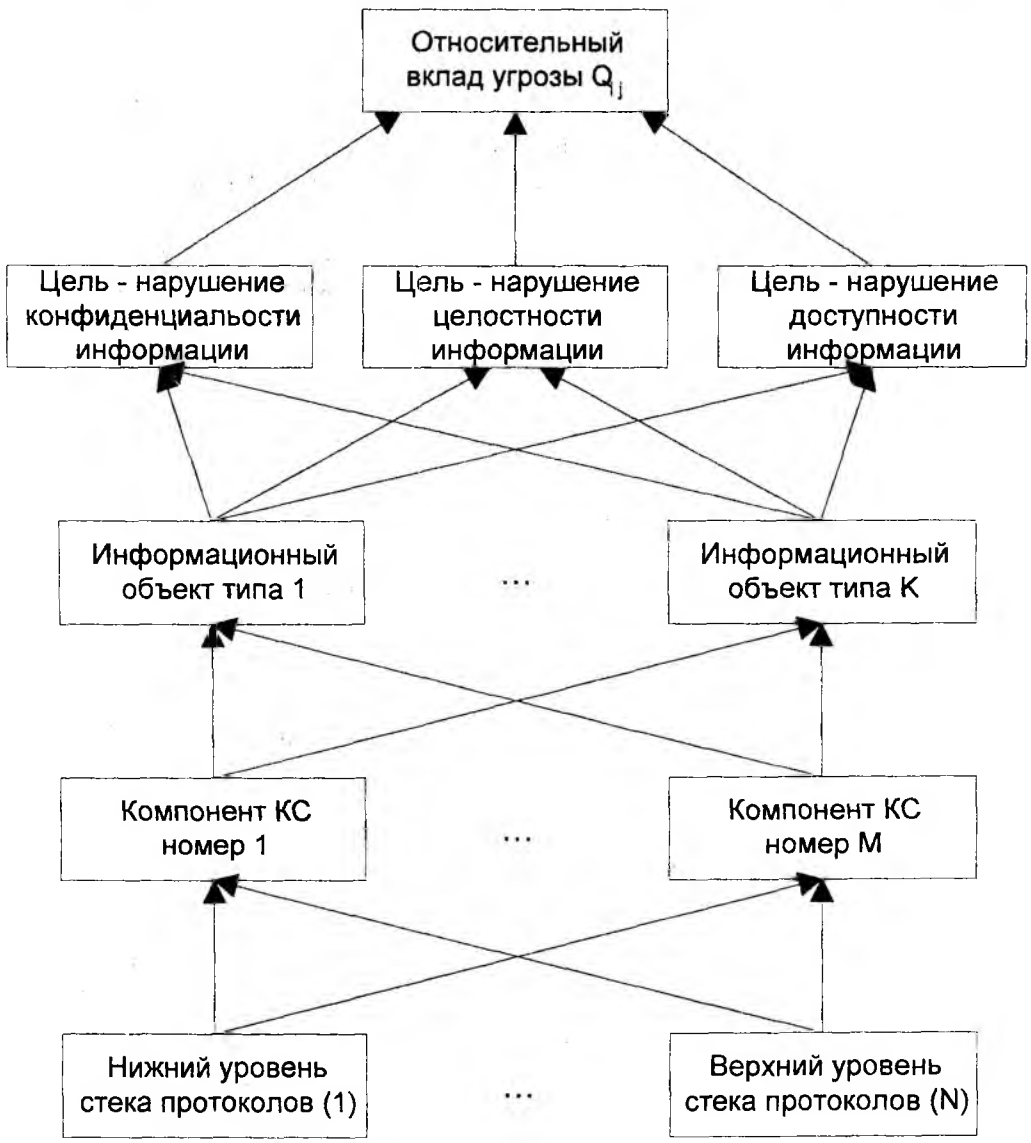



Рис.1

Предлагаемая методика оценки показателя $Q_{ij}$, характеризующего относительный вклад, вносимый $i$-й угрозой информации, реализованной на протоколе $j$-го уровня, в суммарный эффект действий злоумышленника, с использованием МАИ заключается в следующем:

1. Составляется матрица приоритетов типов угроз (угрозы конфиденциальности, целостности, доступности), определяемых классом автоматизированной системы (АС), основой которой является рассматриваемая КС, согласно [10], то есть тем, к обеспечению каких свойств информации (конфиденциальности, целостности или доступности) в данной АС предъявляются повышенные требования, по (3) определяется вектор приоритетов типов угроз $\{T_n\}$, где $n = 1,..,3$ – номер типа угрозы;

2. Составляются матрицы приоритетов типов информационных объектов, по отношению к которым реализуются угрозы, для различных типов угроз, по (3) определяются векторы приоритетов типов информационных объектов для угроз различных типов $\{O_{kn}\}$, где $k = 1,..,K$ – номер типа информационного объекта, $K$ - количество типов информационных объектов в КС, $n = 1,..,3$ – номер типа угрозы;

3. Составляются матрицы приоритетов компонентов КС, в которых реализуются угрозы, по отношению к различным типам информационных объектов, по (3) определяются векторы приоритетов компонентов КС для различных типов информационных объектов $\{C_{mk}\}$ где $m = 1,..,M$ – номер типа компонента КС, $M$ - количество типов компонентов в КС, $k = 1,..,K$ – номер типа информационного объекта, $K$ - количество типов информационных объектов в КС;

4. Составляются матрицы приоритетов уровней стека протоколов, на которых реализуются угрозы в различных компонентах КС, по (3) определяются векторы приоритетов уровней стека протокола для различных компонентов КС $\{P_{jm}\}$, где $j = 1,..,N$ - номер уровня стека протокола, $N$ - количество уровней стека протоколов, $m = 1,..,M$ – номер типа компонента КС, $M$ - количество типов компонентов в КС;

5. Значение показателя $Q_{ij}$ для $i$-й угрозы информации $n$-го типа, реализованной по отношению к $k$-му информационному объекту в $m$-м компоненте КС на протоколе $j$-го уровня, определяется согласно выражению:

$$Q_{ij} = T_n \cdot O_{kn} \cdot C_{mk} \cdot P_{jm}, \qquad (4)$$

где $T_n$ – элемент вектора приоритетов типов угроз $\{T_n\}$, соответствующий $n$-му типу угрозы;

$O_{kn}$ – элемент соответствующего $n$-му типу угрозы вектора приоритетов $\{O_{kn}\}$, соответствующий информационному объекту $k$-го типа;

$C_{mk}$ - элемент соответствующего информационному объекту $k$-го типа вектора приоритетов $\{C_{mk}\}$, соответствующий компоненту КС $m$-го типа;

$P_{jm}$ - элемент соответствующего компоненту КС $m$-го типа вектора приоритетов $\{P_{jm}\}$, соответствующий $j$-му уровню стека протоколов.

**Методика оценки статистической вероятности реализации угроз информации на различных уровнях стека протоколов КС с открытой архитектурой**

Поскольку показатель $R_{ij}$ должен характеризовать статистическую вероятность реализации $i$-й угрозы на протоколе $j$-го уровня, для его оценки необходимо воспользоваться статистическими данными о попытках реализации тех или иных угроз в КС.

Для проектируемых КС такие данные, естественно, отсутствуют, но могут быть заменены данными, полученными для уже функционирующих КС с аналогичной архитектурой. Поскольку, как показано в [11,12], все КС с многоуровневой архитектурой стека протоколов, независимо от конкретного используемого стека протоколов, функционируют по схожим принципам, оценки, полученные для одной КС, с достаточно высокой достоверностью можно применять и для других КС.

Наиболее подходящей для выполнения статистического анализа попыток реализации различных угроз информации является наибольшая из существующих распределенных КС с открытой архитектурой – Internet, функционирующая на основе стека протоколов TCP/IP. Облегчает анализ наличие информации о результатах мониторинга инцидентов с безопасностью, публикуемых различными организациями (например, CERT, CIAC, NIPC), а также ряда исследований, в той или иной мере обобщающих полученные данные.

Для оценки значений показателя $R_{ij}$ предлагается воспользоваться статистическими данными о зарегистрированных NIPC (www.nipc.gov) фактах реализации угроз информации в Internet, классифицированных по уровню стека протоколов TCP/IP, на котором реализовывалась атака. При этом в качестве оценки $R_{ij}$ предлагается использовать среднее по времени значение относительной частоты попыток реализации угроз информации на различных уровнях стека протоколов.

Согласно правилам статистического анализа [13] временных рядов, достоверной можно считать только такую оценку среднего значения, для которой:

- отсутствует автокорреляция остатков (отклонений вычисленных значений относительных частот от среднего по времени), то есть тренд относительных частот по времени является стационарным;
- ряд остатков представляет собой белый шум с распределением, близким к нормальному;
- суммарное относительное отклонение измеренных значений от среднего достаточно мало.

Для проверки наличия/отсутствия автокорреляции остатков можно воспользоваться критерием Дарбина-Уотсона:

$$DU = \frac{\sum_{k=2}^{N}(d_k - d_{k-1})^2}{\sum_{k=1}^{N} d_k^2} \qquad (5)$$

где $d_k = \bar{Y}_k - Y_k$ - разность среднего и фактического значения (остаток) на интервале в $N$ точек, для которых $Y_k$ известно.

Если значение $DU$ близко к 2, то автокорреляция остатков отсутствует, если близко к 0 или к 4, то присутствует.

Для проверки гипотезы о нормальном распределении ряда остатков можно использовать правило Романовского, согласно которому гипотеза о нормальном распределении независимых случайных величин принимается, если выполняется неравенство:

$$\frac{\chi^2 - r}{\sqrt{2r}} < 3 \qquad (6)$$

где $\chi^2$ – значение критерия $\chi^2$;

$r$ – число степеней свободы распределения, равное разности между числом разрядов разбиения всего количества наблюдений и числом наложенных связей (3 для нормального

закона распределения).

Значение критерия $\chi^2$ вычисляется по формуле:

$$\chi^2 = \sum_{i=1}^{k} \frac{(m_i - nP_i^*)^2}{nP_i^*} \tag{7}$$

где $k$ – число разрядов разбиения всего количества наблюдений;

$m_i$ - число значений случайной величины в $i$-м разряде;

$n$ – количество наблюдений;

$P_i^*$ - теоретическая вероятность попадания случайной величины в $i$-й интервал в соответствии с выбранным законом распределения.

Суммарное относительное отклонение измеренных значений от среднего оценим по формуле:

$$\delta = \sum_{k=1}^{N} \left| \frac{Y - Y_k}{Y} \right| \cdot 100\% \tag{8}$$

где $Y$ - среднее значение;

$Y_k$ - $k$-е зарегистрированное значение;

$N$ – количество наблюдений.

На рис. 2-5 приведены графики временных рядов, соответствующих изменению относительных частот (в процентах) зарегистрированных значений во времени, графики экспериментального (часта $P_i$) и теоретического нормального распределения вероятностей ($P_i$ теоретическая), а в таблице 1 приведены результаты анализа по указанным критериям.

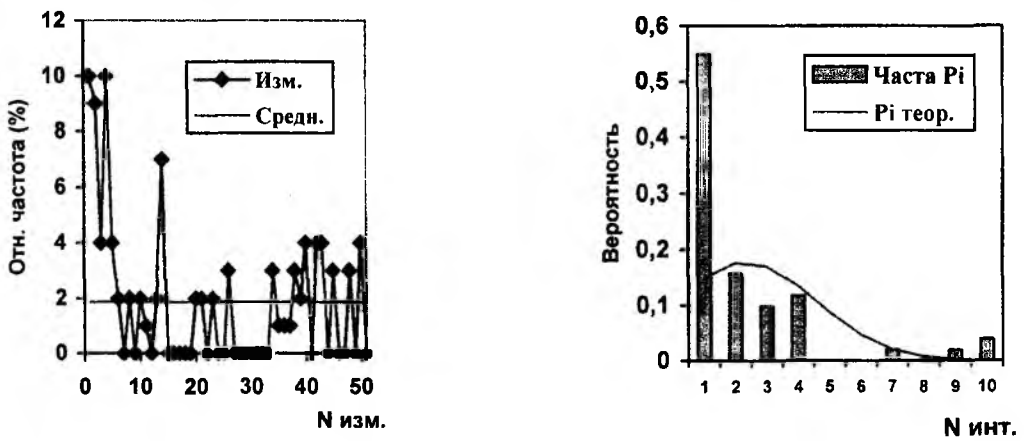



Рис. 2.

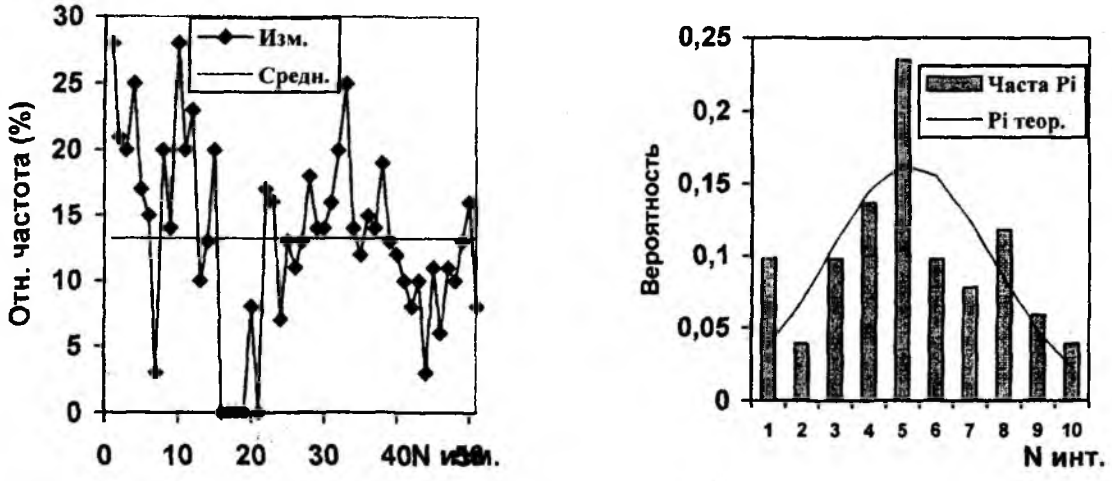



Рис. 3.

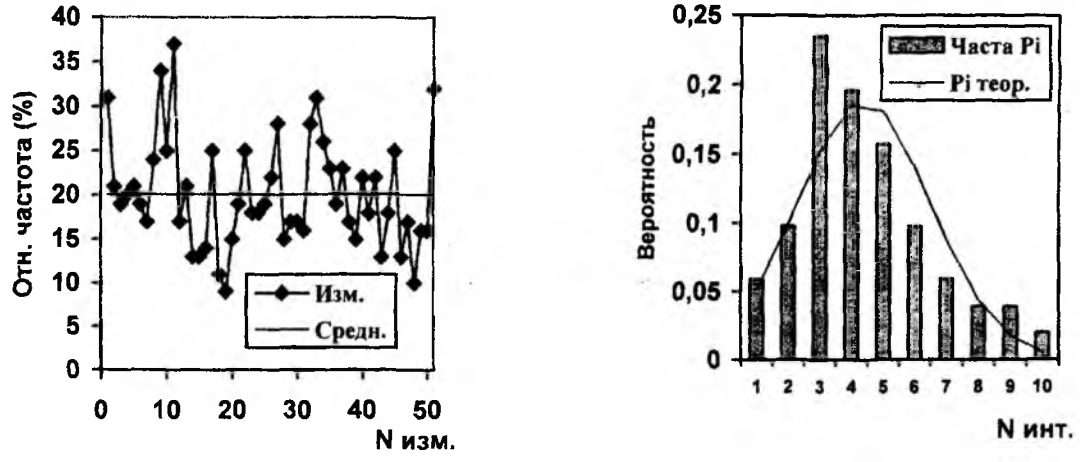



Рис. 4.

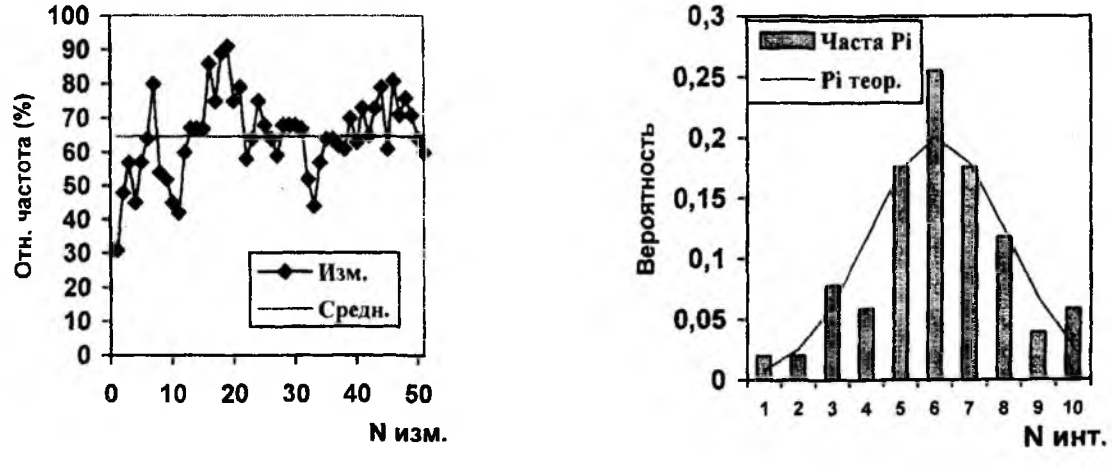



Рис. 5.

**Таблица 1**

| Показатель | Значение для уровней стека TCP/IP | | | |
|---|---|---|---|---|
| | Канальный, $j$=1 | Сетевой, $j$=2 | Транспортный, $j$=3 | Прикладной, $j$=4 |
| Среднее значение, $S_j$ | 1.86 | 13.22 | 20.08 | 64.73 |
| Математическое ожидание, $M_j$ | 1.83 | 13.15 | 19.90 | 64.35 |
| Дисперсия, $D_j$ | 4.97 | 45.51 | 34.46 | 138.93 |
| Среднеквадратическое отклонение, $\sigma_j$ | 2.23 | 6.75 | 5.87 | 11.79 |
| Значение критерия Дарбина-Уотсона, $DU_j$ | 1.12 | 1.02 | 1.38 | 0.77 |
| Значение критерия Романовского | 55.82 | 1.02 | 0.03 | 0.33 |
| Суммарное относительное отклонение (%), $\delta_j$ | 101.38 | 41.13 | 24.29 | 13.99 |

Как видно из приведенных результатов, полученная таким образом оценка среднего значения в целом (то есть для сетевого, транспортного и прикладного уровней) отвечает критериям достоверности. Несоответствие критериям достоверности оценки, полученной на основании данных о попытках реализации угроз на нижнем (канальном) уровне, объясняется большим количеством нулевых значений относительных частот (рис.2), связанных с очень редкой реализацией угроз информации на данном уровне стека протоколов.

На рис. 6 показаны точки, соответствующие экспериментально полученным значениям статистической вероятности реализации угроз информации на разных уровнях стека протоколов TCP/IP ($P_j$), а также график аппроксимации данных значений кривой 3-го порядка вида:

$$R_{ij}(j) = aj^3 \tag{9}$$

где $a$ - нормировочный коэффициент, $a = 1/\sum_{k=1}^{N} k^3$;

$N$ - количество уровней стека протоколов ( для стека TCP/IP – 4);

$j$ – номер уровня стека протоколов TCP/IP, $j = 1 \ldots N$.


Рис. 6.

Для оценки точности предлагаемой аппроксимации оценим отклонения $\Delta_j = | P_j - R_{ij}(j) |$ теоретических значений $R_{ij}$, полученных по выражению (9), от экспериментальных $P_j = M_j / 100$, сравним их со значениями соответствующих среднеквадратических отклонений $\sigma_{P_j} = \sigma_j / 100$ и доверительных интервалов $3\sigma_{P_j}$ (таблица 2).

**Таблица 2**

| Показатель | Значения для уровней стека TCP/IP | | | |
|---|---|---|---|---|
| | Канального | Сетевого | Транспортного | Прикладного |
| $j$ | 1 | 2 | 3 | 4 |
| $P_j$ | 0.0186 | 0.1322 | 0.2008 | 0.6473 |
| $R_{ij}(j)$ | 0.01 | 0.08 | 0.27 | 0.64 |
| $\Delta_j$ | 0.0086 | 0.0522 | 0.0692 | 0.0073 |
| $\sigma_{P_j}$ | 0.0223 | 0.0675 | 0.0587 | 0.1179 |
| $3\sigma_{P_j}$ | 0.0669 | 0.2025 | 0.1761 | 0.3537 |

Как видно из приведенной выше таблицы, значения отклонений $\Delta_j$ не превышает значений доверительных интервалов $3\sigma_{P_j}$, а в большинстве случаев, и соответствующих значений среднеквадратических отклонений $\sigma_{P_j}$. Это дает основания утверждать о достаточной точности аппроксимации и применимости выражения (9) для оценки статистической вероятности реализации угроз информации на различных уровнях стека протоколов TCP/IP.

**Выводы**

Предложенная методика оценки значений показателя $Q_{ij}$, характеризующего относительный вклад, вносимый различными угрозами информации в суммарный эффект действий злоумышленника, с использованием МАИ, может быть применена для КС произвольным стеком протоколов, состоящей из произвольного набора компонентов.

С учетом показанного в [11,12] наличия взаимно-однозначного соответствия между уровнями стека протоколов TCP/IP, уровнями стека протоколов эталонной модели BOC ISO [3] и уровнями других известных стеков протоколов (например, IPX/SPX, Netbios и т.п.), предложенная методика оценки значений показателя $R_{ij}$, характеризующего статистическую вероятность реализации угроз информации на различных уровнях стека протоколов TCP/IP, с использованием (10) , может быть применена для оценки статистической вероятности реализации угроз информации в КС с произвольными стеками протоколов.

Полученные с использованием данных показателей оценки эффективности реализации различных угроз информации позволяют, используя (1), осуществлять выбор множества механизмов защиты, обеспечивающего требуемый уровень защищенности обрабатываемой в КС с открытой архитектурой информации от заданного множества угроз.

**Список литературы:** 1.*НД ТЗИ 1.1-002-99.* Общие положения по защите информации в компьютерных системах от несанкционированного доступа. 2. *Новиков А., Тимошенко А.* Построение логико-вероятностной модели защищенной компьютерной системы//Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2001. Вип. 3. с. 101-105. 3. *ISO/IEC 7498-1:1994*, Information technology – Open Systems Interconnection - Basic Reference Model: The Basic Model. 4. *Саати Т.* Принятие решений. Метод анализа иерархий.– М.: Радио и связь, 1993. 5. *Саати Т., Кернс К.* Аналитическое планирование. Организация систем. М.: Радио и связь, 1991. 6. *Гайкович В., Першин А.* Безопасность электронных банковских систем. М.: Изд-во компании "Единая Европа", 1993. 7. *Охрименко С. А., Черней Г. А.* Угрозы безопасности автоматизированным информационным системам (программные злоупотребления)//НТИ. Сер.1. Орг. и методика информ. работы. 1996, №5, стр.5-13. 8. *Охрименко С., Черней Г., Фотенко В., Руссу В.* Система банковской безопасности. Банковско- финансовый центр Республики Молдова, 1996 . 9. *Медведовский И.Д., Семьянов П.В., Платонов В.В.* Атака через Internet. СПб.: Мир и Семья-95. 1997. 10. *НД ТЗИ 2.5-005-99.* Классификация автоматизированных систем и стандартные функциональные профили защищенности обрабатываемой информации от несанкционированного доступа. 11. *Демченко Ю.В.* Безопасность компьютерных сетей в соответствии с рекомендациями стандарта ISO 7488-2.//Безопасность информации. 1995. № 3. С. 23-35. 12. *Демченко Ю.В.* Архитектура безопасности Интернет и компьютерных сетей на основе протоколов TCP/IP//Безопасность информации. 1996. № 3. С. 34-48. 13. *В.С. Королюк, Н.И. Портенко, А.В. Скороход, А.Ф. Турбин.* Справочник по теории вероятностей и математической статистике. М.: Наука, 1985.

*Физико-технический институт НТУУ «КПИ»,*
*ООО «Институт компьютерных технологий»*

*Поступила в редколлегию 10.04.2002.*

УДК 621.391: 519.7

*А. Н. АЛЕКСЕЙЧУК, канд. техн. наук, А. И. РОМАНОВ, д-р техн. наук*

# РЕГУЛЯРНЫЕ КОНГРУЭНЦИИ И СТРОЕНИЕ АЛГЕБРАИЧЕСКИХ МОДЕЛЕЙ СИММЕТРИЧНЫХ КРИПТОСИСТЕМ

## 1. Введение

Алгебраический подход к решению прикладных задач в области построения и анализа математических моделей электронных устройств, осуществляющих обработку дискретной информации, является наиболее универсальным и естественным при исследовании аналитических и структурных свойств преобразований, реализуемых такими устройствами. В ряду общих математических моделей, используемых в настоящее время при оценке стойкости криптографических систем и алгоритмов, важное место занимают различные алгебраические структуры – многоосновные универсальные алгебры [1 – 3], а также их вероятностные обобщения и аналоги [4 – 6].

Впервые алгебраическая модель симметричной криптосистемы как упорядоченного набора трех множеств (соответственно открытых сообщений, шифрованных сообщений и ключей) и одного отображения, удовлетворяющих определенным естественным ограничениям, была предложена К. Шенноном [7]. Несмотря на свою простоту, эта модель оказалась достаточно полезной при решении на абстрактном уровне разнообразных задач анализа и синтеза криптографических систем, способствовала более глубокому пониманию принципов их построения и функционирования.

Со времен Шеннона алгебраическая модель криптографической системы как составляющая более общей, вероятностной ее модели лежит в основе теоретико-информационного подхода к анализу стойкости симметричных криптосистем [7 – 11]. Вместе с тем, исследованию взаимосвязи между строением криптосистемы как алгебраической структуры и основными характеристиками ее теоретической стройкости (ненадежностью сообщения или ключа, вероятностью правильного дешифрования сообщений) не уделялось, на наш взгляд, должного внимания. В частности, несмотря на достаточно большое число работ, посвященных изучению влияния рандомизации (случайного кодирования) источника сообщений на стойкость защиты информации (см., например, [9 – 12]), не решены в общем случае задачи количественной оценки стойкости и конструктивного построения систем со случайным кодированием в дискретных каналах без памяти. Также практически не исследован вопрос о зависимости вероятности правильного дешифрования сообщений рандомизированного источника от комбинаторных свойств матрицы переходных вероятностей шифра. В этой связи необходимо отметить работы [4 – 6], в которых развивается алгебраический подход к анализу дискретных случайных процессов, используемых в том числе в качестве теоретико-вероятностных моделей устройств, осуществляющих шифрование информации.

Целью настоящей статьи является систематическое исследование строения и общих структурных свойств алгебраических моделей симметричных криптосистем (шифров).

Суть основных результатов, представленных в статье, состоит в следующем. Каждому вполне эндоморфному шифру $A$ (рассматриваемому как двухосновная универсальная алгебра относительно системы его шифрующих преобразований) ставится в соответствие группа подстановок $\wp(A)$ на множестве $X$ открытых сообщений данного шифра (см. определение 2.5). Устанавливается, что строение шифра $A$ в значительной степени определяется свойствами его группы $\wp(A)$. В частности, в п. 3 определяется понятие разложения шифра в сумму подшифров и показывается, что орбиты группы шифра $A$ находятся во взаимно однозначном соответствии с его неразложимыми подшифрами. (Отметим тесную связь указанных результатов с аналогичными предложениями о разложимости конечных полуавтоматов [13] и неотрицательных матриц [6, 14]).

В п. 4 изучается строение решетки регулярных конгруэнций неразложимого шифра $A$. Показывается, что эта решетка изоморфна решетке подгрупп группы $\wp(A)$, содержащих стабилизатор фиксированного открытого сообщения $a \in X$. Далее вводится понятие сплетения шифров и доказывается теорема о представлении шифра с импримитивной группой в виде сплетения двух неразложимых шифров, уточняющая (для случая шифров) результат [2] о расширениях конечных многоосновных универсальных алгебр. Описывается строение групп указанных двух шифров и приводятся необходимые и достаточные условия, при которых эти шифры не имеют нетривиальных регулярных конгруэнций. Обсуждаются практические применения полученных результатов, связанные с возможностью сокращения трудоемкости ряда алгоритмов криптоанализа шифров с импримитивными группами на основе "метода гомоморфизмов" [1, 2]. В п. 5 полученные результаты применяются при анализе строения групповых шифров.

С целью уточнения используемой терминологии обозначений, а также для того, чтобы сделать изложение основных результатов статьи более доступным, в п. 2 приводятся определения ряда понятий и некоторые вспомогательные предложения об алгебраических моделях шифров и конечных группах подстановок. В последующих статьях предполагается изложить применения полученных результатов к анализу и синтезу вероятностно-криптографических систем с дискретным каналом без памяти [15].

## 2. Основные понятия, обозначения и некоторые вспомогательные результаты

Согласно [7], *алгебраическая модель симметричной криптосистемы* (называемая также *алгебраической моделью шифра* или просто *шифром* [16]) представляет собой универсальную алгебру

$$A = (X, K, Y, f) \tag{2.1}$$

с системой носителей $(X, K, Y)$ (где $X$, $K$ и $Y$ – непустые конечные множества) и одной алгебраической операцией $f$: $X \times K \to Y$, удовлетворяющей следующим условиям:

(а) $f$ – сюръекция;

(б) для любых $k \in K$, $x_1, x_2 \in X$: $(x_1 \neq x_2) \Rightarrow (f(x_1, k) \neq f(x_2, k))$.

Множества $X$, $Y$ и $K$ называют соответственно *множеством открытых сообщений*, *множеством щифрованных сообщений* и *множеством ключей* шифра $A$; отображение $f$ называется *функцией шифрования* этого шифра. Говорят, что шифрованное сообщение $y \in Y$ получено путем зашифрования открытого сообщения $x \in X$ на ключе $k \in K$, если выполняется равенство

$$f(x, k) = y. \tag{2.2}$$

Пусть $A = (X, K, Y, f)$ – произвольный шифр. Для любого ключа $k \in K$ определим *частичную функцию* $f_k$: $X \to Y$ функции шифрования $f$, полагая $f_k(x) = f(x, k)$, $x \in X$. Назовем $f_k$ *шифрующим преобразованием* шифра $A$, соответствующим ключу $k \in K$. В дальнейшем для краткости часто будем писать $y = kx$ вместо $y = f_k(x)$. Единственный открытый текст $x \in X$ (если он существует), удовлетворяющий (2.2), будет обозначаться символом $x = k^{-1}y$.

Очевидно, что шифр $A$ по существу представляет собой параметризованное множеством ключей $K$ семейство $(f_k: k \in K)$ шифрующих преобразований множества $X$ во множество $Y$ и может быть таким образом отождествлен с двухосновной универсальной алгеброй $(X, Y, \Pi(K, f))$ с системой операций $\Pi(K, f) \overset{\text{def}}{=} (f_k: X \to Y: k \in K)$. В настоящей статье *подшифром* (*конгруэнцией*, *гомоморфизмом*) *шифра* $A$ называется любая подалгебра (конгруэнция,

гомоморфизм) универсальной алгебры $(X, Y, \Pi(K, f))$ (определения основных понятий теории универсальных алгебр можно найти в [1, 13]).

Ключи $k$ и $k'$ шифра $A$ называются (*криптографически*) *эквивалентными*, если для любого $x \in X$ выполняется равенство $f_k(x) = f_{k'}(x)$. Шифр, полученный путем отождествления попарно эквивалентных ключей шифра $A$, называется *приведенной формой* шифра $A$.

Отметим, что в силу условия (б) шифрующие преобразования шифра $A$ являются инъективными отображениями из $X$ в $Y$. В частности, для любого шифра вида (2.1) справедливо неравенство $|X| \le |Y|$. В большинстве приложений множества $X$ и $Y$ совпадают (в этом случае говорят, что $A$ является *эндоморфным шифром* [7]).

Назовем шифр $A$ вида (2.1) *вполне эндоморфным*, если выполняется равенство $|X| = |Y|$. Очевидно, что каждый вполне эндоморфный шифр $A = (X, K, Y, f)$ подобен (в смысле [7]) эндоморфному шифру $A_h = (X, K, X, hf)$ с множеством открытых сообщений $X$, где $h: Y \to X$ – произвольная биекция, $hf$ – произведение отображений $f$ и $h$. Как правило шифры $A$ и $A_h$ обладают одинаковыми криптографическими свойствами, что позволяет в определенном смысле ограничиться изучением лишь одного из них. Для наших целей, однако, будет удобнее проводить различие между такими шифрами (в частности при изучении строения решеток подшифров и регулярных конгруэнций эндоморфных шифров в пп. 3, 4). На протяжении всего дальнейшего изложения слово "шифр" означает "вполне эндоморфный шифр", если специально не оговорено противное.

Введем в рассмотрение некоторые классы шифров, представляющие наибольший интерес с практической точки зрения.

**2.1. Определение.** Шифр $A$ вида (2.1) называется (а) *транзитивным*, если для любых $x \in X$, $y \in Y$ существует ключ $k \in K$ такой, что $kx = y$;

(б) *регулярным* [8], если для любых $x \in X$, $y \in Y$ существует не более одного ключа $k \in K$ такого, что $kx = y$.

Транзитивный регулярный шифр называется *минимальным шифром* или (*табличным*) *шифром гаммирования* [16].

Отметим, что класс регулярных шифров впервые явно выделен в [7] (К. Шеннон назвал такие шифры замкнутыми секретными системами).

Сформулируем необходимые для дальнейшего свойства транзитивных и регулярных шифров в виде следующих утверждений, доказательства которых могут быть получены непосредственно из определения 2.1.

**2.2. Утверждение.** Для любого транзитивного (регулярного) шифра $A$ вида (2.1) имеет место неравенство $|Y| \le |K|$ ($|K| \le |X|$).

**2.3. Утверждение.** Для любого шифра $A = (X, K, Y, f)$ равносильны условия

(а) $A$ – минимальный шифр;

(б) $A$ – транзитивный шифр, удовлетворяющий условию $|K| = |X| = |Y|$.

(в) $A$ – регулярный шифр, удовлетворяющий условию $|K| = |X| = |Y|$.

Ниже при изучении алгебраических моделей шифров используется ряд понятий и результатов, относящихся к теории конечных групп подстановок. Приведем необходимые для дальнейшего сведения о таких группах. Более подробную информацию можно найти, например, в [17, 18].

Обозначим $S(X)$ симметрическую группу конечного множества $X$. Произведение подстановок $g, h \in S(X)$ определяется равенством $hg(x) = h(g(x))$, $x \in X$.

Пусть $G < S(X)$ – группа подстановок на множестве $X$. Множество $\Omega \subseteq X$ называется *инвариантным* относительно группы $G$, если $g(\Omega) \subseteq \Omega$ для всех $g \in G$. Минимальное (по включению) инвариантное множество группы подстановок называется ее *орбитой*. Каждая орбита группы $G$, содержащая точку $x \in X$, имеет вид $\Omega(x) = \{g(x): g \in G\}$. Множество всех различных орбит группы $G$ образует разбиение (дизъюнктное объединение непустых под-

множеств) множества $X$. Группа $G < S(X)$ называется *транзитивной*, если она имеет единственную орбиту $\Omega = X$.

Для любого $x \in X$ обозначим $G_x = \{g \in G: g(x) = x\}$ – *стабилизатор* точки $x$ в группе $G$. Имеет место равество $(G : G_x) = |\Omega(x)|$, где $(G : G_x) = |G|\,|G_x|^{-1}$ – индекс подгруппы $G_x$ в группе $G$. Группа подстановок $G$ называется *полурегулярной*, если для любого $x \in X$ $G_x = \{e\}$, где $e$ – тождественная подстановка на множестве $X$. Транзитивная полурегулярная группа $G$ называется *регулярной*. Известно [17], что порядок полурегулярной группы $G$ является делителем числа $|X|$; при этом $|G| = |X|$ тогда и только тогда, когда $G$ – регулярная группа.

Множество $\Delta \subseteq X$ называется *блоком* группы $G < S(X)$, если для любого $g \in G$ выполняется одно из условий $g(\Delta) = \Delta$, $g(\Delta) \cap \Delta = \varnothing$. При $1 < |\Delta| < |X|$ блок $\Delta$ называют *нетривиальным*. Группа подстановок, не имеющая нетривиальных блоков, называется *примитивной*, а транзитивная группа, не являющаяся примитивной, – *imprимитивной* группой подстановок.

Пусть $G$ – транзитивная группа подстановок, $\Delta$ – блок группы $G$. Совокупность попарно различных множеств вида $g(\Delta)$, $g \in G$ образует разбиение множества $X$, называемое *системой блоков imprимитивности* группы $G$. Следующая теорема о системах блоков imprимитивности конечной группы подстановок существенно используется в п. 4 настоящей статьи.

**2.4. Теорема** [17]. Пусть $G$ – транзитивная на множестве $X$ группа подстановок, $a$ – фиксированный элемент, принадлежащий $X$. Тогда существует сохраняющее отношение включения взаимно однозначное соответствие между системами блоков imprимитивности

$$\varepsilon = \{X_1, X_2, \ldots, X_r\},\ a \in X_1 \tag{2.3}$$

группы $G$ и подгруппами этой группы, содержащими стабилизатор точки $a$. При данном соответствии системе (2.3) соответствует группа

$$H(\varepsilon) = \{g \in G: g(X_1) = X_1\}. \tag{2.4}$$

Обратное соответствие относит произвольной подгруппе $H$ группы $G$, $H > G_a$ систему $\varepsilon = \varepsilon(H)$ вида (2.3) такую, что $X_1 = \{g(a): g \in H\}$ – орбита группы $H$, содержащая точку $a$,

$$X_i = g_i(X_1),\ i \in \overline{1, r}, \tag{2.5}$$

где

$$G = \bigcup_{i=1}^{r} g_i H \tag{2.6}$$

есть разложение группы $G$ в левые смежные классы по подгруппе $H$, $g_1 = e$. При этом имеют место следующие равенства:

$$r = (G : H),\ |X_1| = (H : G_a). \tag{2.7}$$

Определим понятие группы шифра, лежащее в основе предлагаемого ниже подхода к изучению строения алгебраических моделей симметричных криптосистем.

Пусть $A$ – вполне эндоморфный шифр (отождествляемый с двухосновной универсальные алгеброй $A = (X, Y, \Pi(K, f))$.

**2.5. Определение.** Группа подстановок

$$\wp(A) \stackrel{\text{def}}{=} < f_{k_1}^{-1} f_{k_2} : k_1, k_2 \in K >$$

на множестве $X$, порожденная всевозможными произведениями шифрующих преобразований шифра $A$ на обратные к данным преобразования, называется *группой шифра* $A$.

Для любого эндоморфного шифра $A = (X, X, \Pi(K, f))$ имеет место включение

$$\wp(A) \subseteq \wp_0(A), \tag{2.8}$$

где

$$\wp_0(A) \stackrel{\text{def}}{=} < f_k : k \in K >.$$

Из определения 2.5 непосредственно следует, что группы подобных (в смысле [7]) шифров совпадают. Кроме того, на основании (2.8) транзитивность (примитивность) группы эндоморфного шифра $A$ влечет транзитивность (примитивность) группы $\wp_0(A)$.

### 3. Неразложимые и неприводимые шифры

Определим понятие разложения шифра $A = (X, Y, \Pi(K, f))$ в сумму подшифров. Чтобы исключить из рассмотрения ряд вырожденных случаев, далее будем предполагать, что шифр $A$ имеет по крайней мере два неэквивалентных ключа и удовлетворяет условию $|X| = |Y| \geq 2$.

Напомним (см. п. 2), что алгебра $A' = (X', Y', \Pi(K, f'))$ называется *подшифром* шифра $A$, если $X' \subseteq X$ и для любого ключа $k \in K$ отображение $(f')_k: X' \to Y'$ является сужением шифрующего преобразования $f_k$ на множество $X'$.

Пусть заданы разбиения

$$X = \bigcup_{i=1}^{r} X_i, \quad Y = \bigcup_{i=1}^{r} Y_i \tag{3.1}$$

множеств $X$ и $Y$ соответственно такие, что для любого $k \in K$ выполняется условие

$$f_k(X_i) \subseteq Y_i, \; i \in \overline{1, r}. \tag{3.2}$$

В силу (3.1), (3.2) универсальная алгебра $A_i \stackrel{\text{def}}{=} (X_i, Y_i, \Pi(K, f_i))$, где $f_i = f|X_i \times K$ – сужение функции $f$ на множество $X_i \times K$, является вполне эндоморфным подшифром шифра $A$, $i \in \overline{1, r}$. Будем говорить, что *шифр $A$ является суммой подшифров* $A_i$, $i \in \overline{1, r}$ и писать $A = A_1 + \ldots + A_r$, если выполняются соотношения (3.1), (3.2). Шифр, не допускающий разложений в сумму $r$ ($1 < r < |X|$) подшифров, назовем *неразложимым*.

В силу конечности множеств $X$ и $Y$ каждый шифр $A$ раскладывается в сумму неразложимых подшифров. Ниже показано, что такое разложение единственно, и неразложимые подшифры шифра $A$ находятся во взаимно однозначном соответствии с орбитами группы $\wp(A)$.

**3.1. Лемма.** Множество $X' \subseteq X$ является множеством открытых сообщений некоторого подшифра шифра $A$ тогда и только тогда, когда $X'$ инвариантно относительно группы подстановок $\wp(A)$.

***Доказательство.*** Пусть $A' = (X', Y', \Pi(K, f'))$ – подшифр шифра $A$. Тогда для любых $k_1$, $k_2 \in K$ имеем $Y' = f_{k_1}(X') = f_{k_2}(X')$. Следовательно,

$$X' = f_{k_1}^{-1} f_{k_2}(X') \tag{3.3}$$

и множество $X'$ инвариантно относительно группы $\wp(A)$. Обратно, пусть для любых $k_1, k_2 \in K$ выполняется равенство (3.3). Определим алгебру $A' = (X', Y', \Pi(K, f'))$, полагая $Y' = f_k(X')$, $k \in K$ и $f' = f|X' \times K$. В силу (3.3) множество $Y'$ определено корректно (не зависит от ключа $k$); следовательно, $A'$ является подшифром шифра $A$, что и требовалось доказать.

Из леммы 3.1 следует, что совокупность всех подшифров шифра $A$ является полной решеткой [13], изоморфной решетке подмножеств $X' \subseteq X$, инвариантных относительно группы шифра $A$. В частности, минимальные инвариантные подмножества (орбиты группы $\wp(A)$) совпадают с множествами открытых сообщений неразложимых подшифров шифра $A$. Поскольку $X$ является диъюнктным объединением различных орбит группы $\wp(A)$, то шифр $A$ раскладывается в сумму всех своих неразложимых подшифров. Таким образом, справедлива следующая теорема.

**3.2. Теорема.** Пусть $X_1, X_2, \ldots, X_r$ – все различные орбиты группы $\wp(A)$. Положим $Y_i = f(X_i \times K), f_i = f|X_i \times K, A_i = (X_i, Y_i, \Pi(K, f_i))$, $i \in \overline{1, r}$. Тогда справедливо равенство

$$A = A_1 + \ldots + A_r. \qquad (3.4)$$

При этом шифры $A_i$, $i \in \overline{1, r}$ и только они являются неразложимыми подшифрами шифра $A$.

**3.3. Следствие.** Шифр $A$ является неразложимым тогда и только тогда, когда группа $\wp(A)$ транзитивна на множестве $X$. В частности, каждый транзитивный шифр $A$ неразложим.

Итак, в соответствии с теоремой 3.2 изучение строения (вполне эндоморфных) шифров сводится к изучению неразложимых шифров (группы которых транзитивны на множествах открытых сообщений).

Отметим, что описанная выше конструкция разложения шифра в сумму неразложимых подшифров по существу неявно используется в [7] при изучении свойств так называемых "чистых" (групповых) криптосистем. В частности, нетрудно убедиться в том, что "остаточные классы" (в терминологии [7]) открытых сообщений группового шифра $A$ совпадают с орбитами группы $\wp(A)$ (см. ниже п.5). В [8, 10, 11] и ряде других работ проводится анализ теоретико-информационных характеристик криптосистем, представляющих собой по существу суммы регулярных шифров (с равномерными распределениями вероятностей на множествах ключей). Заметим, что поскольку свойство регулярности шифра сохраняется при переходе к подшифрам (см. определение 2.1 (б)), то, согласно теореме 3.2, каждый регулярный шифр однозначно раскладывается в сумму неразложимых регулярных подшифров. С помощью указанного разложения нетрудно получить выражения или оценки параметров, характеризующих теоретическую стойкость регулярного шифра (например, ненадежности ключа), через аналогичные характеристики стойкости его неразложимых подшифров [8, 10].

Рассмотрим подробнее строение эндоморфных шифров. Очевидно, что не каждый подшифр эндоморфного шифра $A = (X, X, \Pi(K, f))$ является эндоморфным. В связи с этим представляет определенный интерес описание таких разложений вида (3.4) эндоморфного шифра $A$, слагаемые $A_i$, $i \in \overline{1, r}$ которых являются эндоморфными шифрами, не допускающими разложений в сумму собственных эндоморфных подшифров.

Назовем шифр $A = (X, X, \Pi(K, f))$ *неприводимым*, если он не может быть разложен в сумму $r$ $(1 < r < |X|)$ эндоморфных подшифров. Аналогично утверждениям леммы 3.1 и теоремы 3.2 доказывается следующая теорема.

**3.4. Теорема.** Решетка эндоморфных подшифров шифра $A = (X, X, \Pi(K, f))$ изоморфна решетке подмножеств множества $X$, инвариантных относительно группы $\wp_0(A)$. В частности, множества открытых сообщений неприводимых подшифров шифра $A$ находятся во взаимно однозначном соответствии с орбитами указанной группы. Каждый эндоморфный шифр $A$

является суммой всех своих неприводимых подшифров. Шифр $A$ неприводим тогда и только тогда, когда $\wp_0(A)$ – транзитивная группа подстановок на множестве $X$.

Итак, согласно полученным результатам при изучении строения эндоморфных шифров можно ограничиться рассмотрением неприводимых шифров. Структурные свойства эндоморфного шифра $A$ в значительной степени определяются свойствами орграфа $\Gamma(K, f)$, соответствующего системе шифрующих преобразований $(f_k: k \in K)$ [6, 14]. По определению $\Gamma(K, f)$ есть орграф с множеством вершин $X$ и множеством $\Gamma$ дуг таких, что упорядоченная пара вершин $(x_1, x_2)$ принадлежит $\Gamma$ в том и только том случае, когда для некоторого ключа $k \in K$ выполняется равенство $x_2 = kx_1$. На основании теоремы 3.4 заключаем, что эндоморфный шифр $A$ является неприводимым тогда и только тогда, когда орграф $\Gamma(K, f)$ сильно связный. При этом, поскольку $\Gamma(K, f)$ является подстановочным орграфом, то его сильная связность равносильна его связности [6]. Отметим, что свойство неразложимости эндоморфного шифра также может быть охарактеризовано в терминах сильной связности мультиграфа, соответствующего некоторой системе преобразований, определяемых данным шифром (см. следствие 3.3 и теорему 4 работы [6]).

Более подробную информацию о строении неприводимого шифра $A$ можно получить, рассматривая его разложение в сумму так называемых циклических классов, аналогичное разбиению множества состояний неразложимой однородной цепи Маркова на циклические подклассы состояний [14].

Обозначим через $d$ наибольший общий делитель длин простых циклов орграфа $\Gamma(K, f)$. Зафиксируем произвольное открытое сообщение $x(0) \in X$, и для каждого $a \in \overline{0, d-1}$ положим $\Psi_a$ равным множеству всех сообщений $x \in X$, для которых длины путей орграфа $\Gamma(K, f)$, ведущих из $x(0)$ в $x$, сравнимы с $a$ по модулю $d$. Нетрудно убедиться в том, что

$$X = \bigcup_{a=0}^{d-1} \Psi_a \tag{3.5}$$

есть разбиение множества $X$, и для любого $k \in K$ выполняется равенство

$$f_k(\Psi_a) = \Psi_{(a+1) \bmod d}, \; a \in \overline{0, d-1}. \tag{3.6}$$

На основании (3.5), (3.6) шифр $A$ раскладывается в сумму $d$ подшифров: $A = C_1 + \ldots + C_d$, где $C_a = (\Psi_a, \Psi_{(a+1) \bmod d}, \Pi(K, f_a))$, $f_a = f \big| \Psi_a \times K$, $a \in \overline{0, d-1}$. Назовем шифры $C_a$, $a \in \overline{0, d-1}$ *циклическими классами* неприводимого шифра $A$.

Суммируем полученные результаты в виде следующей теоремы (см. [13, 14]).

**3.5. Теорема.** Каждый неприводимый шифр $A = (X, X, \Pi(K, f))$ является суммой $d$ подшифров (циклических классов шифра $A$). Число $d$ есть наибольший общий делитель длин простых циклов орграфа $\Gamma(K, f)$, соответствующего системе шифрующих преобразований шифра $A$. Для каждого $a \in \overline{0, d-1}$ множество $\Psi_a$ открытых сообщений циклического класса $C_a$ является блоком группы $\wp_0(A)$. В частности, справедливо равенство $|X| = d|\Psi_a|$, $a \in \overline{0, d-1}$.

**3.6. Следствие.** Каждый неразложимый эндоморфный шифр является неприводимым и ациклическим (обладает единственным циклическим классом).

### 4. Регулярные конгруэнции и сплетения вполне эндоморфных шифров

Естественным обобщением понятия разложения шифра $A$ в сумму подшифров является понятие конгруэнтного разбиения множества открытых сообщений этого шифра. Напомним, что согласно данным в п.2 определениям *конгруэнцией* шифра $A = (X, Y, \Pi(K, f))$ называется

упорядоченная пара $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ отношений эквивалентности соответственно на множествах $X$ и $Y$ такая, что для любых $k \in K$, $x_1, x_2 \in X$ соотношение $x_1 \varepsilon_X x_2$ влечет соотношение $kx_1 \varepsilon_Y kx_2$. Другими словами, $\varepsilon$ есть конгруэнция шифра $A$, если при зашифровании на любом ключе $\varepsilon_X$-эквивалентных открытых сообщений получаются $\varepsilon_Y$-эквивалентные шифрованные сообщения.

Пусть $\varepsilon_X$, $\varepsilon_Y$ – произвольные отношения эквивалентности на множестве открытых и множестве шифрованных сообщений шифра $A$ соответственно. Обозначим через

$$\overline{X} = X/\varepsilon_X = \{X_1, \ldots, X_r\}, \ \overline{Y} = Y/\varepsilon_Y = \{Y_1, \ldots, Y_s\} \tag{4.1}$$

фактор множества по отношениям эквивалентности $\varepsilon_X$, $\varepsilon_Y$. Будем говорить, что разбиение

$$X = \bigcup_{i=1}^{r} X_i \tag{4.2}$$

является *конгруэнтным разбиением* множества $X$, если $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ есть конгруэнция шифра $A$. Произвольную конгруэнцию вида $(\varepsilon_X, \varepsilon_X)$ эндоморфного шифра $A$ (с множеством открытых сообщений $X$) назовем *x-конгруэнцией* этого шифра, а соответствующее разбиение множества $X$ на классы эквивалентности – *x-конгруэнтным разбиением* данного множества.

Простейшим примером конгруэнтного (x-конгруэнтного) разбиения является совокупность множеств открытых сообщений неразложимых (неприводимых) подшифров (эндоморфного) шифра $A$. Конгруэнтным является также разбиение вида (3.5), индуцированное разложением неприводимого шифра $A$ в сумму его циклических классов.

Для любой конгруэнции $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ шифра $A = (X, Y, \Pi(K, f))$ обозначим $A/\varepsilon = (\overline{X}, \overline{Y}, \Pi(K, \overline{f}))$ факторалгебру универсальной алгебры $A$ по конгруэнции $\varepsilon$. Здесь множества $\overline{X}$ и $\overline{Y}$ имеют вид (4.1), а отображения $\overline{f}_k$: $\overline{X} \to \overline{Y}$, $k \in K$ определяются с помощью равенств

$$\overline{f}_k(\overline{x}) = \overline{f_k(x)}, \ \overline{x} \in \overline{X} \tag{4.3}$$

(символами $\overline{x}$, $\overline{f_k(x)}$ обозначены классы эквивалентности по отношениям $\varepsilon_X$, $\varepsilon_Y$, содержащие элементы $x$, $f_k(x)$ соответственно). Алгебра $A/\varepsilon$ называется *гомоморфным образом* алгебры $A$ при каноническом гомоморфизме, индуцированном конгруэнцией $\varepsilon$ (см. [1, 13]).

С практической точки зрения важность понятия конгруэнции (гоморфного образа), а также различных его обобщений [5] связана, прежде всего, с возможностью сокращения трудоемкости алгоритмов решения (или размерности) ряда задач криптоанализа путем применения так называемого "метода гомоморфизмов" (сведения некоторой задачи для данного шифра $A$ к аналогичной задаче для его гомоморфного образа $A/\varepsilon$). Конкретные применения указанного подхода к анализу стойкости симметричных криптосистем (главным образом, блочных шифров) можно найти в работах, цитированных в [1].

Рассмотрим, например, задачу восстановления ключа шифра $A$ по известным открытому сообщению $x \in X$ и шифрованному сообщению $y \in Y$, которая равносильна нахождению решений $k \in K$ уравнения вида (2.2). Если шифр $A$ обладает гомоморфным образом $A/\varepsilon = (\overline{X}, \overline{Y}, \Pi(K, \overline{f}))$ таким, что $1 < |\overline{X}| < |X|$, $1 < |\overline{Y}| < |Y|$, то возможен следующий подход к решению данной задачи [1]. Вначале определим тем или иным способом все ключи $k \in K$, удовлетворяющие уравнению $\overline{f}_k(\overline{x}) = \overline{y}$ над факторалгеброй $A/\varepsilon$, где $\overline{x}$ и $\overline{y}$ – классы эквивалентности по отношениям $\varepsilon_X$ и $\varepsilon_Y$, содержащие элементы $x$ и $y$ соответственно, а затем

выделим из полученного множества ключей искомые решения уравнения (2.2). В [1, 2] (для более общего класса уравнений над многоосновными универсальными алгебрами) получены оценки трудоемкости алгоритмов решения уравнений вида (2.2) на основе описанного подхода, использующего гомоморфизмы алгебры $A$; приведены также условия оптимальности указанных алгоритмов.

Следует отметить, что, как правило, применение "метода гомоморфизмов" приводит к существенному сокращению трудоемкости алгоритмов криптоанализа в тех случаях, когда факторалгебра $A/\varepsilon$ имеет достаточно простое строение, например, является шифром с большим числом эквивалентных ключей.

Нетрудно показать, что факторалгебра $A/\varepsilon$ является шифром тогда и только тогда, когда выполняется равенство

$$|\overline{X}| = |\overline{Y}|. \tag{4.4}$$

Действительно, из определения конгруэнции, биективности отображений $f_k$, $k \in K$ и равенства (4.3) следует, что отображение $\overline{f}_k$ является сюръекцией $\overline{X}$ на $\overline{Y}$. Таким образом, для инъективности $\overline{f}_k$, $k \in K$ необходимо и достаточно выполнение равенства (4.4).

Назовем конгруэнцию $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ шифра $A$ *регулярной*, если факторалгебра $A/\varepsilon$ является шифром, то есть имеет место равенство (4.4). При выполнении условия $1 < |\overline{X}| < |X|$ регулярная конгруэнция $\varepsilon$ называется *нетривиальной*. Следующие утверждения содержат описание всех регулярных конгруэнций (вполне эндоморфных) шифров.

**4.1. Лемма.** Упорядоченная пара отношений эквивалентности $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ ($\varepsilon = (\varepsilon_X, \varepsilon_X)$) является регулярной конгруэнцией ($x$-конгруэнций) (эндоморфного) шифра $A$ тогда и только тогда, когда фактормножество $\overline{X} = \{X_1, \ldots, X_r\}$ является системой блоков импримитивности группы $\wp(A)$ ($\wp_0(A)$), и для некоторого ключа $\kappa \in K$ имеет место равенство

$$\overline{Y} = \{f\kappa(X_1), \ldots, f\kappa(X_r)\}. \tag{4.5}$$

***Доказательство.*** Пусть $A$ – вполне эндоморфный шифр (случай эндоморфного шифра рассматривается аналогично) и $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ – регулярная конгруэнция шифра $A$. В силу (4.4) выполняется равенство (4.5) и для любых $k_1, k_2 \in K$ отображение $(\overline{f}_{k_1})^{-1}\overline{f}_{k_2}$ является подстановкой на множестве $\overline{X}$. Отсюда на основании (4.3) и определения группы $\wp(A)$ получаем, что элементы множества $\overline{X}$ являются блоками импримитивности данной группы.

Наоборот, предположим, что $\overline{X}$ есть система блоков импримитивности группы $\wp(A)$. Покажем, что в этом случае для любого ключа $k \in K$ выполняется равенство

$$\overline{Y} = \{f_k(X_1), \ldots, f_k(X_r)\}. \tag{4.6}$$

Действительно, согласно предположению, для любого $k \in K$ множество $\{(f_k)^{-1}f\kappa(X_1), \ldots, (f_k)^{-1}f\kappa(X_r)\}$ совпадает с множеством $\overline{X}$. Следовательно, в силу (4.5) имеет место равенство (4.6). Таким образом, $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ является регулярной конгруэнцией шифра $A$, что и требовалось доказать.

На основании леммы 4.1, теоремы 2.4 и результатов п. 3 справедлива следующая теорема.

**4.2. Теорема.** Решетка регулярных конгруэнций ($x$-конгруэнций) неразложимого (неприводимого) шифра $A$ изоморфна решетке подгрупп группы $G = \wp(A)$ ($G = \wp_0(A)$), содержащих стабилизатор произвольного фиксированного открытого сообщения $a \in X$. Указанный изоморфизм ставит в соответствие конгруэнции $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ ($x$-конгруэнции $\varepsilon = (\varepsilon_X, \varepsilon_X)$) шифра $A$ группу $H(\varepsilon) = \{g \in G: g(X_1) = X_1\}$, где $X_1$ – класс $\varepsilon_X$-эквивалентности, содержащий элемент $a$.

**4.3. Следствие.** (Эндоморфный) шифр $A$ не имеет нетривиальных ($x$-) конгруэнций в том и только том случае, когда ($\wp_0(A)$) $\wp(A)$ есть примитивная на множестве $X$ группа подстановок.

Итак, согласно теореме 4.2, шифры, изоморфные гомоморфным образам неразложимого шифра $A$, находятся во взаимно однозначном соответствии с подгруппами группы $\wp(A)$, содержащими стабилизатор $G_a$ элемента $a \in X$. В частности, шифр $A$ обладает нетривиальным гомоморфным образом (по некоторой регулярной конгруэнции $\varepsilon$) в том и только том случае, когда группа $\wp(A)$ является импримитивной.

Прежде чем перейти к более подробному изучению строения шифров с импримитивными группами, рассмотрим, с целью иллюстрации полученных выше общих результатов, простой пример эндоморфного шифра с двухэлементным множеством ключей. Определенный интерес к таким шифрам объясняется широким использованием их в качестве "элементарных конструкций" при построении более сложных криптографических систем, например, поточных шифров многоалфавитной замены [6, 19].

**4.4. Пример.** Пусть $\mathfrak{I} = (X, X, \Pi(K, f))$ – эндоморфный шифр с множеством открытых сообщений $X = \{0, 1, \ldots, n-1\}$, где $n \geq 3$, и множеством (неэквивалентных) ключей $K = \{k_1, k_2\}$. Обозначим $f_1$ и $f_2$ шифрующие преобразования шифра $\mathfrak{I}$, отвечающие ключам $k_1$ и $k_2$ соответственно. Имеем $\wp_0(\mathfrak{I}) = \langle f_1, f_2 \rangle$, $\wp(\mathfrak{I}) = \langle h \rangle$, где $h = (f_1)^{-1} f_2$.

Справедливы следующие утверждения:

1) шифр $\mathfrak{I}$ регулярен тогда и только тогда, когда подстановка $h$ не имеет неподвижных точек (определение 2.1 (б));

2) шифр $\mathfrak{I}$ неразложим тогда и только тогда, когда $h$ – полный цикл (следствие 3.3);

3) шифр $\mathfrak{I}$ не имеет нетривиальных регулярных конгруэнций в том и только том случае, когда $h$ – полный цикл и $n$ – простое число (следствие 4.3).

Предположим, что $\mathfrak{I}$ является неразложимым шифром. Сопрягая подходящим образом подстановки $f_1$ и $f_2$, будем считать в дальнейшем, что полный цикл $h$ имеет вид $h = (0, 1, \ldots, n-1)$. По теореме 4.2 регулярные конгруэнции шифра $\mathfrak{I}$ находятся во взаимно однозначном соответствии с подгруппами циклической группы $\langle h \rangle$ порядка $n$. Следовательно, решетка регулярных конгруэнций данного шифра изоморфна решетке делителей числа $n$ (указанный изоморфизм ставит в соответствие делителю $d$ числа $n$ конгруэнтное разбиение множества $X$, состоящее из смежных классов группы $(\boldsymbol{Z}_n, +)$ по подгруппе, порожденной числом $n/d$). В частности, минимальные (по включению) конгруэнции шифра $\mathfrak{I}$ соответствуют простым делителям $n$.

Приведем необходимые и достаточные условия, при которых неразложимый шифр $\mathfrak{I}$ не имеет нетривиальных регулярных $x$-конгруэнций (группа $\wp_0(\mathfrak{I})$ является примитивной). В силу неразложимости $\mathfrak{I}$ полагаем без ограничения общности $h = (0, 1, \ldots, n-1)$. На основании (2.8) каждый блок группы $\wp_0(\mathfrak{I})$ совпадает с одним из блоков группы $\wp(\mathfrak{I}) = \langle h \rangle$, то есть является смежным классом группы $(\boldsymbol{Z}_n, +)$ по подгруппе, порожденной некоторым делителем числа $n$. Отсюда следует, что шифр $\mathfrak{I}$ не имеет нетривиальных регулярных $x$-конгруэнций тогда и только тогда, когда для каждого собственного делителя $d$ числа $n$ хотя бы одна из подстановок $f_1$, $f_2$ не сохраняет отношение сравнимости по модулю $d$. Отметим, что последнее условие допускает простую практическую проверку.

Рассмотрим подробнее строение шифра $A = (X, Y, \Pi(K, f))$ с импримитивной группой $\wp(A)$. С этой целью определим понятие сплетения шифров, являющееся по существу част-

ным случаем понятия δ-сплетения многоосновной универсальной алгебры (с одной операцией) с семейством универсальных алгебр [2].

Пусть $A_1 = (X_1, K_1, Y_1, \chi)$ и $A_2 = (X_2, K_2, Y_2, h)$ – произвольные шифры (шифр $A_2$ в общем случае не предполагается вполне эндоморфным) и $\delta: X_2 \times K_2 \to K_1$ – некоторое отображение.

**4.5. Определение.** *Сплетением* $A_1 \mathrm{Wr}_\delta A_2$ (или δ-*сплетением*) *шифра* $A_2$ *с шифром* $A_1$ называется шифр

$$A = (X_1 \times X_2, K, Y_1 \times Y_2, f) \tag{4.7}$$

с множеством ключей $K = K_2$ и функцией шифрования $f$ вида

$$f_k(x_1, x_2) = (\chi(x_1, \delta(x_2, k)), h_k(x_2)), \quad k \in K, x_1 \in X_1, x_2 \in X_2. \tag{4.8}$$

Используя равенство (4.8), нетрудно путем непосредственной проверки убедиться в том, что $f$ является сюръекцией из $X_1 \times X_2$ в $Y_1 \times Y_2$, и каждое отображение $f_k$, $k \in K$ инъективно. Таким образом, универсальная алгебра $A$ вида (4.7) действительно является шифром. Если при этом шифр $A_2$ – вполне эндоморфный (что в дальнейшем предполагается выполненным), то вполне эндоморфным является также шифр $A = A_1 \mathrm{Wr}_\delta A_2$.

Существует тесная связь (известная и для более широкого класса универсальных алгебр [2]) между конгруэнциями шифра $A$ и его представлениями в виде сплетения шифров, связанных с данными конгруэнциями.

Пусть, например, $A$ является сплетением шифров: $A = A_1 \mathrm{Wr}_\delta A_2$. Определим отношения $\varepsilon_X$, $\varepsilon_Y$ на множествах $X = X_1 \times X_2$ и $Y = Y_1 \times Y_2$ соответственно, полагая

$$(x_1, x_2)\varepsilon_X(x_1', x_2') \Leftrightarrow x_2 = x_2', \ (y_1, y_2)\varepsilon_Y(y_1', y_2') \Leftrightarrow y_2 = y_2', \tag{4.9}$$

где $(x_1, x_2), (x_1', x_2') \in X, (y_1, y_2), (y_1', y_2') \in Y$. Непосредственно из равенств (4.8), (4.9) следует, что $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ есть регулярная конгруэнция шифра $A$ такая, что факторалгебра $A/\varepsilon$ изоморфна шифру $A_2$.

Итак, каждый шифр $A$, изоморфный сплетению $A_1 \mathrm{Wr}_\delta A_2$, обладает регулярной конгруэнцией ε, удовлетворяющей условию $A/\varepsilon \cong A_2$. Следующая теорема, уточняющая (для случая шифров) один из результатов работы [2], показывает, что справедливо также обратное утверждение.

**4.6. Теорема.** Пусть $A = (X, Y, \Pi(K, f))$ – неразложимый шифр, $\varepsilon = (\varepsilon_X, \varepsilon_Y)$ – регулярная конгруэнция шифра $A$ и $A/\varepsilon = (\overline{X}, \overline{Y}, \Pi(K, \overline{f}))$ – факторалгебра универсальной алгебры $A$ по конгруэнции ε. Тогда существует шифр

$$A_1 = (X_1, \overline{X} \times K, Y_1, \chi) \tag{4.10}$$

такой, что $A \cong A_1 \mathrm{Wr}_\delta(A/\varepsilon)$, где $\delta: \overline{X} \times K \to \overline{X} \times K$ есть тождественное отображение.

***Доказательство.*** Пусть $X = \bigcup_{i=1}^{r} X_i$, $Y = \bigcup_{i=1}^{r} Y_i$ – разбиения множеств открытых и шифрованных сообщений шифра $A$ на классы эквивалентности по отношениям $\varepsilon_X$, $\varepsilon_Y$ соответственно. Обозначим $G$ группу шифра $A$ и положим $H = \{g \in G: g(X_1) = X_1\}$. Рассмотрим

разложение $G = \bigcup_{i=1}^{r} g_i H$ группы $G$ в левые смежные классы по подгруппе $H$ такое, что $X_i = g_i(X_1)$, $i \in \overline{1,r}$, где $g_1$ – тождественная подстановка на множестве $X$ (см. теорему 2.4). Зафиксируем ключ $\kappa \in K$, удовлетворяющий условию $f\kappa(X_i) = Y_i$, $i \in \overline{1,r}$ (лемма 4.1); имеем

$$f\kappa\, g_i\, (f\kappa)^{-1}(Y_1) = Y_i,\ i \in \overline{1,r}. \tag{4.11}$$

Для любых $j \in \overline{1,r}$, $k \in K$ обозначим $\varphi_{kj}$ ограничение функции $f_k$ на множество $X_j$.

Рассмотрим шифр $A_1$ вида (4.10), функция шифрования $\chi$: $X_1 \times (\overline{X} \times K) \to Y_1$ которого определяется равенством

$$\chi(x_1, (X_j, k)) = (f\kappa\, (g_i)^{-1} (f\kappa)^{-1} \varphi_{kj}\, g_j)\, (x_1),\ x_1 \in X_1 \tag{4.12}$$

при условии

$$\varphi_{kj}(X_j) = f_k\, (X_j) = Y_i, \tag{4.13}$$

где $i \in \overline{1,r}$. Очевидно, что в силу равенств (4.11) – (4.13) частичные функции $\chi_{(X_j,k)}$ отображения $\chi$ определены корректно и являются биекциями из $X_1$ в $Y_1$.

Пусть $A' = A_1 \mathrm{Wr}_{\delta}(A/\varepsilon)$ – сплетение шифра $A/\varepsilon$ с шифром $A_1$, где $\delta$ – тождественное отображение множества $\overline{X} \times K$. Согласно определению 4.5 имеем

$$A' = (X_1 \times \overline{X}\,, K, Y_1 \times \overline{Y}\,, f'),$$

где

$$f'_k(x_1, X_j) = (\chi_{(X_j,k)}(x_1),\ \overline{f}_k(X_j)),\ k \in K, x_1 \in X_1, X_j \in \overline{X}\,. \tag{4.14}$$

Покажем, что шифр $A$ изоморфен построенному шифру $A'$. Для этого определим отображения $\psi_X$: $X \to X_1 \times \overline{X}$, $\psi_Y$: $Y \to Y_1 \times \overline{Y}$, полагая

$$\psi_X(x) = (x_1,\ \overline{x}\,), x \in X,\ \psi_Y(y) = (y_1,\ \overline{y}\,), y \in Y, \tag{4.15}$$

где $x_1 \in X_1$, $y_1 \in Y_1$, $\overline{x}$ и $\overline{y}$ – классы эквивалентности по отношениям $\varepsilon_X$ и $\varepsilon_Y$, содержащие элементы $x$ и $y$ соответственно, причем элементы $x_1$, $y_1$ определяются следующим образом:

$$x_1 = (g_j)^{-1}(x),\ y_1 = (f\kappa (g_i)^{-1} (f\kappa)^{-1})(y), \tag{4.16}$$

если $x \in X_j$, $y \in Y_i$, где $i, j \in \overline{1,r}$.

Из равенств (4.15), (4.16) следует, что $\psi_X$ и $\psi_Y$ – биективные отображения. Наконец, с использованием соотношений (4.12) – (4.16) нетрудно непосредственно убедиться в том, что $\psi = (\psi_X, \psi_Y)$ есть гомоморфизм алгебры $A$ в алгебру $A'$.

Теорема доказана.

Конструкция сплетения шифров дает возможность изучать структурные свойства алгебраических моделей симметричных криптосистем путем сведения ряда задач анализа данного

шифра $A$ с импримитивной группой $\wp(A)$ к аналогичным задачам анализа шифров $A/\varepsilon$ и $A_1$, имеющих, как правило, более простое строение. В связи с этим представляет интерес описание групп указанных шифров, характеризующих в определенной степени их структурные свойства.

Рассмотрим вначале группу $\wp(A/\varepsilon)$ шифра $A/\varepsilon$. (В дальнейшем без специальных оговорок используются обозначения, введенные при доказательстве теоремы 4.6). Пусть $\overline{G}$ – гомоморфный образ группы $G = \wp(A)$ при гомоморфизме $\theta$: $G \to S(\overline{X})$, определяемом соотношениями

$$\theta(g) = \overline{g},\ g \in G,\ \ \overline{g}(\overline{x}) = \overline{g(x)},\ \overline{x} \in \overline{X}, \tag{4.17}$$

где $\overline{x}$ и $\overline{g(x)}$ – классы $\varepsilon_X$-эквивалентности, содержащие элементы $x$ и $g(x)$ соответственно. На основании (4.3), (4.17) заключаем, что группа $\wp(A/\varepsilon)$, порожденная подстановками $(\overline{f}_{k_1})^{-1}\overline{f}_{k_2}$, $k_1, k_2 \in K$, совпадает с группой $\overline{G}$.

Отметим, что поскольку $\overline{G}$ является транзитивной группой (в силу транзитивности группы $G$), то согласно следствию 3.3 шифр $A/\varepsilon$ неразложим. При этом (в соответствии с теоремой 4.2) указанный шифр не имеет регулярных конгруэнций тогда и только тогда, когда $\varepsilon$ является максимальной (по включению) регулярной конгруэнцией шифра $A$, то есть подгруппа $H = \{g \in G: g(X_1) = X_1\}$ группы $G$ максимальна в классе подгрупп, содержащих стабилизатор некоторого фиксированного элемента $a \in X_1$.

Опишем теперь строение группы шифра $A_1$, определяемого соотношениями (4.10), (4.12). Введем в рассмотрение группу подстановок $F$ на множестве $X_1$, полагая $F = \{g|X_1: g \in H\}$, и покажем, что

$$\wp(A_1) = F. \tag{4.18}$$

Рассмотрим систему образующих группы шифра $A_1$, соответствующую системе его шифрующих преобразований вида (4.12). Пусть $k(1), k(2) \in K, j(1), j(2), i(1), \quad i(2) \in \overline{1,r}$ и

$$f_{k(1)}(X_{j(1)}) = Y_{i(1)},\ f_{k(2)}(X_{j(2)}) = Y_{i(2)}. \tag{4.19}$$

В силу (4.12) имеем

$$\begin{aligned}(\chi_{(X_{j(2)},k(2))})^{-1}\chi_{(X_{j(1)},k(1))} &= (g_{j(2)})^{-1}(\varphi_{k(2),j(2)})^{-1} f\kappa\, g_{i(2)}\times \\ &\times (g_{i(1)})^{-1}(f\kappa)^{-1}\varphi_{k(1),j(1)}\, g_{j(1)}\end{aligned} \tag{4.20}$$

Из равенства (4.20), в частности, следует, что при $i(1) = i(2)$

$$(\chi_{(X_{j(2)},k(2))})^{-1}\chi_{(X_{j(1)},k(1))} = (g_{j(2)})^{-1}(\varphi_{k(2),j(2)})^{-1}\varphi_{k(1),j(1)}\, g_{j(1)}. \tag{4.21}$$

Для доказательства соотношения

$$\wp(A_1) \subseteq F \tag{4.22}$$

рассмотрим подстановку $g = (g_{j(2)})^{-1}((f_{k(2)})^{-1}f\kappa)\, g_{i(2)}\, (g_{i(1)})^{-1}((f\kappa)^{-1}f_{k(1)})\, g_{j(1)}$. Очевидно, что $g \in G$ и, следовательно, $g \in H$ в силу (4.19). Кроме того, на основании (4.20) имеет место равенство $(\chi_{(X_{j(2)},k(2))})^{-1}\chi_{(X_{j(1)},k(1))} = g|X_1$. Таким образом, все образующие группы $\wp(A_1)$ принадлежат группе $F$, откуда и следует включение (4.22).

Докажем теперь справедливость соотношения

$$\wp(A_1) \supseteq F. \tag{4.23}$$

Пусть $g = \prod_{s=1}^{t} g^{(2s,2s-1)}$ – произвольный элемент группы $H$, представленный в виде произведения $t$ образующих $g^{(2s,\,2s-1)} \stackrel{\text{def}}{=} f_{k_{2s}}^{-1} f_{k_{2s-1}}$ группы $G$, где $k_{2s}, k_{2s-1} \in K$, $s \in \overline{1,t}$. Определим числа $j(1) = 1, j(2), \ldots, j(t+1) \in \overline{1,r}$ с помощью равенств

$$g^{(2s,\,2s-1)}(X_{j(s)}) = (X_{j(s+1)}),\ s \in \overline{1,t}. \tag{4.24}$$

Заметим, что поскольку $g \in H$, то $g(X_1) = X_1$ и, следовательно, $j(t+1) = 1$.

Покажем, что выполняется равенство

$$g|X_1 = \prod_{s=1}^{t}((\chi_{(X_{j(s+1)},k_{2s})})^{-1}\chi_{(X_{j(s)},k_{2s-1})}), \tag{4.25}$$

из которого непосредственно вытекает включение (4.23). На основании (4.24) имеем

$$f_{k_{2s}}(X_{j(s+1)}) = f_{k_{2s-1}}(X_{j(s)}),\ s \in \overline{1,t}, \tag{4.26}$$

откуда в силу (4.20), (4.21) следует, что

$$(\chi_{(X_{j(s+1)},k_{2s})})^{-1}\chi_{(X_{j(s)},k(2s+1))} = (g_{j(s+1)})^{-1}(\varphi_{k_{2s},j(s+1)})^{-1}\varphi_{k_{2s-1},j(s)}\, g_{j(s)}. \tag{4.27}$$

На основании полученных соотношений справедлива следующая цепочка равенств:

$$\prod_{s=1}^{t}((\chi_{(X_{j(s+1)},k_{2s})})^{-1}\chi_{(X_{j(s)},k_{2s-1})}) = \prod_{s=1}^{t}((g_{j(s+1)})^{-1}((\varphi_{k_{2s},j(s+1)})^{-1}\varphi_{k_{2s-1},j(s)}\, g_{j(s)}) =$$

$$= \prod_{s=1}^{t}((\varphi_{k_{2s},j(s+1)})^{-1}\varphi_{k_{2s-1},j(s)}) = \prod_{s=1}^{t}(((f_{k_{2s}})^{-1}f_{k_{2s-1}})|X_{j(s)}) = \prod_{s=1}^{t}(g^{(2s,2s-1)}|X_{j(s)}) =$$

$$= (\prod_{s=1}^{t} g^{(2s,2s-1)})|X_1 = g|X_1.$$

Таким образом, выполняется равенство (4.25), что и требовалось доказать.

Из равенства (4.18), следствий 3.3, 4.3 и известного критерия примитивности группы подстановок $F = \{g \mid X_1 : g \in H\}$ [17] непосредственно получаем следующее утверждение.

**4.7. Следствие.** В условиях теоремы 4.6 шифр $A_1$ является неразложимым. При этом указанный шифр не имеет нетривиальных регулярных конгруэнций тогда и только тогда, когда подгруппа $H = \{g \in G: g(X_1) = X_1\}$ группы $G = \wp(A)$ является минимальной (по включению) среди подгрупп, содержащих стабилизатор фиксированного элемента $a \in X_1$.

Более содержательную информацию о строении шифра $A$ с импримитивной группой можно получить, накладывая разнообразные дополнительные ограничения на подгруппы группы $\wp(A)$, содержащие стабилизатор $G_a$ (см., например, [17]).

## 5. Групповые шифры

Достаточно наглядной иллюстрацией установленной выше тесной связи между строением алгебраической модели шифра и свойствами его группы служит класс групповых ("чистых") шифров, введенных в рассмотрение К. Шенноном [7].

Следуя основной идее определения, предложенного в [7], назовем шифр $A = (X, Y, \Pi(K, f))$, не содержащий эквивалентных ключей, *групповым*, если для любых $k_1, k_2, k_3 \in K$ существует ключ $k \in K$ такой, что

$$k_1 (k_2)^{-1} k_3 = k \tag{5.1}$$

(здесь и далее для сокращения записи используем обозначение $k$ вместо $f_k$, отождествляя ключи шифра $A$ с соответствующими им шифрующими преобразованиями).

Следующая теорема, доказанная, по сути дела, в [7], содержит описание групповых шифров, полнее раскрывающее роль введенного выше понятия группы шифра при изучении строения произвольных вполне эндоморфных шифров.

**5.1. Теорема.** Шифр $A = (X, Y, \Pi(K, f))$ является групповым тогда и только тогда, когда множество $\Pi(K, f)$ его шифрующих преобразований имеет вид

$$\Pi(K, f) = \kappa \wp(A), \tag{5.2}$$

где $\kappa$ – некоторый (равносильно – произвольный) ключ шифра $A$.

Итак, шифрующие преобразования группового шифра образуют "смежный класс" по его группе, что позволяет непосредственно выразить основные свойства групповых шифров в терминах соответствующих свойств конечных групп подстановок.

С использованием теоремы 5.1, а также определений и результатов, изложенных в пп. 2 – 4, нетрудно убедиться в справедливости следующих утверждений.

**5.2. Теорема.** Пусть $A = (X, Y, \Pi(K, f))$ – групповой шифр, $A = A_1 + \ldots + A_r$ – его разложение в сумму неразложимых подшифров $A_i = (X_i, Y_i, \Pi(K, f_i))$, $i \in \overline{1, r}$. Тогда справедливы следующие утверждения:

1) для любого $i \in \overline{1, r}$ число $|X_i| = |Y_i|$ является делителем числа $|K|$, и для любых $x \in X_i$, $y \in Y_i$ существует ровно $|K| \, |X_i|^{-1}$ ключей $k \in K$, удовлетворяющих условию $kx = y$ [7];

2) шифр $A$ является регулярным в том и только том случае, когда $|X_i| = |K|$, $i \in \overline{1, r}$;

3) шифр $A$ является регулярным тогда и только тогда, когда $\wp(A)$ есть полурегулярная группа подстановок;

4) шифр $A$ является неразложимым тогда и только тогда, когда он транзитивен.

**5.3. Следствие.** Каждый регулярный групповой шифр однозначно раскладывается в сумму групповых шифров гаммирования.

**5.4. Следствие.** Групповой шифр гаммирования не имеет нетривиальных регулярных конгруэнций тогда и только тогда, когда он подобен шифру модульного гаммирования (шифру Виженера) в алфавите $\boldsymbol{Z}_p = \{0, 1, \ldots, p-1\}$, где $p$ – простое число.

В заключение докажем теорему, содержащую необходимые и достаточные условия, при которых шифром гаммирования (с точностью до эквивалентных ключей) является факторалгебра группового неразложимого шифра. С практической точки зрения данный результат может представлять интерес, например, при решении задачи восстановления ключа по известным открытому и шифрованному сообщениям с использованием "метода гомоморфизмов" (см. п. 4).

**5.5. Теорема.** Пусть $A = (X, Y, \Pi(K, f))$ – неразложимый групповой шифр, $\varepsilon$ – нетривиальная регулярная конгруэнция шифра $A$, $H = H(\varepsilon)$ – соответствующая ей подгруппа группы $G = \wp(A)$ (см. теорему 4.2). Тогда приведенная форма шифра $A/\varepsilon$ является шифром гаммирования в том и только том случае, когда $H$ есть нормальный делитель группы $G$.

***Доказательство.*** Поскольку $A$ является транзитивным шифром (теорема 5.2, 4)), то шифр $A/\varepsilon$ также транзитивен и, следовательно, (см. утверждение 2.2) число $|\overline{K}|$ неэквивалентных ключей этого шифра удовлетворяет неравенству

$$|\overline{K}| \geq |\overline{X}|. \tag{5.3}$$

Рассмотрим гомоморфизм групп $\theta$: $G \to \overline{G}$, определенный равенством (4.17). Очевидно, что $\mathrm{Ker}\theta = \{g \in G: g(X_i) = X_i,\ i \in \overline{1, r}\}$, где $\mathrm{Ker}\theta$ – ядро гомоморфизма $\theta$. В частности, из определения группы $H$ следует, что

$$\mathrm{Ker}\theta \subseteq H. \tag{5.4}$$

Согласно [17], подгруппа $H$ является нормальным делителем группы $G$ в том и только том случае, когда выполняется равенство

$$\mathrm{Ker}\theta = H. \tag{5.5}$$

Таким образом, на основании утверждения 2.3 для доказательства теоремы достаточно показать, что неравенство (5.3) обращается в равенство тогда и только тогда, когда имеет место (5.5).

Пусть $k_1 = \kappa g_1$, $k_2 = \kappa g_2 \in K$ – произвольные ключи шифра $A/\varepsilon$, где $g_1, g_2 \in G$ (см. (5.2)). Обозначим $\overline{f_1}$ и $\overline{f_2}$ соответственно шифрующие преобразования шифра $A/\varepsilon$, отвечающие ключам $k_1$ и $k_2$. На основании (4.3) равенство $\overline{f_1} = \overline{f_2}$ равносильно соотношениям $\kappa g_1(X_i) = \kappa g_2(X_i)$, $i \in \overline{1, r}$, которые, в свою очередь, равносильны включению $(g_1)^{-1} g_2 \in \mathrm{Ker}\theta$. Таким образом, ключи $k_1$ и $k_2$ шифра $A/\varepsilon$ эквивалентны тогда и только тогда, когда соответствующие им элементы $g_1$ и $g_2$ группы $G$ принадлежат одному смежному классу по подгруппе $\mathrm{Ker}\theta$. Отсюда следует, что

$$|\overline{K}| = (G : \mathrm{Ker}\theta). \tag{5.6}$$

С другой стороны, согласно первому равенству (2.7), имеем

$$|\overline{X}| = (G : H). \tag{5.7}$$

На основании (5.4), (5.6) и (5.7) заключаем, что соотношение $|\overline{K}| = |\overline{X}|$ выполняется в том и только том случае, когда имеет место равенство (5.5).

Теорема доказана.

**Список литературы:** 1. *Горчинский Ю.Н.* О гомоморфизмах многоосновных универсальных алгебр всвязи с криптографическими применениями // Труды по дискретной математике. Т.1. М.: ТВП, 1997. С.67-84. 2. *Шапошников И.Г.* О конгруэнциях конечных многоосновных универсальных алгебр // Дискретная математика. 1999. Т.11. В.3. С.48-62. 3. *Артамонов В.А., Ященко В.В.* Многоосновные алгебры в системах открытого шифрования // Успехи матем. наук. 1994. Т.49. С.149-150. 4. *Горчинский Ю.Н.* Стохастические алгебры // Труды по дискретной математике. Т.2. М.: ТВП, 1998. С.55-87. 5. *Горчинский Ю.Н.* О π-гомоморфизмах конечных многоосновных универсальных алгебр // Дискретная математика. 1999. Т.11. В.2. С.3-19. 6. *Сачков В.Н.* Вероятностные преобразователи и правильные мультиграфы. 1 // Труды по дискретной математике. Т.1. М.: ТВП, 1997. С.227-250. 7. *Шеннон К.* Теория связи в секретных системах. Работы по теории информации и кибернетике. М.: ИЛ, 1963. С. 333- 402. 8. *Ahlswede R.* Remarks on Shannon secret systems // Probl. Control and Inform. Theory. 1982. V.11. №4. P. 301-318. 9. *Massey J.L.* An Introduction to Contemporary Cryptology // Proc. IEEE. 1988. V.76. №5. P.533-549. 10. *Штарьков Ю.М.* Некоторые теоретико-информационные задачи защиты дискретных данных // Проблемы передачи информации. 1994. Т.30. В.2. С.49-60. 11. *Штарьков Ю.М., Юхансон Т., Смитс Б.Дж.М.* О совместной стойкости защиты информации и ключа в секретных системах // Проблемы передачи информации. 1998. Т.34. В.2. С.117-127. 12. *Maurer U.M.* Provable Security in Cryptography: Diss. ETH №9260. 1990. 120p. 13. *Богомолов А.М., Салий В.Н.* Алгебраические основы теории дискретных систем. М.:Физматлит, 1997. 367с. 14. *Сачков В.Н., Тараканов В.Е.* Комбинаторика неотрицательных матриц. М.: ТВП, 2000. 447с. 15. *Алексейчук А.Н.* Математическая модель и задачи анализа стойкости вероятностно-криптографических систем в системах защиты информации // Захист інформації. 2001. №3. С. 5-12. 16. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии. М.: Гелиос АРВ, 2001. 479с. 17. *Супруненко Д.А.* Группы матриц. М.: Наука, 1972. 351с. 18. *Wielandt H.* Finite permutation groups. N-Y. - London, 1964. 116p. 19. *Глухов М.М.* О числовых параметрах, связанных с заданием конечных групп системами образующих элементов // Труды по дискретной математике. Т.1. М.: ТВП, 1997. С.43-66.

*Военный институт телекоммуникаций и информации*
*Национальный технический университет «КПИ»*

*Поступила в редколлегию 17.04.2002*

УДК 681.3.06:519.248.681

*А.А. ЗАМУЛА, канд. техн. наук, Ю.С. ПАВЛЕНКО*

# ВОЗМОЖНОСТЬ ПРИМЕНЕНИЯ СОВРЕМЕННЫХ МЕТОДОВ СЖАТИЯ ИНФОРМАЦИИ ДЛЯ СЖАТИЯ РЕЧИ

## Введение

В современных сетях существенная часть пропускной способности каналов используется для передачи данных и речи. При этом потребности передачи речи значительны. Поэтому важным является эффективное использование полосы пропускания используемые речевыми сигналами. На сегодня передача одной секунды звука с качеством телефонных переговоров требует чуть меньше килобайта данных.

Для обеспечения большей эффективности обмена речевыми сообщениями можно предложить следующие методы:

1) Дальнейшее снижение требований к пропускной способности каналов связи при сохранении достигнутого на данный момент качества.

2) Достижение качества не ниже уровня аналоговых телефонных каналов при сохранении необходимой скорости передачи.

Целью настоящей статьи является проведение сравнительного анализа методов сжатия речи и определение наиболее перспективных из них с целью применения в IP-телефонии.

## 1. Анализ методов сжатия речи

Частотный диапазон человеческого голоса равен примерно 20-20000 Гц и может значительно отличаться у отдельных людей. Исследования показали, что без существенных потерь диапазон может быть уменьшен до 100-6000 Гц. Но в самом начале строительства телекоммуникаций предоставить такую полосу частот каждому абоненту было невозможно, и пришлось ради удешевления услуг связи пойти на большое сокращение частотного диапазона до полосы 200-3400 Гц. Это привело к ухудшению разборчивости, но она осталась все же на достаточном уровне. В этой полосе частота выборок согласно теореме Котельникова должна составить не менее 8 кГц. Если при этом использовать 8-разрядный код, то необходимая пропускная способность должна составить не менее 64 кбит/с. Обзор современных методов сжатия речи, а также их сравнение было проведено в [1]. За последнее время не было произведено радикальных изменений и бесспорным лидером остался стандарт G.723.1 MP-MLQ. Этот алгоритм является совместной разработкой фирм Audiocodec (Израиль) и DSP Group (США) - кодек с линейным предсказанием и скоростью выходного потока информации 6,3 кбит/с[2]. Несмотря на значительно большую, чем у АДИКМ среднеквадратичную погрешность синтеза речи, алгоритмом получена лучшая (3,9 балла MOS) оценка качества, чем у АДИКМ. Это достигнуто благодаря двум усовершенствованиям алгоритма линейного предсказания алгебраического CELP. Первое: длина сглаживающего окна трехкратно увеличена относительно длины анализируемого сегмента речи, что ослабило искажения, вносимые асинхронностью анализируемых сегментов речи и интервалов основного тона. Второе: метод более точного формирования сигнала возбуждения синтезирующего фильтра, получивший название MP-MLQ (Multipulse Maximum Likelihood Quantisation). Еще одним важным преимуществом этого алгоритма является более надежное, по сравнению с известными аналогами вычисление параметров основного тона голоса. Алгоритм MP-MLQ относится к семейству analysis-by-synthesis алгоритмов. В кодере, построенном на MP-MLQ, используется линейное прогнозирование 10-го порядка, и он может работать на скоростях 4,8; 6,4; 7,2 и 8,0 кбит/с. Гибкость алгоритма дает возможность выполнять адаптацию в ходе разговора при ухудшении связи, а также снизить скорость до 4.0 кбит/с. Для сравнения в табл. 1. приведены характеристики некоторых других алгоритмов.

**Таблица 1**

| Скорость передачи, кбит/с | Субъективное качество | Название стандарта | Год выпуска | Алгоритм | Область применения |
|---|---|---|---|---|---|
| 64 | 4,1 | ITU-T G.711 | 1960 | PCM | Телефонные сети |
| 32 | 3,8 | ITU-T G.726 | 1984 | ADPCM | Телефонные сети |
| 6,4 | 3,1 | INMAR-SAT-M | 1990 | IMBE | Спутниковая телефония |
| 13 | 3,3 | ETSI GSM | 1992 | RPE-LTP | Сотовая телефония (Европа) |
| 16 | 3,6 | ITU-T G.728 | 1992 | LD-CELP | Телефонные сети |
| 4,8 | 3,4 | ETSI TETRA | 1996 | ACELP | Сотовая телефония (Европа) |
| 6,3 | 3,9 | ITU-T G.723.1 | 1996 | MP-MLQ | Телефонные сети |
| 5,3 | 3,7 | ITU-T G.723.1 | 1996 | ACELP | Телефонные сети |
| 8 | 3,9 | ITU-T G.729 | 1997 | CS-ACELP | Телефонные сети |
| 2,4 | 3,5 | США (проект) | 1998 | MELP | Мин. Обороны США |

Однако, несмотря на то, что прошло достаточно времени с момента утверждения стандарта G.723.1, и на доступность его реализаций, он не имеет той популярности, какую имеет электронная почта. Частично это можно отнести на счет необходимости микрофона (его нужно купить дополнительно) в то время как клавиатура есть у каждого компьютера. Но это не убедительный факт, так как стоимость достаточной для этих целей гарнитуры не превышает стоимости клавиатуры и существенно не удорожает общую стоимость системы. Очевидно, решающим фактором является значительно большая потребность в пропускной способности. По заявленным на данный момент разработкам существенного уменьшения необходимой пропускной способности не предвидется, если не считать разрабатываемого Министерством обороны США стандарта MELP с заявленной скоростью 2,4 кбит/с.

Проведенный анализ показывает, что основные разработки по сжатию речи на протяжении десятков лет ведутся в двух направлениях:

1) Разложение в ряд Фурье и применение психоакустической модели для отфильтрования невоспринимаемой человеком информации (MPEG).
2) Использование линейного предсказания речи (G.723.1 MP-MLQ).

При этом коэффициент сжатия информации не превышает 20-30. В то же время алгоритмы сжатия графической информации обеспечивают сжатие до 350 раз. Это дает надежду на разработку эффективных алгоритмов сжатия и речи.

## 2. Перспективные методы сжатия речи

Из новых методов сжатия графической информации можно выделить два: с использованием фракталов и с использованием вейвлетов.

Рождение фрактальной геометрии связывают с именем бельгийского математика Мандельброта. В 1977 году он опубликовал книгу «The Fractal Geometry of Nature». Вскоре математики освоили методы построения фрактальных изображений и перешли к вопросу создания системы функций для восстановления заданного изображения. В 1988 году в журнале BYTE Бансли опубликовал статью, в которой приводилась система итерированных функций и получаемое по ним графическое изображение. Коэффициент сжатия при этом составил 10000. Однако, как оказалось, данный алгоритм требовал присутствия человека, а сжатие длилось 100 часов. Полная автоматизация этого процесса снижает коэффициент сжатия до 100.

Преимуществом фрактального метода сжатия является то, что этот метод не анализирует каждый пиксел изображения, поэтому получается лучшее качество изображений любого разрешения. Природа фрактальных преобразований такова, что данные, теряемые при кодировке, наименее существенны для визуального восприятия, например, тонкие детали в текстурах. В отличие от других методов, фрактальное сжатие позволяет контролировать теряемые данные. Наиболее известным из фрактальных форматов файлов является – FIF (Fractal Image Format).

Задача сжатия изображения целиком не реализована и поэтому ее решают поэтапно. В качестве примера можно привести два вида разбиения: Quadtree-разбиение и HV-разбиение.

При Quadtree-разбиении целевой квадратный участок делится на четыре подквадрата в том случае, если он плохо описывается. Процесс продолжается рекурсивно до тех пор, пока ошибка не станет меньше заданной. Слабость метода Quadtree-разбиения в том, что при его реализации нет возможности выбрать исходные элементы контекстно-независимо. Для достижения хороших результатов количество вариантов должно быть очень большим. HV-метод позволяет уменьшить их количество, увеличивая гибкость разбиения. На каждом шаге рассматриваемая прямоугольная область делится на две части вертикальной или горизонтальной линией. Гибкость такой системы выражается в том, что линия разбиения может быть выбрана произвольным образом.

К основным недостаткам фрактальных методов при их применении для сжатия речи можно отнести необходимость накопления больших объемов информации, что приводит к многосекундным задержкам. Это противоречие может быть разрешено с помощью некоторой базы фракталов, аналогичной базе фонем для вокодера.

Другим современным способом сжатия информации является сжатие с использованием вейвлетов[3]. Вейвлеты и вейвлетное преобразование появились как развитие Фурье-преобразования для решения парадокса всегда существующих волн. Вейвлеты так же, как синусы и косинусы, относятся к ортогональным функциям. В остальном вейвлеты отличаются от них. Во-первых, вейвлеты конечны, во-вторых, вместо амплитудного масштабирования в вейвлет-преобразовании используется временная «растяжка», в-третьих, операция сложения в Фурье-преобразовании заменяется временными сдвигами и масштабированием.

Вейвлет-анализ определяется выражением:

$$\gamma(s,\tau) = \int f(t)\Psi^*_{s,\tau}(t)dt\,, \tag{1}$$

где $f(t)$ – исходный сигнал, а $\Psi_{s,\tau}(t)$ – базовая функция, называемая вейвлетом. Вейвлеты получают от одного основного вейвлета $\Psi(t)$, называемого материнским, через масштабирование и сдвиги:

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t-\tau}{s})\,, \tag{2}$$

где s – коэффициент масштабирования, а $\tau$ – временной сдвиг.

Обратное вейвлет-преобразование имеет вид:

$$f(t) = \iint \gamma(s,\tau)\Psi_{s,\tau}(t)dtds\,. \tag{3}$$

Из данных выражений следует возможность выбора в качестве материнского вейвлета функции, которая в наибольшей степени удовлетворяет нуждам задачи, если только она удовлетворяет определенным условиям. Одним из таких условий является условие допустимости и непрерывности[4]:

$$\int \frac{|\Psi(\omega)|^2}{|\omega|}d\omega < +\infty\,. \tag{4}$$

Из выражения (4) следует, что вейвлет должен иметь представление, подобное спектру. Так же важно, чтобы среднее значение вейвлета было равно нулю, т.е.

$$\int \Psi(t)dt = 0. \tag{5}$$

Данные преобразования предназначены для аналоговой обработки. Вместе с тем компьютерные системы имеют дело с дискретными представлениями непрерывных сигналов. В выражении (1) вейвлетное преобразование вычисляется непрерывным сдвигом и масштабированием функции и вычислением корреляции между ними. Но основной проблемой является невозможность аналитически вычислить результат вейвлет-преобразования и их вычисляют численными методами. Из всего вышесказанного следует, что в дискретном преобразовании сигнал масштабируется и сдвигается не непрерывно, а только с определенным дискретным шагом:

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{s_0^j}} \Psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right), \tag{6}$$

где j и k – целые числа и $s_0 > 1$ – фиксированный шаг. Коэффициент сдвига $\tau_0$ определяется шагом. Обычно $s_0$ выбирается равным 2 как наиболее естественный для компьютерных реализаций. Значение для $\tau_0$ обычно выбирается равным 1.

Когда дискретные вейвлеты используются для преобразования непрерывного сигнала, результатом будет серия коэффициентов вейвлета, называемая последовательной декомпозицией вейвлета. Важным в схеме декомпозиции есть вопрос восстановления. В [5] доказывается, что необходимое и достаточное условие для восстановления состоит в том, что энергия коэффициентов вейвлета должна лежать между двумя положительными границами, то есть:

$$A\|f\|^2 \le \sum_{j,k} \left| \left\langle f, \Psi_{j,k} \right\rangle \right| \le B\|f\|^2, \tag{7}$$

где $\|f\|^2$ – энергия сигнала $f(t)$; $A > 0$; $B < \infty$ и A, B – независимы от $f(t)$. Когда условие (7) выполняется, семейство функций $\Psi_{j,k}(t)$ с $j,k \in Z$ называется фреймом, а A и B – границами фрейма. Когда $A = B$, фрейм сжат и дискретные вейвлеты ведут себя подобно ортонормальной основе. Когда $A \neq B$, точное восстановление возможно за счет двойного фрейма. В двойном фрейме дискретного вейвлета преобразующий вейвлет отличается от восстанавливающего вейвлета.

На последнем шаге нужно сделать дискретные вейвлеты ортогональными. Дискретные вейвлеты могут быть сделаны ортогональными в своих растяжениях и сдвигах специальным выборам «материнского» вейвлета:

$$\int \Psi_{j,k}(t)\Psi^*_{m,n}(t)dt = \begin{cases} 1, если \quad j = m \quad и \quad k = n \\ 0, в \quad других \quad случаях \end{cases}. \tag{8}$$

Произвольный сигнал может восстанавливаться суммированием ортогональных базисных функций, с весом коэффициентов вейвлетного преобразования:

$$f(t) = \sum_{j,k} \gamma(j,k)\Psi_{j,k}(t) \tag{9}$$

Ортогональность не всегда необходима для представления сигналов. В некоторых приложениях могут использоваться неортогональные вейвлеты. При этом избыточность может

уменьшать чувствительность к шуму [4]. К недостатку дискретных вейвлетов можно отнести следующее: результирующее вейвлет-преобразование больше не инвариантно к сдвигу, который означает, что вейвлет-преобразование сигнала и сдвинутого во времени того же сигнала, не являются просто сдвинутыми версиями друг друга.

Чтобы сжать с помощью вейвлетов информацию, необходимо уменьшить количество нужных для восстановления вейвлетов. Даже с дискретными вейвлетами нужно большое количество масштабирований и сдвигов, чтобы вычислить вейвлет-преобразование. Для упрощения задачи можно отказаться от использования бесконечного количества дискретных вейвлетов. Исследования показывают, что от выбора количества вейвлетов зависит и коэффициент сжатия и качество восстанавливаемой информации. Сдвиги вейвлетов являются ограниченными длительностью сигнала, и это есть верхняя граница для вейвлета. Но данное обстоятельство не снимает вопроса о необходимом количестве масштабирований для анализа сигнала. Выше было указано, что вейвлеты имеют полосу частот подобную спектру. Из теории Фурье известно, что временное сжатие эквивалентно сжатию спектра и смещению его в верхнюю область частот:

$$F\{f(at)\} = \frac{1}{|a|} F\left(\frac{\omega}{a}\right). \tag{10}$$

Это означает, что временное сжатие вейвлета в 2 раза приведет к сжатию частотного спектра вейвлета в 2 раза, а также сдвинет все частоты в 2 раза в верхнюю область. Указанное предопределяет возможность перекрытия конечного спектра нашего сигнала со спектрами растянутых и сдвинутых вейвлетов. Для более точного описания сигнального спектра, растянутые спектры вейвлетов должны касаться друг друга. Результат масштабирования материнского вейвлета показан на рис. 1.

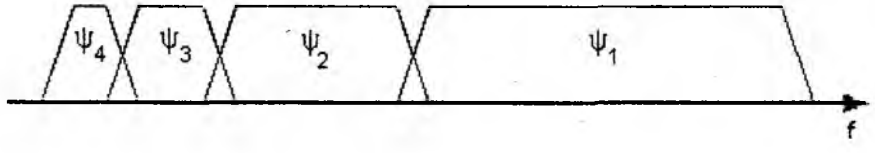



Рис. 1

Из этого следует, что если один вейвлет может быть представлен как частотный фильтр, то серия расширенных вейвлетов может быть представлена как набор фильтров частот. Если вычислить коэффициент, определяющий отношение между центральной частотой спектра вейвлета и шириной этого спектра, то он будет одинаков для всех вейвлетов. Этот коэффициент называют показателем точности Q-фильтра. В случае вейвлетов он называется константой Q-банка фильтров.

Если представить вейвлет-преобразование как группу фильтров, то можно рассматривать вейвлетное преобразование сигнала как прохождение сигнала через эту группу фильтров. Выходами с различных ступеней фильтра являются вейвлет и функция масштабирования превращающих коэффициентов. Следует указать, что анализ сигнала, прошедшего через банк фильтров, идея не новая и используется в течение многих лет в частотном кодирования.

Банк фильтров, необходимый для частотного кодирования, может быть создан различными путями. Одним из них является построение большого числа фильтров частотных полос, чтобы разделить спектр на различные полосы. Преимущество этого метода состоит в том, что ширина каждой полосы может быть выбрана произвольно. Недостатком метода является то, что в этом случае необходимо создавать каждый фильтр отдельно, а это может быть трудоемким процессом. Другой путь состоит в том, чтобы разделить сигнальный спектр на две равные части – низкочастотную и высокочастотную. Высокочастотная часть содержит индивидуальные характеристики и особенности сигнала, которые должны быть сохранены. Тем не менее, низкочастотная часть все еще содержит некоторые особенности

сигнала и, следовательно, мы можем разделить ее снова. Так происходит до тех пор, пока не достигнуто требуемое количество полос. Обычно количество полос ограничивается длиной данных или доступной вычислительной мощностью процессора системы. Процедура разбиения спектра представлена на рис. 2. Преимущество этой схемы в том, что необходимо создать только два фильтра.

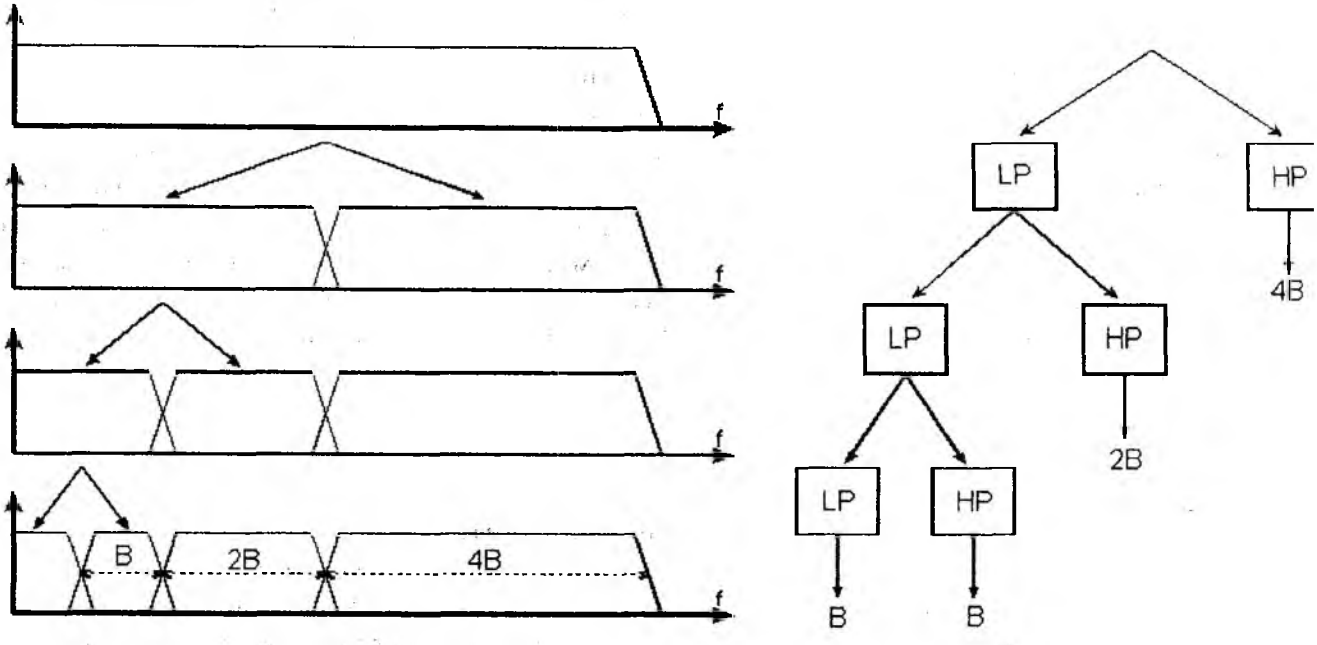



Рис. 2

Фирма Analog Devices выпускает недорогую микросхему ADV601, предназначенную для сжатия видеосигнала с коэффициентом сжатия 350:1 [6]. В ней используется биортогональное (7,9) вейвлет-преобразование и аналогичный рассмотренному выше метод разбиения. Различие состоит в том, что вейвлет-преобразование используется для обработки двухмерного сигнала, и это накладывает свои особенности. Также компания AT&T разработала графический формат хранения информации под названием DJVU.

**Заключение**

К настоящему времени разработаны и используются различные математические методы обработки и представления сигналов, в том числе и с использованием вейвлетов. На их основе созданы аппаратные и программные реализации применительно к сжатию и обработке графической информации. При этом использование этих методов для приложений, связанных с обработкой речевой информации, ограничено. Из указанного следует необходимость проведения дальнейшего исследования о применении теории вейвлетных преобразований в качестве возможной альтернативы существующим методам.

**Список литературы:** 1. *Замула А.А., Павленко Ю.С.* Защита информации в IP-телефонии// Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып.119. С.191-194 2. ITU-T Recommendation G.723.1 Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s. 3. *Valens, C.* A Really Friendly Guide to Wavelets, 1999. 4. *Sheng, Y.* Wavelet Transform, 1996. 5. *Daubechies, I.* Ten Lectures on Wavelets. 2nd ed. Philadelphia: SIAM, 1992. 6. Analog Devices, Inc, 1997. ADV601 Low Cost Multiformat Video Codec.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 15.04.2002*

УДК 681.3.06

*А.А. ЗАМУЛА, канд. техн. наук*

# МЕТОДОЛОГИЯ АНАЛИЗА РИСКОВ И УПРАВЛЕНИЯ РИСКАМИ

В постоянно расширяющейся области использования средств вычислительной техники и передачи данных появляются новые проблемы, связанные с сохранением конфиденциальности, целостности, доступности информации, а также наблюдаемости за действиями пользователей.

К настоящему времени сложилась общепринятая точка зрения на концептуальные основы обеспечения информационной безопасности (ИБ). Суть ее заключается в том, что подход к обеспечению ИБ должен быть комплексным, сочетающим меры следующих уровней: законодательного, административного, программно-технического.

Тем не менее, можно констатировать, что, несмотря на усилия многочисленных организаций, занимающихся решением проблем обеспечения информационной безопасности, тенденция остается негативной.

Модель процесса информационной безопасности может быть представлена так, как показано на рис. 1. В соответствии с данной моделью [1] обработка информации на объекте осуществляется в условиях воздействия на информацию угроз (совокупность дестабилизирующих факторов). Для противодействия угрозам информации могут использоваться специальные средства защиты, оказывающие нейтрализующее воздействие на дестабилизирующие факторы.

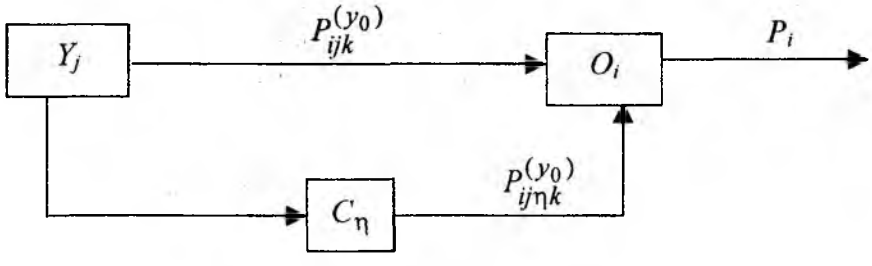



Рис. 1

С учетом обозначений, приведенных на рис. 1, можно вывести такую зависимость:

$$P_i = 1 - \prod_{\forall K} (1 - P_{ik}) a_k \,, \tag{1}$$

где $a_k$ – доля $k$-го состояния (режима работы) компьютерной системы в анализируемый период времени.

Естественным будет предположить, что система защиты информации может быть неполной, т.е. в ней могут отсутствовать средства предупреждения воздействия некоторых угроз. Тогда:

$$P_{ik} = P'_{ik} \cdot P''_{ik} \,, \tag{2}$$

где $P'_{ik}$ - вероятность защищенности информации на $i$-м объекте в $k$-м его состоянии от совокупного воздействия вех тех дестабилизирующих факторов, для противодействия которым в системе защиты не предусмотрены средства защиты; $P''_{ik}$ - то же для тех факторов, для противодействия которым в системе защиты имеются средства защиты.

В свою очередь:

$$P'_{ik} = \prod_{\forall j'} \left(1 - P_{ijk}^{(y_0)}\right), \tag{3}$$

где $j$ – принимает значения номеров угроз, против которых отсутствуют средства защиты, а

$$P''_{ik} = \prod_{\forall \eta} \prod_{\forall j''} \left(1 - P_{ijn\eta}^{(y_0)}\right), \tag{4}$$

где $j''$ принимает значения номеров дестабилизирующих факторов для противодействия которым в системе защиты предусмотрены средства; $\eta''$ - значения номеров тех номеров средств защиты информации, которые оказывают воздействие на угрозу с номером $j''$.

Вероятность надежного обеспечения безопасности информации в группе объектов определяется зависимостью

$$P = \prod_{\forall i} P_i \,. \tag{5}$$

Применительно к рассмотренной выше модели на объект защиты в любом его состоянии имеется потенциальная опасность воздействия некоторой совокупности дестабилизирующих факторов (угроз). Указанное в свою очередь означает, что функционирование объектов сопряжено с рисками как функции вероятности реализации определенной угрозы.

Управление рисками включает в себя два вида деятельности:

- оценку (измерение) рисков;
- выбор эффективных и экономичных защитных регуляторов (контрмер).

Процесс управления рисками можно подразделить на следующие этапы:

1. Определение границ компьютерной системы, в которых предполагается поддерживать режим ИБ.
2. Выбор методологии оценки рисков.
3. Оценивание угроз и оценивание уязвимости в защите компьютерной системы (КС).
4. Оценивание рисков.
5. Выбор контрмер.
6. Аудит системы управления ИБ.

На первом этапе определяется структура организации и степень детализации ее рассмотрения. Для небольшой организации допустимо рассматривать всю информационную инфраструктуру, однако, если организация крупная, всеобъемлющая оценка может потребовать неприемлемых затрат времени и сил. В таком случае следует сосредоточиться на наиболее важных сервисах, заранее соглашаясь с приближенностью итоговой оценки. Если важных сервисов все еще много, выбираются те из них, риски для которых заведомо велики или неизвестны.

Кроме того, на данном этапе осуществляется классификация и описание ресурсов КС (вычислительных систем, способов связи и коммуникаций, информации, ее категорий, вида представления, мест сохранения, технологии обработки и пр.), разработка информационной модели КС (описание информационных потоков КС, интерфейсов между пользователем и КС и т.д.). Все это важно для оценки последствий нарушений ИБ.

Рекомендуется рассмотреть следующие классы ресурсов: средства вычислительной техники, данные, системное и прикладное программное обеспечение. При этом необходимо учитывать ценность ресурсов КС, выраженную, например, в стоимостных показателях (характеристиках).

Известны табличные методы, учитывающие стоимостные характеристики ресурсов [2]. В методах данного типа показатели физических ресурсов оцениваются с точки зрения стоимости их замены или восстановления работоспособности. Существующие или предполагае-

мые программные ресурсы оцениваются тем же способом, что и физические, на основе определения затрат на их приобретение или восстановление.

Количественные показатели информационных ресурсов оцениваются на основе опросов экспертов. При этом учитывается ценность информации для ее владельца, степень критичности информации и другие ее характеристики. На основе результатов опроса производится оценивание показателей и степени критичности информационных ресурсов для наихудшего варианта развития событий. Рассматривается потенциальное воздействие на КС при возможном нарушении конфиденциальности, целостности, доступности информации. Процесс получения количественных показателей дополняется методиками оценивания информационных ресурсов с учетом факторов: безопасность персонала, требования по соблюдению законодательных и нормативных положений; коммерческие и экономические отношения; финансовые потери и нарушения в производственной деятельности; потеря репутации организации и т.д. Система показателей может быть представлена в виде бальной шкалы (например, восьмибальная). Выбор использования количественных показателей определяется удобством или целесообразностью (в ряде случаев количественные оценки затруднены).

Таким образом, на данном этапе оценки рисков должна быть выбрана система критериев и методология получения оценок по этим критериям.

Выбор методологии оценки рисков (второй этап процесса управления рисками) заключается в постановке задачи оценки рисков и обоснования требований к методике оценки рисков.

Выбор подхода к оценке рисков зависит от ряда факторов: требований к режиму ИБ, спектра воздействия угроз, принимаемых во внимание, эффективности контрмер. В случаях, когда в КС требования в области ИБ не являются жесткими, правила обеспечения режима ИБ обычно основываются на концепции базового уровня ИБ. Существует ряд стандартов, в которых рассматривается минимальный (типовой) набор наиболее вероятных угроз. Для нейтрализации этих угроз должны быть приняты контрмеры. Эти меры носят комплексный характер, т.е. охватывают административный, процедурный, программно-технический уровни и все этапы жизненного цикла информационной технологии. Однако такой подход не учитывает многие факторы, которые могут оказать существенное влияние на реализацию ПБ компьютерной системы. Так, не учитываются вероятности осуществления угроз, уязвимости ресурсов. Кроме того, при таком подходе можно упустить из вида специфические для конкретной информационной системы классы угроз.

При наличии повышенных требований к ИБ должен быть проведен так называемый полный вариант оценки рисков, в рамках которого, в дополнение к базовым, рассматриваются следующие аспекты:

- определение ценности ресурсов;
- расширение набора угроз, определенного на базовом информационном уровне ИБ, перечнем угроз, актуальных для исследуемой информационной технологии;
- оценка вероятности угроз;
- определение уязвимости ресурсов.

Методологические вопросы определения ценности ресурсов рассмотрены ранее. Что касается угроз, то необходимо отметить, что первый шаг в анализе угроз – их идентификация. В пределах выбранных видов угроз следует провести их максимально полное рассмотрение. Для этих целей можно использовать существующие инструментальные средства оценки и управления рисками, например, Германский стандарт BSI [3]. Документ включает в себя следующие блоки:

- Методология управления ИБ (организация менеджмента в области ИБ, методология использования Руководства).
- Компоненты информационных технологий:

- Основные компоненты (организационный уровень ИБ, процедурный уровень, организация защиты данных, планирование действий в чрезвычайных ситуациях).
- Инфрастуктура (здания, помещения, кабельные сети, организация удаленного доступа).
- Клиентские компоненты различных типов (DOS, Windows, UNIX, мобильные компоненты, прочие типы).
- Сети различных типов (соединения «точка-точка», сети Novell NetWare, сети с ОС UNIX и Windows, разнородные сети).
- Элементы систем передачи данных (электронная почта, модемы, межсетевые экраны и т.д.).
- Телекоммуникации (факсы, автоответчики, интегрированные системы на базе ISDN, прочие телекоммуникационные системы).
- Стандартное ПО.
- Базы данных.

• Каталоги угроз безопасности и контрмер (около 600 наименований в каждом каталоге). Каталоги структурированы следующим образом.

Угрозы по классам:

- форсмажорные обстоятельства;
- недостатки организационных мер;
- ошибки человека;
- технические неисправности;
- преднамеренные действия.

Контрмеры по классам:

- улучшение инфраструктуры;
- административные контрмеры;
- процедурные контрмеры;
- программно-технические контрмеры;
- уменьшение уязвимости коммуникаций;
- планирование действий в чрезвычайных ситуациях.

Все компоненты рассматриваются по следующему плану: общее описание, возможные сценарии угроз безопасности (перечисляются применимые к данной компоненте угрозы из каталога угроз безопасности), возможные контрмеры (перечисляются возможные контрмеры из каталога контрмер). Фактически, в данном стандарте сделана попытка описать, с точки зрения ИБ, наиболее распространенные компоненты информационных технологий и максимально учесть их специфику.

Целесообразно выявить не только сами угрозы, но и источники их возникновения – это поможет в выборе дополнительных средств защиты. После идентификации угрозы необходимо оценить уровни угроз (вероятность их реализации) и уровни уязвимости (легкости, с которой реализованная угроза способна привести к негативному воздействию). Оценивание, как правило, производится в качественных шкалах. Например, уровень угроз можно оценить по шкале «высокий – низкий». Уровни уязвимости оцениваются таким же образом. Информацию о потенциальных и наиболее вероятных угрозах, уязвимостях информационной технологии, о размерах возможного ущерба от реализации угроз можно получить путем опроса сотрудников.

Собственно уровни рисков, соответствующих показателям (ценности) ресурсов, показателям угроз и уязвимости, относящихся к каждому типу негативных воздействий, сравниваются при помощи матрицы, аналогичной приведенной в табл. 1.

**Таблица 1**

| Показатель (ценность) ресурса | Уровень угроз | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Низкий | | | Средний | | | Высокий | | |
| | Уровни уязвимости | | | Уровни уязвимости | | | Уровни уязвимости | | |
| | Н | С | В | Н | С | В | Н | С | В |
| 0 | 0 | 1 | 2 | 1 | 2 | 3 | 2 | 3 | 4 |
| 1 | 1 | 2 | 3 | 2 | 3 | 4 | 3 | 4 | 5 |
| 2 | 2 | 3 | 4 | 3 | 4 | 5 | 4 | 5 | 6 |
| 3 | 3 | 4 | 5 | 4 | 5 | 6 | 5 | 6 | 7 |
| 4 | 4 | 5 | 6 | 5 | 6 | 7 | 6 | 7 | 8 |

Количественный показатель риска определяется (для данного примера) по шкале от 1 до 8.

Для каждого ресурса рассматриваются относящиеся к нему уязвимые места и соответствующие им угрозы. Если существует уязвимость и нет связанной с ней угрозы или существует угроза, не связанная с какими-либо уязвимыми местами, то в такой ситуации рисков нет. Каждая строка в матрице определяется показателем ресурса, а каждый столбец – степенью опасности угрозы и уязвимости.

Один из этапов процесса оценки и управления рисками сводится к поиску адекватных контрмер, снижающих риски до допустимых уровней и обладающих наибольшей ценовой эффективностью.

Обеспечение повышенных требований к ИБ предполагает соответствующие мероприятия на всех этапах жизненного цикла информационных технологий. Планирование этих мероприятий производится по завершении этапа анализа рисков и выбора контрмер. Обязательной составной частью этих планов является периодическая проверка соответствия существующего режима ИБ политике безопасности, сертификация информационной системы (технологии) на соответствие требованиям определенного стандарта безопасности.

Целью проведения аудита (завершающий этап работ по обеспечению режима ИБ) является проверка соответствия выбранных контрмер декларированным в политике безопасности целям. Вопросы аудита и процедура сертификации информационной технологии на соответствие требованиям ИБ рассматриваются в [4]. В результате выполнения данного этапа должен быть документ «Ведомость соответствия», в котором содержится анализ эффективности контрмер.

Основные разделы этого документа:

- границы проводимого аудита;
- методика оценки;
- соответствие существующего режима ИБ требованиям организации и используемым стандартам;
- несоответствия и их категории;
- общие замечания, выводы рекомендации.

Таким образом, анализ рисков включает в себя идентификацию и вычисление уровней рисков на основе оценок, присвоенных ресурсам, угрозам и уязвимости ресурсов. Контроль рисков состоит в идентификации и выборе контрмер, позволяющих снизить риски до приемлемого уровня.

В конечном итоге у владельца (пользователя) КС в результате реализации процесса анализа и контроля (управления) рисками должна быть уверенность в том, что:

- все возможные риски идентифицированы;
- уязвимости ресурсов идентифицированы и их уровни оценены;
- контрмеры эффективны;
- расходы, связанные с ИБ, оправданы.

В настоящее время на рынке присутствует достаточно широкий спектр инструментальных средств (программных продуктов) проведения полного анализа и управления рисками. Наиболее известные из них: CRAMM [2], LAVA, CONTMAT и др. [4]

Применение указанных средств позволяет получать обоснованные как количественные, так и качественные оценки рисков, уязвимостей, эффективности защиты. Достоинством таких методов является возможность проведения исследований в сжатые сроки и выполнять документирование результатов.

**Список литературы:** 1. *Герасименко В.А.* Защита информации в автоматизированных системах обработки данных. Кн. 1. М.: Энергоатомиздат, 1994. 400 с. 2. *Симонов С.* Анализ рисков. Управление рисками//Jet Info 1999. № 1. 3. *Аудит безопасности информационных систем* //Jet Info, 2000, № 1. 4. Bundesamt fur Sicherheit in der Information – stechnik. It Baseline Protection Manual, 1998, http://www.bsi.bund.de/gshb/english/etc/e-conten.htm 4. *Стенг Д., Мун С.* Секреты безопасности сетей. К.: Диалектика, 1995. 544 с.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 13.05.2002*

УДК 681.3.06:519.248.681

*И.Д. ГОРБЕНКО, д-р техн. наук, С.И. ЗБИТНЕВ, А.А. ПОЛЯКОВ*

# СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЦП В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Введение

В современных автоматизированных системах управления, компьютерных системах и сетях, различных информационных технологиях и системах информационных технологий предъявляются жесткие требования к обеспечению целостности, наблюдаемости и достоверности информации на всех этапах их жизненного цикла. При этом под информацией мы будем понимать совокупность всех данных и программ, которые используются в системе или технологии, независимо от их логического или физического представления. Под информацией будем понимать также и сообщения, циркулирующие в соответствующих системах или технологиях. Опыт применения и проведенные исследования показали, что эти жесткие требования, особенно по реализации функции причастности, могут быть обеспечены только за счет применения цифровой подписи. Цифровая подпись (ЦП), по сути, представляет собой добавленные к информации данные, вычисленные посредством криптографического преобразования защищаемой информации и параметров, наличие которых позволяет удостовериться в целостности информации и подлинности ее источника, а также обеспечить защиту от подлога со стороны получателя.

По существу цифровая подпись представляет собой цифровой эквивалент подписи (штампа, печати, водяного знака и т.д.), наличие которой в сообщении, данных или программе позволяет с высокой вероятность определить источник (источники) этого сообщения или данных и юридически доказать, что с указанной допустимой вероятностью $P_д$ только он мог сформировать эту подпись, но подделать ее в течении заданного времени, при ограниченных ресурсах, злоумышленник может с вероятностью, не превышающей заданной величины $P_з$. Причем ЦП в таком применении вычисляется на основе защищаемой информации с использованием личного (конфиденциального) ключа конкретного субъекта или объекта, являющегося ее источником или отправителем. Проверка целостности и подлинности производится с использованием открытого ключа, причем знание открытого ключа не позволяет подделать ЦП с вероятностью, превышающей $P_з$.

Проведенный анализ показал [1-3], что криптографические преобразования типа ЦП должны удовлетворять ряду требований, основными из них являются следующие:

1) алгоритмы выработки и проверки ЦП должны быть открытыми, т.е. несекретными;
2) алгоритмы выработки и проверки ЦП должны обладать не выше чем полиномиальной сложностью;
3) алгоритм нахождения конфиденциального ключа и/или подделки подписи должен обладать не ниже, чем экспоненциальной, т.е. практически не реализуемой сложностью;
4) ЦП должна обладать чувствительностью к любым изменениям подписанных данных, т.е. обнаруживать нарушения целостности;
5) вероятность появления двух одинаковых подписей в разных сообщениях не должна превышать допустимого значения;
6) вычислительная сложность выработки и проверки ЦП должна быть минимизирована и иметь близкие по величине значения;

7) обеспечивать защиту от подменны, подделки и имитации полной ЦП с требуемой вероятностью;
8) цифровые подписи, полученные для одной и той же информации в разное время и на различных устройствах должны отличаться с большой вероятностью;
9) ключи выработки должны быть конфиденциальными, а ключи проверки ЦП – открытыми;
10) ЦП должна обладать максимальной стойкостью к обнаружению любых изменений, подделок и нарушений;
11) должна существовать возможность принять ЦП с различными уровнями стойкости и сложностью выработки и проверки ЦП;
12) возможность программной, аппаратно-программной и аппаратной реализации ЦП с примерно одинаковой сложностью;
13) возможность использования ЦП как с одинаковыми общесистемными параметрами в сети, так и индивидуальными для отдельной части объектов (субъектов);
14) возможность многоуровневой выработки и проверки ЦП одной и той же информации с использованием различных ключей и при необходимости различных общесистемных параметров.
15) используемая ЦП должна позволять проведение следственных экспериментов с целью обеспечения судебного разбирательства и арбитража.
16) должна существовать возможность хранения ЦП как вместе с защищаемой информацией, так и отдельно от неё.

В конце 20-го века протоколы цифровой подписи получили широкое распространение в силу увеличение компьютеризации бумагооборота. Летом 2000 года президентом США Биллом Клинтоном был подписан и вступил в действие с 1 октября указ «Electronic Signatures in Global and National Commerce Act», приравнивающий в коммерческих документах электронную подпись к чернильной (более того и сам этот указ стал первым документом, подписанным электронной подписью)[4]. Европейский Союз издал распоряжение, согласно которому цифровая подпись в скором времени будет иметь силу во всех странах Союза. Над инициативами в данной области, взаимодействуя друг с другом, работают многие азиатские государства, причём в некоторых из них электронная подпись уже закреплена законодательно. В Российской федерации опубликован и доступен по Internet Проект федерального закона «Об электронной цифровой подписи».Правительство Сингапура объявило, что с 2008 г. электронные деньги в этой стране станут легальной валютой, имеющей хождение наравне с наличными, при этом осуществлять расчёты с любыми торговыми организациями можно будет с помощью карманного компьютера или сотового телефона.

Параллельно с развитием криптографических систем, еще боле интенсивно развиваются математические методы и криптоаналитические системы, что влечет за собой повышение требований к стойкости криптосистем, в частном случае к электронной цифровой подписи.

## 1. Классификация известных ЦП

Абсолютное большинство разрабатываемых и используемых в мире ЦП базируется на использовании несимметричных криптографических преобразований, выполняемых в кольцах [1], полях Галуа [2] и группах точек эллиптических кривых [3]. К ЦП, реализованных в кольцах, необходимо отнести RSA подобные алгоритмы [6], к преобразованиям в полях Галуа – алгоритмы Диффи-Хеллмана [7] и Ель-Гамаля [8]. Опыт применения и проведения исследований ЦП, базирующихся на преобразованиях в кольцах и полях, показали, что они практически исчерпали себя и в ближайшее время не будут обеспечивать требуемой стойкости. Одним из способов решения поставленной задачи является увеличение длины ключа ныне действующих цифровых подписей: *RSA* и *DSA*. Однако увеличение длины ключа повышает требование этих криптосистем к вычислительным возможностям ЭВМ, что не всегда является приемлемым (не все организации в состоянии поменять весь парк компьютеров для

осуществления приемлемого уровня быстродействия обновленных алгоритмов электронной подписи). Для разрешения этого противоречия разработаны и начали внедряться новые или модифицированные криптографические преобразования, выполняемые в группах точек эллиптических кривых. Появилось значительное число методов, на их основе разработаны стандарты и проекты стандартов. Поэтому представляет интерес задача их изучения, выявления особенностей и возможностей, анализа стойкости и сложность выполнения ЦП. Основой при сравнительном анализе конечно же должны быть требования 1-16, приведенные во введении. Вместе с тем, при выполнении сравнительного анализа необходимо задаться видами атак и типами угроз ЦП соответствующей информации [6-9].

На наш взгляд основными видами атак на ЦП являются следующие криптоаналитические атаки на ЦП [9]:

1) *Атака на основе известного открытого ключа* (key-only attack). Самая слабая из атак, практически всегда доступная криптоаналитику (злоумышленнику). Она может выполняться при априорной определенности криптоаналитика относительно реализации ЦП, знании общесистемных параметров, а также действующих открытых ключах.
2) *Атака на основе известных подписанных сообщений* (known-message attack). Для этой атаки полагается, что в распоряжении криптоаналитика имеется некоторое число пар $(m,\langle r,s\rangle)$ подписанных сообщений $m$, при этом он не может выбирать сообщение $m$. Кроме этого криптоаналитик знает систему и параметры ЦП.
3) *Простая атака с выбором подписанных сообщений* (generic chosen-message attack). В этом случае криптоаналитик имеет возможность выбрать некоторое количество подписанных сообщений, знает общесистемные параметры и имеет доступ до открытых ключей после выбора подписанных сообщений.
4) *Направленная атака с выбором сообщения* (direct chosen-message attack). Криптоаналитик знает общесистемные параметры, может по своему усмотрению выбирать открытый ключ и после этого выбирать подписанные сообщения.
5) *Адаптивная атака с выбором подписанного сообщения* (adaptive chosen-message attack). При осуществлении атаки криптоаналитик может выбирать открытый ключ, а также подписанное сообщение. При этом выбор следующего подписанного сообщения он может делать на основе знания допустимой подписи предыдущего выбранного сообщения.

Проведенный анализ показал, что каждая атака направлена на достижение определенной цели. С учетом этого можно выделить следующие виды угроз, в порядке возрастания опасности, для всех схем электронной цифровых подписей [9]:

1) *Экзистенциональная подделка* (existential forgery). Угроза заключается в создании криптоаналитиком для какого-нибудь, возможно бессмысленного сообщения $m'$, отличающегося от перехваченного, реальной (правильной) ЦП.
2) *Селективная подделка* (selective forgery). Представляет угрозу создания для заранее выбранного сообщения $m$ правильной ЦП.
3) *Универсальная подделка* (universal forgery). Эта угроза заключается в нахождении криптоаналитиком алгоритма формирования подписи, функционально эквивалентного действительному алгоритму ЦП, что позволяет создать или модифицировать истинные подписанные сообщения.
4) *Полное раскрытие* (total break). При этой угрозе криптоаналитик может вычислить секретный ключ, возможно отличный от $d$, но соответствующий открытому ключу $Q$. Это позволяет криптоаналитику формировать цифровые подписи для любых сообщений и в дальнейшем навязывать эти сообщения корреспондентам.

## 2. Сравнительный анализ основных ЦП

Проведенный анализ показывает, что наиболее надежными являются схемы ЦП, стойкие против самой слабой из угроз на основе самой сильной из атак, т.е. против экзистенциональ-

ной подделки на основе атаки с выбором подписанных сообщений. Показано [10], что схемы цифровой подписи стойкие против такой атаки существуют только тогда, когда существует коллизионностойкая односторонняя функция.

Кроме доказательств теоретической стойкости цифровой подписи, одним из основных факторов является проверка реально стойкой ЦП временем. В проекте стандарта IEEE, X9-62 предложен вариант цифровой подписи, являющейся модификацией применяющегося стандарта X9.30 (DSA) и получивший название *ECDSA*. В нем в качестве математического аппарата выбрана группа точек эллиптической кривой над простым полем. Использование математического аппарата эллиптических групп позволяет уменьшить длину ключа, что в свою очередь позволяет повысить скорость алгоритма формирования и проверки подписи. Самое же главное это надежда на то, что по мере развития математических методов и производительности криптоаналитических систем криптографические преобразования в группах точек эллиптических кривых будут более устойчивыми к криптоанализу, чем преобразования в кольцах [1] и полях [2]. Кроме того, при разработке преобразований в группах точек эллиптических кривых появилась возможность учесть требование реализации цифровой подписи с различными длинами. Естественно, что попытки удовлетворить противоречивым требованиям к цифровым подписям привели к потоку разработки различных модификаций криптопреобразований, в том числе цифровых подписей.

Другой важнейшей особенностью при анализе и сравнении подписей является применение в ЦП функции хэширования. Дело в том, что для практической реализации состоятельного протокола аутентификации необходимо получить интерактивный случайный запрос от проверяющего [11]. Шамир предложил способ преобразования протокола аутентификации в схему цифровой подписи посредством замены случайного запроса проверяющего на хэш-функцию подписывающего сообщения. В этом случае вместо обращения к проверяющему (он же получатель сообщения), доказывающий (он же подписывающий) вычисляет хэш-функцию от сообщения $M$ – $H(M)$ и использует его в качестве запроса. В результате строится практически однопроходный протокол и обеспечивается групповая передача подписанных сообщений от одного источника многим получателям. В определенном смысле подпись Шнора является классической. Сущность ее в следующем.

1. Подписывающий формирует случайное число $k \in (1,2,...,q-1)$ и вычисляет

$$r = a^k \pmod p,$$

где $a$ – первообразный элемент поля; $p$ – простое число. Параметры используются в качестве первой компоненты цифровой подписи.

Подписывающий вычисляет случайный запрос в виде значения хэш-функции от параметров и подписываемого сообщения $M$

$$e = H(r, M).$$

2. Подписывающий вычисляет вторую компоненту ЦП

$$s = (x \cdot e + k) \bmod q,$$

где $x$ – долговременный секретный ключ, и формирует сообщение $M$ с подписью $(e, s)$.

3. Получатель обрабатывает подписанное сообщение вида $(M';(e',s'))$. Он вычисляет

$$r' = a^{s'} y^{e'} \pmod p$$

и проверяет выполняется ли равенство

$$e' = H(r', M).$$

Если равенство выполняется, то подпись принимается, а информация $M$ считается целостной и подлинной.

Стойкость схемы Шнора в значительной степени зависит от свойств функции $H$. Если криптоаналитик может осуществлять коллизии специального вида, т.е. по заданной паре $(r, M)$ находить другую информацию (сообщение) $M'$, $M' \neq M$, такую что

$$H(r, M) = H(r', M'),$$

то он может по крайней мере осуществлять экзистенциальную подделку подписи. Для этого достаточно получить $M$ и подпись $(e, s)$ для него, а также найти коллизию указанного вида. Тогда пара $(e, s)$ будет подписью также и для сообщения М. Фактически подпись Шнора и другие подписи такого класса будут стойкими, если хэш-функция ведет себя как случайная величина (для различных сообщений).

Алгоритмы ЦП Шнора и ECSS [11,12] в группах точек эллиптической кривой приведен ниже.

<table>
<tr><td colspan="2">Алгоритм Шнора на эллиптической кривой</td></tr>
<tr><td>Формирование цифровой подписи</td><td>Проверка цифровой подписи</td></tr>
<tr><td>Вход: секретный ключ $d$, открытый ключ $Q = -d \times G$, общесистемные параметры.<br>Выход: ЦП $\langle r, s \rangle$ для сообщения $M$.</td><td>Вход: открытый ключ $Q$, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения $M'$.<br>Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. Выбираем $k \in \{1, \ldots, n-1\}$;<br>2. $k \times G = (x, y)$;<br>3. $r = h(x, e) \bmod n$;<br>4. $s = k^{-1}(dr + e) \bmod n$.</td><td>1. $(x, y) = s' \times G + r' \times Q$;<br>2. $v = \pi(x, y) \bmod n$;<br>3. $r' \stackrel{?}{=} v$.</td></tr>
<tr><td colspan="2">ECSS</td></tr>
<tr><td>Вход: секретный ключ $d$, открытый ключ $Q = dG$, общесистемные параметры.<br>Выход: ЦП $\langle r, s \rangle$ для сообщения $M$.</td><td>Вход: открытый ключ $Q$, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения $M'$.<br>Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. $e = h(M)$;<br>2. Выбираем $k \in \{1, \ldots, n-1\}$;<br>3. $k \times G = (x, y)$;<br>4. $r = (x + e) \bmod n$;<br>5. $s = (k - dr) \bmod n$.</td><td>1. $e' = h(M')$;<br>2. $(x, y) = r' \times G + s' \times Q$;<br>3. $v = \pi(x, y) \bmod n$;<br>4. $r' \stackrel{?}{=} v$.</td></tr>
</table>

Здесь общесистемные параметры – параметры эллиптической кривой $a$ и $b$, порядок эллиптической кривой $u = \#E(F_q)$, кофактор $h$, базовая точка $G$ и порядок базовой точки $n = u/h$.

В проекте стандарта ISO/IEC CD 15946-2 включено четыре алгоритма ЦП в группах точек эллиптической кривой. Ниже приведены ЦП вошедшие в этот проект (кроме *ECSS*, так как он описан выше). Рассмотрим их более подробно.

<table>
<tr><td colspan="2">Алгоритм ECDSA</td></tr>
<tr><td>Формирование цифровой подписи</td><td>Проверка цифровой подписи</td></tr>
<tr><td>Вход: секретный ключ $d$, $Q = d \times G$, общесистемные параметры.<br>Выход: ЦП $\langle r, s \rangle$ для сообщения $M$.</td><td>Вход: открытый ключ $Q$, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения $M'$.<br>Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. $e = h(M)$;<br>2. Выбираем $k \in \{1,...,n-1\}$;<br>3. $k \times G = (x, y)$;<br>4. $r = \pi(x, y) \bmod n$;<br>5. $s = k^{-1}(dr + e) \bmod n$.</td><td>1. $e' = h(M')$;<br>2. $w = (s')^{-1} \bmod n$;<br>3. $u_1 = e'w \bmod n$ и $u_2 = r'w \bmod n$;<br>4. $(x, y) = u_1 \times G + u_2 \times Q$;<br>5. $v = \pi(x, y) \bmod n$;<br>6. $r' \stackrel{?}{=} v$.</td></tr>
<tr><td colspan="2">Алгоритм $EC-GDSA$</td></tr>
<tr><td>Вход: секретный ключ $d$, открытый ключ $Q = d^{-1} \times G$, общесистемные параметры.<br>Выход: ЦП $\langle r, s \rangle$ для сообщения $M$.</td><td>Вход: открытый ключ $Q$, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения $M'$.<br>Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. $e = h(M)$;<br>2. Выбираем $k \in \{1,...,n-1\}$;<br>3. $k \times G = (x, y)$;<br>4. $r = \pi(x, y) \bmod n$;<br>5. $s = (kr - e)d \bmod n$.</td><td>1. $e' = h(M')$;<br>2. $w = (r')^{-1} \bmod n$;<br>3. $u_1 = e'w \bmod n$ и $u_2 = s'w \bmod n$;<br>4. $(x, y) = u_1 \times G + u_2 \times Q$;<br>5. $v = \pi(x, y) \bmod n$;<br>6. $r' \stackrel{?}{=} v$.</td></tr>
</table>

| Алгоритм $EC-KCDSA$ | |
|---|---|
| Вход: **секретный ключ** $d$, **открытый ключ** $Q=d^{-1}\times G$, **общесистемные параметры.**<br>Выход: **ЦП** $\langle r,s\rangle$ **для сообщения** $M$. | Вход: **открытый ключ** $Q$, **общесистемные параметры, ЦП** $\langle r',s'\rangle$, **для сообщения** $M'$.<br>Выход: **Подпись действительна или нет.** |
| 1. **Выбираем** $k\in\{1,...,n-1\}$;<br>2. $k\times G=(x,y)$ **и** $c=x\|\|y$;<br>3. $r=H(c)\bmod n$;<br>4. $h=H(Z_A\|\|M)$;<br>5. $e=r\oplus h\bmod n$;<br>6. $s=(k-e)d\bmod n$. | 1. $h'=H(Z_A\|\|M')$;<br>2. $e'=r'\oplus h'\bmod n$;<br>3. $(x,y)=e'\times G+s'\times Q$;<br>4. $v=H(x\|\|y)\bmod n$;<br>5. $r'\overset{?}{=}v$. |

Здесь $\|\|$ – конкатенация двух строк; $\oplus$ – сложение по модулю 2; $\pi$ – функция выделение $x \bmod n$. В случае $EC-KCDSA$ в общесистемные параметры входит дополнительная информация $Z_A$.

В рамках конкурса NESSIE [13] участвует семь алгоритмов цифровой подписи и одна из них $ECDSA$. По предварительному результату отбора $ECDSA$ вышла в следующий этап.

Рассмотрим стойкость всех цифровых подписей к ранее перечисленным угрозам.

*Экзистенциональная подделка.* Этот вид угрозы возникает при использовании хэш-функции. В связи с тем, что хэш-функция производит отображение $m\in M$ на $h\in H$, где множество $H\subset M$, возможны коллизии, при которых для $h=H(m)$, $h'=H(m')$ и $h=h'$, $m\neq m'$. Для защиты от экзистенциальной подделки на хэш-функцию накладывается требование отсутствия полиномиального алгоритма создания коллизий.

Обычно при доказательстве стойкости цифровой подписи предполагается, что хэш-функция является случайным черным ящиком (оракулом), на вход которого поступают случайные запросы $m_0,m_1,m_2...$, а на выходе формируется случайные ответы $h_0,h_1,h_2....$ Все запросы и ответы оракул запоминает, и если на вход поступает $m_i=m_j$ и $i\neq j$, то он выдает ранее вычисленный ответ.

На практике хэш-функция должна удовлетворять, по крайней мере, следующим требованиям [14]:

a) не выше чем полиномиальная сложность вычисления, что позволяет эффективно вычислить значения $h$;
b) однонаправленность, при которой обеспечивается односторонность преобразований. Сущность этого свойства заключается в невозможности вычисления сообщения $m$ по известному $h$ (например, имеет не ниже чем экспоненциальную сложность);
c) защищенность от коллизий, при которой практически невозможно найти $m_1$ и $m_2$ такие, что $H(m_1)=H(m_2)$, так как нахождение $m_1$ и $m_2$ носит не ниже чем экспоненциальную сложность.

Рассмотрим необходимость выполнения этих условий на примере $ECDSA$. Если хэш-функция обратима, то криптоаналитик $B$ может осуществить эффективную атаку на ЦП

следующим образом. Он выбирает случайное число $l$ и вычисляет параметр цифровой подписи $r=\pi(Q+lG)$. Далее $B$ принимает, что $s=r$, и вычисляет $e=r\cdot l(\bmod n)$. Если $B$ может найти сообщение $m$ такое что $e=H(m)$, тогда $\langle r,s\rangle$ действительная ЦП.

Если используемая хэш-функция не обеспечивает защиту от коллизий, то $B$ может найти $H(m_1)=H(m_2)$, где $m_1$ действительное, заранее подписанное сообщение легальным пользователем. Затем он присоединяет ЦП $\langle r,s\rangle$ для сообщения $m_1$ к сообщению $m_2$ и отсылает сообщение $\langle m_2,\langle r,s\rangle\rangle$. Получатель при проверке ЦП не обнаружит подделки, и ему будет навязано ложное сообщение $m_2$.

*Селективная подделка.* Для подписи заранее выбранного сообщения $m$ при неизвестном ключе $d$ необходимо сформировать для сообщения $m$ подпись $\langle r,s\rangle$ так, чтобы проверка на целостность и подлинность этого сообщения $m$ давала положительный результат. Рассмотрим алгоритм подделки подписи для $m$.

1) Формируется или выбирается $k^c\in(1,2,...,n-1)$.
2) Вычисляем $r^c=\pi(k\times G)$.
3) Выбираем или подбираем $s^c\in\{1,...,n-1\}$.
4) Посылаем ложное сообщения $M^c$ с подписью $\langle r^c,s^c\rangle$.

Получатель при приеме проверяет целостность и подлинность сообщения $\langle M^c,\langle r^c,s^c\rangle\rangle$. Для этого он выполняет следующее.

1. Вычисляет значение хэш-функции $e'=h(M^c)$.
2. Вычисляет значение параметров $w=(s^c)^{-1}(\bmod n)$, $u_1=e'w(\bmod n)$ и $u_2=r^cw(\bmod n)$.
3. Находит точку эллиптической кривой $(x,y)=u_1\times G+u_2\times Q$.
4. Преобразует точку эллиптической кривой $v=\pi(x,y)\bmod n$.
5. Сравнивает $r^c\overset{?}{=}v$.

Проверка на 5-м шаге будет выполнена только в том случае, если $s^c=(k^c)^{-1}(dr^c+e)\bmod n$. Анализ этого выражения показывает, что вероятность правильного выбора $s^c$ однозначно определяется вероятностью подбора или угадывания ключа $d$ и составляет для ECDSS очень малую величину. Аналогичной стойкостью против селективной подделки обладают подписи ECSS, ECCDS, ECKCDSA и Шнора.

*Полное раскрытие.* По современным понятиям стойкость всех приведенных алгоритмов ЦП основана на сложности решения дискретного логарифма в мультипликативной группе точек эллиптической кривой. Для нахождения секретного ключа необходимо решить относительно $d$ сравнение

$$Q=d\times G, \tag{1}$$

(в случаях *ECDSA* и *ECSS*), сравнение

$$Q=d^{-1}\times G, \tag{2}$$

(в случаях $EC-GDSA$ и $EC-KCDSA$), и сравнение

$$Q=-d\times G, \tag{3}$$

в случае алгоритма Шнора.

Рассмотрим возможность нахождения $d$ по перехваченным подписанным сообщениям. Пусть перехвачено $i$ подписанных сообщений. Решая для ECDSS сравнение

$$s = k^{-1}(dr + e) \bmod n,$$

относительно $d$ получим

$$d = {(ks - e)}/{r} \pmod n.$$

Для $i$ сообщений получим $i$ сравнений с $i+1$ неизвестными (4), т.е. $k_1, k_2, ..., k_i$ и $d$. По аналогии из сравнения $s = (k - dr) \bmod n$ для ECSS имеем (5), т.е. получим тоже $i$ сравнений с $i+1$ неизвестными $k_1, k_2, ..., k_i$ и $d$.

$$\begin{cases} d = {(k_1 s_1 - e_1)}/{r_1} \bmod n, \\ \vdots \\ d = {(k_i s_i - e_i)}/{r_i} \bmod n. \end{cases} \qquad \begin{cases} d = {(k_1 - s_1)}/{r_1} \bmod n, \\ \vdots \\ d = {(k_i - s_i)}/{r_i} \bmod n. \end{cases} \qquad (4, 5)$$

Для алгоритма Шнора используя сравнение $s = (dr + k) \bmod n$, также получаем $i$ сравнений с $i+1$ неизвестными:

$$\begin{cases} d = {(s_1 - k_1)}/{r_1} \bmod n, \\ \dots \\ d = {(s_i - k_i)}/{r_i} \bmod n. \end{cases} \qquad (6)$$

Аналогично используя алгоритмы ECCDSA и ECKCDSA, можно получить соответственно системами сравнений $i$-го порядка с $i+1$ неизвестными:

$$\begin{cases} d = {s_1}/{(r_1 k_1 - e_1)} \bmod n, \\ \dots \\ d = {s_i}/{(r_i k_i - e_i)} \bmod n. \end{cases} \qquad \begin{cases} d = {s_1}/{(k_1 - e_1)} \bmod n, \\ \dots \\ d = {s_i}/{(k_i - e_i)} \bmod n. \end{cases} \qquad (7, 8)$$

Таким образом, для полного раскрытия, т.е. определенная секретного ключа по $i$ полученным ЦП на ключе $d$, необходимо решать систему $i$-го порядка с $i+1$ неизвестными.

В случае, если сообщение $M$ является зашифрованным, то неизвестными являются значения хэш-функций $e_1, e_2, ..., e_i$. В результате для EC–KCDSA и ECDSA получим систему уравнений с $2i+1$ неизвестными, поэтому шифрование подписанных сообщений позволяет существенно повысить стойкость. Но этим свойством не обладают алгоритмы Шнора и ECSS.

Отличительной особенностью ЦП EC-KCDSA является введение дополнительно параметра $Z_A$. В качестве дополнительного параметра могут использоваться время, идентификаторы, пароли, псевдослучайные данные и др. Причем вследствие стойкости *EC-KCDSA* к экзистенциальной атаке подделать дополнительную информацию для устаревшей ЦП практически не возможно.

### 3. Характеристики известных хэш-функций

Считаем необходимым привести достаточно полный перечень разработанных и используемых в настоящее время функций хэширования [14].

**Таблица 1**

| Название функции | Класс | Длина значения, (бит) |
|---|---|---|
| Whirlpool | Однонаправленная хэш-функция | 512 |
| SHA-2 | Однонаправленная хэш-функция | 512 (256, 384) |
| SHA-2 | Однонаправленная хэш-функция | 256 (384, 512) |
| ГОСТ 34311-95 | Однонаправленная хэш-функция | 256 |
| HAVAL | Однонаправленная хэш-функция | 256 (128, 160,192) |
| Tiger | Однонаправленная хэш-функция | 192 |
| SHA-1 | Однонаправленная хэш-функция | 160 |
| RIPEMD | Однонаправленная хэш-функция | 160 |
| MD4 | Однонаправленная хэш-функция | 128 |
| MD5 | Однонаправленная хэш-функция | 128 |

Указанные хэш-функции могут быть использованы при реализации ЦП. Выбор той или иной ЦП зависит от требований по стойкости и конкретных значений параметров ЦП, прежде всего модуля (порядка базовой точки) $G$. Наличие односторонних коллизионностойких хэш-функций со значениями длин 128, 160, 192, 256, 384, 512 дает возможность строить с такими же длинами ее составляющих $r$ и $s$.

### 4. Анализ стойкости ЦП

Проведенный анализ показывает [3-9], что для каждой из расмотренных ЦП существует возможность правильного выбора или подбора ключей $k_i$, $d$, $d^{-1}$, но вероятности этих событий чрезвычайно малы, поэтому обычно их не рассматривают. По сути, очень малая вероятность этих событий обеспечивается выбором величины модуля $n$. Поэтому атаки типа "грубая сила" маловероятны и не могут быть реализованы для подделки ЦП или формирования ЦП для ложных сообщений.

На наш взгляд важнейшим фактором оценки стойкости ЦП является ее устойчивость против появления со временем эффективных криптоаналитических атак или обнаружения лазеек. Алгоритм DSA уже боле 10 лет выдержал испытание временем, поэтому следует ожидать, что и его модификации, реализованные в группах точек эллиптических кривых, будут наследовать реальную криптостойкость.

Рассмотрим более подробно криптографические атаки, основанные на решении дискретного логарифмического уравнения (9)-(11) в группах точек эллиптических кривых. Задача формируется следующим образом.

Криптоаналитику известен вид ключевых уравнений

$$Q = d \times G\left(\mathrm{mod}\, f(x), p\right),\ Q = d^{-1} \times G\left(\mathrm{mod}\, f(x), p\right) \qquad (9, 10)$$

или

$$Q = -d \times G\left(\mathrm{mod}\, f(x), p\right), \qquad (11)$$

где $Q$ – открытый ключ, а $G$ – базовая точка порядка $n$. Необходимо вычислить или найти личный ключ $d$.

Наиболее простым методом поиска $d$ есть атака «грубая сила». В этом случае криптоаналитик вычисляет $Q_i$ изменяя $d = 1, 2, 3, \ldots, n-1$, пока не получим значение $Q_j$ открытого ключа. Возможность выполнения этой атаки существует всегда. Выбирая величину $n$, эту атаку можно сделать чрезвычайно маловероятной.

- Алгоритм Полинга-Хеллмана. Впервые предложен в [15]. В его основе лежит решение задачи факторизации порядка $n$ базовой точки $G$. Алгоритм понижает сложность нахождения ключа за счет решения дискретного логарифма по каждому из модулей, полученных при факторизации $n-1$. При нахождении ключа используется китайская теорема об остатках. Для защиты от этой атаки необходимо использовать базовую точку с простым порядком $n$ и большим простым делителем порядка $n-1$.
- Алгоритм «маленьких» и «больших» шагов [16]. В отличие от «грубой силы», этот алгоритм обеспечивает компромисс между быстродействием и используемой памятью. Он требует хранения порядка $\sqrt{n}$ точек и времени выполнения в худшем случае $\sqrt{n}$ шагов. При $n \geq 2^{160}$ и соответственно $\sqrt{n} \geq 2^{80}$ этот метод практически не реализуем.
- Алгоритм $\rho$-Полларда. Алгоритм описан в [17] и представляет собой вероятностную версию алгоритма «маленьких» и «больших» шагов. Он требует почти таких же вычислений, но значительно меньше объема памяти для хранения промежуточных результатов.

В работе [18-19] показано, как ускорить алгоритм $\rho$-Полларда в $\sqrt{2}$ раз.

- Распределенный алгоритм $\rho$-Полларда. В работе [20] предложен метод распределения алгоритма на $r$ процессоров, что позволяет увеличить скорости в $\sqrt{\pi n}/(2r)$ раз.
- Алгоритм $\lambda$-Полларда [18]. Этот метод, как и $\rho$-метод, является вероятностным и рассмотрен в [18]. Как и $\rho$-Поллард $\lambda$-Поллард может быть распределен на $r$ процессоров. Метод быстрее $\rho$-метода, если искомый логарифм находиться в подинтервале $[0 \ldots 0{,}39n]$.
- Составной логарифм. В [21] показано, как при известном разложении дискретного логарифма в группах точек эллиптической кривой $E$ с базовой точкой $G$ можно ускорить метод $\rho$-Полларда с использованием тех же эллиптической кривой и базовой точки. Более точно, если в первом случае поиск дискретного логарифма занимает $t$ времени, то во втором случае понадобится только $\left(\sqrt{2}-1\right)t \approx 0.41t$. Имея решение двух случаев, ожидаемое время в третьем случае составляет $\left(\sqrt{3}-\sqrt{2}\right)t \approx 0.32t$. Строя последовательность известных разложений для заданной эллиптической кривой и базовой точки, можно уменьшить сложность дискретного логарифма в последующих шагах, т.е. для нахождения последующих ключей при тех же общесистемных параметрах.

Во избежании атаки данного типа, необходимо выбирать параметры эллиптической кривой, чтобы первый шаг был невыполним.

- Суперсингулярные эллиптические кривые. В работах [22-24] показана возможность понижения сложности дискретного логарифма на эллиптической кривой $E$, определенной над $F_q$, до сложности дискретного логарифма в мультипликативной группе в расширении $F_{q^k}$, где $k \geq 1$, для которого может быть применен алгоритм квадратичного решета. Такая атака возможна при существовании малых $k$. Для проверки устойчивости кривой к этой атаке необходимо проверить, что бы $n$ не делилось на $q^k - 1$.
- Аномальные кривые над простым полем. Эллиптическая кривая $E$ над полем $F_p$ называется аномальной, если порядок $\#E(F_p) = p$. В работах [25-27] приведены эффективные алгоритмы разложения дискретного логарифма для аномальных кривых. Атака не применима к кривым, у которых порядок $\#E(F_p) \neq p$.

- Кривые, определенные над малым полем. Предполагается, что $E$ определена над $F_{2^e}$. В [18,28] показано, как метод ρ-Полларда на эллиптической кривой $E\left(F_{2^{ed}}\right)$ в этом случае может быть ускорен в $\sqrt{d}$ раз.

Из проведенного анализа, а также из [17-27] следует, что наилучшим алгоритмом является распараллеленный алгоритм ρ-Полларда. На данный момент максимальная длина "взломанной" эллиптической кривой по данным [30] составляет 109 битов. Для решения этой задачи использовался метод ρ-Полларда и около 50000 компьютеров Pentium Pro 200Mhz на протяжении 2 месяцев.

Важной характеристикой различных ЦП есть сложность выработки и проверки ЦП. Наибольшей сложностью обладают алгоритмы *EC-KCDSA* и Шнора. Это связанно с возможностью выноса процедуры хэширования в алгоритмах *ECDSA*, *EC-GDSA* и *ECSS* и невозможностью в *EC-KCDSA* и Шнора.

Для ускорения алгоритмов можно произвести *некоторую* модификацию. Так в алгоритме Шнора можно заменить $e=h(M\|x)$ на $e=h(h(M)\|x)$, в $EC-KCDSA$ заменить $H=h(Z_A\|M)$ на $H=(Z_A\|h(M))$, что позволит понизить зависимость производительности ЦП от хэш-функции.

Вторым фактором, влияющим на производительность, – есть нахождения при выработке и проверке подписи нахождения обратного элемента. В алгоритме EC-GDSA удалось избежать этой сложной операции, подписи за счет особого способа вычисления открытого ключа $Q=d^{-1}G$. Применение такого способа в EC-KCDSA позволяет уменьшить сложность выработки и проверки ЦП.

Наименее сложным и следовательно наиболее быстрым может быть алгоритм ЦП Шнора, если вычисление хэш-значения вынести за пределы алгоритма ЦП. Если это не сделать, то минимальная вычислительная сложность обеспечивается в алгоритме ECSS.

Во всех алгоритмах ЦП существует возможность существенного повышения производительности за счет выноса самой сложной операции скалярного умножения за пределы алгоритма ЦП. Это можно сделать за счет предварительного формирования таблиц $(k_i,k_iG)$ пар. Одна из пар может быть выбрана случайно по необходимости. Однако такие таблицы должны формироваться, храниться и использовать как секретные, аналогичные личным ключам.

Алгоритм решения сравнения на основе ρ-Полларда, примеры нахождения личного ключа и результаты оценки сложности криптоанализа приведены в [31].

### 5. Проблемные вопросы и перспективы развития преобразований в группах точек эллиптической кривой

Ряд источников [32, 33] показывает, что возможным направлением усовершенствования методов и средств криптографических преобразований может быть применение гиперэллиптических кривых [33] и эллиптических кривых над оптимальным расширенным полем [32].

Гиперэллиптической кривой С степени $g\geq 1$ над полем $F_q$ называется кривая вида

$$y^2+h(x)y=f(x), \tag{12}$$

где $h(x)$ – полином степени $\leq g$ (может быть $h(x)=0$); $f(x)$ – полином степени $2g+1$.

**Пример:** $y^2=x^5+1$.

При $g=1$ получаем частный случай – обычную эллиптическую кривую.

Множество рациональных точек на гиперэллиптической кривой не формирует группу, поэтому в качестве элементов группы выбраны так называемые делители [33]:

$$D = \sum_{P \in C} m_p P,$$

где $m_p$ – целые числа.

Групповая операция обладает свойствами:

$$\sum_{P \in C} m_p P + \sum_{P \in C} n_p P = \sum_{P \in C} (m_p + n_p) P .$$

Достоинства эллиптических кривых:

1) порядок поля $F_q$ может быть малым (50-80 бит);

2) большее множество возможных кривых.

Недостаток – большая вычислительная сложность групповой операции.

Одна из проблем, возникающих при реализации эллиптических кривых над полем $GF\left(2^n\right)$, – большая вычислительная сложность операций в этом поле. Одно из предложений – использование поля $GF\left(\left(2^n\right)^m\right)$. Однако показано, что такие кривые являются нестойкими. Поэтому предлагается применять поле $GF\left(p^m\right)$, где $p$ – простое число вида $2^n \pm c$. При этом m выбирается таким образом, чтобы существовал неприводимый полином $P(x) = x^m - w$, $w \in GF(p)$.

Существует 2 типа OEF:

- тип $1 - c = 1$;
- тип $2 - w = 2$.

Примеры OEF: $GF\left(\left(2^{31}-1\right)^6\right)$, $GF\left(\left(2^8-17\right)^{17}\right)$, $GF\left(\left(2^{13}-1\right)^{13}\right)$.

**Заключение**

В настоящее время вместо криптографических преобразований в кольцах и полях начали применять преобразования в группах точек эллиптической кривы. Разработаны и рекомендованы к применению ЦП Шнора, *ECSS*, *EC-DSA*, *EC-GDSA* и *EC-KCDSA*. Основными преимуществами ЦП, реализованных за счет преобразований в группах точек эллиптической кривых, является возможность уменьшения длин ключей при одновременном уменьшении сложности прямых и обратных преобразований.

Все рассмотренные алгоритмы ЦП обладают практически одинаковой стойкостью. Алгоритм *EC-KCDSA* обеспечивает защиту от навязывания ранее переданных подписанных сообщений, что достигается за счет применения дополнительной информации – например, времени формирования ЦП.

Наибольшая минимизация сложности прямого и обратного преобразований может быть достигнута с использованием алгоритмов Шнора или *ECSS*. Дальнейшее уменьшение сложности преобразований может быть достигнуто за счет использования проективного базиса.

Использование нескольких значений $n$ порядка базовой точки эллиптической кривой позволят обеспечивать различные уровни стойкости. В качестве базовых порядков и точек эллиптических кривых можно выбрать значения $n = 196$, 224, 256, 320, 384, и 512. Одним из факторов выбора этих значений $n$ есть наличие статических алгоритмов вычисления хэш-функции.

**Список литературы:** 1. IEEE P1363 / D9 (Draft Version 9). Standard Specifications for Public Key Cryptography. Number-Theoretic Background. 1999 2. X9.62 – 1999 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA) 3. X9.42 - 1998, Public Key Cryptography for The Financial Service Industry : Agreement of Symmetric Keys on Using Diffie-Hellman and MQV Algorithms 4. По материалам Chip Magazine 5. *A. Menezes, P. van Oorschot, S. Vanstone* Handbook of Applied Cryptography. CRC Press, 1997. 6. PKCS#1: RSA Encryption Standard. Version 2.0. RSA Data Security Inc. 1997 7. *Diffie W., Hellman M.E.* New Direction in Cryptography / IEEE Trans. Inf. Theory., Nov. 1976, IT-22, 644-654 8. *ElGamal T.A.* Public-Key Cryptosystem and Signature Scheme Based on Discrete Logarithm // Proc. Of CRYPTO'84 Lecture Notes in Comp. Sci. Springer-Verlag, 1985, V. 196, 10-18 9. *Иванов М.А.* Криптографические методы защиты информации в компьютерных системах и сетях. М.: КУДИЦ-ОБРАЗ, 2001. 368с. 10. *Pointcheval D., Stern J.* "Security proofs for signature scheme", Eurocrypt'96, Springer-Verlag, 387-398 11. *Beaver D., Feigenbaum J., Shoup V.* "Hiding instance in zero-knowledge proof system", Crypto'90, Springer-Verlag, 326-338 12. Введение в криптографию. / Под общей редакцией *Ященко В.В.*. М.:МЦНМО-ЧеРо, 2000. 268с. 13. *Don Johnson , Alfred Menezes, Scott Vanstone* "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Certicom Research, Canada 14. . E. Biham. On the Applicability of Differential Cryptoanalysis to Hash Functions. In E.I.S.S Workshop on Cryptographic Hash Functions, pages 25-27, March 1992. 15. *D. Pointcheval, J. Stern* "Security proofs for signature schemes", Advancer in Cryptology – Eurocrypt '96, Lecture Notes in Computer Science, 1070 (1993), Springer-Verlag, 387-398 16. *S. Pohlig and M. Hellman* "An improved algorithm for computing logarithm over $GF(p)$ and its cryptographic significance", IEEE Translation on Information Theory, 24 (1978) 106-110. 17. *J. Pollard* "Monte Carlo methods for index computation mod $p$ " , Mathematics of Computation, 32 (1978), 918-924 18. *R. Gallant , R. Lambert, S. Vanstone* "Improving the parallelized Pollard lambda search on binary anomalous curves " to appear in Mathematics of Computation. 19. *M. Wiener, R. Zuccherato* "Faster attacks on elliptic curve cryptosystem", Selected Areas in Cryptography, Lecture Notes in Computer Science, 1556 (1999), Springer-Verlag 252-266 20. *P. van Oorschot, M. Wiener* "Parallel collision search with cryptanalytic application" Journal of Cryptography, 12 (1999), 1-28 21. *R. Silverman and J. Stapleton*, Contribution to ANSI X9F1 working group, 1997. 22. *A. Menezes*, Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston, 1993 23.*A.Menezes, T. Okamoto and S. Vanstone*, "Reducing elliptic curve logarithms to logarithms in a finite field", IEEE Transactions on Information Theory, 39 (1993), 1639-1646. 24. *G. Frey and H. Riick*, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", Mathematics of Computation, 62 (1994), 865-874. 25.I.Semaev, "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", Mathematics of Computation, 67 (1998), 353-356. 26. N. Smart, "The discrete logarithm problem on elliptic curves of trace one", Journal of Cryptology, 12 (1999), 193-196. 27. T. Satoh and K. Araki, "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", Commentarii Mathematici Universitatis Sancti Pauli, 47 (1998),81-92. 28. M. Wiener and R. Zuccherato, "Faster attacks on elliptic curve cryptosystems", Selected Areas in Cryptography, Lecture Notes in Computer Science, 1556 (1999), Springer-Verlag, 190-200. 29. A. Escott, J. Sager, A. Selkirk and D. Tsapakidis, "Attacking elliptic curve cryptosystems using the parallel Pollard rho method", CryptoBytes, The Technical Newsletter of RSA Laborato- ries, volume 4, number 2, Winter 1999, 15-19. Also available at http://www.rsasecurity.com 30.*Michael J. Wiener, Robert J. Zuccherato* Faster Attacks on Elliptic Curve Cryptosystems, 1998, 8 с.31. *Горбенко И.Д., Збитнев С.И., Поляков А.А.* Криптоанализ криптографических преобразований в группах точек эллиптических кривых методом Поларда // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С.43-50 32. *N. Kobliz* "Hyperelliptic cryptosystem", Journal of Cryptology, 1 (1989), 139-150. 33. *M. Petersen* "Hyperelliptic cryptosystem", Technical Report, University of Aarhus, Denmark, 1994

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 22.04.2002*

УДК 681.3.06:519.248.681

*И.Д. ГОРБЕНКО, д-р техн. наук, А.А. ПОЛЯКОВ, С.И. ЗБИТНЕВ*

# ПРОТОКОЛЫ – ПРИМИТИВЫ УПРАВЛЕНИЯ КЛЮЧАМИ В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ

**Введение**

В защищенных информационных технологиях определяющей процедурой является процедура выработка общего секрета (ключа). Для решения этой задачи разработаны и применяются различной степени сложности состоятельные протоколы, прежде всего Диффи-Хелмана, реализованные на основе преобразований в полях Галуа [1]. Однако развитие методов и средств криптоанализа таких криптосистем и криптопротоколов вынуждает увеличивать размеры общесистемных параметров и ключей, вследствие чего увеличивается сложность выполнения базовых операций в полях, в ряде случаев до недопустимых величин. Разрешение этого противоречия может быть достигнуто за счет выработки общего секрета в группах точек эллиптических кривых над полем Галуа $GF(q)$. Целью данной статьи есть рассмотрение основных проблемных задач реализации состоятельных протоколов, определение и анализ базовых параметров протоколов и условий их реализации с использованием преобразований в группах точек эллиптических кривых.

## 1. Основные понятия

В зависимости от приложений существует несколько определений протокола. Наиболее приемлемым на наш взгляд есть следующее. Протокол – это распределенный алгоритм решения некоторой совокупности объектов и субъектов любой задачи, каждый из которых достигает цели (решает задачу) с использованием частных (распределенных) алгоритмов, причем при выполнении распределенных алгоритмов все объекты и субъекты используют одинаковую спецификацию данных и действий, процедуры синхронизации и восстановление работы после сбоев и др. Можно сказать, что протокол это многосторонний алгоритм, заданный последовательностью шагов, точно и однозначно описывающий действия двух или более сторон (объектов) которые должны быть выполнены для достижения частных и общих целей. Особенностью криптографического протокола есть то, что при его выполнении с целью обеспечения конфиденциальности, целостности, наблюдаемости и доступности до информации и/или ресурсов используются криптографические преобразования [6].

Для согласованного выполнения криптографических преобразований все взаимодействующие объекты и субъекты должны выполнять процедуру установления ключей [7]. Под установлением ключей понимается процесс или протокол, посредством выполненная которого общий секрет становится доступным объектам и/или субъектам системы (технологии), что позволяет им выполнять криптографические преобразования с необходимым качеством.

Установления ключей может быть четко разделено на передачу (транспортировку) ключей и согласование ключей [2]. При передачи ключей один объект или субъект создает или получает соответствующим образом секретное значение ключа и затем передает его другим объектам и/или субъектам безопасным образом т.е. с обеспечением конфиденциальности, целостности, подлинности, доступности и наблюдаемости. При выполнении протокола согласования ключей общий секрет вырабатывается двумя или более объектами или субъектами как функция информации, связанная с каждым из них. По сути протокол согласования ключей представляет собой процедуру разделения секрета, при реализации которой только $i > t$ из $n$ объектов и субъектов могут совместно выработать общий секрет.

Протоколы установления ключей, включающие аутентификацию, обычно требуют фазы настройки, посредством которой осуществляется распределение подлинных и возможно секретных начальных ключевых данных. Большинство протоколов имеют своей целью создание

различных ключей при каждом выполнении протокола. В некоторых случаях начальные ключевые данные задают фиксированный ключ, который каждый раз будет приводить к выполнению протокола данной парой или группой пользователей. Системы, использующие такие статические ключи, являются незащищенными от атак с известным ключом.

Многие протоколы установления ключей требуют участия централизованной или доверенной стороны для начальной системной настройки или для *интерактивных* действий (то есть в реальном времени), либо для обеих целей. Доверенная сторона обычно называется различными именами, в зависимости от выполняемых ею функций, например, *доверенная третья сторона*, *доверенный сервер*, *аутентификационный сервер*, *центр распределения ключей* (ЦРК), *центр преобразования ключей* (ЦПК) и *сертификационный орган* (СА).

К протоколам предъявляются требования, чтобы каждая из сторон при установлении ключей могла определять истинную подлинность другой(-их) стороны, что бы предотвратить несанкционированное использование результирующего ключа. В этом случае считается, что метод обеспечивает *безопасное установление ключей*. Это требуется как для секретности ключа, так и для идентификации сторон, выполняющих доступ к нему. Кроме того, требование идентификации сторон несколько, но очень важным образом, отличается от требований аутентификации объекта – здесь требованием является скорее знание подлинности сторон, которые могут получить доступ к ключу, чем подтверждение факта установления фактической связи с участием таких сторон.

*Аутентификация ключей* – свойство, дающее одной стороне уверенность в том, что никакая другая сторона, кроме конкретной второй стороны, не сможет получить доступ к конкретному секретному ключу. Аутентификация ключей не зависит от фактического владения таким ключом второй стороной или осведомленности первой стороны о таком фактическом владении. Фактически она не требует вообще никаких действий от второй стороны. По этой причине иногда используется более точное ее название (*неявная*) *аутентификация ключей*.

*Подтверждение ключей* – свойство, дающее уверенность одной стороне в том, что вторая сторона на самом деле владеет конкретным секретным ключом.

*Явная аутентификация ключей* – свойство, использование которого позволяет осуществлять аутентификацию и подтверждение ключей.

Главное внимание в процедуре аутентификации ключей сосредотачивается на подлинности второй стороны, а при подтверждении ключей – на знании значения обратного ключа. Подтверждение ключей базируется на участии объекта или субъекта, принимающего сообщение в получении и демонстрации факта владения этим ключом.

На практике факт владения ключом может быть подтвержден различными способами, включая создание односторонней хэш-функции самого ключа, использование ключа в ключевой хэш-функции и шифрование известных данных с использование этого ключа. При этом может раскрываться некоторая информация относительно значения самого ключа. Методы, использующие протоколы с нулевым знанием, позволяют подтверждать факт владения ключом, не давая при этом никакой дополнительной информации относительно его значения.

Протокол *установления аутентифицированных ключей* является протоколом установления ключей, который обеспечивает аутентификацию ключей.

Анализ показывает, что в протоколе установления ключей, который включает также и аутентификацию объектов, взаимодействие необходимо построить так, чтобы можно было давать гарантии, в том, что сторона, подлинность которой подтверждена, является той же самой стороной, с которой устанавливается ключ. Если это не обеспечивается, то криптоаналитик может осуществить ложную аутентификацию, а затем имитировать ее в протоколе установлении ключей.

Криптографические протоколы, включающие обмены сообщениями, требуют точного определения как используемых при этом сообщений, так и действий, предпринимаемых ка-

ждой стороной. На основании указанных целей можно выделить следующие протоколы аутентификации, протоколы установления ключей и протоколы аутентифицированных ключей.

*Протокол аутентификации* – дает одной стороне некоторую степень гарантии относительно подлинности другой стороны, с которой она намерена вести информационный обмен.

*Протокол установления ключей* – устанавливает общий секрет с целью осуществления в дальнейшем защищенного информационного обмена.

*Протокол установления аутентифицированных ключей* – устанавливает общий секрет со стороной, чья подлинность была (или может быть) подтверждена.

Протоколы установления ключей используются для создания общих секретов, которые обычно называются *сеансовыми ключами* или используются для их получения. Идеально сеансовый ключ является *временным секретом*, то есть секретом, использование которого ограничено коротким периодом времени. Например, единственное телекоммуникационное соединение, после которого сеанс разрывается. Использование сеансовых ключей объясняется следующими причинами [3]:

- необходимостью ограничения объема шифротекста (зашифрованного на фиксированном ключе), который может использоваться для выполнения криптоаналитической атаки;
- необходимостью ограничения скомпрометированных данных по периоду времени и количеству данных в случае компрометации ключа;
- необходимостью отказа от долговременного хранения большого количества различных секретных ключей, например в случае, когда один терминал обменивается сообщениями с большим числом других терминалов посредством создания ключей только когда это действительно требуется;
- необходимостью создания сеансов или приложений, независимых на протяжении всех коммуникационных передач.

При проектировании или выборе метода установления ключей для использования важно учитывать требуемые гарантии и свойства для предполагаемого применения. Необходимо делать различия между функциональностью, обеспечиваемой для пользователя, и техническими характеристиками, отличающими механизмы на уровне реализаций. Характеристики, отличающие методы установления ключей, включают:

1. *Характер* аутентификации, под которой понимается возможность аутентификации объектов, ключей и подтверждение ключей.
2. *Взаимность* аутентификации, при которой аутентификация объектов и аутентификация ключей или подтверждение ключей может обеспечиваться для обеих сторон.
3. *Новизна ключей*. Ключ считается *новым*, если он отличается от ранее использованного ключа или использованных ключей.
4. *Управление ключами*. Под управлением понимается процедура выработки, распределения, хранения, передачи, приема, ввода, использования и уничтожения ключей.
5. *Эффективность*. При оценке эффективности учитывается:
   a) число обменов сообщениями между взаимодействующими сторонами;
   b) ширина канала пропускания, требуемая для сообщений (объем передаваемых сторонами данных);
   c) сложность вычислений, выполняемых каждой из сторон;
   d) возможность предварительных вычислений, выполняемых с целью уменьшения *интерактивной* вычислительной сложности.
6. *Требования к третьей стороне*. Основными из этих требований являются:
   a) требование *интерактивного* (в реальном времени), автономного участия третьей стороны или без участия третьей стороны;
   b) установление требуемой степени доверия к третьей стороне.

7. *Тип используемого сертификата,* если он используется. Под ним подразумевают способ распределения долговременных ключей, подлинность и целостность которых подтверждается третей стороной.
8. *Неоспоримость авторства.* Протокол обеспечивает возможность доказательства причастности объектов или субъектов к процедуре выполнения протокола.

## 2. Основные алгоритмы выработки общего секрета

Анализ показывает, что в основе протоколов управления ключами лежат два математических алгоритма вычисления общего секретного значения – простой алгоритм Диффи-Хелмана (ДХ) [9] и сложный алгоритм (MQV) [1, 2, 11]. Простой алгоритм ДХ обеспечивает выработку общего секрета на основе знаний одного личного ключа $d$, используемого, как правило, многократно.

В алгоритме MQV один из ключей, например $d_2$, является сеансовым. Функция $avf(Q)$ определяет преобразованное (связанное) значение точки $P$. Несмотря на повышенную, по сравнению с алгоритмом ДХ, вычислительную сложность, алгоритм MQV в большинстве случаев является более предпочтительным. Он позволяет вырабатывать на каждый сеанс или файл сеансовый ключ, что обеспечивает защиту от компрометации ключей и осуществления криптоаналитических атак. В табл. 1 приведено описание алгоритмов выработки общего секрета ДХ и MQV для абонентов A и B.

**Таблица 1**

| *Алгоритм Диффи-Хелмана* | *Алгоритм MQV* |
|---|---|
| • $d_A$ – личный ключ объекта $A$;<br>• $Q_B$ – открытый ключ объекта $B$. | • Две пары ключей, долговременный $\{d_{1,A}, Q_{1,A}\}$ и сеансовый $\{d_{2,A}, Q_{2,A}\}$, принадлежащих объекту $A$.<br>• Два открытых ключа $Q_{1,B}$ и $Q_{2,B}$, принадлежащих объекту $B$. |
| 1. Вычислить точку $P=(x,y)=d_A\times Q_B$.<br>2. Проверить $P\neq O$, где $O$ – ноль аддитивной группы (точка бесконечности). Если $P=O$, то вывести "недостоверный" и остановка.<br>3. Установить $Z=\pi(x_P,y_P)$, где $\pi$ – функция преобразования координаты точки. В простейшем случае $Z=x_P$, $\pi(x_P,y_P)=x_P$. | 1. Вычислить целое число:<br>$r=d_{2,A}+(\mathrm{avf}(Q_{2,A})\times d_{1,A})(\mathrm{mod}\,n)$,<br>где $n$ – порядок базовой точки $G$ на ЭК.<br>2. Вычислить точку на эллиптической кривой:<br>$P=h\times r\times(Q_{2,B}+(\mathrm{avf}(Q_{2,B})\times Q_{1,B}))$,<br>где $h$ коэффициент связи порядка ЭК $u$ и порядка базовой точки $n$.<br>3. Проверить $P\neq O$. Если $P=O$, то вывести "недостоверный" и остановка.<br>4. Установить $Z=x_P$, где $x_P$ – $x$-координата точки $P$. |

Из предварительного анализа приведенных алгоритмов видно, что вычислительно более сложным является алгоритм *MQV*. При выполнении алгоритма *MQV* необходимо выполнить как минимум две вычислительно сложных операции, определение значения $r$ и вычисление точки $P$, используя операцию скалярного умножение. В алгоритме Диффи-Хелмана выполняется только одного скалярное умножение.

## 3. Основные требования к протоколам

Проведенный анализ показал [2], что протоколы управления ключами можно разделить на два класса:

- *протоколы согласования ключей*, задачей которых является выработка общего секрета (секретного ключа) на основе известных открытых ключей объектов;
- *протоколы транспортировки ключей*, задачей которых является доставка, ввод в действие и использование ключей с обеспечением их целостности, подлинности и при необходимости, конфиденциальности.

Названные протоколы по своему функциональному предназначению очень похожи, поэтому на них как правило накладываются одни и те же функциональные требования. Вместе с тем существуют различия в методах и, как следствие, средствах их реализации.

В практическом аспекте основным требованием к рассматриваемым протоколам является требование их состоятельности в смысле обеспечения целостности, подлинности, конфиденциальности, доступности и наблюдаемости ключей на всех этапах их жизненного цикла. В теоретическом смысле каждый из протоколов должен обладать свойствами полноты, корректности, а также в некоторых случаях свойством нулевого разглашения знаний.

Представляет интерес рассмотрение протоколов, реализуемых в группах точек эллиптических кривых с позиции их состоятельности, целостности, подлинности ключей и параметров, что может обеспечиваться за счет включения в протокол и выполнения при каждом обращении к ключам и параметрам дополнительных алгоритмов проверки, которые строятся на основе математических свойств ключей и параметров. Такие дополнительные проверки позволяют защититься от ряда угроз, прежде всего:

- *подмены параметров эллиптической кривой;*
- *преднамеренного или вынужденного использования слабых кривых;*
- *несоответствия и несогласованности открытых ключей с параметрами эллиптической кривой.*

Проведенный анализ показал, что в существующих протоколах управления ключами выработка конкретного значение секретного ключа производится из общего секрета посредством использования специальных функций, обозначаемых как *kdf* .

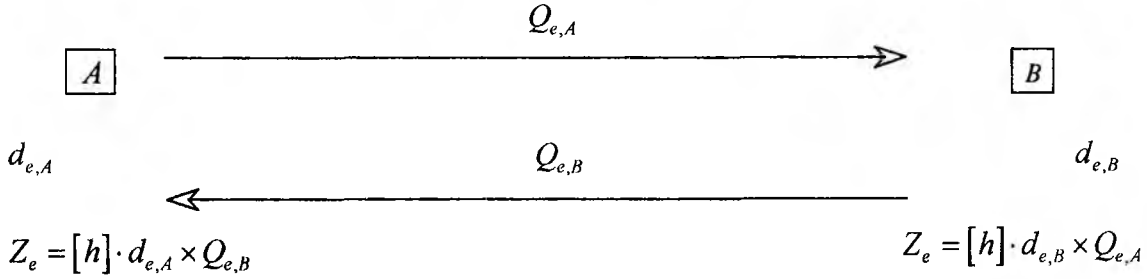
Поэтому важным является определение требований к таким функциям и выполнение этих требований.

## 4. Стандартные протоколы согласования ключей

Рассмотрим основные стандартные протоколы согласования ключей с целью их классификации и анализа. При этом выделим два объекта, один из которых является источником (инициатором), а другой ответчиком (приемником). В таком протоколе ключи состоят из пары сеансовых ключей.

*Протокол 1. Сеансовый протокол согласования ключей*

Секретными являются ключи $d_{e,A}$ и $d_{e,B}$, открытые $Q_{e,A}$ и $Q_{e,B}$.



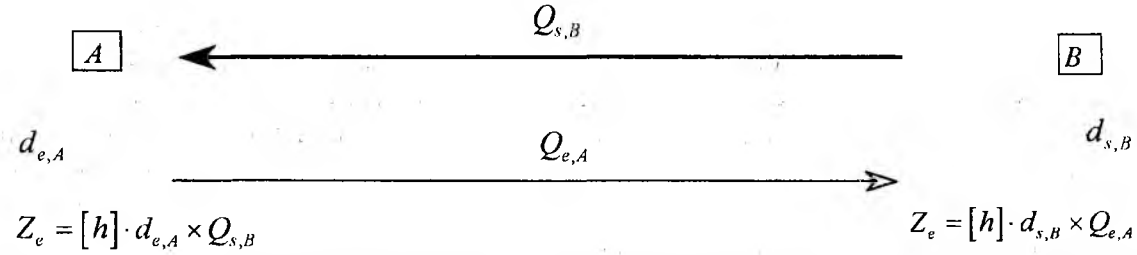

Общим секретом является $Z_e$, вырабатываемый пользователями $A$ и $B$, где $h$-кофактор. Конкретное секретное значение вырабатывается как

$$\textit{ключ} = \mathrm{kdf}\left(Z_e\right).$$

Таким образом, протокол реализует выработку сеансовых пар ключей и производит обмен открытыми сеансовыми ключами, на основании которых вырабатывается общее секретное значение.

*Протокол 2. Однопроходной протокол Диффи-Хелмана*

В протоколе используются две пары ключей $\{d_{e,A}, Q_{e,A}\}$ и $\{d_{s,B}, Q_{s,B}\}$, одна из которых является сеансовой, другая главной. Открытый главный ключ $Q_{s,B}$ передается заранее.



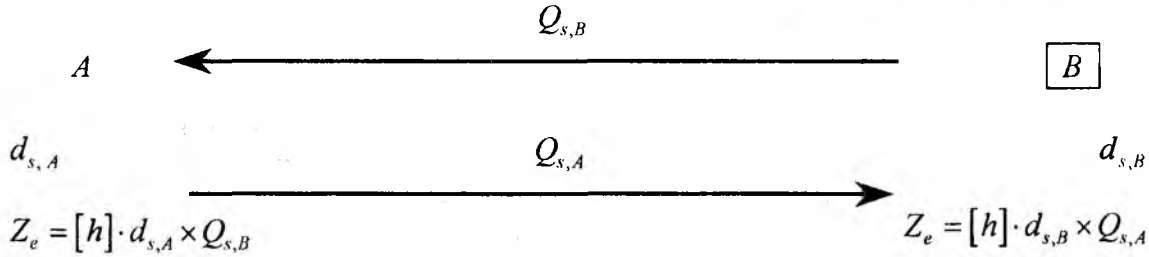

Жирными линиями выделены те передачи ключей, которые не участвуют в протоколе, они были произведены заранее. Их илюстрация является информативной.

В протоколе выполняется только одна передача сеансового ключа $Q_{e,A}$. Значение секретного ключа вырабатывается как

$$ключ = \mathrm{kdf}(Z_e).$$

*Протокол 3. Протокол на главных ключах*

В протоколе используются только главные пары ключей $\{d_{s,A}, Q_{s,A}\}$ и $\{d_{s,B}, Q_{s,B}\}$.

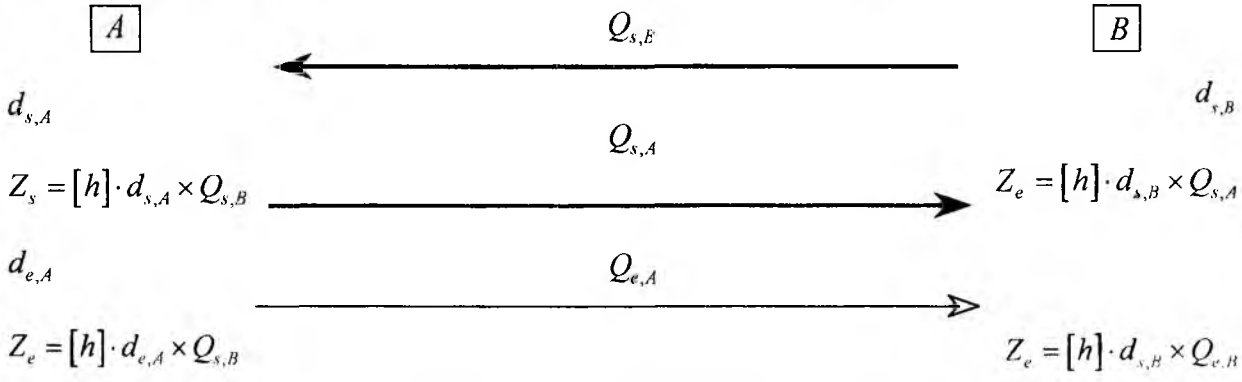



Протокол выполняет вычисление общего секрета, основываясь только на главных ключах, что позволяет не производить передачи открытых ключей во время выполнения протокола. Секретный ключ вычисляется с использованием следующей функции:

$$ключ = \mathrm{kdf}(Z_e).$$

*Протокол 4. Однопроходной протокол с использованием главных ключей*

Ключевой материал, использующийся в протоколе, состоит из двух пар главных ключей $\{d_{s,A}, Q_{s,A}\}$, $\{d_{s,B}, Q_{s,B}\}$ и одной пары сеансовых ключей $\{d_{e,A}, Q_{e,A}\}$.

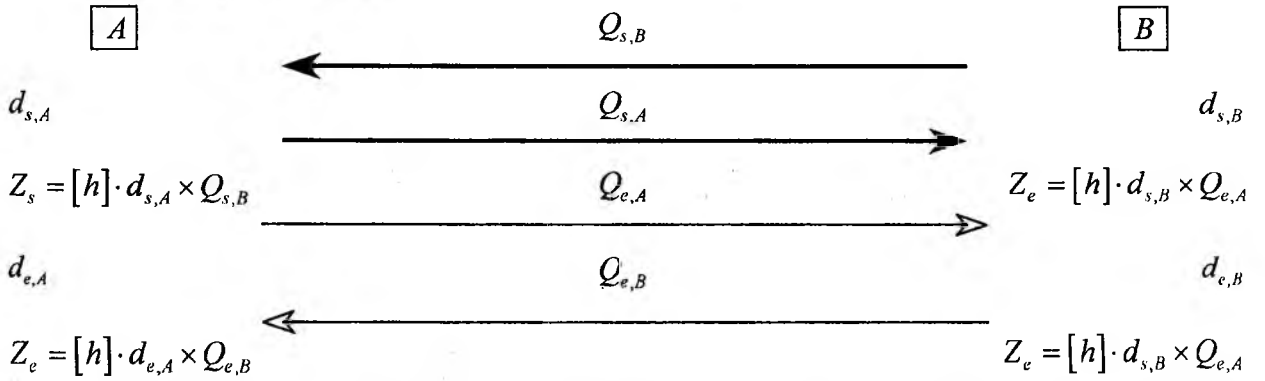



Протокол является усиленной версией однопроходного протокола ДХ. В протоколе выполняется одна передача открытого сеансового ключа. На основании используемого ключевого материала вычисляется два общих секретных значения, $Z_s$ и $Z_e$. Вычисление секретного ключа осуществляется на основании этих значений, вначале выполняется операция конкатенация (операция объединения) над значениями $Z_s$ и $Z_e$, а затем вычисляется секретный ключ

$$ключ = \mathrm{kdf}(Z_s \,||\, Z_e).$$

*Протокол 5. Полный протокол согласования ключей*

Данный протокол является полным протоколом согласования ключей. Ключевой материал состоит из двух пар главных ключей и $\{d_{s,A}, Q_{s,A}\}$, $\{d_{s,B}, Q_{s,B}\}$ и двух пар сеансовых ключей $\{d_{e,A}, Q_{e,A}\}$, $\{d_{e,B}, Q_{e,B}\}$.




В данном протоколе выполняется две передачи сеансовых открытых ключей. С использованием сеансовых и долговременных открытых и личных ключей, формируемых общие секретные значения $Z_s$ и $Z_e$. Вычисление секретного ключа производится по формуле

$$ключ = \mathrm{kdf}(Z_s \| Z_e).$$

Приведенные выше базовые протоколы-примитивы являются основными для формирования прикладных протоколов:

На рис. 1 в качестве примера приведена схема, реализующая полный протокол согласования ключей. Данный протокол является полным протоколом согласования ключей с подтверждением целостности и аутентификацией ключей. Ключевые данные состоят из двух пар главных ключей $\{d_{s,A}, Q_{s,A}\}$, $\{d_{s,B}, Q_{s,B}\}$ и двух пар сеансовых ключей $\{d_{e,A}, Q_{e,A}\}$, $\{d_{e,B}, Q_{e,B}\}$, а также ключей цифровой подписи $\{d_{sig,A}, Q_{sig,A}\}$ [3].

## 5. Стойкость криптопротоколов

Криптопротоколы, использующие криптопреобразования в группах точек ЭК, должны обеспечивать требуемую стойкость, т.е. относятся к классу теоретически стойких или доказуемо стойких алгоритмов.

Алгоритм согласования ключей Диффи-Хеллмана реализует скалярное умножение точки, которое возможно реализовать в рамках математического аппарата ЭК. Для компрометации протоколов установления ключей необходимо произвести обратную операцию к скалярному умножению, т.е. решить задачу дискретного логарифма. Таким образом, криптографическая стойкость протоколов основана сложности решения задачи ДЛЭК[4] с известной наилучшей атакой $\rho$-метод Полларда. В табл. 2 приведены числовые характеристики.

**Таблица 2**

| Длина в битах | Количество операций в группе. |
|---|---|
| 128 | $1{,}63480 \times 10^{19}$ |
| 192 | $7{,}02141 \times 10^{28}$ |
| 256 | $3{,}01567 \times 10^{38}$ |
| 320 | $1{,}29522 \times 10^{48}$ |
| 352 | $8{,}48836 \times 10^{52}$ |
| 384 | $5{,}56293 \times 10^{57}$ |
| 416 | $3{,}64572 \times 10^{62}$ |
| 448 | $2{,}38926 \times 10^{67}$ |
| 480 | $1{,}56583 \times 10^{72}$ |
| 512 | $1{,}02618 \times 10^{77}$ |
| 1024 | $1{,}18824 \times 10^{154}$ |

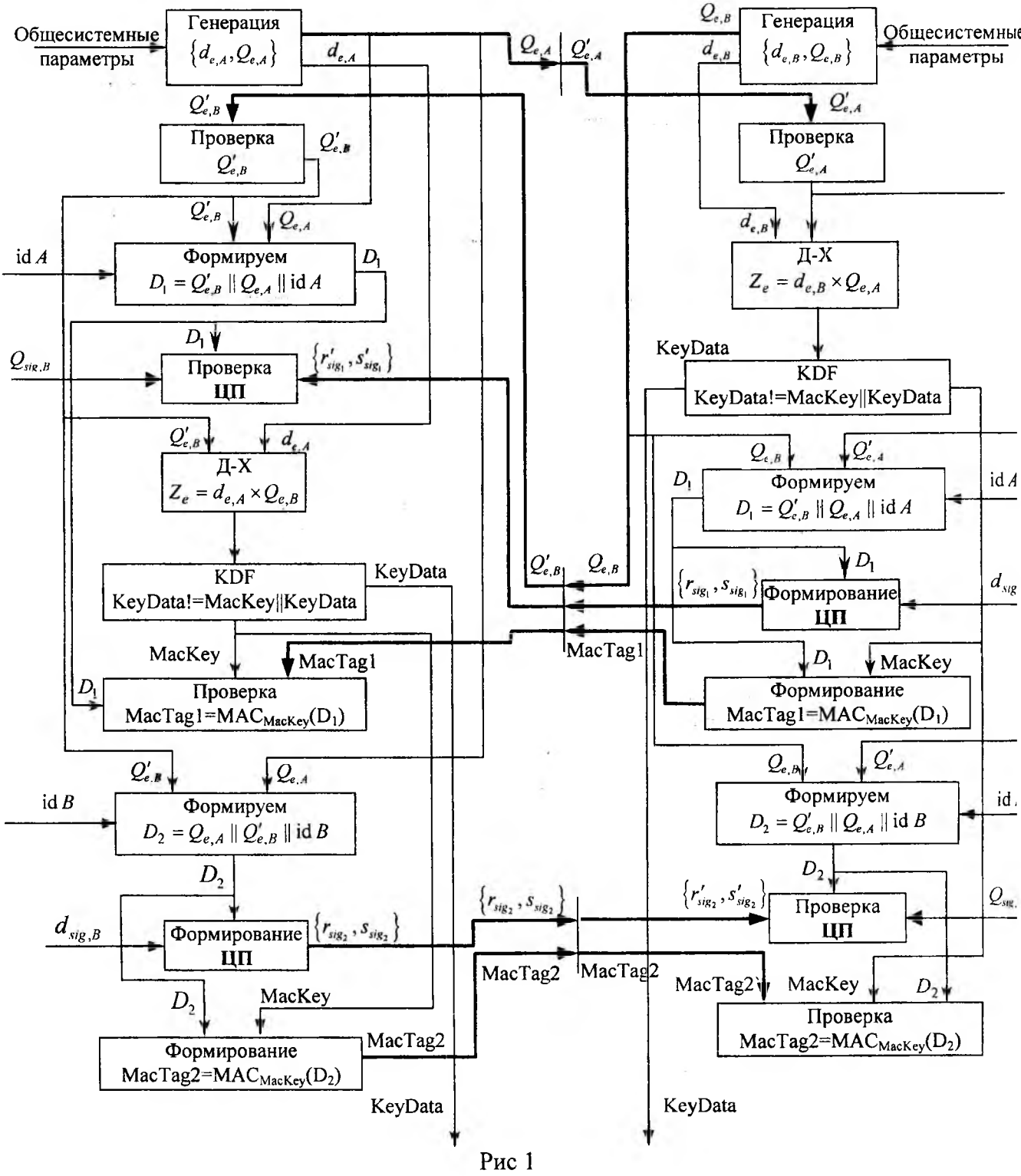



Рис 1

## 6. Требования к длине ключа

Задача схем установления ключей состоит в установлении секретных ключевых данных, разделяемых двумя объектами. Сложность атаки схемы (протокола) установления ключей должна быть не меньше сложности атаки полного перебора ключей. То есть, когда кто-нибудь устанавливает симметричные ключи, он хочет иметь гарантию, что схема установления ключей будет иметь ту же криптоаналитическую сложность, что и для симметричного алгоритма.

Такое основное условие должно быть выполнено при выборе размера параметров ЭК. Это требование связано с тем, что условие $n > 2^{160}$ в настоящее время является недостаточным для обеспечения защиты, оно не дает требуемого уровня защиты для 256-битного ключа симметричного шифрования. Поэтому минимальным рекомендуемым значением является $n \geq 2^{512}$ [2, 5].

В практически применяемых системах обеспечивается вычислительная сложность. Так в [2] определены требования к длине ключей криптопреобразований для симметричных криптоалгоритмов. Они заключаются в том, что для первого класса сложности, в симметричных криптоалгоритмах, длина ключа должна быть не менее 256 бит, 2-го – 128 бит и 3-го – 128 бит.

При решении данной проблемы необходимо знать сложность наилучшей криптоаналитической атаки на эллиптические кривые. Наилучшей атакой на ЭК в настоящее время считается алгоритм ρ-Полларда [4]. Его сложность можно оценить количеством операций сложения на эллиптической кривой

$$I_{ЭК} = \sqrt{\frac{\pi \cdot n}{4}}.$$

**Таблица 3**

| Название симметричного криптоалгоритма | Длина ключа симметричного криптоалгоритма, бит | Длина модуля преобразования ЭК, бит |
|---|---|---|
| DEA | 56 | 112 |
| 2-ключевой 3-DES | 112 | 224 |
| RIJNDAL | 128 | 256 |
| 3-ключевой 3-DES | 168 | 336 |
| RIJNDAL | 192 | 384 |
| RIJNDAL | 256 | 512 |

В таб. 3 приведены длины модулей преобразований в группах точек эллиптической кривой, при которых обеспечивается такая же стойкость ключа, как и в симметричных криптоалгоритмах.

## 7. Общесистемные параметры ЭК

Для использования эллиптических преобразований в криптографии необходимо иметь средства генерации общесистемных параметров: параметров эллиптической кривой $a$, $b$, поле $F_q$, над которым определена кривая, порядок эллиптической кривой $u = \#E(F_q)$, кофактор $h$, базовая точка $G$, порядок базовой точки $n = u/h$, в случае $q = 2^m$ примитивный полином $f(x)$. В настоящее время используются следующие алгоритмы формирования общесистемных параметров [1-2].

- Через подполя, случай когда $q = 2^{de}$, $d$ и $e$ целые числа.
- Комплексное умножение.
- Алгоритм «малых» и «больших» шагов.
- Алгоритм Скуфа.
- Построение общесистемных параметров через якобиан.

Существует два способа распространения параметров:

- Централизованное, через центр управления и сертификации ключей
- Каждая группа пользователей генерирует общесистемные параметры себе.

Учитывая то, что генерация общесистемных параметров является трудоемком процедурой, вариант централизованной поставки позволяет существенно уменьшить вычислительные и временные затраты при смене общесистемных параметров. Кроме того сертификационный центр или центральный орган распространения общесистемных параметров (ОП) имеет возможность заранее сформировать ОП. Недостатком централизованной рассылки ОП является необходимость доверительных отношений всех участников, использующих распространяемые параметры. Если отсутствует уверенность корректности полученных ОП, пользователь может провести ряд проверочных процедур:

- Проверить условие MOV [10].
- Сотсутствие аномальности кривой $\#E(F_q) \neq q$.

- Проверить, что для $q = 2^m$ $b \neq 0$, $q = p$ $4a^3 + 27b^2 \neq 0 \bmod p$.
- Порядок базовой точки $n$ простой и $n > 4\sqrt{p}$.
- $p - 2\sqrt{p} \leq \#E(F_q) \leq p + 2\sqrt{p}$.
- $nG = O$.
- Координаты точки $G$ удовлетворяют неравенству $y^2 + xy = x^3 + ax^2 + b$ при $q = 2^m$ или $y^2 = x^3 + ax + b$ при $q = p$.

В случае недоверия сертификационному центру группа пользователей может организовать свой центр распределения и генерации общесистемных параметров. Ниже приведены некоторые общесистемные параметры, удовлетворяющие вышеуказанным требованиям:

**Параметры для $F(2^m)$:**

$m$ =193, $f(x) = x^{193} + x^{15} + 1$

$a$ =0017858FEB7A98975169E171F77B4087DE098AC8A911DF7B01,
$b$ = 00FDFB49BFE6C3A89FACADAA7A1E5BBC7CC1C2E5D831478814,
$G(x, y)$ = 0301F481BC5F0FF84A74AD6CDF6FDEF4BF6179625372D8C0C5E1 – сжатая форма
$G(x, y)$ = 401F481BC5F0FF84A74AD6CDF6FDEF4BF6179625372D8C0C5E10025E399F29037\
12CCF3EA9E3A1AD17FB0B3201B6AF7CE1B05,
$n$ = 010000000000000000000000C7F34A778F443ACC920EBA49,
$h$ = 02.

$m$ =283, $f(x) = x^{283} + x^{12} + x^7 + x^5 + 1$

$a$ = 000000000000000000000000000000000000000000000000000000000000000000000000000001,
$b$ = 027B680AC8B8596DA5A4AF8A19A0303FCA97FD7645309FA2A581485AF6263E313B79\
A2F5,
$G(x, y)$ = 0305F939258DB7DD90E1934F8C70B0DFEC2EED25B8557EAC9C80E2E198F8CDB\
ECD86B12053 – сжатая форма,
$G(x, y)$ = 0405F939258DB7DD90E1934F8C70B0DFEC2EED25B8557EAC9C80E2E198F8CDB\
ECD86B1205303676854FE24141CB98FE6D4B20D02B4516FF702350EDDB0826779\
C813F0DF45BE8112F4,
$n$ = 03FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEF90399660FC938A90165B042A7CEFAD\
B307,
$h$ = 02.

$m$ =571, $f(x) = x^{571} + x^{10} + x^5 + x^2 + 1$

$a$ = 000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000001,
$b$ = 02F40E7E2221F295DE297117B7F3D62F5C6A97FFCB8CEFF1CD6BA8CE4A9A18AD84F\
FABBD8EFA59332BE7AD6756A66E294AFD185A78FF12AA520E4DE739BACA0C7FFE\
FF7F2955727A,
$G(x, y)$ = 030303001D34B856296C16C0D40D3CD7750A93D1D2955FA80AA5F40FC8DB7B2\
ABDBDE53950F4C0D293CDD711A35B67FB1499AE60038614F1394ABFA3B4C850
D927E1E7769C8EEC2D19 – сжатая форма,
$G(x, y)$ = 040303001D34B856296C16C0D40D3CD7750A93D1D2955FA80AA5F40FC8DB7B2\
ABDBDE53950F4C0D293CDD711A35B67FB1499AE60038614F1394ABFA3B4C850
D927E1E7769C8EEC2D19037BF27342DA639B6DCCFFFEB73D69D78C6C27A6009\
CBBCA1980F8533921E8A684423E43BAB08A576291AF8F461BB2A8B3531D2F048\
5C19B16E2F1516E23DD3C1A4827AF1B8AC15B,

$n$ = 03FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF\
FFFE661CE18FF55987308059B186823851EC7DD9CA1161DE93D5174D66E8382E9BB2F\
E84E47,

$h$ = 02.

**Параметры для $F(p)$:**

$|p| = 192$,

$p$ = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFEE37,

$a$ = 000000000000000000000000000000000000000000000000,

$B$ 000000000000000000000000000000000000000000000003,

$G(x, y)$ = 03DB4FF10EC057E9AE26B07D0280B7F4341DA5D1B1EAE06C7D – сжатая форма,

$G(x, y)$ = 04DB4FF10EC057E9AE26B07D0280B7F4341DA5D1B1EAE06C7D9B2F2F6D9C562\
8A7844163D015BE86344082AA88D95E2F9D,

$n$ = FFFFFFFFFFFFFFFFFFFFFFE26F2FC170F69466A74DEFD8D,

$h$ = 01.

$|p| = 256$,

$p$ = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F,

$a$ = 000000000000000000000000000000000000000000000000000000000000000000,

$b$ = 000000000000000000000000000000000000000000000000000000000000000007,

$G(x, y)$ = 0279BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81\
798 – сжатая форма,

$G(x, y)$ = 0479BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81\
798483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10\
D4B8,

$n$ = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141,

$h$ = 01.

$|p| = 521$,

$p$ = 01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF\
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,

$a$ = 01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF\
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC,

$b$ = 0051953EB9618E1C9A1F929A21A0B68540EEA2DA725B99B315F3B8B489918EF109E15\
6193951EC7E937B1652C0BD3BB1BF073573DF883D2C34F1EF451FD46B503F00,

$G(x, y)$ = 0200C6858E06B70404E9CD9E3ECB662395B4429C648139053FB521F828AF606B4D\
3DBAA14B5E77EFE75928FE1DC127A2FFA8DE3348B3C1856A429BF97E7E31C2E\
5BD66,

$G(x, y)$ = 0400C6858E06B70404E9CD9E3ECB662395B4429C648139053FB521F828AF606B4D\
3DBAA14B5E77EFE75928FE1DC127A2FFA8DE3348B3C1856A429BF97E7E31C2E\
5BD66011839296A789A3BC0045C8A5FB42C7D1BD998F54449579B446817AFBD1\
7273E662C97EE72995EF42640C550B9013FAD0761353C7086A272C24088BE94769\
FD16650,

$n$ = 01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA\
51868783BF2F966B7FCC0148F709A5D03BB5C9B8899C47AEBB6FB71E91386409,

$h$ = 01.

## 8. Вывод

Использование протоколов установления и выработки ключей в группах точек эллиптической кривой позволяет согласованно выработать ключи и обеспечить функцию причастности. Использование преобразований в группах точек ЭК по сравнению с преобразованиями в кольцах и полях [3] позволяет в 4 – 6 и более раз сократить длины открытых ключей и общесистемных параметров или при тех же параметрах существенно повысить стойкость.

Используемые на практике состоятельные протоколы, реализуемые за счет преобразований в кольцах и полях, являются состоятельными и при использовании в группах точек эллиптических кривых.

Следует ожидать, что в ближайшие годы при реализации состоятельных протоколов будут использоваться алгоритмы направленного шифрования, цифровой подписи и выработки ключей, построенные на основе преобразований в группах точек эллиптической кривой.

**Список литературы:** 1. X9.42 Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Algorithm Keys Using Diffie-Hellman, 1996. Working Draft. 2. X9.63 Public Key Cryptograph For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. 1999. 207 c. 3. IEEE P1363 / D9 (Draft Version 9). Standard Specifications for Public Key Cryptography. Number-Theoretic Background. 1999. 4. *И.Д. Горбенко, С.И. Збитнев, А.А. Поляков* Криптографические преобразования в группах точек эллиптических кривых методом Полларда // Радиотехника: Всеукр. межвед. науч-тех. сб 2001. Вып. 119. С. 43-50. 5. ISO/IEC CD 15946-3 Information Technology – Security Techniques – Cryptographic Techniques Based on Elliptic Curves – Pert 3http://crypto.nessie.org. 6. *A. Menezes, P. van Orschot, S. Vanstone* Handbook of Applied Cryptography. CRC Press, 1997. 7. *S. Blake-Wilson, D. Johnson, A. Menezes* Key agreement protocols and their Security Analysis. 1997. 8. M. Bellare, R. Canetti, H. Krawczyk. Keying hash functions for message authentication. In *Advances* in Cryptology: Crypto `96, v 1-15, 1996. 9. *Diffie W., Hellman M.E.* New Direction in Cryptography / *IEEE Trans. Inf. Theory.*, Nov. 1976, IT-22, 644-654. 10. *A.J. Menezes, T. Okamoto, and S.A. Vanstone.* Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39, pages 1639-1646, 1993. 11. *A.J. Menezes, M. Qu, and S.A. Vanstone.* Some new key agreement protocols providing implicit authentication. Workshop record, *2nd Workshop on Selected Areas in Cryptography (SAC '95)*, Ottawa, Canada, May 18-19, 1995.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 22.04.2002*

УДК 681.3.06:519.248.681

*И.Д. ГОРБЕНКО,д-р.техн.наук, Д.С.БАЛАГУРА*

# СХЕМЫ НАПРАВЛЕННОГО ШИФРОВАНИЯ В ГРУППАХ ТОЧЕК НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Введение

При информационном взаимодействии часто возникает задача зашифрования информации таким образом, чтобы расшифровать её мог только получатель. Эту задачу успешно решают схемы направленного шифрования, суть которых состоит в том, что информация шифруется на открытом ключе получателя либо на ключе, полученном с помощью открытого ключа, а расшифровывается на секретном ключе получателя, либо на ключе, полученном с помощью секретного ключа. Сейчас известно несколько схем направленного шифрования. Основными проблемами в этом классе криптографических преобразований является невысокая скорость таких преобразований, а также уменьшающаяся стойкость вследствие развития математических методов и средств криптоанализа. Целью настоящей статьи является рассмотрение состояния вопроса в области направленного шифрования, обоснование возможности реализации направленного шифрования в группах точек на эллиптических кривых [1] и разработка концептуальных вопросов их реализации и оценки стойкости.

## 1. Анализ применяемых алгоритмов направленного шифрования

Вначале рассмотрим реализацию **RSA направленного шифрования**[2]. При использовании этого алгоритма получатель должен знать общесистемные параметры $P_j$ и $Q_j$, причем $P_j$ и $Q_j$ должны быть "сильными" простыми числами. Этим повышается стойкость алгоритма. Используя эти числа, получатель должен выполнить следующие действия:

- сформировать открытый ключ $E_k$ такой, что $1 \le E_k \le \varphi(N_j)$, где $N_j = P_j \cdot Q_j$, а $\varphi(N_j)$ - функция Эйлера, и $(E_k, \varphi(N_j)) = 1$;
- вычислить личный ключ $D_k$ как обратный элемент к $E_k$ в кольце, причём $D_k$ вычисляется из соотношения $E_k \cdot D_k \equiv 1 (\mathrm{mod}\ \varphi(N))$;
- передать отправителю открытый ключ получателя $E_k$ и модуль преобразования $N_j$, обеспечивая их целостность и подлинность.

Зашифрование выполняется по следующей схеме: всё сообщение делится на блоки, длина которых равна длине модуля преобразований $M = M_1 \| M_2 \| ... \| M_n$, затем выполняется зашифрование каждого блока сообщения по формуле $C_j = M_i^{E_k} (\mathrm{mod}\ N)$.

Расшифрование каждого блока выполняется с использованием секретного ключа как $M_i' = C_i^{D_k} (\mathrm{mod}\ N)$ $M_i' = C_i^{D_k} (\mathrm{mod}\ N)$, а затем все блоки объединяются в сообщение.

Считается [3], что стойкость метода базируется на сложности факторизации модуля преобразований $N$.

Метод обладает двумя существенными недостатками:

1) в связи с разработкой новых методов и средств криптоанализа сложность факторизации модуля становится субэкспоненциальной [3], например: $I = \exp\left(\delta (\ln N)^{\nu} (\ln \ln N)^{1-\nu}\right)$, где $(\delta, \nu)$ - параметры используемого метода факторизации;
2) для повышения стойкости алгоритма необходимо увеличивать длину модуля преобразований, что приводит к повышению сложности прямого и обратного преобразований.

Разновидностью направленного шифрования RSA являются комбинированная схема направленного шифрования и схема RSA-OAEP [4].

Комбинированная схема направленного шифрования разработана с целью ускорения процедур зашифрования/расшифрования. Суть изменений заключается в том, что информация шифруется при помощи какого-либо симметричного алгоритма, а направленно шифруется лишь ключ симметричного шифрования и некоторая служебная информация.

Схема RSA-OAEP разработана как замена для стандартного направленного шифрования RSA. В схеме была введена специальная процедура предварительного шифрования данных на случайном ключе. Введенная процедура сводит на нет возможность атаки с выбранным *криптотекстом*.

Вторым алгоритмом направленного шифрования является **алгоритм, использующий схему Диффи-Хеллмана** [5]. Алгоритм позволяет вырабатывать сеансовые ключи динамически, то есть непосредственно перед началом передачи данных. Для согласования ключей используется схема Диффи-Хеллмана [5]. При использовании данной схемы каждый из абонентов должен обладать сертифицированными общесистемными параметрами домена: большим простым числом $P_j$ (модуль вычислений) и первообразным корнем $\theta_V$. Генерация общего секрета выполняется по формулам $Y_A = \theta_V^{X_A} \left(\mathrm{mod}\ P_j\right)$, $K_{BA} = Y_A^{X_B} = \theta_V^{X_A X_B} \left(\mathrm{mod}\ P_j\right)$, где $Y_A$ - открытый ключ, $K_{AB}$ - общий секрет. Абонент В выполняет те же действия.

После согласования ключей выполняется шифрование либо при помощи симметричного алгоритма шифрования, либо с помощью шифрующего устройства, выполняется разворачивание ключа до необходимой длины, а затем осуществляется направленное шифрование.

Стойкость метода базируется на сложности решения дискретного логарифма $X_A = \log_{\theta_\gamma} Y_B (\mathrm{mod}\ P)$ [4].

Но этот метод также обладает недостатком, так как в связи с разработкой новых методов криптоанализа сложность решения дискретного логарифмического уравнения уже носит субэкспоненциальный характер.

При использовании **алгоритма Эль-Гамаля** [6] отправитель и получатель должны знать открытые параметры $p$ и $g$ и открытый ключ получателя $Y$, где $p$-большое простое число, число $g$ входит в диапазон $1 < g < p-1$ и имеет в мультипликативной группе $Z_p^*$ большой порядок. В идеальном варианте $g$ - первообразный элемент по модулю $p$. После этого получатель должен выполнить следующие действия:

- выбрать себе произвольное число $k$ в диапазоне от 1 до $p-1$ (секретный ключ);
- вычислить открытый ключ как $h = g^k \bmod p$;
- передать отправителю открытый ключ получателя $Y$, обеспечивая его целостность и подлинность.

Зашифрование выполняется по схеме: все сообщение $M$ делится на блоки $M_i$ таким образом, что $M_i \in Z_p^*$, после чего каждый блок шифруют следующим образом:

Выбирают случайное число $r$ такое, что $1 \le r \le p-1$;

Вычисляют пару $C = (c_1, c_2)$, где $c_1 = g^r \bmod p$, $c_2 = MY^r \bmod p$;

Пара $C = (c_1, c_2)$ передаётся получателю.

Расшифрование выполняется получателем по правилу $D(C) = c_2 \cdot \left(c_1^{\,k}\right)^{-1} \bmod p$. Получив все пары $C = (c_1, c_2)$ и расшифровав их, получатель собирает сообщение $M$.

Сложность системы базируется на сложности определения ключа $k$, произвольной составляющей $r$ и информации $M$ при знании открытых параметров $p$, $g$ и ключа $Y$. Сложность криптоанализа определяется сложностью решения дискретного логарифма $r = \log_g Y \bmod p$, сложностью решения дискретного логарифма $r = \log_g c_1 \bmod p$ и определения значения сообщения $M$ из соотношения $c_2 = MY^r \bmod p$.

Этот метод также обладает недостатком, так как в связи с разработкой новых методов криптоанализа сложность решения дискретного логарифмического уравнения уже носит субэкспоненциальный характер. Для увеличения криптостойкости системы необходимо увеличивать модуль преобразований. Кроме того, при зашифровании для каждого блока необходимо выполнять два возведения в степень при большом модуле преобразований.

Указанные недостатки в значительной мере могут быть устранены за счёт реализации направленного шифрования в группах точек эллиптических кривых.

## 2. Простая схема шифрования с использованием аппарата эллиптических кривых [1]

Необходимыми условиями для использования простой схемы направленного шифрования с использованием аппарата ЭК является набор параметров ЭК $q$, $a$, $b$, $G$, $n$ и $h$, а также хэш-функция вместе с функцией генерирования ключей, где $q = p$ или $q = 2^m$ - порядок поля, a и b – коэффициенты ЭК, Q – базовая точка, n – порядок базовой точки на ЭК, h – кофактор.

**2.1. Алгоритм зашифрования данных (рис.1).** Входные данные: строка шифруемых данных $M$ длиной $l_M$. Открытый ключ $Q$ получателя зашифрованных данных, строка дополнительных данных $д\_д$, используемая совместно отправителем и получателем (необязательно). Открытый ключ $Q$ должен соответствовать параметрам ЭК и быть подлинным. Для выполнения зашифрования необходимо использовать примитив Диффи-Хеллмана (ДХ), примитив генерирования ключей (ГК) и функции генерирования ключей на основе хеш-функции, например, SHA-1 или SHA-2 [7].

Зашифрование битовой строки $M$ выполняется следующим образом:

1. Сгенерировать динамическую (сеансовую) пару ключей $(d_e, Q_e)$, соответствующую параметрам ЭК, используя примитив ГК, где $d_e$ - личный ключ, $Q_e$ - открытый ключ.
2. Используя примитив ДХ, выработать из $d_e$ и $Q_e$ общий секрет $z \in F_q$.
3. Используя общий секрет $Z$ и строку $д\_д$ (необязательно), сгенерировать ключ зашифрования $K^3$.
4. Используя ключ зашифрования, зашифровать открытые данные $M$: $C_i = M_i \oplus K_i^3$, где $C$ – зашифрованные данные.

Выходные данные: битовая строка $QE||C$, где $QE$ – битовая строка, сформированная из ключа $Q_e$

**2.2. Алгоритм расшифрования (рис.1).** Входные данные: битовая строка из ключа и зашифрованных данных $QE||C$, личный ключ $d$, принадлежащий получателю зашифрованных данных $C$, битовая строка $д\_д$. Личный ключ $d$ должен быть сгенерирован с помощью примитива ГК. При выполнении преобразования расшифрования используется модуль подтверждения подлинности открытого ключа, модуль ДХ и функции генерирования ключей. Непосредственное формирование гаммы зашифрования осуществляется в блоке KDF по алгоритму, базирующемуся на преобразовании общего секрета, дополнительных данных и системных данных с помощью многократного применения однонаправленной безключевой хэш-функции SHA-1. Количество вызовов функции зависит от необходимой длины ключевой по-

следовательности. При использовании алгоритма на базе SHA-1 псевдослучайная последовательность представляет собой последовательность символов, формирующихся как

$$\Gamma_i = SHA(Z \| д\_д \| с\_д \| count), \quad (1)$$

где $\Gamma_i$ - 160-битная последовательность развёрнутого ключа,

*SHA-1* – функция хэширования,

*Z* – общий секрет абонентов,

*д_д* дополнительные данные,

*с_д* системные данные,

*count*– счётчик, который изменяется при каждом вызове хэш-функции SHA-1.

Расшифрование выполняется согласно схеме (рис.1)

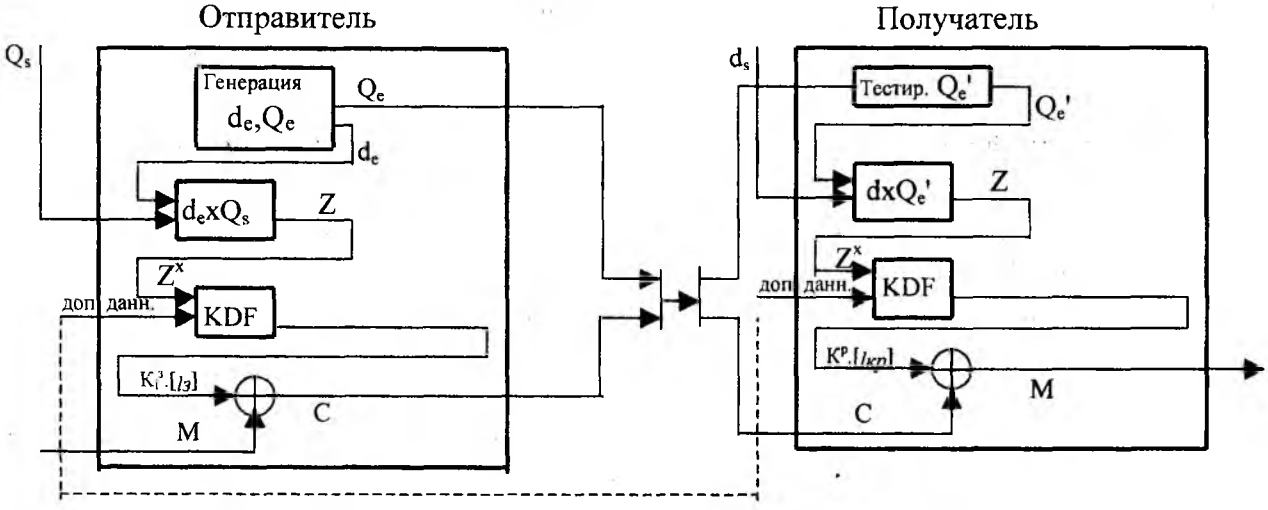



Рис. 1

Проведенный анализ показал, что более высокий уровень стойкости может быть обеспечен при использовании усиленной СНШ.

### 3. Усиленная схема направленного шифрования с использованием эллиптических кривых[1]

При использовании усиленной схемы СНШ в группах точек на ЭК корреспонденты должны обладать теми же знаниями, что и при использовании простой СНШ. Кроме того, объекты, использующие схему, должны согласовать схему вычисления кода аутентификации MAC.

**3.1. Алгоритм зашифрования (рис.2).** Входные данные такие же как и в простой схеме, только возможно использование двух строк дополнительных данных *дд1* и *дд2*.

Зашифрование битовой строки *M* выполняется следующим образом:

1. Генерируется сеансовая пара ключей $(d_e, Q_e)$, с использованием параметров ЭК.
2. Используя примитив ДХ, выработать из $d_e$ и $Q_e$ общий секрет $z \in F_q$.
3. Из *Z* формируются ключевые данные КД=К³||*мак_ключ* длиной $l_{кз} + l_{мак}$, где *мак_ключ* –ключ зашифрования MAC.
4. Зашифровываются открытые данные как $C_i = M_i \oplus K_i^3$.
5. Вычисляется контрольная сумма тег *мак_тэг* для битовой строки: *мак_тэг* =*C* ||[ $дд_2$] используя *мак_ключ* и MAC -схему.

Выходные данные: битовая строка *QE*||*C*||*мак_тэг*.

**3.2. Алгоритм расшифрования (рис.2).** Входные данные: битовая строка *QE*||C'||*мак_тэг'*, личный ключ *d*, принадлежащий получателю, битовые строки $дд_1$ и $дд_2$. При выполнении алгоритма используются те же модули, что и в простой схеме.

Расшифрование битовой строки *QE*||C'||*мак_тэг'* выполняется согласно рис.2

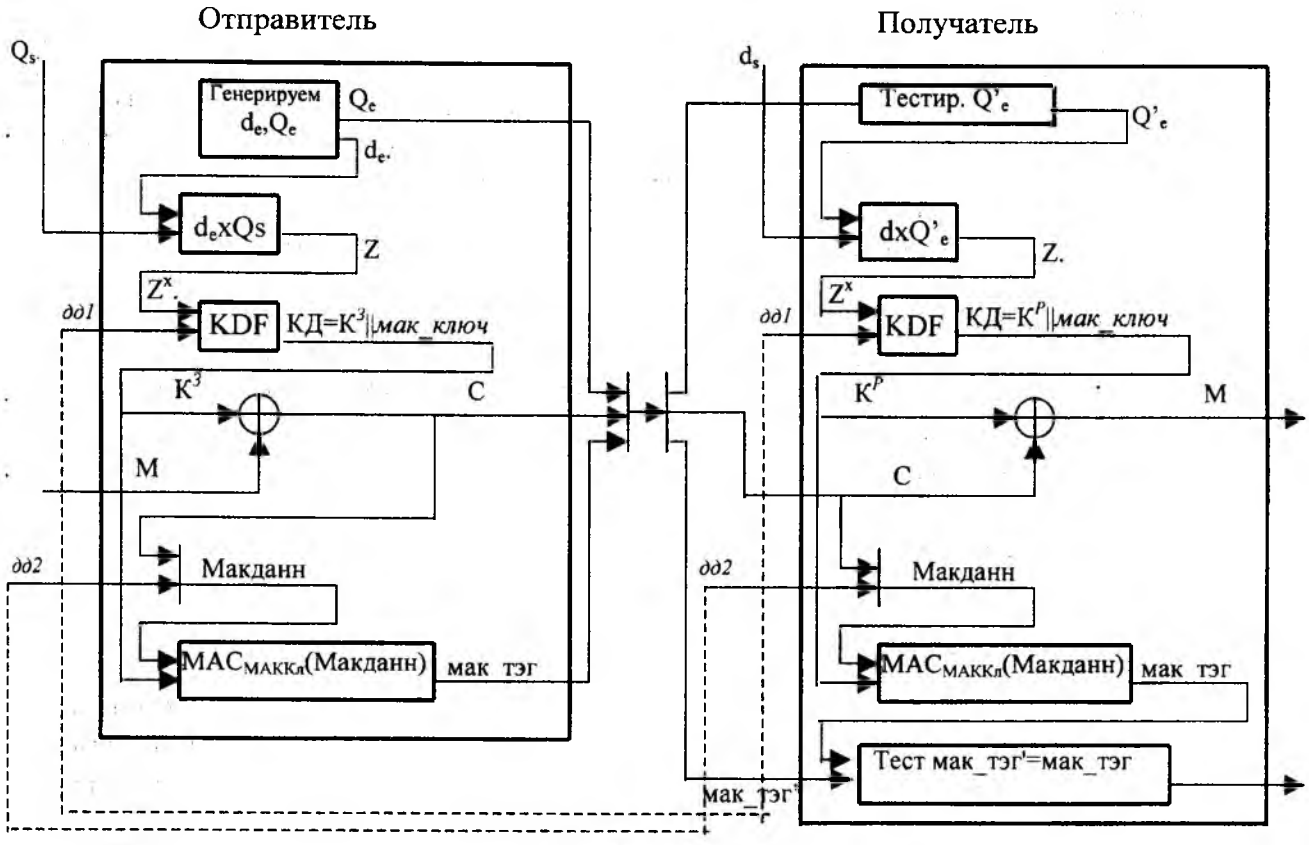



Рис. 2

## 4. Анализ схем, условия их применения

Рассмотрим основные угрозы, которые предотвращают эти схемы.

**Простая схема направленного шифрования** [1]. Основная угроза, которую предотвращает эта схема – нарушения конфиденциальности информации, то есть схема реализует услугу конфиденциальности. Услуга реализуется за счет поточного симметричного шифрования информации на развёрнутом сеансовом ключе.

Аутентификация получателя. Эта услуга заложена в саму суть схем направленного шифрования, так как расшифровать сообщение может только определённый абонент.

Дополнительная услуга, предоставляемая простой схемой – аутентификация отправителя. Данная услуга реализуется в том случае, если дополнительные данные известны только двум абонентам. В этом случае, верно расшифровав информацию, получатель может быть уверен, что она послана определённым абонентом, иначе, если получатель неверно расшифровал информацию, то либо при передаче информации была допущена ошибка, либо соединение было атаковано злоумышленником, либо отправитель сообщения не тот, за кого себя выдаёт.

Основные атаки – повторение сообщений, модификация сообщений, удаление пакетов. Разработчики схемы не предусматривали защиту от этих атак, поэтому защита от них полностью лежит на протоколах более низкого уровня.

**Усиленная схема направленного шифрования** [1]. Основные угрозы, предотвращаемые этой схемой – нарушение конфиденциальности информации и нарушение целостности зашифрованной информации и ключа. Кроме того, схема гарантирует аутентификацию получателя. Услуга конфиденциальности реализуется при помощи поточного симметричного шифрования информации на развёрнутом сеансовом ключе. Услуга целостности обеспечива-

ется при помощи ключевой MAC – схемы. Аналогично простой схеме может быть введена аутентификация отправителя.

Как показывает анализ, простая схема может применяться, в том случае, если канал связи целостный, т.е. надёжно защищён на системном либо канальном уровнях. В случае, если целостность канала связи не гарантируется, необходимо использовать усиленную СНШ, т.к. она защищает данные от модификации или подмены. Но, учитывая небольшую разницу в необходимых вычислительных ресурсах на выполнение каждой из схем, в приложениях рекомендуется использовать усиленную СНШ.

По сравнению со схемами RSA и Диффи-Хеллмана схема на эллиптических кривых обладает рядом преимуществ.

Математический аппарат групп точек на эллиптических кривых над полем Галуа обеспечивает значительно более высокую стойкость. Стойкость в зависимости от метода криптоанализа определяется через сложность, измеряемую в необходимом числе операций сложения на эллиптической кривой [8]:

- Метод $\lambda$-Полларда. Сложность криптоанализа $I_\lambda = 2\sqrt{n}$.
- Метод $\rho$-Полларда. Сложность криптоанализа $I_\rho = \sqrt{\frac{\pi n}{2}}$.
- Метод $\rho$-Полларда оптимальный. Сложность криптоанализа $I_\rho = \sqrt{\frac{\pi n}{4}}$.

В связи с тем, что при использовании групп точек на эллиптической кривой стойкость к криптоанализу достаточно высока, появляется возможность использовать модули преобразования меньших размеров, чем при преобразованиях в полях и кольцах. Так сложность факторизации 1024-битного модуля RSA приблизительно равна сложности криптоанализа ЭК с модулем преобразований 160 бит. Для эффективной защиты [8] сейчас вполне достаточно размера модуля от $2^{256}$ и более. Данные по стойкости приведены в табл. 1. Указанное позволяет ускорить процесс вычислений (криптографических преобразований). Кроме того, для вычислений может использоваться проективное представление (базис), применение которого позволяет ускорить вычисления.

**Таблица 1**

| Длина модуля преобразований | Сложность факторизации модуля RSA | Сложность криптоанализа ЭК |
|---|---|---|
| 192 бита | $2^{40,41} \approx 10^{12,32}$ | $2^{95,82} \approx 10^{29,21}$ |
| 256 битов | $2^{40,56} \approx 10^{14,5}$ | $2^{127,82} \approx 10^{39}$ |
| 512 битов | $2^{65,15} \approx 10^{19,86}$ | $2^{255,82} \approx 10^{78}$ |
| 1024 бита | $2^{88,47} \approx 10^{27}$ | $2^{511,82} \approx 10^{156}$ |

Кроме рассмотрения стойкости определения общего секрета необходимо произвести и анализ непосредственно шифрующей последовательности, вырабатываемой функцией KDF. В стандарте X9.63 предлагается алгоритм KDF, базирующийся на преобразовании общего секрета, дополнительных данных и системных данных с помощью многократного применения однонаправленной безключевой хэш-функции SHA-1, для хэширования можно использовать и появившийся алгоритм SHA-2 [7]. Количество вызовов функции зависит от необходимой длины ключевой последовательности. При использовании алгоритма на базе SHA-1 псевдослучайная последовательность представляет собой последовательность символов, формирующихся по закону (1).

Однако, мы не смогли найти результатов исследования таких последовательностей и доказательства их свойств (статистические характеристики, статистическая безопасность,

структурные свойства, период повторения). Поэтому предложенный алгоритм формирования псевдослучайной последовательности требует дополнительного исследования, так как если функция развёртывания ключа обладает небольшим периодом повторения либо плохими статистическими свойствами, то использование этой функции может значительно снизить стойкость всей схемы.

Исследование стандартного алгоритма развёртывания ключа требуется ещё и потому, что кроме стандартного алгоритма на основании хэш-функции могут быть использованы алгоритмы, построенные на основе алгоритмов блочного симметричного шифрования в режимах выработки псевдослучайной последовательности $\Gamma_i$, алгоритмы, построенные на основе линейных рекуррентных регистров, а также алгоритмы, базирующиеся на комбинации симметричных алгоритмов и ЛРР, хэш-функции и ЛРР. Для этого можно использовать:

- примитив, основанный на алгоритме ГОСТ 28147-89 в режиме выработки гаммы шифрующей ($\Gamma_i$);
- примитив, основанный на алгоритме RIJNDAEL в режиме выработки $\Gamma_i$;
- примитив, основанный на линейном рекуррентном регистре;
- примитив, основанный на комбинации ЛРР и ГОСТ 28147-89 в режиме выработки $\Gamma_i$;
- примитив, основанный на комбинации ЛРР и RIJNDAEL в режиме выработки $\Gamma_i$;
- примитив, основанный на комбинации ЛРР и стандартного примитива на SHA-1.

Перечисленные выше примитивы были протестированы на соответствие всем необходимым требованиям.

Полученные результаты показывают, что применение алгоритма НШ в группах точек на ЭК, когда $\Gamma_i$ формируется с использованием соотношения $\Gamma_i = SHA(Z \| \partial\_\partial \| c\_\partial \| count)$ обладает следующими особенностями:

1) Реализация $\Gamma_i$ зависит как от общего секрета, так и дополнительных данных, которые могут вводить взаимодействующие объекты по своему усмотрению.

2) Обеспечивается достаточно высокая скорость формирования $\Gamma_i$.

3) Период повторения зависит от составляющих, участвующих в формировании входных параметров функции генерации ключа и является случайным.

4) Если в качестве дополнительных данных использовать последовательность со строгим периодом повторения и нумерацию блоков осуществлять с использованием возрастающей функции, то можно надеяться, что период $\Gamma_i$ будет не меньше периода повторения дополнительной последовательности. Для того, чтобы период повторения дополнительной последовательности был гарантированно высоким, необходимо использовать выборки из ЛРР гарантировано большого периода повторения.

5) Такая последовательность обладает хорошей статистической безопасностью (независимостью), что объясняется с одной стороны использованием конкатенации общего секрета, дополнительных и системных данных, а с другой стороны использованием однонаправленной хэш-функции.

Рассмотрим сложность преобразования в группах точек на ЭК. Сравнение операции скалярного умножения на ЭК с операцией модульного возведения в степень показывает, что производительность операции скалярного умножения на ЭК с модулем 160 битов приблизительно на треть выше, чем производительность модульного возведения в степень с соответствующей по сложности криптоанализа длиной модуля. И с ростом модуля преобразования выигрыш в производительности существенно возрастает (для 256 битов уже более чем в 3 раза).

Значения времени выполнения полных процедур зашифрования и расшифрования (в качестве функции развёртывания ключа используется SHA-1) на компьютере с процессором К6-233 приведены в табл. 2.

**Таблица 2**

| M | | 191 | 191 | 409 | 409 |
|---|---|---|---|---|---|
| $L_k$ | | 190 | 190 | 408 | 408 |
| $L_д$ | | 100байт | 100кбайт | 100байт | 100кбайт |
| T, с | Обычная схема | 0,40 | 0,43 | 2,96 | 3 |
| | Усиленная схема | 0,44 | 0,47 | 3 | 3,03 |

При рассмотрении данных таблицы необходимо учесть, что большинство времени вы полнения процедур уходит на генерацию общего секрета и ключа. Конечно, время, затрачи ваемое на выполнение зашифрования, при модуле преобразования 409 бит достаточно высо ко, но при этом необходимо учитывать, что увеличение размера шифруемого текста не при ведет к значительному увеличению суммарного времени шифрования (время, затрачиваемо на зашифрование 100 байт на 0,03 секунды меньше, чем время, затрачиваемое на зашифро вание 100кбайт). Кроме того, модуль преобразования 409 битов является слишком большим При нынешнем развитии средств вычислительной техники и математическом аппарате крип тоанализа при шифровании не требуется использования модулей такой длины. Заметим, что операции вычисления ключевой пары $d_e$ и $Q_e$, а также вычисления общего секрета $Z$ мож но производить либо предварительно и затем хранить их соответствующим образом, либо возлагать на специальный сопроцессор. Более того, процедура генерации $d_e$ и $Q_e$ може быть вынесена за пределы самого алгоритма шифрования, а также может выполняться пред варительно. В этом случае скорость шифрования может быть увеличена на несколько поряд ков. Кроме того, операции вычисления ключа и зашифрования могут быть распараллелены Так же многое зависит от производительности алгоритма выработки ключевой последова тельности. В качестве алгоритма может использоваться любой алгоритм симметричного шифрования, работающий в режиме обратной связи по шифротексту, и даже ЛРР с общим секретом в качестве начальной установки.

### 5. Возможности модификации схем с целью улучшения их защищённости

Как мы выяснили, наиболее предпочтительными для направленного шифрования с точ ки зрения защищённости являются схемы на эллиптических кривых. Но и они не гарантиру ют защиты от всех возможных угроз. Как сказано выше, рекомендуется использовать уси ленную схему направленного шифрования на ЭК, но она не обеспечивает такой важной ус луги, как аутентификация отправителя.

Для реализации услуги аутентификации отправителя может быть использован механизм цифровой подписи. Так как все схемы НШ базируются на несимметричной криптографии, то внедрение цифровой подписи не представляет никакой сложности. Правда, применение циф ровой подписи может повлечь за собой значительное увеличение объёмов ключевой инфор мации, но этого можно избежать. Направленное шифрование базируется на наличии у субъ екта – получателя секретного и открытого ключей, это же условие является необходимым для реализации механизма цифровой подписи. В сети в большинстве случаев существуе возможность связаться каждому субъекту с каждым, это значит, что каждый субъект облада ет парами открытый – личный ключи, следовательно, не возникает необходимость введени новых ключевых данных, так как ключи из одной пары могут использоваться и для направ ленного шифрования, и для цифровой подписи. Что допускается согласно стандарту ISO 11166. Информацией, подписываемой цифровой подписью, может являться MAC – код схемах использующих общий секрет, либо зашифрованный блок в схемах с поблочной пере дачей данных.

Перечислим достоинства и недостатки состоятельных протоколов НШ по сравнению с стандартными протоколами НШ. Основным недостатком введения в схемы НШ механизмов выполняющих вычисление MAC – кода и цифровой подписи, является увеличение времен

вычислений. Например, в схеме НШ на основе RSA для обеспечения целостности и аутентификации информации, вычисления MAC – кода и цифровой подписи необходимо выполнять для каждого блока. Это увеличивает временные затраты на обработку одного блока на время вычисления MAC – кода и время вычисления цифровой подписи, то есть более чем в два раза. Для схем Д-Х и Д-Х на ЭК увеличение времени на обработку данных не столь значительно по сравнению с классическими алгоритмами НШ, так как в этих схемах вычисление MAC – кода и цифровой подписи выполняется один раз. Вторым недостатком является необходимость увеличения количества ключей в случае, если система требует использования разных ключей для шифрования и выработки цифровой подписи.

К достоинствам состоятельных протоколов НШ можно отнести защищённость от атак всех типов (в том числе и от атаки man-in-the-middle, если открытые ключи абонентов распространяются в виде сертификатов).

Анализируя основные свойства состоятельных протоколов НШ, можно сделать вывод, что их использование является более предпочтительным, чем использование обычных схем НШ.

**Заключение**

Следует ожидать, что в ближайшее время в качестве направленного шифра будут использоваться алгоритмы (схемы), реализованные в группах точек на эллиптической кривой. Их использование позволяет с одной стороны обеспечить необходимый уровень стойкости, с другой уменьшить сложность, т. е. повысить скорость преобразований. Эти схемы могут применяться в банковских системах, а также в Интернет.

**Список литературы:** 1. X9.63-199x Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. 2. ISO 11166 – 1995 Banking – Key management by means of asymmetric algorithms. Part 2: Approved algorithms using the RSA cryptosystem. 3. *И.Д. Горбенко, П.В. Колесников* Оценка стойкости RSA систем, в которых открытые ключи или параметры являются личными// Радиотехника: Всеукр. межвед. научн.-техн. сб. 2001г. вып. 119. 4. *Jakob Jonsson, Burt Kaliski:* RSA-OAEP Encryption scheme. B: Primitive specification and supportingdocumentation, 2000, by. 5. X9.42 - 1998, Public Key Cryptography for The Financial Service Industry : Agreement of Symmetric Keys on Using Diffie-Hellman and MQV Algorithms. 6. *Вербіцький О.В.* Вступ до криптології. Львів: ВНТЛб. 1998р.7.E. Biham and A. Shamir. Differential cryptoanalysis of FEAL and N-Hash. In Advances in Cryptology - Eurocrypt '91, pages 1-16, 1991.8. *И.Д. Горбенко, С.И. Збитнев, А.А. Поляков* Криптоанализ криптографических преобразований в группах точек эллиптических кривых методом Полларда// Радиотехника: Всеукр. межвед. научн.-техн. сб. №119. 2001г.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 25.04.2002*

УДК 681.3.06:519.248.681

*И.Д. ГОРБЕНКО, д-р. техн. наук., А.А.ПОЛЯКОВ, асп.*
*Е.В. ПОПОВИЧ, асп., Е.С. КОРОБЕЛЬНИКОВ, магистр.*

# СХЕМЫ РАЗДЕЛЕНИЯ СЕКРЕТА: СУЩНОСТЬ, ОСНОВНЫЕ МЕТОДЫ И СРЕДСТВА РЕАЛИЗАЦИИ

**Введение**

Важнейшей интенсивно развивающейся в последние годы областью криптологии являются специфические протоколы, которые получили название схем разделения секрета. По сути схемы разделения секрета являются многосторонними протоколам, основной функцией которых является установления ключей. Под установлением ключей понимается процесс или прикладной протокол посредством выполнения которого общий секрет становится доступным объектам системы (технологии), что позволяет им выполнять криптографические преобразования с необходимым качеством [1-3]. Разделение секрета первоначально осуществлялось для защиты криптографических ключей от потери или для создания резервной копии ключей. Применение схемы разделения секрета позволяет также решить задачи обеспечения криптографической живучести или совместного управления действиями в системе (технологии) с использованием криптографических преобразований.

Анализ показывает, что важнейшей составляющей обеспечения реальной криптостойкости в условиях возможной компрометации одной или нескольких частей общего секрета (ключа) является криптографическая живучесть. Кроме того, анализ показал, что основным предназначением схем разделения секрета [ ] является распределенное управление доверием или совместный контроль за критичными действиями (например открытие банковских хранилищ, подписание множественных соглашений, корпоративных, чеков и д.р.). При этом управление такими действиями осуществляется по принципу участия (согласия) не менее $k$ объектов или субъектов из общего их числа $n$.

Идея разделения общего секрета заключается в том, чтобы общий секрет разделить на $n$ – частей которые называются *частями секрета.* При объедении не менее $k$ объектов или субъектов, общий секрет восстанавливается однозначно. По своей сути в схемах разделения секрета реализуется метод предварительного распределения ключей, обеспечивающий одноразовое установление ключей, когда ключ задается предварительно и может быть одинаковым или различным для различных групп.

Одним из важнейших приложений схем разделения секрета является совместное управление $k$ объектами критичными действиями в системе, неправильное выполнение которых может привести к существенным или катастрофическим последствиям. В этом случае восстановленный секрет является сигналом на разрешение выполнения критического действия. Примерами таких приложений является управление статическим или тактическим оружием, атомными станциями, беспилотными самолетами и космическими аппаратами, особо опасными и опасными производствами и др.

Проведенный анализ показал, что известные схемы разделения секрета можно классифицировать следующим образом.

1. Схемы предварительного разделения секрета. В таких схемах вся конфиденциальная информация, кроме одной, части (доли), заранее устанавливается в общий секрет. При необходимости эта часть может быть предоставлена для формирования общего секрета, например с использованием специальных средств управления ключами или других средств.
2. Схемы динамического разделения секрета. К таким схемам относятся схемы, в которых секреты, восстанавливаемые различными санкционированными подмножествами, являются или могут быть различными в зависимости от предоставленных значений частей секрета.

3. Многозначные схемы разделения секрета. В таких схемах различные подмножества восстанавливают различные общие секреты, которые могут активизировать различные критические действия. То есть основным признаком таких систем является связывание различных подмножеств с различными санкционированными подмножествами.
4. Схемы разделения секрета со взаимным недоверием предназначены для защиты от злоумышленных действий со стороны одного или нескольких объектов каждого подмножества санкционированных объектов, а также доверительной стороны, которая вырабатывает общий и части секрета, и распределяет части общего секрета.
5. Криптографические пороговые схемы разделения секрета. К ним можно отнести пороговые $(k,n)$ схемы, которые обладают тем свойством, что при компрометации одной из частей, $(k,n)$ пороговая схема становится $(k-1,n)$ схемой. Причем $(k-1,n)$ схема обеспечивает надежное восстановление общего секрета.

Проведенный анализ показал [1–7], что известные схема разделения могут быть реализованы с использованием преобразований по модулю, полиномиальной интерпретации, геометрической интерпретации, с использованием известных избыточных кодов, например Рида-Саломона и др.

Целью статьи является проведение классификации и сравнительного анализа схем разделения секрета, а также рассмотрение основных приложений, в которых они могут применяться, естественно с попытками оценить их возможности и особенности применения.

**1. Схемы выработки общего секрета на основе преобразований по модулю**

Такие схемы является одними из первых схем. Они позволяет вырабатывать общий секрет в случае предоставления всем и $n$ субъектам своих частей секрета. Общий секрет и доли секрета вырабатываются доверенной стороной. Пусть общий секрет есть $M$, он имеет определенный смысл если или является ключом. Доверенная сторона генерирует $n$ случайных долей ключа $R_1, R_2, ..., R_n$, например являются ключами типа "отрывной блокнот". Затем доверенная сторона вычисляет защищенный открытый общий секрет $S_i$, причем

$$S_i = M_i \oplus \sum_{j=i}^{n} R_i, \; i = \overline{1,l}, \tag{1}$$

где $l$ – длина общего и частей секрета.

Здесь мы применили термин защищенный открытый секрет в том смысле, что относительно $S_i$ должны быть обеспечены его целостность, подлинность, доступность и наблюдаемость. Требование же обеспечения конфиденциальности к нему предъявляются не всегда. К общему же секрету $M$ предъявляются требования обеспечения всех услуг безопасности, в том числе конфиденциальности как наиболее важнейшей услуги.

Затем доверенная сторона рассылает части секрета $R_i$ с обеспечением их конфиденциальности, целостности, подлинности, доступности и наблюдаемости всем объектам. Подчеркнем, что $S_i$ значение тоже является частью общего секрета и может использоваться для выработки общего секрета. Она при необходимости может использоваться также как и $R_i$, и выдаваться объектам.

При необходимости общий секрет вырабатывается на основании предоставленных конфиденциальных частей $R_1, R_2, ..., R_n$ и $S_i$. Значение $S_i$ может быть также и контрольным значением и хранится у особо доверенного субъекта или объекта. При формировании общего секрета в специальном устройстве выработки общего секрета вычисляется сумма по модулю

$$\sum_{j=1}^{n} R'_j \oplus S'_i = \sum_{j=1}^{n} R'_j \oplus M_i \oplus \sum_{j=1}^{n} R_j, \; i = \overline{1,l}. \qquad (2)$$

Если все $R'_i = R_i$ и $S'_i = S_i$, то в результате вычисления (2) в специальном устройстве формируется общий секрет $M_i$, который может использоваться в качестве ключа, пароля, общего секрета состоятельного протокола, начального приближения вычисления ключа, команды на выполнение критического действия и др.

Рассматриваемая схема может обеспечивать различные уровни стойкости: безусловную стойкость, т.е. теоретическую недешифруемость, вычислительную стойкость, доказуемую [8] и даже временную стойкость. Уровень стойкости определяется свойствами частей секрета $R_i$. Так если $R_i$ есть случайная последовательность, то в системе при длине каждого $R_i$, $l_R > l_{ДОЛ}$, т.е. больше допустимой, обеспечивается безусловная стойкость и схема разделения *секрета абсолютно безопасная. Если* $R_i$ *– псевдослучайные последовательности,* то стойкость схемы определяется стойкостью этих последовательностей. По сути, если $R_i$ – случайные (хотя бы одна), то в схеме разделения секрета может быть обеспечена безусловная стойкость. если $j = 1$, т.е. используется одна $R_i$ последовательность, тогда как следует из (1) и (2), можно сформировать две части общего секрета $R_i$ и $S_i$. Если $R_i$ – случайное с длиной $l_R$, то единственной, а такой на схему будет атака «грубая сила», сложность которой определяется

$$I = 2^{\ell_R}. \qquad (3)$$

При разделении секретов на $n$ частей стойкость определяется как:

$$I_{CX} = n\left(2^{\ell_R}\right)^{n+1} = 2^{l_R(n+1)}. \qquad (4)$$

К основным недостаткам такой схемы можно отнести:

- необходимость полного доверия к доверительной стороне, в смысле отсутствия подозрений на его злоумышленные действия;
- зависимость стойкости схемы выработки общего секрета от режима секретности у доверенной стороны;
- невозможность проверки каждым из объектов подлинности и целостности частей секрета;
- при злоумышленном или непреднамеренном искажении части секрета, общий секрет восстановить невозможно.

**2. Пороговые схемы разделения секрета**

В пороговой схеме общий секрет делится на $n$-частей. Однако восстановление секрета может выполнено по $k \le n$ частным подлинным секретам [ ]. В этой схеме доверенная сторона также формирует общий секрет $S$ и из него вычисляет частные секреты $S_i$ каждого объекта $P_i$. При этом на $S_i$ накладываются также ограничения, чтобы каждые $k$ объектов, представив $k$ подлинных секретов $S_i$ могли бы вычислить общий секрет $S$. Более того, если при вычислении $S$ используется $k+v$ частных секретов, то $v$ из них могут быть ложными или искаженными, а $k$ подлинных всё равно обеспечивают формирование общего секрета. Подчеркнём, что на этапе разделения и использования секрета значения $S_i$ должны распространятся и хранится с обеспечением целостности, подлинности, конфиденциальности, доступности и наблюдаемости. Кроме того, в такой схеме ни одна

группа, знающая только $k-1$ частный секрет, восстановить S не может (безусловно или вычислительно).

В дальнейшем будем использовать также понятие совершенной пороговой схемы разделения секрета. Совершенной будем называть такую пороговую схему, в которой знание $k-1$ или менее частных секретов не даёт злоумышленнику никакой информации как о частных так и о общем секрете. Здесь никакой используется в информационно-теоретическом смысле. Это свойство обеспечивается и при многократном формировании общего секрета, однако в этом случае необходимо использовать разные частные секреты. Следует заметить, что в таких схемах нужно обеспечить управление доступом к общему секрету, например за счет использования доверенного устройства выработки общего секрета.

Построение известной пороговой схемы Ади Шамира базируется на полиномиальной интерполяции и на том факте, что одномерный полином $f(x)$ степени $k-1$ над полем Галуа *уникально задаётся по* $k$ *точкам. Полиномы могут быть заданы над* $p$*-ичным расширенным* полем. При этом коэффициенты полинома $f(x)=a_0+a_1x+a_2x^2+...+a_{k-1}x^{k-1}$, $a_i$ – задаются над полем $GF(p)$ как элементы поля $Z_p$. Основными параметрами такой схемы являются числа $(k,n)$, где $k$ есть минимальное число частей секрета, с использованием которых может быть восстановлен общий секрет, а $n$ – общее число долей секрета, причем $1\le k\le n$.

Коэффициенты $a_i$ определяются или задаются числом $n$ долей секрета. Затем случайным образом формируется общий секрет $S$, который должен быть разделен на доли секрета $S_i$, $i=\overline{1,n}$. Предлагаемая схема должна быть такой, чтобы любые $k$ объектов или субъектов, объединив свой $k$ частных секретов могли однозначно восстановить общий секрет $S$. При этом все доли секрета $S_i$ являются конфиденциальными, и на протяжении их жизненного цикла должны быть обеспечены целостность, подлинность, конфиденциальность, и доступность.

При выполнении приведенных выше требований и условий пороговая схема разделения секрета А. Шамира реализуется следующим образом.

1. Формируется большое простое число $P$, которое заведомо больше допустимого $P_{\text{Д}}$, т.е.
$$P>P_{\text{доп}}$$
2. Формируется случайным образом общий секрет $S$, который является элементом поля $GF(p)$, т.е. целое $S$ которое удовлетворяет условию.
$$1<S<P$$
3. Случайно формируется $k-1$ коэффициентов полинома $f(x)$ – $a_1,a_2,...,a_{k-1}$ которые объявляются конфиденциальными.
4. В качестве $a_0$ принимается значение общего секрета $S$, т.е. $a_0=S$.
5. Доверенная сторона разделяет общий секрет, вычислив доли секрета $S_i=f(i)$, где $i$ – числовой идентификатор или номер каждого из объектов или субъектов, причем $1\le i\le p-1$. Разделение секрета может заключаться в присвоении каждому из объектов или субъектов уникального случайного идентификатора.
6. Все доли секрета $S_i$ транспортируются и устанавливаются или вкладываются каждому из объектов или субъектов с обеспечением конфиденциальности, подлинности, целостности, доступности и наблюдаемости.

В дальнейшем мы рассмотрим отдельно алгоритм контроля подлинности каждой из частей секрета.

Восстановление общего секрета производится на основе использования не менее $k$ целостных и подлинных долей секрета, или $k+v \le n$ частных секретов, не более чем $v$ из которых могут быть сформированы объектом или субъектом злоумышленником или искажены. Эти $v$ и менее частных секретов с нарушенной целостностью обнаруживаются и не учитываются при выработке общего секрета. Восстановление общего секрета выполняется в следующем порядке.

1. Каждый из объектов(субъектов) передает и/или устанавливает частные секрет $S_i = f(i)$ в доверенное устройство выработки общего секрета с обеспечением конфиденциальности, целостности и подлинности.
2. Доверенное устройство контролирует целостность и подлинность частных секретов, если эта функция реализована в схеме разделения секрета, а затем выбирает из них $k$ подлинных.
3. По $k$ значениям $f(i_1),...,f(i_k)$ в доверенном устройстве производится восстановление $f(x)$ с использованием интерполяционной формулы Лагранжа:

$$f(x) = \sum_{e=1}^{k} f(i_e) \prod_{j \ne l} \frac{x - i_j}{i_e - i_j}. \qquad (5)$$

4. Общий секрет формируется в виде

$$S = a_0 = f(0).$$

В дальнейшем $S$ может использоваться в качестве ключа, пароля, общего секрета и др.

Таким образом выработка общего секрета в доверенном(исполняющем) устройстве производится на основе восстановления полинома $f(x)$, т.е. вычисления вектора коэффициентов $a_1, a_2, ..., a_{k-1}$, а затем определении общего секрета как $S = a_0 = f(0)$

Проведенный анализ показывает [ ], что свойства пороговой схемы разделения секрета Ади-Шамира позволяют построить протокол с нулевыми знаниями. При соответствующем выборе параметров знание $k-1$ значения $f(i_1),...,f(i_{k-1})$ не дает никакой информации об общем секрете. Его стойкость базируется на интерполяционной формуле Лагранжа, а также зависит от длины модуля преобразований $P$ и длин $S_i$-ых долей секрета. Рассмотрим возможные атаки на схему Шамира. Основной задачей атак является определение общего секрета $S = a_0$. Значение $a_0$ можно определить непосредственно или через определение значений частных секретов $f(l_1),...,f(l_k)$. Если $a_0 = S$ и формируется доверенной стороной случайно, то сложность атаки типа "грубая сила" по определению $a_0$ можно оценить через вероятность $P_0$ ее осуществления

$$P_0 = \frac{1}{p-2} \approx \frac{1}{p} = p^{-1}. \qquad (8)$$

Сложность атаки "грубая сила" по определению $a_0$ через значения $f(i_1), f(i_2),..., f(i_k) \in GF(p)$ можно оценить

$$P_f = \left( \frac{1}{(p-1)^k} \right) = (p-1)^{-k} \approx p^{-k}. \qquad (7)$$

Предварительные сравнения (6) и (7) показывают, что более предпочтительной является атака по непосредственному определению $a_0$. Сложность этой атаки зависит только от

величины модуля $p$. Если $p$ есть открытый общесистемный параметр, известный криптоаналитику, то сложность атаки можно определить так же через безопасное время

$$\mathrm{T_б} = T_б = \frac{I_0}{\zeta K} \approx \frac{P}{\zeta K} \tag{8}$$

где $I_0 \approx p$ есть число попыток подбора значения $a_0$ с вероятностью 1, $\zeta$ – производительность криптоаналитической системы, $K = 3{,}1 \cdot 10^7$ сек/год – количество секунд в году. При этом условии $T_б$ измеряется в годах. Если $a_0$ должно быть определено с вероятностью $\mathrm{P}_q$, то $T_б$ с такой вероятностью определяется из соотношения

$$T_б^{\mathrm{P}_q} = \frac{p}{\zeta K}\mathrm{P}_q. \tag{9}$$

В таблице 1 приведены значения $I_0 = p$ и $T_б$ при $\zeta_k = 10^{12}$ и $10^{16}$ вар/сек. (в знаменателе)

Сложность восстановления общего секрета схемы Ади – Шамира

**Таблица 1**

| $p$ | $2^{64}$ | $2^{128}$ | $2^{256}$ | $2^{512}$ | $2^{1024}$ |
|---|---|---|---|---|---|
| $T_б$ **(лет)** $P_д = 1$ | $\frac{6*10^{-1}}{6*10^{-5}}$ | $\frac{10^{19}}{10^{15}}$ | $\frac{4*10^{58}}{4*10^{54}}$ | $\frac{10^{134}}{10^{130}}$ | $\frac{10^{288}}{10^{284}}$ |
| $T_д$ **(лет)** $P_д = 10^{-16}$ | $\frac{6*10^{-17}}{6*10^{-21}}$ | $\frac{10^{3}}{10^{-1}}$ | $\frac{4*10^{42}}{4*10^{38}}$ | $\frac{10^{118}}{10^{114}}$ | $\frac{10^{272}}{10^{268}}$ |

Анализ данных таблицы показывает что применение значения $T_б$ для криптографических преобразований достигаются уже при величине модуля $p$ порядка $2^{256}$. Так из таблицы следует, что при длине модуля $p = 2^{256}$ вероятность с которой может осуществлён криптоанализ с $\mathrm{P} = 10^{-16}$ и производительности криптоаналитической системы $10^{16}$, безопасное время составляет не менее $10^{38}$ лет. Поэтому в перспективных схемах разделения секрета величины модулей $p$ должны составлять порядка $2^{256} - 2^{512}$.

Основным и свойством и пороговой схемы Ади-Шамира являются следующие

1. Совершенность – при знании любых $k-1$ и менее долей секрета $S_i$ все значения общего секрета $S$ остаются равновероятными и теоретически могут выбираться из интервала $0 \le S \le p-1$.
2. отсутствие не доказанных допущений. В отличие от вероятностно-стойких схем, схема А. Шамира не базируется ни на каких недоказанных допущениях (например сложности решения таких задач как факторизация модуля, нахождения дискретного логарифма и т.д.).
3. Расширяемость при появлении новых пользователей. Это свойство заключается в том, что новые части секрета могут быть вычислены и распределены без влияния на уже существующие части.

4. Идеальность, под которой понимается тот факт, что все части общего секрета и сам общий секрет имеют одинаковый размер и могут принимать значения над полем $GF(p)$ с равной вероятностью.

Особенностью пороговой схемы разделения секрета является то, что она требует выполнения модульных операций над большим полем $GF(p)$, сложность которых имеет полиномиальный характер. Кроме того, доверенное устройство должно иметь возможность контролировать целостность и подлинность частей секрета перед выработкой общего секрета.

**3. Конструкция и свойство протокола проверяемого секрета.**

Вначале рассмотрим конструкцию протокола проверяемого разделения секрета над простым полем Галуа $GF(p)$. Предназначением этого протокола является проверка целостности и подлинности каждой из частей секрета, а также проверка частей секретов при их поступлении в доверенное устройство, т.е. перед выработкой общего секрета. Протокол может быть построен следующим образом. Доверенная сторона выбирает случайный полином

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1} (\mathrm{mod}\, p).$$

Как и ранее $a_0 = S$.

Все объекты или субъекты системы знают общесистемные параметры $\theta_v$ и $p_j$; где $\theta_v$ – первообразный элемент поля $GF(p_j)$, причем $p_j$ и $p$ разные простые числа соответствующего размера. Затем доверенная сторона по $a_i$ вычисляет отдельные составляющие

$$R_i = \theta_v^{a_i} (\mathrm{mod}\, p),\ i = \overline{1, k-1} \qquad (10)$$

Открытые составляющие преобразуются в сертификаты, которые опубликовываются или хранятся в общедоступной базе сертификатов (данных) и являются доступными субъектам и объектам, которые разделяют секрет.

После этого вычисляются части секрета $S_i = f(i)$, для необходимого числа $n$ объектов(субъектов). Далее части секрета доставляются всем объектам(субъектам), разделяющим секрет, с обеспечением конфиденциальности, подлинности, целостности, наблюдаемости и доступности.

Каждый из объектов(субъектов) может проверить подлинность и целостность своей части секрета, проверяя равенство

$$\theta_v^{S_i} = R_0 \cdot (R_1)^{i} \cdot (R_2)^{i^2} \cdot \ldots \cdot (R_{k-1})^{i^{k-1}} (\mathrm{mod}\, p_j) \qquad (11)$$

$$\Theta_v^{S_i} = \mathrm{R_0}*(\mathrm{R_1})^{\mathrm{I}} * (\mathrm{R_{1\backslash 2}})^{i^2} * \ldots * (\mathrm{R_{k\text{-}1}})^{i^{k-1}} (\mathrm{mod}\ \mathrm{P_j}) \quad (**)$$

Подставив (10) в (11) имеем

$$R_0 \cdot R_1^{i} \cdot R_2^{i^2} \cdot \ldots \cdot R_{k-1}^{i^{k-1}} = \theta_v^{a_0} \cdot \theta_v^{a_1 i} \cdot \theta_v^{a_2 i^2} \cdot \ldots \cdot \theta_v^{a_{k-1} i^{k-1}} = \theta^{a_0 + a_1 i + a_2 i^2 + \ldots + a_{k-1} i^{k-1}} (\mathrm{mod}\, p) = \theta_v^{f(i)} (\mathrm{mod}\, p) \qquad (12)$$

Значит $S_i = f(i)$ и по набору $R_i$ обеспечивается проверка частных секретов $S_i$.

Таким образом, каждый объект, используя только свою часть секрета $S_i$, общие для системы параметры $\theta_v$ и $p_i$, а так же базу открытых ключей $R_0, R_1, \ldots, R_{k-1}$, может проверить целостность и подлинность своего части секрета. Рассмотренный протокол обеспечивает

контроль целостности и подлинности частных секретов, т.е. от различных злоумышленных действий доверенной стороны и в процессе их транспортировки.

Восстановление секрета. Если доверенное устройство не является злоумышленником, то протокол восстановления общего секрета выполняется следующим образом.

Каждый объект $A_j$ посылает доверенному устройству – объекту $A_v$ свою часть $S_i$ секрета, обеспечивая его целостность, подлинность, конфиденциальность, наблюдаемость и доступность. Доверенная сторона может проверить подлинность и целостность всех принимаемых $S$ частей секрета, используя описанный выше алгоритм выражения (10) – (11).

Части секрета, которые не прошли проверку, не используются. Если честных объектов предоставивших частные секреты, не менее чем $K$, то доверенное устройство получает не менее $K$ частей общего секрета и может восстановить общий секрет используя схему Шамира, описанную выше.

**4. Другие схемы разделения секрета**

Среди других схем разделения секрета необходимо выделить схему Беркли и схемы построенные на основе кодов Рида Саломона.

Схема разделения секрета Блэкли имеет геометрическую основу [5]. Секрет представляет собой точку в $k$-мерном пространстве. При этом для случая $k > 2$ все геометрические построения выполняются над конечным полем $GF(p)$.

Каждая из $n$ проекций задается как гиперплоскость в $m$-мерном пространстве. Определение секрета сводится к нахождению точки пересечения $m$ гиперплоскостей. В исходном виде схема не является совершенной. На рис. 1 представлен специальный случай схемы Блэкли.

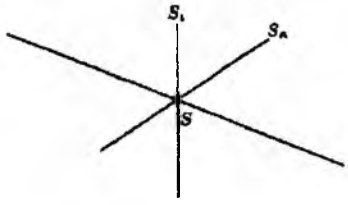


Рисунок 1. Схема Блэкли

Для восстановления секрета необходимо иметь две проекции. Секрет задается как точка на плоскости. Каждая проекция — прямая, проходящая через эту точку. Таким образом, секрет может быть получен по двум проекциям (как точка пересечения двух прямых).

Схема Шамира тесно связана с кодами Рида-Соломона, широко известными в теории помехоустойчивого кодирования [ ]. Обозначим через $(a_1, a_2, ..., a_{k-1})$ список ненулевых элементов конечного поля $GF(p)$. Последовательность входных символов $a = (a_0, a_1, ..., a_{p-1})$,

$$a_i \in GF(p)$$

кодируется в кодовое слово $D = (D_1, D_2, ..., D_{p-1})$, где $a_i$ примитивный элемент поля

$$a_0 = -\sum_{i=1}^{p-1} D_i .$$

Секрет задается как

$$D_i = \sum_{j=0}^{k-1} a_j a_i^j .$$

В качестве долей выбираются $D_i$. Предположим, $t$ из $s$ проекций содержат ошибки (ошибки, например, могли возникнуть при передаче по каналу связи). Согласно существующей модели, ошибки, возникающие в канале связи, рассматриваются как сумма по модулю 2 ошибочного значения и символа кодового слова. Таким образом, суть процедуры декодирования с исправлением ошибок заключается в определении позиции и значения ошибки (для кода Рида-Соломона и то и другое – элементы поля). Помимо ошибок

различают стирания. В отличие от ошибок при исправлении стираний позиции известны Применяя алгоритм декодирования с исправлением ошибок и стираний, можно гарантированно восстановить $D$, а следовательно, и $a_0$, при условии что $s-2t \geq k$. Схема Шамира представляет собой специальный случай кода Рида-Соломона, где p простое число, $a_i = i$ и $t = 0$. Рассмотрим ситуацию, в которой злоумышленник препятствует получению секрета легальными пользователями. Для этого он искажает $D_i$. Однако секрет будет восстановлен при условии, что кроме $t$ искаженных, легальные пользователи предоставят $k$ неискаженных проекций. В общем случае для блокирования работы *(k,n)* пороговой схемы злоумышленник должен исказить более $\lfloor (n-k)/2 \rfloor$ долей. Алгоритм декодирования может применяться для борьбы с ошибками, возникающими вследствие несовершенства носителей (магнитных дисков, лент и т.д.) и устройств хранения информации. Применение декодирования не приводит к существенному возрастанию вычислительной сложности схемы Шамира. Вычислительная сложность алгоритма декодирования с исправлением ошибок и стираний (алгоритм Берлекэмпа-Мэсси в модификации Форни) оценивается по сложности как $O(n^2)$ (известен и более эффективный алгоритм со сложностью $O(n\log^2 n)$ [3].

**Заключение**

Приведём протокол разделения секрета в системах управления ключами и управления доступом на рисунке 2.

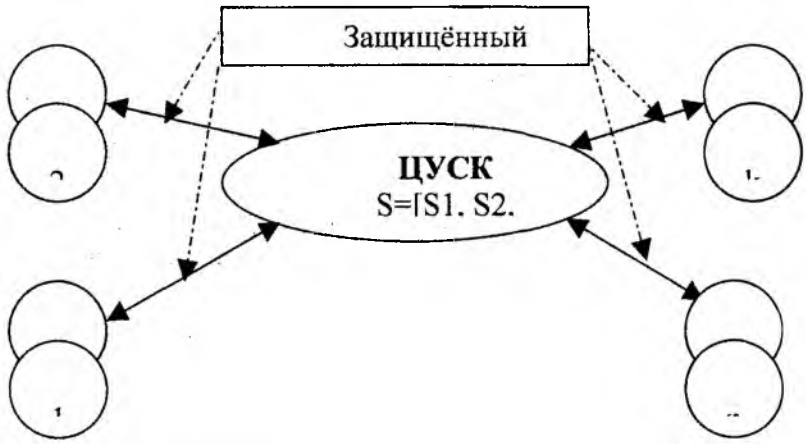



Рисунок 2. Схема установления секретного симметричного ключа путём разделения секрета по схеме Ади - Шамира

Проведённый сравнительный анализ показал, что наиболее предпочтительный криптопротокол разделения секрета, с точки зрения стойкости и надежности, является протокол Ади-Шамира, схема которого является совершённой. Протоколы такого класса могут использоваться в системах и технологиях, в которых требуется совместное управление доступом к критической информации и ресурсам.

**Список литературы:** 1. Simmons G.J. Contemporary Cryptology – The Science of Information Integrity. IEEE Press, 1992. 2. Shamir A. How to share a secret //Coom. Of the ACM, 1979, v22, № 11, pp. 612-613. 3. Rivest R.L. Multigrade cryptography Manuscript, 1996, ftp://theory.Ics.mit.edu/~rivest/publication.html. 4. McEliece R.J., Sarvate D.V. On sharing secrets and Reed-Solomon codes // Comm. of the ACM, Aug 1981, v. 24, № 9, pp. 583-584. 5. Blakley G.R. Safeguarding cryptographic keys // Proc of AFIPS National Computer Conference, 1979, 48, pp. 313-317. 6. Liu C.L. Introduction to combinatorial mathematics. New-York McGraw-Hill. 1968. 7. DeSantis A., Desmedt Y., Frankel Y., Yung M. How to share a function securely // Proc. of the 26th ACM Symposium on the Theory of Computing, 1994, pp. 522-533. 8. Вильямс криптография и защита систем. Принципы и практика 2-е издание. Изд. Москва-Санкт-Петербург-Киев 2001, 669с.

*Харьковский национальный университет радиоэлектроники* *Поступила в редколлегию 13.04.2002*

УДК 681.3.06:519.248.681

*Т.А. ГРИНЕНКО, С.И. ЗБИТНЕВ, Д.В. МЯЛКОВСКИЙ*

# МЕТОДЫ ФОРМИРОВАНИЯ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ

## Введение

Выходные данные генераторов случайных (ГСЧ) и псевдослучайных чисел (ГПСЧ) используются во многих криптографических приложениях, например, при генерации ключей, общесистемных параметров и др. В соответствии с требованиями криптографических приложений эти генераторы должны удовлетворять ряду сложных и противоречивых требований. На наш взгляд, этим требованиям в значительной мере могут удовлетворять ГПСЧ, реализованные на основе использования преобразований в группах точек эллиптических кривых. Метод и конкретные алгоритмы построения ГПСЧ в группах точек эллиптических кривых рассмотрены в [1].

В общем случае для построения генератора псевдослучайной последовательности (ПСП) используется односторонняя функция. Для построения таких односторонних функций используются функции, сложность которых основывается на сложности дискретного логарифма [2] или на сложности факторизации большого числа [3].

В данной статье мы рассмотрим ГПСП, основывающийся на сложности дискретного логарифма в группах точек эллиптической кривой. В случае Блюм – Микали сложность дискретного логарифма заключается [4] в сложности нахождения целого $a$, такого что $y \equiv g^a \pmod p$. В нашем случае сложность дискретного логарифма на эллиптической кривой заключается в сложности нахождения целого $d$ из сравнения $Q = d \times G \pmod q$, где $Q$, $G \in E(\mathrm{F}_q)$ и являются известными.

Математический аппарат в группе точек эллиптической кривой (ЭК) получил широкое признание и находит применение при реализации цифровых подписей для направленного шифрования и в ряде состоятельных протоколов управления ключами. В [1] предложен конкретный метод и алгоритмы построения ПСП в группах точек ЭК над простым полем $GF(P)$. Представляет интерес решение этой задачи на общий случай расширенного поля в различных представлениях ЭК и с различными реализациями алгоритмов построения ПСП. Кроме того, необходимо провести более глубокие исследования статистических характеристик различных реализаций, а также сравнительного анализа различных методов и алгоритмов построения ПСП, их сравнения с уже ставшим классическим BBS-генератором [5].

Существует несколько методов построения ПСП в группах точек ЭК [1].

## 1. Методы построения ПСП в группах точек ЭК

Пусть даны эллиптические кривые: $E_A$

$$y^2 + xy = x^3 + ax^2 + b \pmod{f(x), 2}, \tag{1}$$

определенная в аффинных координатах и $E_P$

$$Y^2 + XYZ = X^3 + aX^2Z^2 + bZ^6 \pmod{f(x), 2}, \tag{2}$$

определенная в проективных координатах. Пусть дана базовая точка $G$ с координатами $(x, y) \in E_A$ или $(X, Y, Z) \in E_P$. Рассмотрим два метода построения ПСП:

$$Z_i = Z_{i-1} + G, \text{ где } Z_i, Z_{i-1} \in E; \tag{3}$$

$$Z_i = a \times Z_{i-1}, \text{ где } Z_i, Z_{i-1} \in E. \tag{4}$$

В первом случае мы получаем последовательность значений $Z_i$ путем многократного суммирования точки $Z_{i-1}$ и базовой точки $G$, а во втором – путем скалярного умножения точки $Z_{i-1}$ на число $a$. Во втором случае возникает вопрос выбора $a$, оно может быть константой либо получаться из другого ГПСП, в нашем случае $a=\pi(Z_{i-1})$, где $\pi$ – функция преобразования $x/X$-координаты в число. Таким образом, мы получаем $Z_i=\pi(Z_{i-1})\times Z_{i-1}$.

Для обоих методов построение ПСП можно выполнить несколькими способами. Основными из них, на наш взгляд, являются:

$$Num(Z_i)=X_i\,||\,Y_i\,||\,Z_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{5.1}$$

$$Num(Z_i)=X_i\,||\,Y_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{5.2}$$

$$Num(Z_i)=X_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{5.3}$$

$$Num(Z_i)=x_i\,||\,y_i, \text{ если } Z_i(x_i,y_i)\in E_A, \tag{5.4}$$

$$Num(Z_i)=x_i, \text{ если } Z_i(x_i,y_i)\in E_A. \tag{5.5}$$

В выражениях (5.1)-(5.5) знак $||$ – конкатенация значений координат точек ЭК, а $Num(Z_i)$ – обозначение способа формирования ПСП.

Проведенный анализ показал, что сложность (скорость) функционирования ГПСП зависит от выбранного метода и способа формирования ПСП. Минимальная сложность достигается для метода (3) и способа формирования ПСП (5.1). В этом случае за один шаг ГПСП формируется псевдослучайное число трехкратной длины.

Координаты точки имеют корреляционную функцию, соответствующую уравнению эллиптической кривой, что может привести к корреляции значений ПСП. Проведенный анализ показал, что декорреляцию можно осуществить посредством вычисления значений хэш-функций от текущего числа $Num(Z_i)$.

С учетом (3) и (4), а также пяти способов формирования чисел (5.1)-(5.5) проведены исследования следующих конкретных алгоритмов формирования ПСП:

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=X_i\,||\,Y_i\,||\,Z_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.1}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=X_i\,||\,Y_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.2}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=X_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.3}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=x_i\,||\,y_i, \text{ если } Z_i(x_i,y_i)\in E_A, \tag{6.4}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=x_i, \text{ если } Z_i(x_i,y_i)\in E_A, \tag{6.5}$$

$$Z_i=\pi(Z_{i-1})\times Z_{i-1},\ Num(Z_i)=X_i\,||\,Y_i\,||\,Z_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.6}$$

$$Z_i=\pi(Z_{i-1})\times Z_{i-1},\ Num(Z_i)=X_i\,||\,Y_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.7}$$

$$Z_i=\pi(Z_{i-1})\times Z_{i-1},\ Num(Z_i)=X_i, \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.8}$$

$$Z_i=\pi(Z_{i-1})\times Z_{i-1},\ Num(Z_i)=x_i\,||\,y_i, \text{ если } Z_i(x_i,y_i)\in E_A, \tag{6.9}$$

$$Z_i=\pi(Z_{i-1})\times Z_{i-1},\ Num(Z_i)=x_i, \text{ если } Z_i(x_i,y_i)\in E_A, \tag{6.10}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=H(X_i\,||\,Y_i\,||\,Z_i), \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.11}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=H(X_i\,||\,Y_i), \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.12}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=H(X_i), \text{ если } Z_i(X_i,Y_i,Z_i)\in E_P, \tag{6.13}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=H(x_i\,||\,y_i), \text{ если } Z_i(x_i,y_i)\in E_A, \tag{6.14}$$

$$Z_i=Z_{i-1}+G,\ Num(Z_i)=H(x_i), \text{ если } Z_i(x_i,y_i)\in E_A, \tag{6.15}$$

$$Z_i = \pi(Z_{i-1}) \times Z_{i-1},\ \ Num(Z_i) = H(X_i \| Y_i \| Z_i), \text{ если } Z_i(X_i, Y_i, Z_i) \in E_P, \quad (6.16)$$

$$Z_i = \pi(Z_{i-1}) \times Z_{i-1},\ \ Num(Z_i) = H(X_i \| Y_i), \text{ если } Z_i(X_i, Y_i, Z_i) \in E_P, \quad (6.17)$$

$$Z_i = \pi(Z_{i-1}) \times Z_{i-1},\ \ Num(Z_i) = H(X_i), \text{ если } Z_i(X_i, Y_i, Z_i) \in E_P, \quad (6.18)$$

$$Z_i = \pi(Z_{i-1}) \times Z_{i-1},\ \ Num(Z_i) = H(x_i \| y_i), \text{ если } Z_i(x_i, y_i) \in E_A, \quad (6.19)$$

$$Z_i = \pi(Z_{i-1}) \times Z_{i-1},\ \ Num(Z_i) = H(x_i), \text{ если } Z_i(x_i, y_i) \in E_A. \quad (6.20)$$

**2. Экспериментальные исследования свойств ПСП**

Для сравнения ПСП были выбраны эллиптическая кривая $E$ над полем $F_{2^{191}}$, примитивный полином $f(x) = 0x\,800000000000000000000000000000000000000000000201$, параметры ЭК $a = 0x\,2866537B676752636A68F56554E12640276B649EF7526267$, $b = 0x\,2E45EF571F00786F67B0081B9495A3D95462F5DE0AA185EC$, базовая точка $G$ с координатами $x/X = 0x\,36B3DAF8A23206F9C4F299D7B21A9C369137F2C84AE1AA0D$, $y/Y = 0x\,765BE73433B3F95E332932E70EA245CA2418EA0EF98018FB$, $Z = 1$, порядок базовой точки $n = 0x\,400000000000000000000004a20e90c39067c893bbb9a5$, кофактор $k = 2$.

Для тестирования генератора ПСЧ на эллиптических кривых использовалась методика NIST STS [6], рекомендованная Национальным институтом по стандартизации и технологиям США, разработанная для статистического тестирования алгоритмов-кандидатов на AES ( NIST SP 800-22). Пакет NIST STS включает в себя 16 статистических тестов. Эти тесты используются для проверки гипотезы о случайности двоичных последовательностей произвольной длины, порождаемых ГСЧ или ГПСЧ. По совокупности результатов всех тестов принимается решение о том, будет ли заданная последовательность нулей и единиц «случайной» или нет.

Рассмотрим двоичную последовательность $S = \{s_1, s_2, \ldots, s_n\}$, $s_i \in \{0,1\}$ произвольной длины $n$ бит. Для фиксированного значения $n$ сформируем множество из $m$ двоичных последовательностей, т.е. сформируем выборку объемом $N = m \times n$. Каждую последовательность подвергаем тестированию с использованием пакета NIST STS. В результате тестирования сформируем статистический портрет генератора $P_{ij} \in [0,1]$, $i = \overline{1,m}, j = \overline{1,q}$ ($m$ – количество тестируемых двоичных последовательностей, $q$ – количество статистических тестов). По полученному статистическому портрету определяем долю последовательностей, прошедших каждый статистический тест. Для этого задаем уровень значимости $\alpha \in [0{,}001; 0{,}01]$ и осуществляем подсчет значений вероятности $P$, превышающих заданный уровень $\alpha$ для каждого из $q$ тестов. В результате сформируем вектор коэффициентов $R = \{r_1, r_2, \ldots r_q\}$, элементы которого характеризуют в процентном соотношении прохождение последовательности $S_i$ всех статистических тестов. Далее осуществляем статистический анализ статистического портрета. Полученные значения вероятностей $P_{ij} \in [0,1]$ должны подчиняться равномерному закону распределения на интервале [0,1]. Наконец, принимаем окончательное решение – прошел ли генератор статистическое тестирование пакетом NIST STS или нет.

С использованием методики NIST STS было осуществлено тестирование 20 ПСП (алгоритмы (6.1)-(6.20), а также проведено сравнение свойств этих ПСП со свойствами ПСП генератора псевдослучайных чисел BBS [6] (тестовая выборка, рекомендованная NIST).

Для осуществления тестирования были выбраны следующие параметры:

1. Длина тестируемой последовательности $n = 10^6$ бит.

2. Количество тестируемых последовательностей $m$ = 100. Таким образом, объем тестируемой выборки составил $N = 10^6 \times 100 = 10^8$ бит.

3. Уровень значимости $\alpha$ = 0,01

4. Количество тестов $q$ = 189. Таким образом, статистический портрет генератора содержит 18900 значений вероятности P.

В идеальном случае при $m$ = 100 и $\alpha$ = 0,01 может быть отвергнута только одна последовательность из ста, т.е. коэффициент прохождения каждого теста должен составлять 99%. Но это слишком жесткое правило. Поэтому и применяется правило на основе доверительного интервала для $r_j$. Нижняя граница в этом случае составит значение $r_{\min}$ = 0,96015. С этих позиций проанализируем результаты тестирования ПСП, представленные на диаграммах (рис. 1-7).

В табл. 1 приводятся данные по прохождению ПСП тестов по Правилу 1 [6].

**Таблица 1**

| Генератор | Количество тестов, в которых тестирование прошли более 99% последовательностей | Количество тестов, в которых тестирование прошли более 96% последовательностей |
|---|---|---|
| BBS | 134 (70,8%) | 189 (100%) |
| 6.1 | 119 (63%) | 175 (92,6%) |
| 6.2 | 126 (66,7%) | 171 (90,5%) |
| 6.4 | 128 (67,7%) | 181 (95,8%) |
| 6.5 | 137 (72,5%) | 187 (98,9%) |
| 6.9 | 118 (62,4%) | 180 (95,2%) |
| 6.10 | 123 (65,1%) | 187 (98,9%) |
| 6.11 | 139 (73,5%) | 187 (98,9%) |
| 6.12 | 141 (74,6%) | 188 (99,5%) |
| 6.13 | 138 (73%) | 189 (100%) |
| 6.14 | 134 (70,9%) | 189 (100%) |
| 6.15 | 124 (65,6%) | 187 (98,9%) |
| 6.16 | 126 (66,7%) | 188 (99,5%) |
| 6.17 | 146 (77,2%) | 188 (99,5%) |
| 6.18 | 131 (69,3%) | 189 (100%) |
| 6.19 | 121 (64%) | 187 (98,9%) |
| 6.20 | 127 (67,2%) | 188 (99,5%) |

Генераторы (6.13), (6.14), (6.18) прошли все тесты. Генератор BBS прошел все тесты. Если применять жесткий критерий, т.е. когда может быть отброшена лишь одна последовательность из ста, то лучший результат показал генератор (6.13), он имеет лучшие характеристики, чем BBS. Генератор (6.14) имеет такую же статистику, как и BBS.

В табл. 2 представлены сводные результаты по прохождению генераторами тестов по Правилу 2 [6].

**Таблица 2**

| Генератор | Количество тестов, в которых значение вероятности $P \le 0,01$ | Количество тестов, у которых значение вероятности $P \le 0,001$ |
|---|---|---|
| BBS | 0 | 0 |
| 6.1 | 19 | 16 |
| 6.2 | 28 | 18 |
| 6.4 | 21 | 14 |
| 6.5 | 4 | 2 |
| 6.9 | 11 | 7 |
| 6.10 | 6 | 4 |
| 6.11 | 2 | 0 |
| 6.12 | 5 | 0 |
| 6.13 | 2 | 0 |
| 6.14 | 3 | 2 |
| 6.15 | 1 | 0 |
| 6.16 | 1 | 0 |
| 6.17 | 4 | 0 |
| 6.18 | 2 | 0 |
| 6.19 | 2 | 1 |
| 6.20 | 2 | 1 |

В таблице значения вероятности $P$ сравниваются с уровнями значимости $\alpha = 0{,}01$ и $\alpha = 0{,}001$, т.к. это достаточно малые значения.

Для (6.13) малые значения вероятности $P$ получены:

- для неперыкрывающихся шаблонов $P = 0{,}008879$,
- для проверки случайных отклонений $P = 0{,}004301$.

Однако эти значения не совпадают с отрицательными выводами по правилу один.

Для (6.14) малые значения вероятности $P$ получены:

- для последовательного теста $P = 0{,}009535$,
- для проверки случайных отклонений $P = 0{,}005166$,
- для проверки случайных отклонений (вариант) $P = 0{,}004045$,
- для неперыкрывающихся шаблонов $P = 0{,}00017$ и $P = 0{,}000082$.

Для (6.18) малые значения вероятности $P$ получены:

- для неперыкрывающихся шаблонов $P = 0{,}007160$ и $P = 0{,}001112$.

Однако эти значения не совпадают с отрицательными выводами по правилу один.

На рис. 1-7 представлены статистические портреты некоторых генераторов ПСП с указанием их параметров и способов формирования.

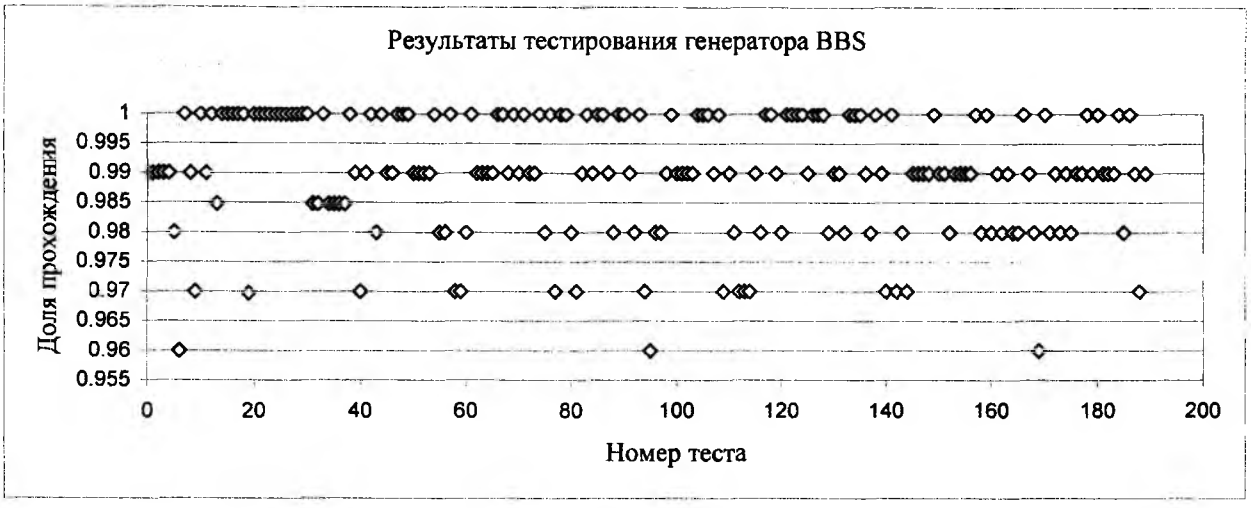



Рис. 1

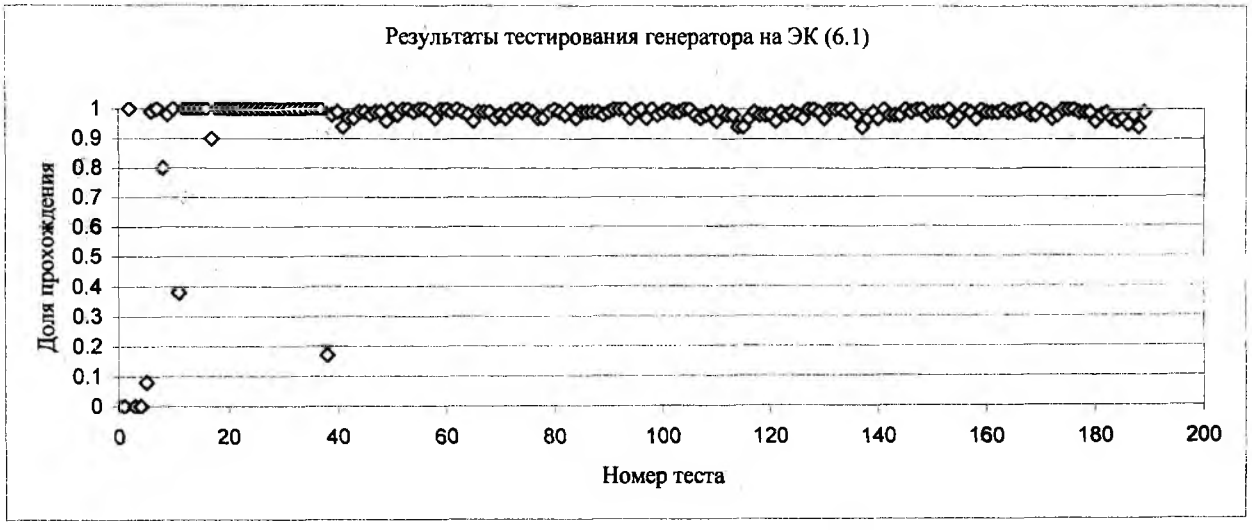



Рис.2

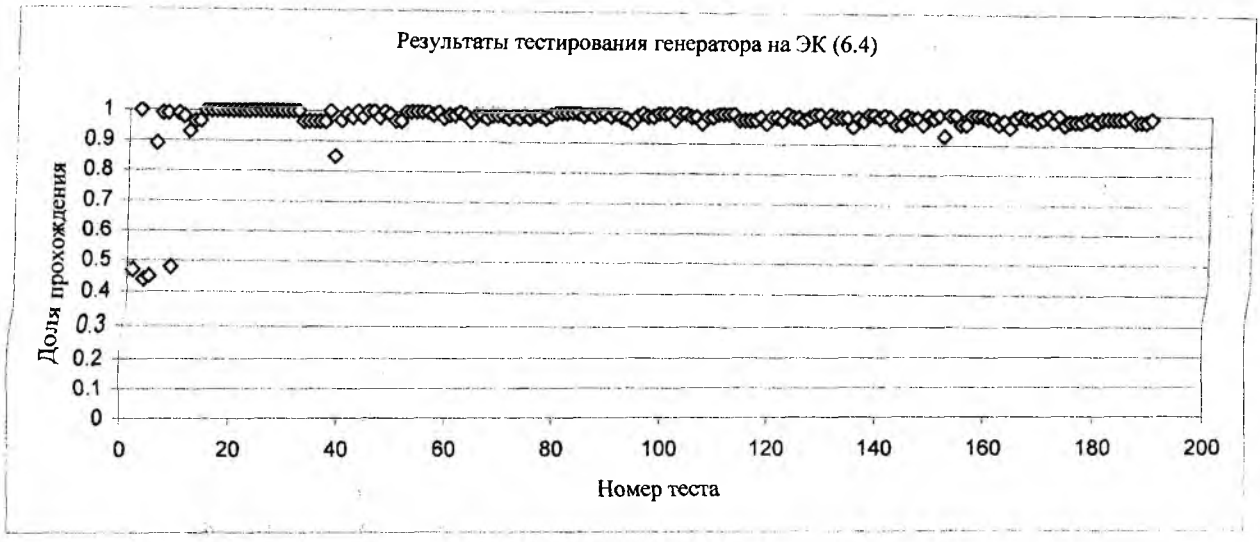



Рис.3

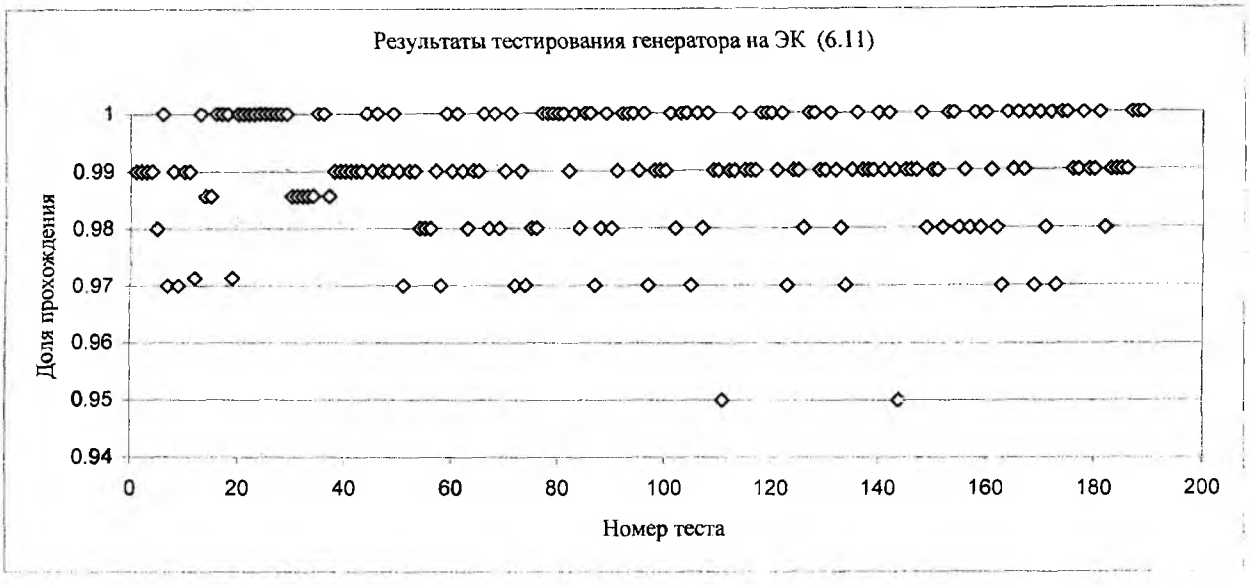



Рис.4


Рис.5


Рис.6

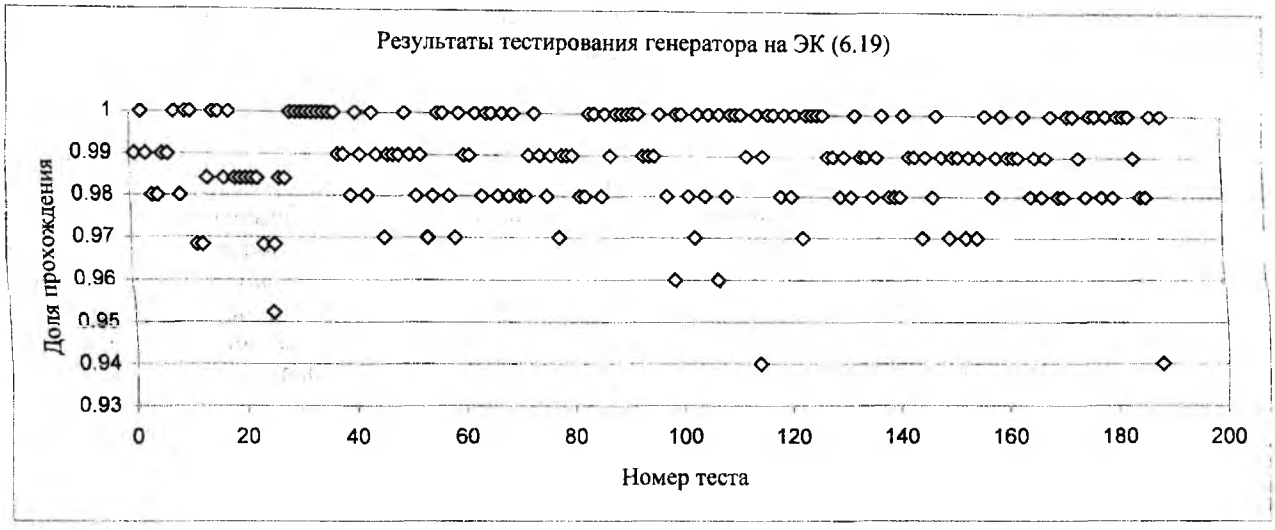



Рис.7

В табл.3 приведены результаты экспериментальной оценки скорости формирования ПСП для различных алгоритмов.

**Таблица 3**

| Генератор | Количество полученных битов за секунду |
|---|---|
| 6.1 | 57 600 000 |
| 6.2 | 38 400 000 |
| 6.3 | 19 200 000 |
| 6.4 | 93 203 |
| 6.5 | 46 601 |
| 6.6 | 15 368 |
| 6.7 | 10 245 |
| 6.8 | 5 122 |
| 6.9 | 390 |
| 6.10 | 195 |
| 6.11 | 12 800 000 |
| 6.12 | 12 800 000 |
| 6.13 | 12 800 000 |
| 6.14 | 63 366 |
| 6.15 | 63 366 |
| 6.16 | 6 835 |
| 6.17 | 6 835 |
| 6.18 | 6 835 |

*Окончание табл. 3*

| | |
|---|---|
| 6.19 | 258 |
| 6.20 | 258 |

Наибольшую скорость обеспечивают генераторы (6.1), (6.2), (6.3), (6.11), (6.12), (6.13). В тоже время у генератора (6.13) и лучшие статистические характеристики случайности.

**Заключение**

Применение математического аппарата групп точек позволяет построить различные генераторы ПСП. Основными методами формирования ПСП являются методы, основанные на операциях сложения и умножения в группах точек эллиптических кривых.

Результаты статистического тестирования предложенных алгоритмов с использованием методики NIST SP 800-22 показали, что лучшими являются генераторы (6.13), (6.14) и (6.18). Причем наилучшим является генератор (6.13), характеристики которого даже лучше, чем у принятого в качестве классического BBS генератора.

Генератор ПСП, построенный по алгоритму (6.13), кроме вполне приемлемых статистических характеристик, обеспечивает приемлемую сложность (скорость) формирования псевдослучайных чисел. Так на ПЭВМ Celeron 600 скорость составляет порядка 12 800 000 бит./с.

**Список литературы:** 1. *Гриненко Т.А., Горбенко Ю.И., Орлова С.Ю.* Метод формирования и свойства псевдослучайных последовательностей на эллиптических кривых // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С. 119-123. 2. *Leonard Adleman.* A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In Proceeding of the 20th Annual Symposium on Foundation of Computer Science, page 55-60, IEEE Computer Society, 1979. 3. *Werner Alexi, Benny Chor, Oded Goldreich, and Claus P.Schnorr.* RSA and Rabin functins: Certain parts are as hard as the whole. To appear, SIAM Journal of Computing. 4. *Manuel Blum and Silvio Micali.* How to generate cryptographically strong sequences of psevdo-random bits. SIAM Journal of Computing, 13 (4): 850-864, 1984. 5. *Alfred Menezes, et. al.* Handbook af Applied Cryptography – CRC Press, 1997. 6. *Потий А.В., Орлова С.Ю., Гриненко Т.А.* Статистическое тестирование генераторов случайных и псевдослучайных чисел с использованием набора статистических тестов NIST STS // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2001. Вип. 2. С. 206-214.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 29.04.2002*

УДК 681.3.06:519.248.681

*С.И. ЗБИТНЕВ*

# ПРОЕКТИВНАЯ ГЕОМЕТРИЯ – НЕ ВСЕ ТАК ГЛАДКО

## Введение

По современным взглядам разрешение ряда противоречий несимметричной криптографии может быть осуществлено за счет использования криптографических преобразований в группах точек эллиптической кривой [15]. Если эллиптическая кривая (ЭК) удовлетворяет условию MOV [1,2] и FR-условию [3] и свободна от $p$-делителя над $F_{p^r}$ [4,5,6], то для криптопреобразований в группах точек ЭК известны три атаки на дискретный логарифм в группах точек ЭК – методы $\rho$-Полларда[7,8], Полига-Хелмана [9] и малых – больших шагов [9, 17]. Криптосистемы на основе эллиптических кривых с длиной ключа 160 бит имеет одинаковую стойкость с криптосистемами ElGamal и RSA с длиной ключа 1024 бита [16]. По этой причине криптосистемы на ЭК обсуждались в ISO/IEC CD 14883-3, ISO/IEC DIS 11770-3, ANSI ASC X.9.63, X.9.62, IEEE p1363 [2], ГОСТ-34.2001 РФ и NESSIE и в настоящее время получили развитие.

Несмотря на уменьшения длины блока преобразования, основные операции в группах точек ЭК требуют значительных вычислительных затрат. Поэтому важным является уменьшение вычислительной сложности преобразований в группах точек ЭК. Наиболее распространенными методами уменьшения вычислительной сложности являются:

- Использование специфических кривых, в которых в ущерб стойкости мы достигаем существенного повышения быстродействия [18,19,20];
- Использование эллиптических кривых, определенных над $F_q$. При программной реализации достигается максимальная производительность, если $q = p$, а при $q = 2^m$ достигается существенное ускорение аппаратной реализации криптопреобразования [2];
- Использование различных базисов представления элементов поля для $E(F_{2^m})$. Полиномиальное представление эффективней при программной реализации преобразований на ЭК, а нормальный базис Гауса предпочтительней при аппаратной реализации [2];
- Различное представление точек на ЭК [2,10-14].

Последний способ позволяет повысить производительность на порядок, без ущерба безопасности криптосистем, и поэтому, возможно, является основным в решении задач уменьшения сложности преобразований в группе точек ЭК.

Целью настоящей статьи является проведение сравнительного анализа вычислительной сложности операций сложения в группах точек ЭК в различных известных координатных базисах, а также определение основных ограничений и условий применения различных координатных базисов при выполнение криптографических преобразований в группах точек эллиптических кривых.

## 1. Представление точек на эллиптической кривой

Точка на эллиптической кривой может быть представлена в нескольких координатных базисах. Основными из них являются аффинные координаты [10], проективные координаты [11], якобиановые координаты [12,13], координаты Чудновского(Chudnovsky Jacobian) [12] и модифицированные якобиановые координаты [14].

**Метрика сложения в аффинных координатах [10].** Пусть

$$E: y^2 = x^3 + ax + b\left(a, b \in \mathrm{F}_p, 4a^3 + 27b^2 \neq 0\right) - \tag{1}$$

уравнение эллиптической кривой $E$ над $\mathrm{F}_p$. Пусть даны точки $P_1 = (x_1, y_1) \in E(\mathrm{F}_p)$ и $P_2 = (x_2, y_2) \in E(\mathrm{F}_p)$, тогда суммой двух точек $P_1$ и $P_2$ называется точка $P_3 \in E(\mathrm{F}_p)$, такая что $P_3 = P_1 + P_2 = (x_3, y_3)$. Если $P_1 \neq P_2$, то координаты точки $P_3 = (x_3, y_3) = P_1 + P_2 = (x_1, y_1) + (x_2, y_2)$ образуются как в [10]

$$x_3 = \lambda^2 - x_1 - x_2 \pmod p,\ y_3 = \lambda(x_1 - x_3) - y_1 \pmod p, \tag{2}$$

где $\lambda = (y_2 - y_1)/(x_2 - x_1) \pmod p$.

Если $P_1 = P_2$, то операцию $P_1 + P_2$ называют удвоением и вычисляется как $P_3 = (x_3, y_3) = 2P_1 = 2(x_1, y_1)$, причем

$$x_3 = \lambda^2 - 2x_1 \pmod p,\ y^3 = \lambda(x_1 - x_3) - y_1 \pmod p, \tag{3}$$

где $\lambda = (3x_1^2 + a)/(2y_1) \pmod p$.

Для дальнейшего сравнения сложности операций сложения и удвоения введем переменные $t(B+B)$ и $t(2B)$, сложение и удвоение точек соответственно, где $B$ – координатный базис. Сложность операций сложения и удвоения выражаются в количестве умножений $(M)$, возведение в квадрат $(S)$ и инверсий $(I)$, операция суммирования игнорируется в силу незначительной сложности. В результате получим, что $t(A+A) = I + 2M + S$ и $t(2A) = I + 2M + 2S$, где $A$ – аффинный базис представления точки на ЭК.

Проведенный анализ показал, что при программной реализации $I \in [9M, 30M]$ и $S \approx 0.8M$.

**Метрика сложения в проективных координатах [11].** Для проективных координат $x = X/Z$ и $y = Y/Z$, а уравнение ЭК имеет вид:

$$E_P: Y^2Z = X^3 + aXZ^2 + bZ^3. \tag{4}$$

Пусть даны точки $P_1 = (X_1, Y_1, Z_1) \in E_P(\mathrm{F}_p)$ и $P_2 = (X_2, Y_2, Z_2) \in E_P(\mathrm{F}_p)$, тогда суммой двух точек $P_1$ и $P_2$ называется точка $P_3 \in E_P(\mathrm{F}_p)$ такая, что $P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$. Если $P_1 \neq P_2$, то координаты точки $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2 = (X_1, Y_1, Z_1) + (X_2, Y_2, Z_2)$ образуются как в [11]

$$X_3 = vA \pmod p,\ Y_3 = u\left(v^2X_1Z_2 - A\right) - v^3Y_1Z_2 \pmod p,\ Z_3 = v^3Z_1Z_2 \pmod p, \tag{5}$$

где $u = Y_2Z_1 - Y_1Z_2 \pmod p$; $v = X_2Z_1 - X_1Z_2 \pmod p$; $A = u^2Z_1Z_2 - v^3 - 2v^2X_1Z_2 \pmod p$.

Если $P_1 = P_2$, то операцию $P_1 + P_2$ называют удвоением и $P_3 = (X_3, Y_3, Z_3) = 2P_1 = 2(X_1, Y_1, Z_1)$, причем [11]

$$X_3 = 2hs \pmod p,\ Y_3 = w(4B - h) - 8Y_1^2s^2 \pmod p,\ Z_3 = 8s^3 \pmod p, \tag{6}$$

где $w = aZ_1^2 + 3X_1^2 \pmod p$; $s = Y_1Z_1 \pmod p$; $B = X_1Y_1s \pmod p$; $h = w^2 - 8B \pmod p$.

Время выполнения операции сложения $t(P+P) = 12M + 2S$ и удвоения $t(2P) = 7M + 5S$, где $P$ обозначает проективное представление точки.

**Метрика сложения в якобиановых координатах [12,13].** Для якобиановых координат $x = X/Z^2$ и $y = Y/Z^3$, а уравнение ЭК имеет вид:

$$E_J : Y^2 = X^3 + aXZ^4 + bZ^6. \tag{7}$$

Пусть даны точки $P_1 = (X_1, Y_1, Z_1) \in E_J(\mathrm{F}_p)$ и $P_2 = (X_2, Y_2, Z_2) \in E_J(\mathrm{F}_p)$, тогда суммой двух точек $P_1$ и $P_2$ называется точка $P_3 \in E_J(\mathrm{F}_p)$ такая, что $P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$. Если $P_1 \neq P_2$, то координаты точки $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2 = (X_1, Y_1, Z_1) + (X_2, Y_2, Z_2)$ образуются как в [12, 13]

$$X_3 = -H^3 - 2U_1H^2 + r^2 \pmod p,\ Y_3 = -S_1H^3 + r(U_1H^2 - X_3) \pmod p,\ Z_3 = Z_1Z_2H \pmod p, \tag{8}$$

где $U_1 = X_1Z_2^2 \pmod p$, $U_2 = X_2Z_1^2 \pmod p$, $S_1 = Y_1Z_2^3 \pmod p$, $S_2 = Y_2Z_1^3 \pmod p$, $H = U_2 - U_1 \pmod p$, $r = S_2 - S_1 \pmod p$.

Если $P_1 = P_2$, то операцию $P_1 + P_2$ называют удвоением и $P_3 = (X_3, Y_3, Z_3) = 2P_1 = 2(X_1, Y_1, Z_1)$, причем

$$X_3 = T \pmod p,\ Y_3 = -8Y_1^4 + M(S - T) \pmod p,\ Z_3 = 2Y_1Z_1 \pmod p, \tag{9}$$

где $S = 4X_1Y_1^2 \pmod p$; $M = 3X_1^2 + aZ_1^4 \pmod p$; $T = -2S + M^2 \pmod p$.

Время выполнения составляет $t(J+J) = 12M + 4S$ и $t(2J) = 4M + 6S$, где $J$ обозначает якобиановое представление точки.

**Метрика сложения в координатах Чудновского [12].** Для координат Чудновского $x = X/Z^2$ и $y = Y/Z^3$, а уравнение ЭК имеет вид:

$$E_{J^c} : Y^2 = X^3 + aXZ^4 + bZ^6. \tag{10}$$

Пусть даны точки $P_1 = (X_1, Y_1, Z_1, Z_1^2, Z_1^3) \in E_{J^c}(\mathrm{F}_p)$ и $P_2 = (X_2, Y_2, Z_2, Z_2^2, Z_2^3) \in E_{J^c}(\mathrm{F}_p)$, тогда суммой двух точек $P_1$ и $P_2$, называется точка $P_3 \in E_{J^c}(\mathrm{F}_p)$ такая, что $P_3 = P_1 + P_2 = (X_3, Y_3, Z_3, Z_3^2, Z_3^3)$. Если $P_1 \neq P_2$, то координаты точки $P_3 = (X_3, Y_3, Z_3, Z_3^2, Z_3^3) = P_1 + P_2 = (X_1, Y_1, Z_1, Z_1^2, Z_1^3) + (X_2, Y_2, Z_2, Z_2^2, Z_2^3)$ образуются как в [12]

$$X_3 = -H^3 - 2U_1H_2 + r^2 \pmod p,\ Y_3 = -S_1H^3 + r(U_1H^2 - X_3) \pmod p,\ Z_3 = Z_1Z_2H \pmod p,$$

$$Z_3^2 = Z_3^2 \pmod p,\ Z_3^3 = Z_3^3 \pmod p, \tag{11}$$

где $U_1 = X_1(Z_2^2) \pmod p$; $U_2 = X_2(Z_1^2) \pmod p$; $S_1 = Y_1(Z_2^3) \pmod p$; $S_2 = Y_2(Z_1^3) \pmod p$; $H = U_2 - U_1 \pmod p$; $r = S_2 - S_1 \pmod p$.

Если $P_1 = P_2$, то операцию $P_1 + P_2$ называют удвоением и $P_3 = \left(X_3, Y_3, Z_3, Z_3^2, Z_3^3\right) = 2P_1 = 2\left(X_1, Y_1, Z_1, Z_1^2, Z_1^3\right)$, причем:

$$X_3 = T \pmod p,\ Y_3 = -8Y_1^4 + M(S-T) \pmod p,\ Z_3 = 2Y_1Z_1 \pmod p,\ Z_3^2 = Z_3^2 \pmod p,$$

$$Z_3^3 = Z_3^3 \pmod p, \tag{12}$$

где $S = 4X_1Y_1^2 \pmod p$; $M = 3X_1^2 + a\left(Z_1^2\right)^2 \pmod p$; $T = -2S + M^2 \pmod p$.

Время выполнения составляет $t\left(J^c + J^c\right) = 11M + 3S$ и $t\left(2J^c\right) = 5M + 6S$, где $J^c$ обозначает представление точки в координатах Чудновского.

**Метрика сложения в модифицированных якобиановых координатах [14].** Для модифицированных якобиановых координат $x = X/Z^2$ и $y = Y/Z^3$, а уравнение ЭК имеет вид:

$$E_{J^m} : Y^2 = X^3 + aXZ^4 + bZ^6. \tag{13}$$

Пусть даны две точки $P_1 = \left(X_1, Y_1, Z_1, aZ_1^4\right) \in E_{J^m}\left(\mathrm{F}_p\right)$ и $P_2 = \left(X_2, Y_2, Z_2, aZ_2^4\right) \in E_{J^m}\left(\mathrm{F}_p\right)$, тогда суммой двух точек $P_1$ и $P_2$ называется точка $P_3 \in E_{J^m}\left(\mathrm{F}_p\right)$ такая, что $P_3 = P_1 + P_2 = \left(X_3, Y_3, Z_3, aZ_3^4\right)$. Если $P_1 \neq P_2$, то координаты точки $P_3 = \left(X_3, Y_3, Z_3, aZ_3^4\right) = P_1 + P_2 = \left(X_1, Y_1, Z_1, aZ_1^4\right) + \left(X_2, Y_2, Z_2, aZ_2^4\right)$ образуются как в [14]:

$$X_3 = -H_3 - 2U_1H^2 + r^2 \pmod p,\ Y_3 = -S_1H^3 + r(U_1H^2 - X_3) \pmod p,\ Z_3 = Z_1Z_2H \pmod p,$$

$$aZ_3^4 = aZ_3^4 \pmod p, \tag{14}$$

где $U_1 = X_1Z_2^2 \pmod p$; $U_2 = X_2Z_1^2 \pmod p$; $S_1 = Y_1Z_2^3 \pmod p$; $S_2 = Y_2Z_1^3 \pmod p$; $H = U_2 - U_1 \pmod p$; $r = S_2 - S_1 \pmod p$.

Если $P_1 = P_2$, то операцию $P_1 + P_2$ называют удвоением и $P_3 = \left(X_3, Y_3, Z_3, aZ_3^4\right) = 2P_1 = 2\left(X_1, Y_1, Z_1, aZ_1^4,\right)$, причем:

$$X_3 = T \pmod p,\ Y_3 = M(S-T) - U \pmod p,\ Z_3 = 2Y_1Z_1 \pmod p,\ aZ_3^4 = 2U\left(aZ_1^4\right) \pmod p, \tag{15}$$

где $S = 4X_1Y_1^2 \pmod p$, $U = 8Y_1^4 \pmod p$, $M = 3X_1^2 + \left(aZ_1^4\right) \pmod p$, $T = -2S + M^2 \pmod p$.

Время выполнения составляет $t\left(J^m + J^m\right) = 13M + 6S$ и $t\left(2J^m\right) = 4M + 4S$, где $J^m$ обозначает представление точки в модифицированных якобиановых координатах.

Анализируя время выполнения каждого алгоритма, однозначно видно неэффективность применения "чистых" аффинных координат в криптопреобразования в группах точек ЭК. Что касается проективных, якобиановых, координат Чудновского и модифицированных якобиановых координат, то тут нет явного лидера, некоторые координатные базисы выигрывают при удвоении точки, но проигрывают при сложении, некоторые наоборот.

## 2. Сложность и особенности выполнения преобразований в проективных координатах

Пусть дана эллиптическая кривая $E$ над $\mathrm{F}_q$, коэффициенты уравнения ЭК $a,b \in \mathrm{F}_q$, порядок ЭК $\#E(\mathrm{F}_q)$, базовая точка $G \in E(\mathrm{F}_q)$, порядок базовой точки $n$ ($nG = \mathrm{O}$) и секретный ключ $d$. В криптосистемах, основанных на эллиптических кривых, основной операцией является скалярное умножение, т.е. выполняется операции умножения большого целого числа $d$ на базовую точку $G$ с координатами $X,Y,Z \in \mathrm{F}_q$. В результате вычисляется открытый ключ $Q$, представляющий собой для $E(\mathrm{F}_{2^m})$ точку на ЭК.

$$Q = d \cdot G(\mathrm{mod}\, f(x),\ 2) \text{ для } q = 2^m, \tag{16}$$

$$Q = d \cdot G(\mathrm{mod}\, p) \text{ для } q = p, \tag{17}$$

где $f(x)$ – примитивный полином степени $m$ над полем $\mathrm{F}_2$.

Значение $Q$ вычисляется посредством многократного выполнения операций сложения и удвоения, одна из реализаций алгоритма скалярного умножение приведена ниже.

**Вход:** большое целое число $d$ и точка на эллиптической кривой $G$.

**Выход:** точка на эллиптической кривой $Q = d \cdot G(\mathrm{mod}\, f(x),\ 2)$.

1. Если $d = 0$, выдать O и остановить алгоритм.

2.Если $d < 0$, установить $R \leftarrow (-G)$ и $k \leftarrow (-d)$, иначе $R \leftarrow G$ и $k \leftarrow d$.

3. Пусть $h_l h_{l-1} \ldots h_1 h_0$ - бинарное представление числа $3k$, где самый крайний бит $h_l$ равняется 1.

4. Пусть $k_l k_{l-1} \ldots k_1 k_0$ - бинарное представление числа $k$.

5. Установить $S \leftarrow R$.

6. Цикл $i$ от $l-1$ до 1 делать:

6.1. Установить $S \leftarrow 2S$.

6.2. Если $h_i = 1$ и $k_i = 0$, вычислить $S \leftarrow S + R$

6.2. Если $h_i = 0$ и $k_i = 1$, вычислить $S \leftarrow S + (-R)$

7. Выдать результат $S$.

После нахождения $Q$ в проективных координатах значение этой точки необходимо преобразовать в аффинные координаты. Эта операция выполняется один раз в конце скалярного умножения.

Из пункта 6.1 алгоритма скалярного умножения следует, что количество удвоений точки зависит от длины множителя, то есть удвоение производиться $l-2$ раз. По теории вероятности в случайном числе распределение битов 1 и 0 равновероятностное, следовательно, количество единиц в случайном числе приблизительно $l/2$, следовательно, и количество пар $h_i = 1$, $k_i = 0$ и $h_i = 0$, $k_i = 1$ приблизительно $l/2$. Этот факт дает возможность утверждать, что количество удвоений точки выполняется приблизительно в два раза чаще, чем сложение в группе точек ЭК.

Для построения криптосистем на основе ЭК предпочтительным является использование представления точек в модифицированных якобиановых координатах, так как они обеспечивают минимизацию сложности операции удвоения точки на ЭК. Но при большом количестве единиц в бинарном представлении множителя необходимо использовать более сбалансированное представление координат – якобианово представление (ниже все примеры

приведены в якобиановом и аффинном представлениях точки над расширенным полем Галуа).

Использование одного координатного базиса не всегда позволяет достичь максимальной производительности. Перспективным направлением является использование смешанных координат. К примеру, на этапах 6.2 и 6.3 алгоритма скалярного умножения сложность суммирования в модифицированных координатах составляет $t\left(J^m + J^m\right) = 13M + 6S$. Если перед выполнением алгоритма скалярного умножения представить точки $R$ и $-R$ в аффинном базисе, мы получим суммирование в смешанных координатах и сложность такой операции составляет $t\left(J^m + A\right) = 9M + 5S$, при условии представления результата в базисе $J^m$. В табл. 1 приведены всевозможное использование смешанных координат и их сложность.

**Таблица 1**

| Удвоение | | Сложение | |
|---|---|---|---|
| Операция | Временные затраты | Операция | Временные затраты |
| $t(2P)$ | $7M + 5S$ | $t\left(J^m + J^m\right)$ | $13M + 6S$ |
| $t\left(2J^c\right)$ | $5M + 6S$ | $t\left(J^m + J^c = J^m\right)$ | $12M + 5S$ |
| $t(2J)$ | $4M + 6S$ | $t\left(J + J^c = J^m\right)$ | $12M + 5S$ |
| $t\left(2J^m = J^c\right)$ | $4M + 5S$ | $t(J + J)$ | $12M + 4S$ |
| $t\left(2J^m\right)$ | $4M + 4S$ | $t(P + P)$ | $12M + 2S$ |
| $t\left(2A = J^c\right)$ | $3M + 5S$ | $t\left(J^c + J^c = J^m\right)$ | $11M + 4S$ |
| $t\left(2J^m = J\right)$ | $3M + 4S$ | $t\left(J^c + J^c\right)$ | $11M + 3S$ |
| $t\left(2A = J^m\right)$ | $3M + 4S$ | $t\left(J^c + J = J\right)$ | $11M + 3S$ |
| $t(2A = J)$ | $2M + 4S$ | $t\left(J^c + J^c = J\right)$ | $10M + 2S$ |
| | | $t\left(J + A = J^m\right)$ | $9M + 5S$ |
| | | $t\left(J^m + A = J^m\right)$ | $9M + 5S$ |
| | | $t\left(J^c + A = J^m\right)$ | $8M + 4S$ |
| | | $t\left(J^c + A = J^c\right)$ | $8M + 3S$ |
| | | $t(J + A = J)$ | $8M + 3S$ |
| | | $t\left(J^m + A = J\right)$ | $8M + 3S$ |
| | | $t\left(A + A = J^m\right)$ | $5M + 4S$ |
| | | $t\left(A + A = J^c\right)$ | $5M + 3S$ |
| $t(2A)$ | $2M + 2S + I$ | $t(A + A)$ | $2M + S + I$ |

Рассмотрим пример вычисления открытого ключа.

**Пример 1.** Пусть дана точка $G = (23, 0, 1)$, которая принадлежит кривой

$$Y^2 + XYZ = X^3 + aX^2Z^2 + bZ^6 \left(\mathrm{mod}\, f(x), 2\right),$$

где $a=1$; $b=3$; $n=38$; $m=5$; $f(x)=x^5+x^2+1$. Необходимо найти скалярное умножение $Q=dG$, причем $d=7$.

Используя алгоритм скалярного умножения, мы получим $Q=(2,29,25)$, в аффинных координатах эта точка имеет вид $Q=(28,2)$.

Заметим, что при использовании различных алгоритмов скалярного умножения можно получить различные значения точки $Q$. Так при семикратном сложении точки $G$ самой с собой получим $Q'=(31,16,2)$, которая в аффинных координатах имеет вид $Q'=(28,2)$.

Из выше приведеного видно, что точки $Q$ и $Q'$ в проективном представлении принадлежат одному классу точек, так как в аффинном представлении это одна и таже точка. Этот эффект обусловлен тем, что порядок кривой в проективном представлении [21] $\#E_p(\mathrm{F}_{2^m})=(2^m-1)(\#E_a(\mathrm{F}_{2^m}))$. Если известна хотя бы одна точка в проективном представлении (и в других представлениях, кроме аффинного), то все множество точек, принадлежащее одному классу, можно получить как $(tX,tY,tZ)$, где $t\le n$ есть простое число.

**Пример 2.** Подсчитать количество точек (порядок кривой), удовлетворяющих уравнениям:

1. $y^2+xy=x^3+x^2+3(\mathrm{mod}\, f(x),2)$.
2. $Y^2+XYZ=X^3+aX^2Z^2+bZ^6(\mathrm{mod}\, f(x),2)$,

где $a=1$; $b=3$; $n=38$; $m=5$; $f(x)=x^5+x^2+1$. Причем второе уравнение получено из первого, используя соответствующие формулы перехода.

Учитывая, что порядок поля небольшой, можно перебрать все возможные комбинации $(x,y)$ в первом уравнении и $(X,Y,Z)$ во втором. Получим:

1. $\#E=37+1$.
2. $\#E=1147+1$.

Если проанализировать полученный результат в проективном представлении, то мы увидим 37 классов точек. Точки принадлежат одному классу, если при переходе в аффинное представление мы получаем одну и туже точку. Из табл. 2 следует, что точке $(5,14)$ в аффинных координатах соответствует 31 различная точка в проективных координатах.

**Таблица 2**

| Проективное преставление 1 | Аффинн. представление 1 | Проективное преставление 2 | Аффинн. представление 2 | Проективное преставление 1 | Аффинн. представление 1 | Проективное преставление 2 | Аффинн. Представление 2 |
|---|---|---|---|---|---|---|---|
| x:1 y:16 z:28 | x:5 y:14 | x:17 y:6 z:3 | x:5 y:14 | x:1 y:29 z:25 | x:14 y:2 | x:17 y:9 z:18 | x:14 y:2 |
| x:2 y:24 z:9 | x:5 y:14 | x:18 y:3 z:22 | x:5 y:14 | x:2 y:1 z:17 | x:14 y:2 | x:18 y:22 z:26 | x:14 y:2 |
| x:3 y:1 z:21 | x:5 y:14 | x:19 y:29 z:10 | x:5 y:14 | x:3 y:17 z:8 | x:14 y:2 | x:19 y:30 z:3 | x:14 y:2 |
| x:4 y:20 z:29 | x:5 y:14 | x:20 y:18 z:2 | x:5 y:14 | x:4 y:19 z:23 | x:14 y:2 | x:20 y:26 z:28 | x:14 y:2 |
| x:5 y:11 z:1 | x:5 y:14 | x:21 y:10 z:30 | x:5 y:14 | x:5 y:10 z:14 | x:14 y:2 | x:21 y:27 z:5 | x:14 y:2 |
| x:6 y:19 z:20 | x:5 y:14 | x:22 y:31 z:11 | x:5 y:14 | x:6 y:11 z:6 | x:14 y:2 | x:22 y:25 z:13 | x:14 y:2 |
| x:7 y:5 z:8 | x:5 y:14 | x:23 y:14 z:23 | x:5 y:14 | x:7 y:31 z:31 | x:14 y:2 | x:23 y:21 z:20 | x:14 y:2 |
| x:8 y:30 z:18 | x:5 y:14 | x:24 y:12 z:13 | x:5 y:14 | x:8 y:8 z:7 | x:14 y:2 | x:24 y:18 z:12 | x:14 y:2 |
| x:9 y:2 z:14 | x:5 y:14 | x:25 y:23 z:17 | x:5 y:14 | x:9 y:7 z:30 | x:14 y:2 | x:25 y:5 z:21 | x:14 y:2 |
| x:10 y:28 z:27 | x:5 y:14 | x:26 y:4 z:4 | x:5 y:14 | x:10 y:15 z:22 | x:14 y:2 | x:26 y:14 z:29 | x:14 y:2 |
| x:11 y:9 z:7 | x:5 y:14 | x:27 y:22 z:24 | x:5 y:14 | x:11 y:13 z:15 | x:14 y:2 | x:27 y:20 z:4 | x:14 y:2 |
| x:12 y:8 z:15 | x:5 y:14 | x:28 y:13 z:16 | x:5 y:14 | x:12 y:28 z:16 | x:14 y:2 | x:28 y:3 z:27 | x:14 y:2 |
| x:13 y:27 z:19 | x:5 y:14 | x:29 y:25 z:12 | x:5 y:14 | x:13 y:23 z:9 | x:14 y:2 | x:29 y:16 z:2 | x:14 y:2 |
| x:14 y:21 z:6 | x:5 y:14 | x:30 y:26 z:25 | x:5 y:14 | x:14 y:2 z:1 | x:14 y:2 | x:30 y:6 z:10 | x:14 y:2 |
| x:15 y:15 z:26 | x:5 y:14 | x:31 y:7 z:5 | x:5 y:14 | x:15 y:4 z:24 | x:14 y:2 | x:31 y:24 z:19 | x:14 y:2 |
| x:16 y:17 z:31 | x:5 y:14 | | | x:16 y:12 z:11 | x:14 y:2 | | |

Результаты таблицы можно проверить, используя формулы перехода из проективных в аффинные координаты. При этом переход из аффинных координат в проективные однозначен, каждая точка аффинного представления отображается в одну точку проективного. При обратном отображении 31 точка проективного представления должна проецироваться в одну точку аффинных координат.

В процессе выполнения арифметических операций над группой точек эллиптической кривой существует вероятность появления операции сложения двух точек из одного класса, что приводит к некорректным результатам. В этом и состоит вынесенный в название термин «не все так гладко»

Рассмотрим пример ошибочной операции из-за неоднозначности отображения между координатными базисами.

**Пример 3.** Пусть даны точки $G=(23,23,1)$ и $Q=(14,2,1)$, которые принадлежат кривой

$$Y^2+XYZ=X^3+aX^2Z^2+bZ^6\left(\mathrm{mod}\, f(x),2\right),$$

где $a=1$; $b=3$; $n=38$; $m=5$; $f(x)=x^5+x^2+1$. Необходимо вычислить точку $P=6G+16Q$, используя алгоритм скалярного умножения

$$P=6(23,23,1)+16(14,2,1)=(11,29,7)+(24,5,13)=(0,0,0).$$

Некорректность заключается в том, что на последнем этапе сложения точек мы использовали функцию сложения точек, так как они в проективных координатах не равны, но они принадлежат одному классу. Точки $(11,29,7)$ и $(24,5,13)$ соответствую точке $(5,14)$ в аффинных координатах, что подтверждает табл. 2.

Для получения правильного результата на последнем этапе необходимо использовать формулу удвоения любой из двух точек, тогда мы получим:

$$2(11,29,7)=(3,30,3)_{проект}=(28,2)_{аффин},$$

$$2(24,5,13)=(25,6,10)_{проект}=(28,2)_{аффин}.$$

В результате получен тот же результат, что и при вычислении $P=6G+16Q$ в аффинных координатах. Очевидно это и есть кардинальное решение, которое позволяет исключить некорректность результата сложения точек.

При реализации криптопреобразований в группах точек ЭК для повышения производительности более предпочтительным является выполнение операций сложения и удвоения в смешанных координатах, что позволяет минимизировать число операций умножения и возведения в квадрат при скалярном умножении. В таблице 2 приведены различные варианты смешанных координат и их сложность. Использование смешанных координат повышает вероятность получения некорректных результатов, и поэтому при криптопреобразованиях необходимо вводить дополнительные ограничительные условия.

В силу эпитоморфизма между множеством точек кривой $E$ в аффинном представлении и множеством точек, удовлетворяющих уравнению кривой $E$ в проективном представлении, в алгоритм сложения следует внести изменения: при появлении, в процессе расчета, точки $(0,0,0)$ прервать операцию сложения и выполнить операцию удвоения точки; перед операцией сложения выполнить проверку на равенство точек; пренебречь вероятностью возникновения ситуации сложения двух точек из одного класса $P=(q-1)/\#E_a(\mathrm{F}_q)$, при условии некритичности возникновения ошибки вычислений.

**Выводы**

Использование смешанных координат позволяет повысить производительность криптопреобразований в группах точек ЭК. Криптопреобразования, построенные с использованием представления точек в аффинных координатах, крайне неэффективны из-за присутствия в алгоритмах сложения и удвоения точек операции деления в поле.

В процессе криптопреобразований в группе точек ЭК для избежания некорректного использования операции сложения необходимо ввести дополнительную проверку, либо пренебречь ею.

Перспективными направлениями изучения проективной геометрии являются нахождение новых алгоритмов скалярного умножения, расширяющие возможности использования смешанных координат и минимизирующие возможности некорректного использования операции сложения.

**Список литературы:** 1.*A. Menezes, T. Okamoto, S. Vanstone,* "Reducing elliptic curve logarithm to logarithm in finite field", Proceeding of the 22nd Annual ACM Symposium on The Theory of Computing (1991), 80-89. 2. IEEE P1363 Working Draft, June 16, 1998 3. *G. Frey, H.G. Ruck,* "A remark concerning $m$-divisibility and the discrete logarithm in the divisor class group of curves", Mathematic of computation, 62 (1994), 865-874 4. *I.A. Semaev* "Evaluation of discrete logarithm in a group of $p$ - torsion point of an elliptic curve in characteristic $p$ ", Mathematic of computation, 67(1998), 353-356 5. *T. Stoh, K. Araki* "Fermat quotients and polynomial time discrete logarithm for anomalous curve", Comentarii Math. Univ. St. Pauli., vol. 47 (1998), 81-92 6. *N.P. Smart* "The discrete logarithm problem on elliptic curve of trace one", to appear in J. Cryptology 7. *J. Pollard* "Monte Carlo method for index computation $(\mathrm{mod}\, p)$ ", Mathematic of computation, 32 (1978), 918-924 8. *Горбенко И.Д., Збитнев С.И., Поляков. А.А.* Криптоанализ криптографических преобразований в группах точек эллиптических кривых методом Поларда // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С.43-50 9. *S.C. Pohlig, M.E. Hellman* "An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance", IEEE Trans. Inf. Theory, IT-24 (1978), 106-110 10. *J.H. Silverman* "The Arithmetic of Elliptic Curve", GTM 106, Springer-Verlag, New York, 1986 11. *K. Koyama, Y. Tsuruoka* "Speeding up elliptic cryptosystem by using a signed binary window method", Advanced in Cryptology - Proceeding of Crypto'92, Lecture Notes in Computer Science, 740 (1993), Springer-Verlag, 345-357 12. *D.V. Chudnovsky, G.V. Chudnovsky* "Sequence of number generated by addition in formal group and new primality and factorization test", Advanced in Applied Math., 7 (1986), 385-434 13. *H. Cohen, A. Miyaji, T. Ono* "Efficient elliptic curve exponentiation", Advanced in Cryptology – Proceeding of ICICS'97, Lecture Notes in Computer Science, 1334 (1997), Springer-Verlag, 282-290 14. *H. Cohen, A. Miyaji, T. Ono* "Efficient elliptic curve exponentiation using mixed coordinates", Advanced in Cryptology , 1998 15. *A. Menezes,* Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston, 1993 16. *Иванов М.А.* Криптографические методы защиты информации в компьютерных системах и сетях. М.: КУДИЦ-ОБРАЗ, 2001. 368с. 17. *R. Schoof* "Counting points on elliptic curves over finite fields", Journal de Theorie des Nombres de Bordeaux 7 (1995), 219-254 18. *N. Koblitz* "CM-Curve with good cryptographic properties", Dept. of Mathematics, 1991,279-287 19. *J. Guajardo, C. Paar* "Efficient Algorithms for Elliptic Curve Cryptosystem", GTE Corporation, Springer-Verlag, 1998, 342-356 20. *J. Solinas* "An Improved Algorithm for Arithmetic in a Family of Elliptic Curves", National Security Agency, Springer-Verlag, 1998, 357-371 21. *J.H. Silverman* "The Arithmetic of Elliptic Curves", GTM 106, Springer-Verlag, New-York, 1986

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 22.04.2002*

УДК 681.3.06

*А.В. ПОТИЙ, канд. техн. наук, Ю.А. ИЗБЕНКО*

# ОБОСНОВАНИЕ ВЫБОРА МЕТОДА ПОСТРОЕНИЯ КРИПТОГРАФИЧЕСКИ СТОЙКИХ БУЛЕВЫХ ФУНКЦИЙ

## Введение

При конструировании поточных шифров важной задачей является выбор нелинейной функции, которая отвечает за обеспечение требуемой криптографической стойкости формируемого бегущего ключа.

На сегодняшний день наиболее широкое распространение получили два подхода к построению поточных шифров: нелинейный фильтр-генератор, в котором выход линейного рекуррентного регистра (ЛРР) фильтруется выходной функцией нелинейной фильтрации с целью внесения нелинейности, и комбинирующий генератор, в котором нелинейная функция комбинирует выходы нескольких ЛРР.

В качестве нелинейных функций в основном используются нелинейные булевы функции. Задачей данных функций является противостояние криптографическим атакам с целью недопущения просачивания на выход функции информации о ее входных данных. Функции такого рода называются криптографически стойкими функциями (КСФ), отыскание методов конструирования подобных функций является областью активных исследований в криптографии. При построении и выборе таких функций разработчики используют ряд показателей, основными из которых являются [1-6]: 1)сбалансированность, 2)нелинейность, 3)критерий распространения (строгий лавинный критерий), 4)корреляционный иммунитет, 5)алгебраическая степень.

Целью данной статьи является анализ существующих показателей стойкости КСФ; сравнительная оценка на основе проведенного анализа трех классов функций, используемых при построении КСФ, методы построения которых описаны в [1-6]: корреляционно-иммунных функций, эластичных функций, булевых функций на основе бент-функций; формулирование рекомендаций относительно использования основных показателей стойкости и выбора класса методов для построения КСФ. Вводится дополнительный показатель стойкости булевых функций.

## 1. Основные показатели стойкости криптографически стойких функций

Как указывалось выше, при построении и выборе булевых функций основными показателями стойкости являются:

1. Сбалансированность.
2. Нелинейность.
3. Критерий распространения (строгий лавинный критерий).
4. Корреляционный иммунитет.
5. Алгебраическая степень.

В случае использования корреляционно-иммунных и эластичных функций используются показатели 1, 2, 4, 5, в случае использования булевых функций на основе бент-функций используются показатели 1, 2, 3, 5.

Для дальнейшего рассмотрения вопроса введем некоторые понятия и определения.

**Определение 1.** Пусть $f$ является функцией на $V_n$, $V_n=(0,1)^n$, где $\alpha_0=(0,\ldots,0,0)$, $\alpha_1=(0,\ldots,0,1),\ldots,$ $\alpha_{2^n-1}=(1,\ldots,1,1)$ – все векторы на $V_n$. Тогда (1,-1)-последовательность, определенная как $((1)^{f(\alpha_0)}, (1)^{f(\alpha_1)},\ldots, (1)^{f(\alpha_{2^n-1})})$, называется *последовательностью* функции $f$.

(0,1)-последовательность, определенная как $(f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{2^n-1}))$, называется *таблицей истинности* функции $f$.

**Пример 1.** Пусть $f = x_1x_2x_3 \oplus x_1x_3 \oplus x_2 \oplus x_3 \oplus 1$ на $V_3$.

Тогда $V_3=(0,1)^3$, где $\alpha_0=(0,0,0)$, $\alpha_1=(0,0,1)$, $\alpha_2=(0,1,0)$, $\alpha_3=(0,1,1)$, $\alpha_4=(1,0,0)$, $\alpha_5=(1,0,1)$, $\alpha_6=(1,1,0)$, $\alpha_7=(1,1,1)$ – все векторы на $V_n$.

Таблица истинности функции $f$: $f(000)=1$, $f(001)=0$, $f(010)=0$, $f(011)=1$, $f(100)=1$, $f(101)=1$, f(110)=0, f(111)=1. Последовательность функции $f$: -1,1,1,-1,-1,1,-1.

**Определение 2.** Функция $f$ на $V_n$ является *сбалансированной* функцией, если ее таблица истинности (последовательность) содержит $2^{n-1}$ нулей / единиц (единиц / минус единиц).

**Пример 2.** Пусть $f = x_1x_2 \oplus x_3$ на $V_3$. Тогда данная функция сбалансирована, т.к. ее таблица истинности имеет вид 0,1,0,1,0,1,1,0 и содержит $2^{3-1} = 4$ нулей / единиц.

Сбалансированность функции является важным показателем, поскольку одним из требований, предъявляемым к поточным шифрам, являются хорошие статистические свойства шифрующей гаммы. Несоблюдение данного критерия делает криптосистему уязвимой к статистическим атакам.

**Определение 3.** Аффинной функцией $f$ на $V_n$ является функция вида $f = a_1x_1 \oplus \ldots \oplus a_nx_n \oplus c$, где $a_j$, $c \in GF(2)$, $j=1,2,\ldots,n$. Функция $f$ называется линейной, если $c = 0$.

**Пример 3.** Аффинная функция на $V_3$ $f_1 = x_3 \oplus x_1 \oplus 1$, линейная – $f_2 = x_3 \oplus x_1$.

**Определение 4.** Криптографически слабыми функциями являются аффинные (линейные) функции и функции с линейной структурой, а также нелинейные функции, которые могут быть сведены к вышеназванным функциям с помощью аффинных преобразований.

**Определение 5.** *Весом Хэмминга* вектора $\alpha$, обозначаемым как $W(\alpha)$, является количество единиц в векторе. Для данных функций $f$ и $g$ *расстоянием Хэмминга* является $d(f,g) = W(f(x \oplus g(x))$, где $x=(x_1, x_2, \ldots, x_n) \in (0,1)$.

**Пример 4.** Для $\alpha = (101)$ $W(\alpha) = 2$.

Для $f(x) = x_1x_2$ и $g(x) = x_1 \oplus x_2$ $d(f,g) = W(x_1x_2 \oplus x_1 \oplus x_2) = 3$.

**Определение 6.** *Нелинейность функции* $f$ -- минимальное расстояние Хэмминга $N_f$ между функцией $f$ и всеми криптографически слабыми функциями на $V_n$:

$$N_f = \min \{d(f,\varphi)\},$$

где $\varphi$ - множество криптографически слабых функций.

Для произвольной функции $f$ нелинейность $N_f$ на $V_n$ может достигать [5]:

$$N_f \le 2^{n-1} - 2^{n/2-1}. \qquad (1)$$

Нелинейность функции является важным критерием, поскольку несоблюдение данного критерия делает возможным проведение корреляционных атак, использующих корреляцию данной функции со множеством криптографически слабых функций. При построении КСФ необходимо обеспечить ее минимальную корреляцию со множеством всех криптографически слабых функций, т.е. стремиться, чтобы нелинейность данной функции стремилась к верхней границе нелинейности, определенной в (1).

**Определение 7.** Пусть $f$ - функция на $V_n$. Тогда говорят, что $f$ удовлетворяет

1. *критерию распространения относительно вектора* $\alpha$, *КР(*$\alpha$*),* если функция $f(x) \oplus f(x \oplus a)$ является сбалансированной, где $x = (x_1,x_2,\ldots,x_n)$ на $V_n$.
2. *критерию распространения степени k, КР(k),*если удовлетворяется критерий распространения относительно всех векторов $\alpha \in V_n$ при $1 \le W(\alpha) \le k$. При этом функция $f$ является совершенно нелинейной.
3. *строгому лавинному критерию, СЛК,* если $f$ удовлетворяет критерию распространения степени 1.

**Пример 5.** Пусть $f = x_1x_2 \oplus x_3$ на $V_3$. Пусть $a = (1,1,0)$, тогда

$$f(x) \oplus f(x \oplus a) = (x_1x_2 \oplus x_3) \oplus ((x_1 \oplus 1)(x_2 \oplus 1) \oplus x_3) = x_1 \oplus x_2 \oplus 1$$

является сбалансированной и удовлетворяет критерию распространения относительно вектора $a = (1,1,0)$.

**Пример 6.** Пусть $f = x_1 \oplus x_1x_5 \oplus x_2x_4 \oplus x_2x_5 \oplus x_2x_4x_5 \oplus x_3x_4x_5$ на $V_5$. Пусть $a = (0,0,1,0,0)$. тогда

$$f(x) \oplus f(x \oplus a) = x_3x_4x_5 \oplus (x_3 \oplus 1)\, x_4x_5 = x_4x_5$$

не является сбалансированной и не удовлетворяет критерию распространения относительно вектора $a = (0,0,1,0,0)$.

Следует отметить, что по сути своей критерий распространения связан с корреляционными свойствами последовательности. Критерий распространения характеризует зависимость выходных значений нелинейной функции от входных векторов с различным весом Хэмминга. Таким образом, критерий распространения является важным конструктивным критерием, пренебрежение которым делает возможным применение статистических атак.

**Определение 8.** Пусть $f$ - функция на $V_n$. Тогда говорят, что к*орреляционный иммунитет порядка k, КИ(k),* удовлетворяется в том случае, если случайная величина $Y$, порождаемая функцией $f$, статистически не зависит от любого подмножества $X_{i_1}, \ldots X_{i_k}$ ее $k$ входных координат, где $X$ - случайная величина, принимающая значения $x \in V_n$ с равномерной вероятностью $2^{-n}$, $X_i$ – случайная величина, соответствующая значению $i$-ой координаты $x_i \in GF(2)$ [1]. Эквивалентное определение корреляционного иммунитета в терминах преобразования Уолша : функция $f$ над $V_n$ имеет *корреляционный иммунитет порядка k, КИ(k),* если ее преобразование Уолша удовлетворяет равенству $F(\omega) = 0$ для всех $\omega \in V_n$ таких, что $1 \le W(\omega) \le k$. Преобразование Уолша $F(\omega)$ функции $f$ над $V_n$ определяется как принимающая действительные значения функция

$$F(\omega) = 2^{-n} \sum_x (-1)^{f(x) \oplus \langle \omega, x \rangle}, \qquad (2)$$

где $\omega \in V_n$. Отметим, что в данной сумме $f(x)$ и $\langle \omega, x \rangle$ рассматриваются как функции, принимающие действительные значения.

Функция, обладающая корреляционным иммунитетом порядка $k$, называется корреляционно-иммунной $k$-го порядка. Корреляционно-иммунные функции являются частным случаем эластичных функций.

Как будет показано ниже, данный критерий, по мнению авторов, не является конструктивным.

**Определение 9.** Алгебраическая степень $\deg(f)$ является степенью самого длинного слагаемого функции, представленной в алгебраической нормальной форме. Высокая алгебраическая степень позволяет противостоять различным аналитическим атакам, призванным свести данную функцию к криптографически слабой.

**Пример 7.** Пусть $f = x_1 \oplus x_1x_5 \oplus x_2x_4 \oplus x_2x_5 \oplus x_2x_4x_5 \oplus 1$ на $V_5$. Тогда $\deg(f) = 3$.

### 2. Сравнительная оценка используемых классов КСФ

Рассмотрев показатели стойкости КСФ, обратимся к цели нашей статьи. Поскольку наиболее распространенной и эффективной атакой на поточные шифры являются корреляционные атаки, целью которых является выявить зависимость выходной гаммы нелинейной функции от поступающих на нее данных, наше внимание привлек тот факт, что любая булева функция $f$ имеет определенную корреляцию с некоторыми линейными функциями $\ell_i$ из множества всех линейных функций $L$. В [7] вводится коэффициент кросс-корреляции $c(f, \ell_i)$, определенный как

$$c(f, \ell_i) = 2^{-n} F(\omega) = 2^{-n} \sum_x (-1)^{f(x) \oplus \langle \omega, x \rangle}. \quad (3)$$

Суммарная корреляция булевой функции $f$ с множеством линейных функций $L$ имеет вид

$$\sum_L c(f, L)^2 = 1, \quad (4)$$

что демонстрирует тот факт, что корреляция с множеством линейных функций существует всегда независимо от выбора функции $f$. Как видно из (4), коэффициент кросс-корреляции равен нулю в том случае, если $F(\omega) = 0$. Однако это противоречит идее корреляционно-иммунных (эластичных) функций, согласно которой данные функции имеют нулевую корреляцию с линейными функциями. Следовательно, корреляционный иммунитет не является конструктивным критерием стойкости булевых функций. Согласно [7], для совершенно нелинейных функций $| F(\omega) | = 2^{n/2}$. Следовательно, формула (3) принимает вид

$$c(f, \ell_i) = 2^{-n} F(\omega) = 2^{-n} \cdot 2^{n/2} = 2^{-n/2}, \quad (5)$$

из чего следует сделать вывод, что для всех совершенно нелинейных функций абсолютная величина коэффициента кросс-корреляции – величина постоянная, а значит, данный класс функций имеет минимальную корреляцию с линейными функциями. Кроме того, как видно из (5), при расширении векторного пространства $c(f, L)$ будет стремиться к нулю. Для остальных же булевых функций абсолютная величина коэффициента кросс-корреляции с некоторыми линейными функциями будет выше, чем $2^{-n/2}$. Отметим, что нижняя граница коэффициента кросс-корреляции может достигать нуля, но в силу равенства (4) это неизбежно повлечет за собой увеличение корреляции с другими линейными функциями.

В связи с этим в качестве дополнительного показателя стойкости булевых функций предлагается ввести абсолютное значение кросс-корреляции функции

$$C_f = \max | c(f, \ell_i) |, \quad (6)$$

равное максимальному значению коэффициента кросс-корреляции функции $f$ на $V_n$, $\ell_l \in L$. Критерием для данного показателя будем полагать минимально возможное значение $C_f$. При прочих равных показателях стойкости некоторых функций данный показатель позволит нам выбрать функцию, минимально коррелирующую с множеством всех аффинных функций.

Известно [5], что совершенной нелинейностью обладают бент-функции. Функция $f$ на $V_n$ называется бент-функцией, если

$$2^{-n/2} \sum_{x \in V_n} (-1)^{f(x) \oplus \langle \beta, x \rangle} = \pm 1 \quad \text{для всех } \beta \in V_n. \quad (7)$$

Помимо этого, они обладают максимальной нелинейностью и максимальным расстоянием до линейных структур. Однако две причины препятствуют использованию бент-функций в чистом виде: их последовательности несбалансированны, что делает их уязвимыми к статистическому анализу; они существуют, согласно (7), лишь на четных векторных пространствах.

В качестве примера рассмотрим три булевы функции (рис.1):

1. Типичная бент-функция $f_1(x) = x_1x_2 \oplus x_3x_4$.
2. Сбалансированная функция, построенная на основе бент-функции $f_1(x)$ согласно [5], $f_2(x) = x_1x_2 \oplus x_3x_4 \oplus x_5$.

3. Корреляционно-иммунная функция 2-го порядка, построенная согласно [2], $f_3(x) = x_1x_4 \oplus x_1x_5 \oplus x_2 \oplus x_3 \oplus x_4$.

На рис. 1 представлено распределение абсолютного значения преобразования Уолша, в табл. 1 приведены расчетные показатели стойкости данных функций.

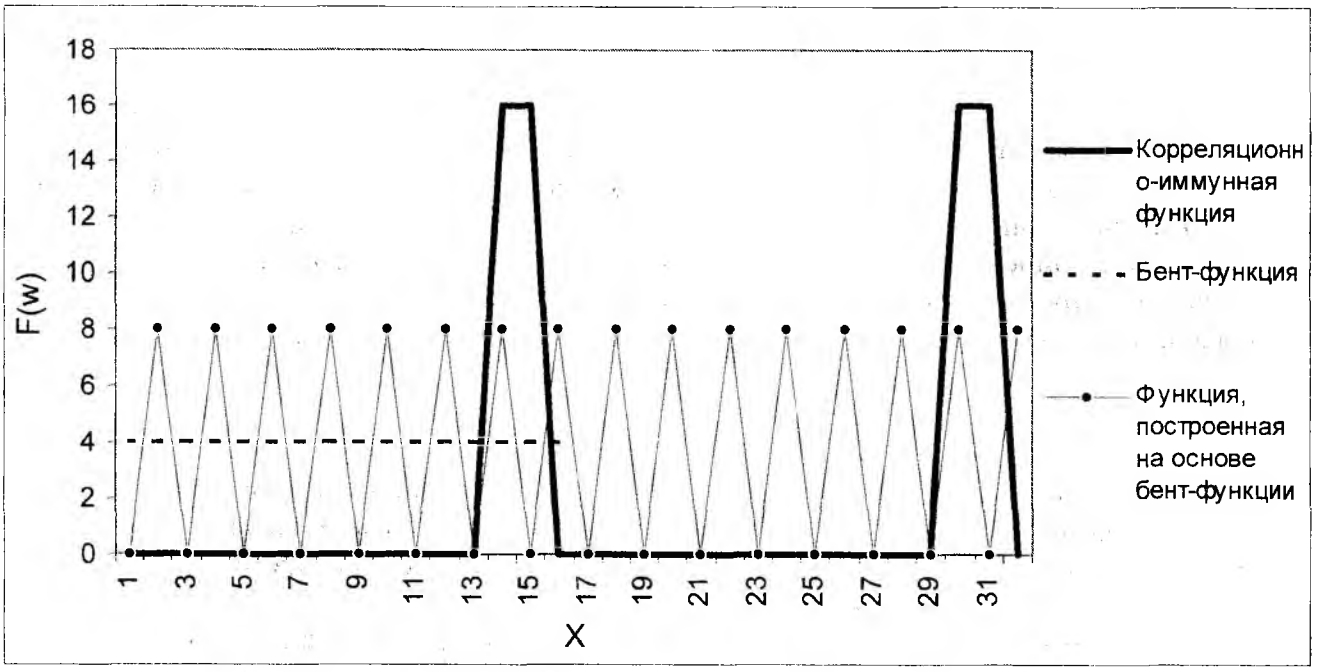



Рис. 1

**Таблица 1**

| | Сбалансированность | $N_f$ | | КР($k$) | КИ($k$) | deg($f$) | $C_f$ |
|---|---|---|---|---|---|---|---|
| | | Полученное значение | Верхняя граница | | | | |
| $f_1(x)$ | нет | 6 | 6 | 4 | нет | 2 | 0,25 |
| $f_2(x)$ | да | 12 | 12 | удовл. всем векторам, кроме (00001) | нет | 2 | 0,25 |
| $f_3(x)$ | да | 8 | 12 | не удовл. $W(\alpha)$=1,2,3,4, удовл $W(\alpha)$=5 | 2 | 2 | 0,5 |

Как видно из приведенных результатов, бент-функция $f_1(x)$ имеет в качестве абсолютного значения преобразования Уолша константу, равную $|F(\omega)| = 2^{n/2} = 2^{4/2} = 4$; у функции $f_2(x)$, построенной на основе данной бент-функции, значения $|F(\omega)|$ имеют диапазон 0÷8; у корреляционно-иммунной функции $f_3(x)$ значения $|F(\omega)|$ имеют диапазон 0÷16, нулевая корреляция корреляционно-иммунной функции с некоторыми линейными функциями влечет более высокую корреляцию с остальными линейными функциями. Все функции обладают высокой нелинейностью, причем бент-функция достигает верхней границы нелинейности на заданном пространстве. Приведенные функции не являются криптографически стойкими, однако служат хорошим наглядным материалом для подтверждения наших идей.

Таким образом, на основе выше изложенного, наиболее предпочтительным классом функций, используемых для построения КСФ, является класс совершенных нелинейных функций (бент-функций), так как данный класс функций обладает привлекательными криптографическими свойствами. Из этого следует, что в качестве методов построения КСФ

рекомендуется использовать класс методов, конструирующих высоконелинейные сбалансированные функции на основе бент-функций, так как данный класс методов изначально "стартует" с хорошими криптографическими свойствами, имея в качестве недостатка лишь несбалансированность. При отборе же КСФ на заданном векторном пространстве, помимо основных конструктивных критериев, следует выбирать $f$ насколько возможно близкой к совершенной нелинейной функции с равномерно минимизированной кросс-корреляцией и с минимально-возможным абсолютным значением $C_f$. Использование классов методов, генерирующих корреляционно-иммунные и эластичные функции, менее предпочтительно, однако они также дают высокие показатели криптостойкости.

**Список литературы:** **1.** P.Camion, C. Carlet, P. Charpin and N. Sendrier, "On correlation-immune functions," in Lecture Notes in Computer Science vol.576; Advances in Crypology: Crypto '91 3roc., pp 87-100. Berlin: Springer-Verlag, 1991. **2.** J. Seberry, X.M. Zhang and Y. Zheng, "On Constructions and Nonlinearity of Correlation Immune Functions" In T. Helleseth, editor, Advances in Cryptology - Eurocrypt '93, pages 181-199, Springer-Verlag, Berlin, 1994. **3.** P. Camion and A. Canteaut "Construction of t-Resilient Functions over a Finite Alphabet", in Lecture Notes in Computer Science; Advances in Crypology: Eurorypt '96 Proc.,Springer-Verlag 1996, pp. 283-293. **4.** X.-M. Zhang and Y.Zheng, "On nonlinear resilient functions," Advances in Cryptology - Eurocrypt '95, Lecture Notes in Computer Science, vol.921, L.C. Guillou ed., Springer-Verlag, pp. 274-288, 1995. **5.** Jennifer Seberry, Xian-Mo Zhang, Yuliang Zheng. Nonlinearity and Propagation Characteristics of Balanced Boolean Functions. Information and Computation, Vol. 119, No 1, pp 1-13, 1995. **6.** B. Preneel, R. Govaerts, and J. Vandewalle, "Boolean functions satisfying higher order propagation criteria" in Lecture Notes in Computer Science 547; Advances in Cryplology: Proc. Eurocrypt'91,1991, pp. 141-152. Berlin: Springer-Verlag. **7.** W.Meier, O.Staffelbach. Nonlinearity criteria for cryptographic functions. Lecture Notes in Computer Science 434, pp.549-562, Springer-Verlag, 1990.

*Харьковский военный университет*

*Харьковский национальный*

*университет радиоэлектроники*

*Поступила в редколлегию 19.03.02*

УДК 681.3.06: 519.248.681

*И.В. ЛИСИЦКАЯ канд.техн.наук, А. С. БОНДАРЕНКО, Т.В.ЦЕПУРИТ, А. И. КОЛЫБЕЛЬНИКОВ*

# ОБЕСПЕЧЕНИЕ СТОЙКОСТИ ШИФРА DES К АТАКАМ ЛИНЕЙНОГО КРИПТОАНАЛИЗА. ТРЕБОВАНИЯ К ОТБОРУ S-БЛОКОВ, ЗАЩИЩЕННЫХ ОТ АТАК НА ДЕСЯТИЦИКЛОВЫЕ ИТЕРАТИВНЫЕ ЛИНЕЙНЫЕ АППРОКСИМАЦИОННЫЕ ХАРАКТЕРИСТИКИ.

Продолжим обсуждение условий отбора таблиц S-блоков стандарта DES, защищенных от атак линейного криптоанализа, начатое в работе [1]. Рассмотрим теперь возможные атаки на десятицикловую характеристику с тождественными циклами, представленную на рис. 1 под номером 6(десятицикловая характеристика под номером 7 рис.1, составленная только из активных S-блоков, описывается графом переходов под номером 4 на рис.2, а такой граф переходов, как уже было показано выше, для шифра DES не реализуем).

Нас будут интересовать характеристики, каждая из симметричных половинок которых содержат общее число активных S-блоков, не превышающее восьми:

$$\left[\left(\frac{16}{64}\right)^8 \cdot 2^7\right]^3 \cdot 2^2 = 2^{-25}, \left[\left(\frac{16}{64}\right)^9 \cdot 2^6\right]^3 \cdot 2^2 = 2^{-34}.$$

Будем сначала рассматривать характеристики, в которых каждый из символов, использованных при их описании, обозначает однобитный вход или выход S-блока, участвующего в их образовании. Пусть будет допустимой характеристика 6, рис. 1. Из этой характеристики следует, что выполняются переходы $Z \leftarrow \Phi$, $\Phi \leftarrow \Psi$, $Z \oplus \Psi \leftarrow \Theta$ и $\Phi \oplus \Theta \leftarrow Z \oplus \Psi$. Для шифра DES (для однобитных значений $\Gamma$, $\Psi$, $\Phi$ и $\Theta$) это обозначает, что не выполняются обратные переходы $\Phi \leftarrow \Gamma$ и $\Theta \leftarrow \Psi$. Но тогда из выполнимости перехода $\Phi \oplus \Theta \leftarrow Z \oplus \Psi$ в нижней части характеристики при условии, что существует переход $\Phi \leftarrow \Psi$, следует, что должен выполняться и переход $\Theta \leftarrow Z$ (так как $\Phi \neq \Theta$). Выполнение же перехода $Z \oplus \Psi \leftarrow \Theta$ (при однобитных входах и выходах S-блоков первые три цикла рассматриваемой характеристики будут одноблочными) обязывает, чтобы существовали одновременно раздельные переходы $\Gamma \leftarrow \Theta$ и $\Psi \leftarrow \Theta$.

Приведенные соображения позволяют имеющиеся и установленные связи между входами и выходами S-блоков представить в виде графа переходов, изображенного под номером 6.1 на рис. 2.

Но характеристика вида 6, рис. 1 и соответствующий ей граф переходов не являются единственно возможными для десятицикловых характеристик. Наряду с рассматриваемой допустимыми будут еще несколько вариантов характеристик, получающихся путем варьирования допустимыми композициями входов и выходов S-блоков, задействованных при построении характеристики. Симметричные половины некоторых из них без своих зеркальных дополнений представлены вместе с "исходной" на рис. 3.

Заметим теперь, что, например, характеристики 2, 4 и 6 могут быть получены соответственно из характеристик 1, 2 и 3, если символы $\Psi$ заменить на символы $\Gamma$, а символы $\Theta$ заменить на символы $\Phi$, и поэтому в данном случае можно рассматривать только характеристики трех типов. Анализ и других возможных вариантов характеристик

1

$Z \leftarrow \Phi$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$\Psi \leftarrow \Phi$

$\Theta \leftarrow Z \oplus \Psi$

4-х цикловая итеративная характеристика

2

$Z \leftarrow \Phi$

$\Phi \leftarrow Z$

$0_x \leftarrow 0_x$

$\Phi \leftarrow Z$

$Z \leftarrow \Phi$

$0_x \leftarrow 0_x$

6-ти цикловая итеративная характеристика

3

$Z \leftarrow \Phi$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$\Psi \leftarrow \Theta$

$\Phi \leftarrow Z$

$Z \oplus \Psi \leftarrow \Phi \oplus \Theta$

$\Theta \leftarrow \Psi$

6-ти цикловая итеративная характеристика

4

$Z \leftarrow \Phi$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$\Psi \leftarrow \Theta$

$\Phi \leftarrow Z \oplus \Psi$

$Z \leftarrow \Theta$

$\Phi \oplus \Theta \leftarrow \Psi$

$Z \oplus \Psi \leftarrow \Phi$

$\Theta \leftarrow \Psi$

8-ми цикловая итеративная характеристика

5

$Z \leftarrow \Phi$

$\Phi \leftarrow Z \oplus \Psi$

$\Psi \leftarrow \Phi$

$0_x \leftarrow 0_x$

$\Psi \leftarrow \Phi$

$\Phi \leftarrow Z \oplus \Psi$

$Z \leftarrow \Phi$

$0_x \leftarrow 0_x$

8-ми цикловая итеративная характеристика

6

$Z \leftarrow \Phi$

$\Phi \leftarrow \Psi$

$Z \oplus \Psi \leftarrow \Theta$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$0_x \leftarrow 0_x$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$Z \oplus \Psi \leftarrow \Theta$

$\Phi \leftarrow \Psi$

$Z \leftarrow \Phi$

$0_x \leftarrow 0_x$

10-ти цикловая итеративная характеристика

7

$Z \leftarrow \Phi$

$\Phi \oplus \Theta \leftarrow \Psi$

$Z \oplus \Psi \leftarrow \Phi$

$\Phi \leftarrow Z$

$\Psi \leftarrow \Theta \oplus k$

$\Theta \leftarrow Z \oplus \Psi$

$Z \leftarrow \Phi$

$\Phi \oplus \Theta \leftarrow \Psi$

$Z \oplus \Psi \leftarrow \Phi$

$\Theta \leftarrow \Psi$

10-ти цикловая итеративная характеристика

Рис.1

на рис. 2. Что касается характеристик с графом переходов 6,1, который построен с использованием одного циклического перехода, то при однобитной интерпретации символов обозначений в соответствии с реальным видом характеристики следует считать, что $\Phi$ и $\Theta$ – это входные биты одного и того же S-блока, точнее, одноименных S-блоков, в то время как $\Gamma$ и $\Psi$ – это входные биты разных S-блоков. Поэтому речь должна идти не об однобитном циклическом переходе между двумя одноименными S-блоками, а о циклическом переходе $\Phi \oplus \Theta \leftarrow Z \oplus \Psi \leftarrow \Phi \oplus \Theta$, который может и не быть объединением однобитных циклических переходов $\Phi \leftarrow \Gamma$ и $\Theta \leftarrow \Psi$ (в нем отсутствуют переходы, но может существовать переход $\Psi \leftarrow \Phi$, так как при наличии еще хотя бы одного из переходов $\Phi \leftarrow \Gamma$ или $\Theta \leftarrow \Psi$ результирующая характеристика для шифра DES становится нереализуемой).

Переход $\Phi \oplus \Theta \leftarrow Z \oplus \Psi \leftarrow \Phi \oplus \Theta$ для шифра DES может состояться лишь с использованием трех различных S-блоков, при этом один из его полупереходов должен

быть одноблочным ("зайти" в один и тот же S-блок, т.е. попасть в одну и ту же вершину графа,

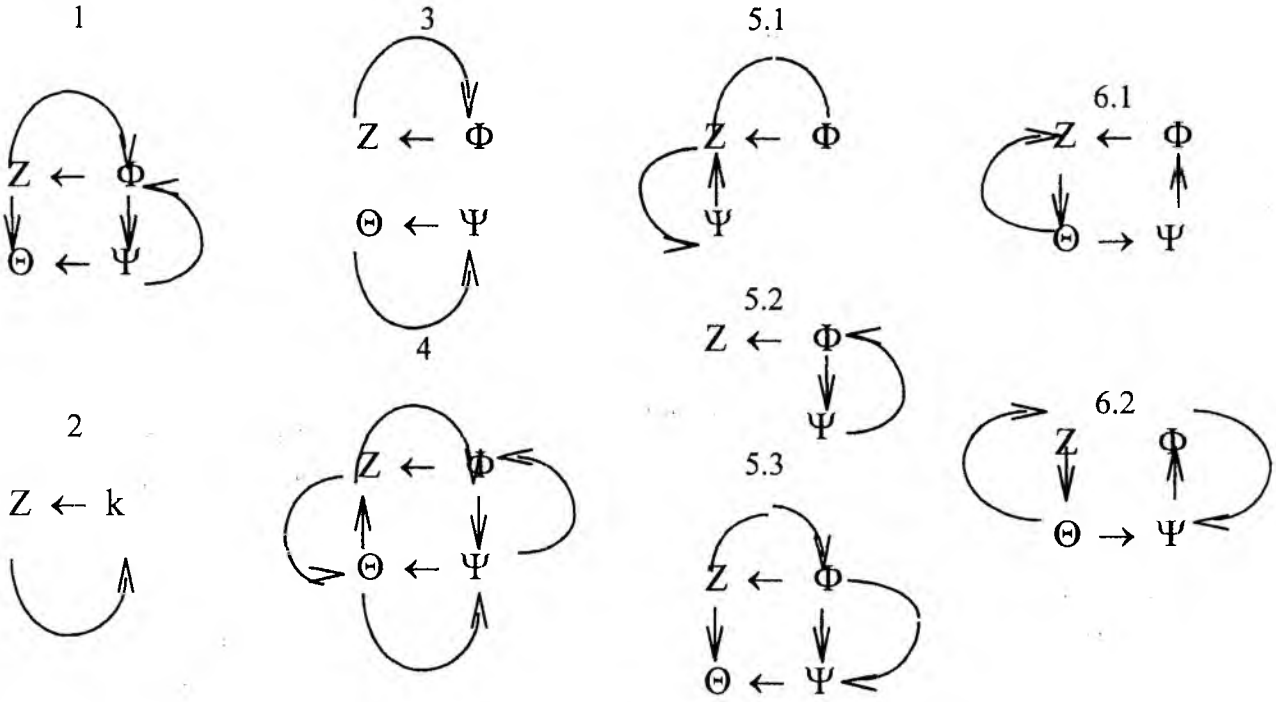



Рис.2

1

$Z \leftarrow \Phi$

$\Phi \leftarrow \Psi$

$Z \oplus \Psi \leftarrow \Theta$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$0_x \leftarrow 0_x$

2

$\Psi \leftarrow \Theta$

$\Theta \leftarrow Z$

$Z \oplus \Psi \leftarrow \Phi$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$0_x \leftarrow 0_x$

3

$\Phi \leftarrow \Psi$

$\Psi \leftarrow \Theta$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$Z \leftarrow \Phi \oplus \Theta$

$0_x \leftarrow 0_x$

4

$\Theta \leftarrow Z,$

$Z \leftarrow \Phi,$

$\Phi \oplus \Theta \leftarrow Z \oplus \Psi$

$\Psi \leftarrow \Phi \oplus \Theta$

$0_x \leftarrow 0_x$

5

$\Phi \leftarrow \Psi$

$\Psi \leftarrow \Phi \oplus \Theta$

$\Theta \leftarrow Z$

$Z \oplus \Psi \leftarrow \Theta$

$0_x \leftarrow 0_x$

6

$\Theta \leftarrow \Gamma$

$Z \leftarrow \Phi \oplus \Theta$

$\Phi \leftarrow \Psi$

$Z \oplus \Psi \leftarrow \Phi$

$0_x \leftarrow 0_x$

Рис. 3

можно только из вершин, "принадлежащих" разным S-блокам). Для иллюстрации на рис. 4 представлены различные варианты построения таких характеристик при использовании трех S-блоков $S_1$, $S_2$ и $S_5$:

$$\left.\begin{matrix} S_1(4_x, 4_x) \\ S_2(2_x, 1_x) \end{matrix}\right\} \Leftrightarrow S_5(18_x, 9_x)$$ или в побитовой записи $\begin{matrix} 3 \leftarrow 17 \\ 8 \leftarrow 18 \end{matrix} \Leftrightarrow 17,18 \leftarrow 3,8.$

1

$8 \leftarrow 17$ 1

$17 \leftarrow 3$ 1

$3 \oplus 8 \leftarrow 18$ 1

$17 \oplus 18 \leftarrow 3 \oplus 8$ 2

$0_x \leftarrow 0_x$

2

$3 \leftarrow 18$ 1

$18 \leftarrow 8$ 1

$3 \oplus 8 \leftarrow 17$ 1

$17 \oplus 18 \leftarrow 3 \oplus 8$ 2

$0_x \leftarrow 0_x$

3

$17 \leftarrow 3$ 1

$3 \leftarrow 18$ 1

$17 \oplus 18 \leftarrow 3 \oplus 8$ 2

$8 \leftarrow 17 \oplus 18$ 1

$0_x \leftarrow 0_x$

4

$18 \leftarrow 8$ 1

$8 \leftarrow 17$ 1

$17 \oplus 18 \leftarrow 3 \oplus 8$ 2

$3 \leftarrow 17 \oplus 18$ 1

$0_x \leftarrow 0_x$

5

$17 \leftarrow 3$ 1

$3 \leftarrow 17 \oplus 18$ 1

$18 \leftarrow 8$ 1

$3 \oplus 8 \leftarrow 18$ 1

$0_x \leftarrow 0_x$

6

$18 \leftarrow 8$ 1

$8 \leftarrow 17 \oplus 18$ 1

$17 \leftarrow 3$ 1

$3 \oplus 8 \leftarrow 17$ 1

$0_x \leftarrow 0_x$

Рис. 4

Здесь задействованы циклические однобитные переходы $18 \leftarrow 8 \leftarrow 18$, $17 \leftarrow 3 \leftarrow 17$ и циклический двубитный переход $3,8 \leftarrow 17,18 \leftarrow 3,8$, при этом допустимыми являются также переходы $18 \leftarrow 3$ и $8 \leftarrow 17$, в то время как переходы $17 \leftarrow 8$ и $18 \leftarrow 3$ являются нереализуемыми. Четыре первых характеристики рис. 4 состоят из трех одноблочных циклов и одного двухблочного, остальные – являются одноблочными. Первые четыре характеристики кроме двух одноблочных циклов с однобитными переходами также включают двублочный цикл с однобитными переходами. При этом минимум два однобитных перехода для этих характеристик имеются в списке ограничений У-4, что и обеспечивает их "неуязвимость" атакам линейного криптоанализа (если один из однобитных переходов не из списка ограничений У-4, то второй однобитный переход для шифра DES нереализуем):

$$\left[\left(\frac{16}{64}\right)^3 \cdot \left(\frac{4}{64}\right)^2 \cdot 2^4\right]^3 \cdot 2^2 = 2^{-28}.$$

В одноблочных характеристиках под номерами 5 и 6, рис.4 сохраняются однобитными одновременно два перехода. Они попадают в циклы без свободных выходов, и поэтому даже если один из переходов оказывается не из списка ограничений У-4, то второй непременно имеет вероятность, равную нулю - задействуются входы S-блоков $1_x$ или $20_x$. Если оба эти однобитных перехода оказываются из списка ограничений У-4, то тогда ограничений У-4 для перекрытия подобных характеристик оказывается уже недостаточно. Действительно, в этом случае для вероятности пятнадцатицикловой характеристики приходим к оценке:

$$\left[\left(\frac{4}{64}\right)^2 \cdot \left(\frac{16}{64}\right)^2 \cdot 2^3\right]^3 \cdot 2^2 = 2^{-25}, \qquad (1)$$

чего для перекрытия подобных характеристик явно недостаточно.

Заметим, однако, что эти две характеристики (как и предыдущие) удовлетворяют также введенному ранее ограничение У-5.

**Условие У-5** (условие защиты от атак ЛК теперь уже на десятицикловые итеративные аппроксимации). Элементы ТРЛА S-блоков, удовлетворяющие условиям $W(\alpha)$, $W(\beta) \le 2$, должны подчиняться ограничению $|NS(\alpha,\beta)| \le 10$.

С учетом этого ограничения приходим к результату

$$\left[\left(\frac{4}{64}\right)^2 \cdot \left(\frac{10}{64}\right)^2 \cdot 2^3\right]^3 \cdot 2^2 = 2^{-29}.$$

Заметим, однако, что ограничение на однобитные переходы, введенное корейскими учеными, обеспечивает перекрытие всех рассмотренных выше десятицикловых характеристик.

Если идти дальше, то на основе конкатенации (объединения) характеристик простейшего типа, рассмотренных выше, могут быть построены характеристики, использующие большее число циклических переходов между S-блоками. Пример построения таких характеристик иллюстрирует рис. 5.

1

$22 \oplus 32 \leftarrow 24 \oplus 28$

$24 \oplus 28 \leftarrow 7 \oplus 12$

$7 \oplus 12 \oplus 22 \oplus 32 \leftarrow 27 \oplus 29$

$24 \oplus 27 \oplus 28 \oplus 29 \leftarrow 7 \oplus 12 \oplus 22 \oplus 32$

$0_x \leftarrow 0_x$

2

$7 \oplus 12 \leftarrow 27 \oplus 29$

$27 \oplus 29 \leftarrow 22 \oplus 32$

$7 \oplus 12 \oplus 22 \oplus 32 \leftarrow 24 \oplus 28$

$24 \oplus 27 \oplus 28 \oplus 29 \leftarrow 7 \oplus 12 \oplus 22 \oplus 32$

$0_x \leftarrow 0_x$

3

$24 \oplus 28 \leftarrow 7 \oplus 12$

$7 \oplus 12 \leftarrow 27 \oplus 29$

$24 \oplus 27 \oplus 28 \oplus 29 \leftarrow 7 \oplus 12 \oplus 22 \oplus 32$

$22 \oplus 32 \leftarrow 24 \oplus 27 \oplus 28 \oplus 29$

$0_x \leftarrow 0_x$

4

$27 \oplus 29 \leftarrow 22 \oplus 32$

$22,32 \leftarrow 24 \oplus 28$

$24 \oplus 27 \oplus 28 \oplus 29 \leftarrow 7 \oplus 12 \oplus 22 \oplus 32$

$7 \oplus 12 \leftarrow 24 \oplus 27 \oplus 28 \oplus 29$

$0_x \leftarrow 0_x$

5

$27 \oplus 29 \leftarrow 22 \oplus 32$

$22 \oplus 32 \leftarrow 24 \oplus 27 \oplus 28 \oplus 29$

$24 \oplus 28 \leftarrow 7 \oplus 12$

$7 \oplus 12 \oplus 22 \oplus 32 \leftarrow 24 \oplus 28$

$0_x \leftarrow 0_x$

6

$24 \oplus 28 \leftarrow 7 \oplus 12$

$7 \oplus 12 \leftarrow 24 \oplus 27 \oplus 28 \oplus 29$

$27 \oplus 29 \leftarrow 22 \oplus 32$

$7 \oplus 12 \oplus 22 \oplus 32 \leftarrow 27 \oplus 29$

$0_x \leftarrow 0_x$

Рис. 5

Из приведенных на рис. 5 характеристик четыре: 1, 2, 3 и 4–содержат восемь активных S-блоков, а 5 и 6 – шесть. Все шестиблочные характеристики имеют двублочные циклы, которые состоят из однобитных переходов, причем, хотя бы три из этих переходов имеются в списке ограничений условий У-4. Легко убедиться, что использование при построении таких характеристик хотя бы двух однобитных переходов из списка ограничений У-4 уже обеспечивает полную их безопасность для атак линейного криптоанализа:

$$\left[\left(\frac{16}{64}\right)^4 \cdot \left(\frac{4}{64}\right)^2 \cdot 2^5\right]^3 \cdot 2^2 = 2^{-29}.$$

Но все же существуют десятицикловые характеристики, которые требуют дополнительных ограничений. Они могут быть построены на основе использования циклических двубитных переходов. Условием их образования является использование S- блоков, выходы которых инициируют входы двух смежных S-блоков очередного цикла. Таких S-блоков три: $S_5$ (3 и 8 биты), $S_6$ (4 и 11 биты), и $S_7$ (7 и 12 биты). Примеры построения таких характеристик представлены на Рис. 6.

| 1 | 3 | 5 |
|---|---|---|
| 17,18 ← 5 2 | 5 ← 28,31 1 | 5 ← 28,31 1 |
| 5 ← 28,31 1 | 28,31 ← 3,8 2 | 28,31 ← 3,5,8 2 |
| 17,18,28,31 ← 3,8 2 | 3,5,8 ← 17,18,28,31 2 | 3,8 ← 17,18 1 |
| 3,5,8 ← 17,18,28,31 2 | 17,18 ← 3,5,8 2 | 17,18,28,31 ← 3,8 2 |
| $0_x \leftarrow 0_x$ | $0_x \leftarrow 0_x$ | $0_x \leftarrow 0_x$ |
| 2 | 4 | 6 |
| 28,31 ← 3,8 2 | 3,8 ← 17,18 1 | 3,8 ← 17,18 1 |
| 3,8 ← 17,18 1 | 17,18 ← 5 2 | 17,18 ← 3,5,8 2 |
| 17,18,28,31 ← 5 2 | 3,5,8 ← 17,18,28,31 2 | 5 ← 28,31 1 |
| 3,5,8 ← 17,18,28,31 2 | 28,31 ← 3,5,8 2 | 17,18,28,31 ← 5 2 |
| $0_x \leftarrow 0_x$ | $0_x \leftarrow 0_x$ | $0_x \leftarrow 0_x$ |

Рис. 6

Действительно, как показывает анализ завершающей цикловую функцию *P*-подстановки, существуют пары смежных S-блоков, входящие в двублочный циклический переход, которые можно активизировать дополнительным входным битом (одним), входящим в еще один одноблочный циклический переход, причем последний имеет общие биты с исходным двублочным циклическим переходом. Всего существует три S-блока, выходы которых активизируют одновременно два смежных S-блока: $S_5$ (выходные биты *P*-подстановки 3,8 – входы S-блоков $S_1$, $S_2$), $S_6$ (выходные биты *P*-подстановки 4,11 – входы S блоков $S_2$, $S_3$) и $S_7$ (выходные биты *P*-подстановки 7,12 – входы S-блоков $S_3$, $S_4$). В первом случае возникает дополнительные циклические переходы 5 ← 28,31 ← 5, во втором – 8 ← 16,18 ← 8 и в третьем – 8 ← 16,18 ← 8, причем, нас интересуют только характеристики, при построении которых используются одноблочные циклы. Для десятицикловой характеристики из двублочных циклов хотя бы один будет состоять из S- блоков с однобитными переходами.

Характеристики под номерами 1 и 2 содержат по два однобитных перехода, а для характеристики 1 один из переходов имеет нулевую вероятность. И тогда для вероятности пятнадцатицикловой характеристики получаем оценку

$$\left[\left(\frac{4}{64}\right)^2 \cdot \left(\frac{16}{64}\right)^5 \cdot 2^6\right]^3 \cdot 2^2 = 2^{-34}.$$

Но зато все другие характеристики действительно строятся или могут быть построены без однобитных переходов. Следовательно, здесь уже ограничение У-4 не работает. В то же время рассматриваемые характеристики гарантированно имеют минимум два

цикла с двубитными входами, при этом выходы S-блоков этих циклов могут иметь не более чем двубитные выходы. Общее число активных S-блоков, приходящееся на симметричную половину такой характеристики, равно 6 или 7. Здесь можно воспользоваться условием $L$-4, введенным корейскими учеными для перекрытия восьмицикловых характеристик (в наших обозначениях – это условие У-5, введенное для перекрытия шестицикловых характеристик и уже использованное выше для перекрытия десятицикловых характеристик).

Для характеристик 3-6, рис.7, все S-блоки удовлетворяют условию У-5. В этом случае для вероятности пятнадцатицикловой характеристики получим оценку

$$\left[\left(\frac{10}{64}\right)^7 \cdot 2^6\right]^3 \cdot 2^2 = 2^{-36}.$$

В приведенных выше расчетах не учитывается еще один дополнительный цикл и возможность свободного выбора начального и заключительного циклов линейной аппроксимации. Однако во всех рассмотренных случаях имеется запас, позволяющий заключить, что десятицикловые характеристики являются защищенными от атаки линейного криптоанализа.

Других дополнительных ограничений в виде Условия $L$-5, использованного корейскими учеными, здесь уже не требуется.

**Требования к отбору S-блоков, защищенных от атак ЛК на двенадцатицикловые итеративные линейные аппроксимационные характеристики**

В этом случае анализу подлежат характеристики с общим число S-блоков, приходящихся на двенадцать, четырнадцать и шестнадцать циклов меньшее 17, 19,22 соответственно

$$\left(\frac{16}{64}\right)^{22} \cdot 2^{15} = 2^{-29}.$$

Для этих характеристик выполняется циклический переход $Z \oplus \Psi \leftarrow \Phi \oplus \Theta \leftarrow Z \oplus \Psi$, при этом для первой из них являются допустимыми переходы $Z \leftarrow \Phi$, $Z \leftarrow \Theta$, $\Phi \leftarrow \Psi$ и $\Theta \leftarrow \Psi$. Но тогда являются допустимыми переходы $\Psi \leftarrow \Phi$ и $\Psi \leftarrow \Theta$. Граф переходов для этой характеристики представлен под соответствующим номером на рис.8. Из этого графа следует, что рассматриваемая характеристика может быть построена на основе двух однобитных переходов с общим битом. Все такие пары циклических однобитных переходов (их всего семь) имеют несовпадающие биты, принадлежащие входам одного и того же S-блока, и, следовательно, в этом случае переход $Z \oplus \Psi \leftarrow \Phi \oplus \Theta$ является одноблочным. В результате рассматриваемая двенадцатицикловая харктеристика является одноблочной – ее т.е. и характеристики этого типа оказываются не опасными для атак ЛК.

Пример характеристики второго типа также приведен на рис.8, а на рис.7 под соответствующим номером приведен ее граф переходов.

Симметричная половина содержит 5 S-блоков. Ограничение У-4 на однобитные переходы в этом случае обеспечивает полное перекрытие таких характеристик.

Понятно, что если хотя бы один из однобитных переходов имеет нулевую вероятность, то такая характеристика нереализуема. Пример построения характеристики рассматриваемого вида приведен на рис. 8.

$$\left(\frac{4}{64}\right)^{21} \cdot 2^{15} = 2^{-69}.$$

Подчеркнем, что и в характеристиках неминимального типа (использующих свободу в выборе выходов S-блоков) имеются однобитные переходы (обеспечивающие согласование с тождественными циклами). Их сохраняется четыре на двенадцатицикловую характеристику, что приводит к оценке

$$\left(\frac{16}{64}\right)^{12}\cdot\left(\frac{4}{64}\right)^{9}\cdot 2^{15}=2^{-45},$$

1

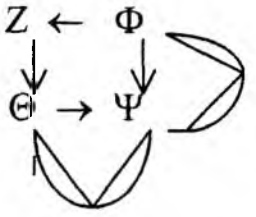
$$Z \leftarrow \Phi$$
$$\Phi \leftarrow \Psi$$
$$Z\oplus\Psi \leftarrow \Phi\oplus\Theta$$
$$\Theta \leftarrow \Psi$$
$$Z \leftarrow \Theta$$
$$0_x \leftarrow 0_x$$
$$Z \leftarrow \Theta$$
$$\Theta \leftarrow \Psi$$
$$Z\oplus\Psi \leftarrow \Phi\oplus\Theta$$
$$\Phi \leftarrow \Psi$$
$$Z \leftarrow \Phi$$
$$0_x \leftarrow 0_x$$

12-цикловая итеративная характеристика 1

2

$$Z \leftarrow \Phi$$
$$\Phi \leftarrow \Psi$$
$$Z\oplus\Psi \leftarrow \Phi\oplus\Theta$$
$$\Theta \leftarrow Z$$
$$\Psi \leftarrow \Theta$$
$$0_x \leftarrow 0_x$$
$$\Psi \leftarrow \Theta$$
$$\Theta \leftarrow Z$$
$$Z\oplus\Psi \leftarrow \Phi\oplus\Theta$$
$$\Phi \leftarrow \Psi$$
$$Z \leftarrow \Phi$$
$$0_x \leftarrow 0_x$$

12-цикловая итеративная характеристика 2



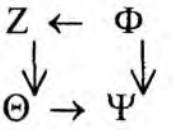




Рис. 7

Для этой характеристики оказываются в силе все высказанные выше соображения, и она также оказывается не опасной для атак ЛК.

Остается заметить, что в принципе можно построить двенадцатицикловую итеративную характеристику и без тождественных циклов, но она попадает в число нереализуемых – содержит нереализуемую композицию циклических однобитных переходов. Подобные же общие соображения можно высказать и об итеративных четырнадцатицикловых и шестнадцатицикловых характеристиках, и, значит, никаких дополнительных ограничений для итеративных характеристик с числом циклов превышающем десять не требуется.

Подводя итоги результатам исследований, представленных в этой и предшествующих работах [1,2], можно прийти к общему выводу, состоящему в том, что конечный набор критериев для отбора таблиц подстановок шифра DES,

1
$23 \leftarrow 1\ 1$

$1 \leftarrow 17$ 1
$17,23 \leftarrow 1,3$ 1

$3 \leftarrow 17$ 1
$23 \leftarrow 3$ 1

$0_x \leftarrow 0_x$

2.1

$17 \leftarrow 3$ 1

$3 \leftarrow 18$ 1

$17,18 \leftarrow 3,8$ 2

$8 \leftarrow 17$ 1

$18 \leftarrow 8$ 1

$0_x \leftarrow 0_x$

2.2

$8 \leftarrow 17$ 1

$17 \leftarrow 3$ 1

$3,8 \leftarrow 17,18$ 1

$18 \leftarrow 8$ 1

$8 \leftarrow 18$ 1

$0_x \leftarrow 0_x$

Рис. 8

устойчивых к атакам линейного криптоанализа, кроме требований разработчиков, на наш взгляд, должен включать в себя также следующие дополнительные ограничения:

**Условие У-1′** (объединенные У-4 и У-6; условие перекрытия шестицикловых и восьмицикловых итеративных аппроксимаций с однобитными переходами). Для ТРЛА S -блоков необходимо выполнить следующие (общее число 28 случаев) условия:

- *S*1-блок: $|NS_1(4_x, 4_x)| \le 4$, $|NS_1(2_x, 2_x)| \le 4$, $|NS_1(8_x, 8_x)| \le 4$ $|NS_1(10_x, 4_x)| \le 4$;
- *S*2-блок: $|NS_2(4_x, 4_x)| \le 4$, $|NS_2(2_x, 1_x)| \le 4$, $|NS_2(8_x, 8_x)| \le 4$ $|NS_2(10_x, 4_x)| \le 4$;
- *S*3-блок: $|NS_3(8_x, 4_x)| \le 4$, $|NS_3(4_x, 8_x)| \le 4$, $|NS_3(2_x, 8_x)| \le 4$;
- *S*4-блок: $|NS_4(8_x, 4_x)| \le 4$, $|NS_4(2_x, 2_x)| \le 4$, $|NS_4(4_x, 1_x)| \le 4$;
- *S*5-блок: $|NS_5(16_x, 1_x)| \le 4$, $|NS_5(8_x, 8_x)| \le 4$, $|NS_5(2, 4)| \le 4$ $|NS_5(4_x, 2_x)| \le 4$;
- *S*6-блок: $|NS_6(16_x, 4_x)| \le 4$, $|NS_6(4_x, 8_x)| \le 4$, $|NS_6(2_x, 2_x)| \le 4$ $|NS_6(8_x, 4_x)| \le 4$;
- *S*7-блок: $|NS_7(4_x, 8_x)| \le 4$, $|NS_7(2_x, 1_x)| \le 4$, $|NS_7(8_x, 4_x)| \le 4$;
- *S*8-блок: $|NS_8(16_x, 1_x)| \le 4$, $|NS_8(2_x, 4_x)| \le 4$, $|NS_8(4_x, 8_x)| \le 4$.

**Условие У-2′** (условие У-7 перекрытия восьмицикловых итеративных характеристик). Элементы ТРЛА пар S-блоков со выходными масками, удовлетворяющие условию $W(\alpha) = 1$, $W(\beta_1 \oplus \beta_2) = 1$, должны подчиняться ограничению:

$$\left|NS_k(\alpha, \beta_1) \cdot NS_k(\alpha, \beta_2)\right| \le 48.$$

**Условие У-3'** (условие У-3; условие защиты от атак на четырехцикловые и восьмицикловые итеративные характеристики). Элементы ТРЛА пар S-блоков, имеющие входные и выходные маски, удовлетворяющие условию $W(\alpha) \le 2$, $W(\beta_1 \oplus \beta_2) \le 2$, кроме ситуации $W(\alpha) = 2$, $W(\beta_1 \oplus \beta_2) = 2$ должны подчиняться ограничению:

$$\left|NS_k(\alpha,\beta_1) \cdot NS_k(\alpha,\beta_2)\right| \le 80.$$

**Условие У-4′** (условие У-5 защиты от атак ЛК на шестицикловые, восьмицикловые и десятицикловые итеративные аппроксимации). Элементы ТРЛА S-блоков, удовлетворяющие условиям $W(\alpha)$, $W(\beta) \le 2$, должны подчиняться ограничению $|NS(\alpha, \beta)| \le 10$.

В приведенных соотношениях $\alpha \in GF(2)^6$, $\beta \in GF(2)^4$, $W(\alpha)$ – вес битового входа, а $W(\beta)$ – вес битового выхода S блока.

**Список литературы:** 1. *И .В. Лисицкая, А. С. Бондаренко, А. И. Колыбельников* Обеспечение стойкости шифра DES к атакам линейного криптоанализа. Требования к отбору S-блоков, защищенных от атак на характеристики обнуляющего типа, четырехцикловые и шестицикловые итеративные аппроксимации // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып 119. С. 117-190. 2. *И. В. Лисицкая, А. С. Бондаренко, А. И. Колыбельников* Обеспечение стойкости шифра DES к атакам линейного криптоанализа. Требования к отбору S-блоков, защищенных от атак на восьмицикловые линейные итеративные аппроксимации // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып 124. С. - .

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 19.03.2002*

УДК. 681.3.06:519.248.681

*И.В. РУБАН, канд. техн. наук, С.В. ДУДЕНКО*

# ОЦЕНКА ПУТЕЙ ИСПОЛЬЗОВАНИЯ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ ХАРТЛИ В КРИПТОГРАФИЧЕСКИХ СИСТЕМАХ

В настоящее время вопрос актуальности защиты информации как в локальных, так и в глобальных информационных системах обсуждается во всех источниках, связанных с вопросами безопасности информации. Постоянный рост возможностей технического доступа к информационным ресурсам вызывает необходимость разработки новых методов и средств криптографической защиты информации. Подходы к решению этой задачи ограничиваются, как правило, двоичной арифметикой, что определяется классической теорией построения криптосистем [1, 2].

Выбор в данной работе преобразования Хартли обусловлен использованием вещественных последовательностей данных и рядом интересных на наш взгляд свойств данного преобразования. В отличие от преобразования Фурье, отображающего вещественные функции в комплексную область, и несимметричного по $i$ (происходит изменение знака при переходе от прямого к обратному преобразованию), преобразование Хартли осуществляет прямое и обратное преобразование только в вещественной области и обладает указанной симметрией.

Использование преобразования Хартли при разработке криптографической системы защиты информации вызвано определенным рядом преимуществ данного преобразования:

– преобразование Хартли относится к классу унитарных преобразований;

– преобразование Хартли обладает свойством симметрии;

– преобразование позволяет получить равномерное спектральное распределение коэффициентов по всей области преобразования данных;

– использование десятичной арифметики значительно расширяет алфавит и затрудняет криптоанализ;

– преобразование Хартли оперирует $N$ вещественными числами в отличие от $2N$ вещественных чисел других дискретных преобразований.

Прямое и обратное дискретное преобразование Хартли (ДПХ) определяется соотношениями [3, 4]

$$H(\nu) = N^{-1} \sum_{\tau=0}^{N-1} f(\tau) cas(2\pi\nu\tau / N),$$
$$f(\tau) = \sum_{\nu=0}^{N-1} H(\nu) cas(2\pi\nu\tau / N), \qquad (1)$$

где $cas(\Theta) = \cos(\Theta) + \sin(\Theta)$ .

Заметим, что любая секретная система $\boldsymbol{T}$ может быть записана как сумма фиксированных операций

$$T = \sum_{i=1}^{n} p_i \cdot Q_i,$$

где $\boldsymbol{Q_i}$ – определенная операция преобразования в системе $\boldsymbol{T}$, соответствующая выбору ключа $\boldsymbol{i}$, причем вероятность такого выбора равна $\boldsymbol{p_i}$.

Алгебра секретных систем раскрывает два способа комбинирования секретных систем с целью получения новой секретной системы [5,6]:

1. "Взвешенная сумма"

$$S = \sum_{i=1}^{n} p_i \cdot T_i \ , \ \sum_{i=1}^{n} p_i = 1,$$

где $T_i$ – $i$-я секретная система;

$p_i$ – вероятность выбора $i$-ой секретной системы.

Этот способ состоит из предварительного выбора системы $T_i$ с некоторой вероятностью $p_i$. После того как выбор сделан, система $T_i$ применяется в соответствии с определением. Полный ключ системы $S$ указывает, какая из систем используется и с каким ключом. При этом новая система имеет множество отображений, состоящее из совокупности всех множеств отображений, используемых секретных систем с вероятностями их использования, равными произведению вероятностей выбора этих отображений на вероятность выбора секретной системы.

2. "Произведение"

$$S = \prod_{i=1}^{n} T_i$$

при условии, что система $T_{i+1}$ имеет область определения (пространство языка) такую, что ее можно отождествить с областью определения (пространством криптограмм) системы $T_i$.

Этот способ состоит в последовательном применении секретных систем. Полный ключ системы $S$ состоит из ключей всех используемых систем. Если $V_i$ – количество всех ключей системы $T_i$ с вероятностью выбора $p_i$, тогда система $S$ имеет $\prod_{i=1}^{n} V_i$ ключей с вероятностью выбора $\prod_{i=1}^{n} p_i$.

Построим секретную систему, в которой в качестве фиксированных операций используется дискретное преобразование Хартли. Систему $T$ можно представить как:

$$T=(p_1 \cdot X_1 + p_2 \cdot X_2 + \ldots + p_m \cdot X_m \ ) \ , \qquad (2)$$

где $X_m$ – $N$-элементное дискретное преобразование Хартли, причем $N=2^m$ , $m=\overline{1,\infty}$ .

Чтобы обеспечить криптостойкость системы (2) к методу полного перебора, необходимо иметь количество ключей (а, следовательно, и количество фиксированных операций), исчисляемое порядком $10^{20}$ и более, что на практике не осуществимо.

"Возведение в степень" секретной системы позволяет расширить множество ключей до необходимого, используя две и более фиксированные операции:

$$S = T^l,$$

где $l$ – количество циклов, и при ограничении, что $T$ является эндоморфной системой – системой, у которой пространство языка и криптограмм можно отождествить [1]. (Этот случай является частым, например, если происходит преобразование последовательности букв в последовательность букв.) ДПХ оперирует вещественными числами, следовательно, является эндоморфным.

В общем случае такую систему можно представить в виде

$$S = \prod_{i=1}^{l} \left[ \sum_{j=1}^{n} p_j \cdot H_j(E_i) \right], \quad \sum_{i=1}^{n} p_j = 1, \tag{3}$$

где $H_j$ – $j$-я инволюция, такая что $H_j(E) \neq H_i(E)$, для всех $i \neq j$;

$p_j$ – вероятность выбора инволюции $H_j$.

Множество ключей системы (3) равно $m^l$, где $m$ – количество фиксированных операций, причем, если выбор инволюции на каждом цикле происходит равновероятно, то вероятность выбора $K_i$ – го ключа $p(K_i) = m^{-l}$.

Количество циклов преобразования влияет на криптостойкость шифра, может быть произвольным и задается пользователем. Ключ такой системы имеет формат

$$\{a_1 a_2 a_3 \cdots a_l\},$$

где $a_i \ \forall i = \overline{1,l}$ – символ, соответствующий выбору определенного ДПХ,

$l$ – количество циклов преобразования (длина ключа).

Таким образом, секретная система на основе ДПХ будет иметь вид:

$$T = (p_1 \cdot X_1 + p_2 \cdot X_2 + \ldots + p_m \cdot X_m)^l, \tag{4}$$

где $X_m$ - $N$-элементное дискретное преобразование Хартли, причем $N = 2^m$, $m = \overline{1, \infty}$.

Одним из недостатков преобразования Хартли является линейность оператора ДПХ, определяемого выражением

функция $f(\tau) = f_1(\tau) + f_2(\tau)$ имеет ДПХ $H(\nu) = H_1(\nu) + H_2(\nu)$.

Это свойство приводит к необходимости введения нелинейных преобразований при построении криптографических систем. Единственной информацией, которая увеличивает апостериорные знания криптоаналитика при анализе шифртекста, является экспонента чисел, получаемых на $k$-м цикле преобразования, так как она позволяет изменить апостериорную вероятность выбора ключа [7, 8]. При проведении статистических исследований было замечено, что в случае использования в качестве алфавита открытых сообщений целых неотрицательных чисел получается равномерное спектральное распределение коэффициентов по всей области преобразования данных; таким образом, умножая каждый символ шифртекста на усредненную экспоненту, возможно приведение результата преобразований к целочисленному формату. При этом производится округление чисел до такого количества знаков числа, которое позволяет производить однозначное восстановление открытого текста (в связи с по-

грешностью в вычислениях). В процессе шифрования параметры округления, а именно степень экспоненты, являются частью ключа.

При исследовании корреляционной зависимости между символами открытого текста и шифртекста криптосистемы (4) (табл. 1), использующей один тип ДПХ на всех циклах, были получены результаты, представленные в табл. 2. Анализ проводился с использованием 256 выборок.

**Таблица 1**

| № п/п | открытый текст | шифртекст с использованием 2-х элементного ДПХ (2ДПХ) | шифртекст с использованием 4-х элементного ДПХ (4ДПХ) | шифртекст с использованием 8-ми элементного ДПХ (8ДПХ) | шифртекст с использованием 16-ти элементного ДПХ (16ДПХ) |
|---|---|---|---|---|---|
| 1 | 1178882085 | 27447987464 | 6390732588 | 14879583820 | 4794898561 |
| 2 | 858665261 | 19992358540 | 4654833707 | 1998783358 | 6589009473 |
| 3 | 3823248653 | 89016944473 | 20725872478 | 48256182293 | -3422254577 |
| 4 | 168678351 | 3927348903 | 914407173 | 23468485364 | -6906244101 |
| 5 | 540024881 | 12573434063 | 2927480746 | 7635039233 | 9006365679 |
| 6 | 225075823 | 5240454874 | 1220138481 | 10860985233 | 4705832951 |
| 7 | 220216380 | 5127312150 | 1193795388 | 802360321 | 13028566458 |
| 8 | 1886999599 | 43935133121 | 10229445324 | 4454979297 | -6610817978 |

**Таблица 2**

| | Математическое ожидание | Коэффициент корреляции | константа | коэффициент |
|---|---|---|---|---|
| | | | b | k |
| открытый текст | 1858360217 | | | |
| шифртекст (2ДПХ) | 43268320556 | 1 | 0,0023906 | 0,0429496729599519 |
| шифртекст (4ДПХ) | 10074190925 | 1 | 0,0072695 | 0,1844674407367620 |
| шифртекст (8ДПХ) | 23455803573 | 0,7437944546833 | 476122192,903884 | 0,0589294679329340 |
| шифртекст (16ДПХ) | 5508634488 | 0,3103411857746 | 1735542418,63848 | 0,0222955070037177 |

Из табл. 2 видно, что с увеличением количества одновременно обрабатываемых символов коэффициент корреляции уменьшается.

Корреляционную зависимость можно выразить как [9]

$$y=k\cdot x+b \ ,$$

где: $y$ – случайная величина (символ открытого текста);

$x$ – случайная величина (символ шифртекста).

Проведенный анализ показал, что использование 2- либо 4-элементного ДПХ на каждом цикле преобразования приводит к однозначному восстановлению открытого текста без знания ключа, использование 8- и 16-элементного ДПХ позволяет криптоаналитику частично восстановить открытый текст. Использование в качестве фиксированных операций только разноэлементных ДПХ, имеющих по своей сути ячеистую структуру, приводит к появлению в некоторых случаях значимых коэффициентов корреляции. Диаграмма корреляционной зависимости "открытый – шифрованный текст" представлена на рис. 1.

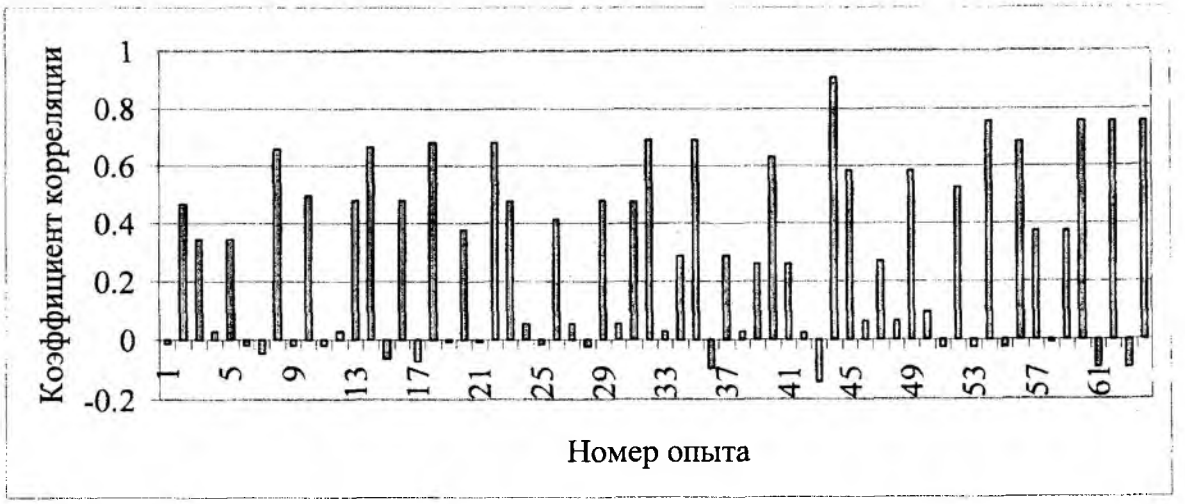



Рис.1

Улучшить показатели нелинейной зависимости (в пределах одного блока) позволяет ведение дополнительной перестановки, что также увеличивает скорость наступления лавинного эффекта и позволяет сократить время, затрачиваемое на шифрование одного блока ткрытого текста. Диаграмма корреляционной зависимости "открытый – шифрованный екст" представлена на рис. 2.


Рис.2

Оценка корреляционной зависимости показала, что коэффициент корреляции не является постоянным для различных ключей и что для сгенерированных случайным образом лючей имеем незначимые коэффициенты.

***Преимущества предлагаемого криптографического алгоритма:***

- большой размер шифрвеличин приводит к отсутствию идентичных криптограмм;

- отсутствие методики проведения криптоанализа предлагаемого криптографического алоритма в современной литературе, так как криптопротокол работает не с битовой последоательностью, а с вещественными числами;

- с увеличением количества циклов преобразований наблюдается лавинный эффект (завиимость каждого элемента блока шифртекста от исходного);

- при определенных параметрах исключается возможность силовой атаки;

- незначимые коэффициенты корреляционной зависимости между элементами открытого текста и шифртекста.

**Список литературы:** 1. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии. М.: Гелиос АРВ, 2001. 480 с. 2. *Столлингс В.* Криптография и защита сетей: принципы и практика. М.: Издательский дом "Вильямс", 2001. 672с. 3. *Брейсуэлл Р.* Преобразование Хартли. М.: Мир, 1990. 175с. 4. *Bracewell R.N.*. The Fourier Transform and Its Applications. McGraw-Hill, 1965. 234p. 5. *Шеннон К.* Работы по теории информации и кибернетике. М.: Изд. ин. лит., 1963. 829с. 6. Введение в криптографию./ Под ред *В.В. Ященко.* М.: МЦНМО: «ЧеРо», 2000. 288с. 7. *Чмора А.Л.* Современная прикладная криптография. М.: Гелиос АРВ, 2001. 256с. 8. *Зегжда Д.П., Ивашко А.М.* Как построить защищенную систему. СПб: Мир и семья-95, 1997. 312с. 9. *Гмурман В.Е.* Теория вероятностей и математическая статистика. М.: Высш. школа, 1977. 479с.

*Харьковский военный университет*

*Поступила в редколлегию 23.04.2002*

УДК 681.3.06: 519.248.681

*И.В. ЛИСИЦКАЯ канд. тех. наук, Т.В. ЦЕПУРИТ, С.В. СУПРОНЮК*

# АЛГОРИТМ ПОРОЖДЕНИЯ ПРОТИВОРЕЧИВЫХ ПОДСТАНОВОК

В работе [1] обоснована возможность применения для построения долговременных ключей алгоритма ГОСТ 28147-89 противоречивых подстановок. Показано, что порождать таблицы, построенные из таких подстановок случайным образом, безнадежно. В этой работе будет обоснован один из практически реализуемых алгоритмов порождения таблиц, составленных из противоречивых подстановок.

Будем рассматривать таблицу из $m$ подстановок $n$-ой степени в виде матрицы $S_{m,n}$, нулевая строка которой – упорядоченная перестановка 1,2,3, . . . , $n$ в сочетании с любой другой строкой-перестановкой матрицы образует подстановку. Нас будет интересовать задача построения матрицы $S_{m,n}$, в столбцах которой отсутствуют совпадения чисел, а в строках нет повторений, т.е. латинского прямоугольника размером $m \times n$ [2]. Она может быть интерпретирована как задача о нахождении максимального паросочетания в двудольном графе [2] (в нашем случае – графе $B_{n,n}$ с равными долями). Иногда для наглядности эту задачу представляют также как задачу нахождения максимальной системы браков: вершины одной доли графа интерпретируются как мужчины, вершины другой доли - как женщины, ребра графа задают допустимые браки. Нужно найти максимальное паросочетание в смысле числа брачных пар [3].

Продемонстрируем идею предлагаемого метода на примере определения системы подстановок $S_{4,5}$.

Для латинского прямоугольника 4 × 5 первая подстановка (нулевая и первая строка матрицы) не должна содержать идентичных элементарных переходов $i$ в $i$. В терминах браков нулевая строка, которая не проставляется, – это мужчины 1, 2, 3, 4, 5, а другие строки – это женщины 1′, 2′, 3′, 4′, 5′. Для $i = 1 \div 5$, $i$-ый мужчина может жениться на любой женщине кроме той, которая имеет одинаковый с ним номер, т.е. в графе $B_{5,5}$ присутствуют все ребра, кроме (1, 1), (2, 2), (3, 3), (4, 4) и (5, 5). Представим такой граф матрицей $E$ диагонального типа, с элементами 0 и 1 в виде:

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Номера столбцов матрицы соответствуют номерам вершин одной доли графа (номерам женщин), а номера строк соответствуют номерам вершин второй доли графа (номерам мужчин). Наличие 1 на позициях $E_{ii}$, $i = 1 \div 5$ обозначает отсутствие соответствующих ребер графа, наличие нуля на пересечении $i$-ого столбца и $j$-ой строки обозначает возможность соединения соответствующих вершин ребром (возможность брака $i$-ой женщины и $j$-ого мужчины).

Породим случайное число из интервала [1, n-k ], где $n$ –число столбцов матрицы, а $k$ – номер цикла алгоритма (на данный момент - это первый цикл). Под циклом будем понимать построение полного паросочетания графа. Пусть для первого цикла это будут числа $r_1 = 4, r_2 = 1, r_3 = 1, r_4 = 2$.

Далее алгоритм продолжается так: для $i$-ой вершины ($i$-ого мужчины) $i = 1, 2, 3, 4, 5$, просматриваются в порядке возрастания номеров (с циклическим возвратом в наименьший номер) все ребра, исключая те, которые ведут к запрещенным вершинам другой доли графа (занятым женщинам). В качестве состоявшегося ребра (брака) будет считаться соединение $i$-ой вершины одной доли графа с $r_i$-той вершиной из числа свободных вершин другой доли графа. Таким образом, мужчина 1 женится

на женщине 5; мужчина 2 женится на женщине 3; мужчина 3 женится на женщине 4; мужчина 4 женится на женщине 2; мужчина 5 женится на женщине 1, как это показано на рисунке 1.

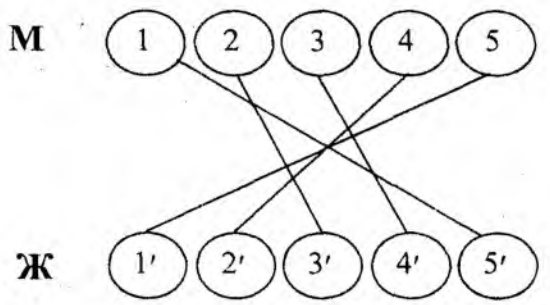



Рис. 1

Эта процедура применительно к матрице $E$ может быть интерпретирована как последовательный просмотр каждой из строк $j = 1, 2, 3, 4, 5$, начиная со столбца, соответствующего номеру мужчины, для которого ищется женщина. Отсчитывается с циклическим возвратом $r_i$ нулей в строке, и $r_i$-тый ноль при таком просмотре заменяется на единицу, обозначающую, что эта вершина уже занята (заключается брак между $i$-ым мужчиной и $j$-ой женщиной). Для того чтобы обозначить для последующих просмотров строк, что $j$-ая женщина уже занята, единицы проставляются для всего столбца, соответствующего рассмотренному мужчине. В результате мы последовательно получим:

$$\begin{bmatrix} 1&0&0&0&[1] \\ 0&1&0&0&1 \\ 0&0&1&0&1 \\ 0&0&0&1&1 \\ 0&0&0&0&1 \end{bmatrix} \rightarrow \begin{bmatrix} 1&0&1&0&1 \\ 0&1&[1]&0&1 \\ 0&0&1&0&1 \\ 0&0&1&1&1 \\ 0&0&1&0&1 \end{bmatrix} \rightarrow \begin{bmatrix} 1&0&1&1&1 \\ 0&1&1&1&1 \\ 0&0&1&[1]&1 \\ 0&0&1&1&1 \\ 0&0&1&1&1 \end{bmatrix} \rightarrow \begin{bmatrix} 1&1&1&1&1 \\ 0&1&1&1&1 \\ 0&1&1&1&1 \\ 0&[1]&1&1&1 \\ 0&1&1&1&1 \end{bmatrix} \rightarrow \begin{bmatrix} 1&1&1&1&1 \\ 1&1&1&1&1 \\ 1&1&1&1&1 \\ 1&1&1&1&1 \\ 1&1&1&1&1 \end{bmatrix}.$$

| шаг 1 | шаг 2 | шаг 3 | шаг 4 |
|---|---|---|---|
| $r_1 = 4$ | $r_2 = 1$ | $r_3 = 1$ | $r_4 = 2$ |

На последнем шаге мы выбираем единственный оставшийся ноль. Единицы, отмеченные в матрицах квадратными скобками, фиксируются как реализованные при рассматриваемом варианте паросочетания (ребра графа). В терминах построения системы подстановок это означает, что в первом столбце первой строки ставится число 5, во втором столбце - 4 и т.д., т.е.

$$S_{1,5} = \begin{pmatrix} 1&2&3&4&5 \\ 5&3&4&2&1 \end{pmatrix}.$$

Затем снова решается задача о браках, но теперь в графе $B_{5,5}$ нет ребер (1, 1) и (1, 5), (2, 2) и (2, 3), (3, 3) и (3, 4), (4, 4) и (4, 2), (5, 5) и (5, 1), что соответствует исходной матрице типа $E$ вида:

$$E = \begin{bmatrix} 1&0&0&0&1 \\ 0&1&1&0&0 \\ 0&0&1&1&0 \\ 0&1&0&1&0 \\ 1&0&0&0&1 \end{bmatrix}$$

Породим случайные числа из интервала [1; 3], так как это делалось ранее. Пусть, например, это будут числа $r_1 = 2$, $r_2 = 3$, $r_3 = 1$, $r_4 = 1$. Применяя вышеописанную процедуру, получаем последовательно:

$$\begin{bmatrix} 1 & 0 & [1] & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ [1] & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & [1] \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \rightarrow \blacksquare$$

| шаг 1 | шаг 2 | шаг 3 | шаг 4 |
|---|---|---|---|
| $r_1 = 2$ | $r_2 = 3$ | $r_3 = 1$ | $r_4 = 1$ |

На шаге 4 для четвертого мужчины женщины заняты (нет допустимых вершин второй доли графа). Приведенному результату соответствует граф, представленный на рис. 2:

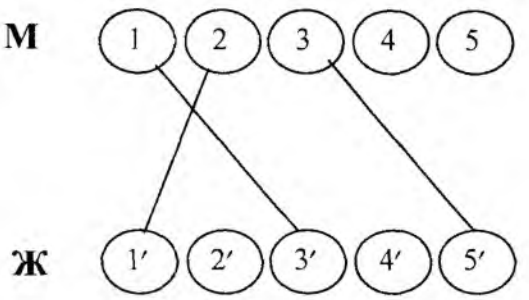



Рис. 2

В этом случае мы не получим полного паросочетания, так как мужчина под номером 4 не нашел пары. Его потенциальные невесты 1′, 2′, 3′, 5′ оказались уже занятыми с учетом состоявшихся выборов невест.

Здесь начинается другая часть алгоритма, связанная с текущей коррекцией выполняемой процедуры с тем, чтобы она реализовала вариант полного паросочетания.

Эта коррекция становится понятной, если возвратиться к матрице $E$ на 2-ом шаге. Из нее видно, что после выполнения второго шага в матрице появилась строка с одним нулем, т.е. альтернативы выбора для 4-ого мужчины нет. Здесь вполне естественно пойти по пути назначения сразу 4-ому мужчине 5-ой женщины, т.е. если в матрице появилась строка с одним нулем, то этот нуль принудительно следует заменить единицей (в графе проставляется ребро).

Теперь появилась новая строка с одним нулем, который снова принудительно меняется на единицу и далее опять остается строка с одним нулем, и мы приходим к графу:

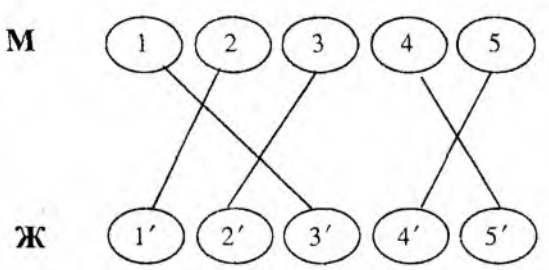



Рис. 3

В терминах подстановок мы получим допустимую систему:

$$S_{2,5} = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 5\,3\,4\,2\,1 \\ 3\,1\,2\,5\,4 \end{pmatrix}.$$

Снова решаем задачу о браках для графа заданного теперь матрицей:

$$E = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Пусть получены случайные числа из интервала [1, 2] $r_1 = 1$, $r_2 = 2$, $r_3 = 1$, $r_4 = 1$. Применяя оговоренные выше процедуры, последовательно получим:

$$\begin{bmatrix} 1\,[1]\,1\,0\,1 \\ 1\,1\,1\,0\,0 \\ 0\,1\,1\,1\,0 \\ 0\,1\,0\,1\,1 \\ 1\,1\,0\,1\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,0\,1 \\ 1\,1\,1\,0\,[1] \\ 0\,1\,1\,1\,1 \\ 0\,1\,0\,1\,1 \\ 1\,1\,0\,1\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,0\,1 \\ 1\,1\,1\,0\,1 \\ [1]\,1\,1\,1\,1 \\ 1\,1\,0\,1\,1 \\ 1\,1\,0\,1\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,0\,1 \\ 1\,1\,1\,0\,1 \\ 1\,1\,1\,1\,1 \\ 1\,1\,[1]\,1\,1 \\ 1\,1\,1\,1\,1 \end{bmatrix}.$$

| Шаг 1 | шаг 2 | шаг 3 | шаг 4 |
|---|---|---|---|
| $r_1 = 1$ | $r_2 = 2$ | $r_3 = 1$ | $r_4 = 1$ |

И снова не получено полного паросочетания. Здесь, когда нулей в матрице остается мало, на первый план выступают не статистические, а детерминированные переходы, а поэтому выходом из создавшегося положения будет использование альтернативного решения, заключающегося в выборе тех значений подстановки, используя которые вероятность полного паросочетания увеличивается. Необходимо в процессе построения подстановки отслеживать количество нулей в строках графа. В случае, когда в строке оказывается сравнительно малое количество нулей, необходимо в первую очередь фиксировать соответствующий переход. Когда выбор перехода не однозначен, необходимо фиксировать то значение подстановки, которое минимально ограничивает выбор последующих, т.е. фиксируется то значение, при котором вычеркивается меньшее количество нулей в столбце графа.

Используя приведенные соображения, в нашем случае получим:

$$\begin{bmatrix} 1\,[1]\,1\,0\,1 \\ 1\,1\,1\,0\,0 \\ 0\,1\,1\,1\,0 \\ 0\,1\,0\,1\,1 \\ 1\,1\,0\,1\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,1\,1 \\ 1\,1\,[1]\,0 \\ 0\,1\,1\,1\,0 \\ 0\,1\,0\,1\,1 \\ 1\,1\,0\,1\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,1\,1 \\ 1\,1\,1\,1 \\ 0\,1\,1\,1\,[1] \\ 0\,1\,0\,1\,1 \\ 1\,1\,0\,1\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,1\,1 \\ 1\,1\,1\,1\,1 \\ 1\,1\,1\,1\,1 \\ [1]\,1\,0\,1\,1 \\ 1\,1\,01\,1 \end{bmatrix} \rightarrow \begin{bmatrix} 1\,1\,1\,1\,1 \\ 1\,1\,1\,1\,1 \\ 1\,1\,1\,1\,1 \\ 1\,1\,1\,1\,1 \\ 1\,1\,[1]\,1\,1 \end{bmatrix}$$

Этому решению соответствует граф, представленный на рис. 4.

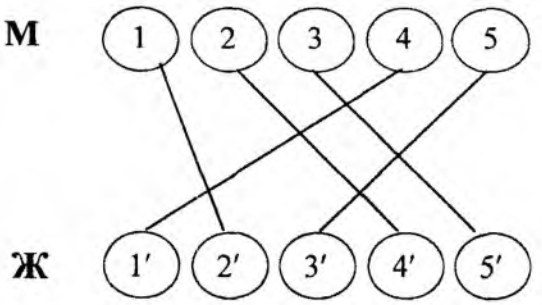



Рис. 4

В других обозначениях построена система:

$$S_{4,5} = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 5\,3\,4\,2\,1 \\ 3\,1\,2\,5\,4 \\ 2\,4\,5\,1\,3 \end{pmatrix}.$$

Последняя подстановка получается единственным образом:

$$S_{5,5} = \begin{pmatrix} 1\,2\,3\,4\,5 \\ 5\,3\,4\,2\,1 \\ 3\,1\,2\,5\,4 \\ 2\,4\,5\,1\,3 \\ 4\,5\,1\,3\,2 \end{pmatrix}.$$

Подобные способы порождения таблиц противоречивых подстановок значительно ограничивают их случайность, поэтому рекомендуется использовать минимальное количество ограничений. Для увеличения случайности таблицы необходимо выбирать разрешенные позиции переходов в произвольном (случайном) порядке.

В реальном случае для шифра ГОСТ требуется не квадратная таблица, а прямоугольник размером 8x16. В этом случае анализ графа Е при генерации подобных таблиц можно не производить, задача будет решена за один проход без возвратов. Однако исключить анализ графа при построении таблиц большего размера нельзя. Вышеупомянутые методы построения противоречивых подстановок могут быть использованы для построения таблиц с другими свойствами, такими как цикличность подстановок. Отличие будет сказываться лишь на построении графа Е.

Программа позволяет, при использовании ПЭВМ, формировать матрицу 256x256 противоречивых подстановок за 1 минуту или 10-15 таблиц подстановок за одну минуту.

**Список литературы:** 1. *Лисицкая И.В.* Противоречивые подстановки в алгоритме ГОСТ 28147-89//. Информационные системы: Сб. научн. тр.-Харьков: НАНУ, ПАНУ, ХВУ, 1995. 9 с. 2. Математическая энциклопедия. Сов. энциклопедия – М.: 1979.Т. 2.1103с.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 24.04.2002*

УДК 681.3.06: 519.248.681

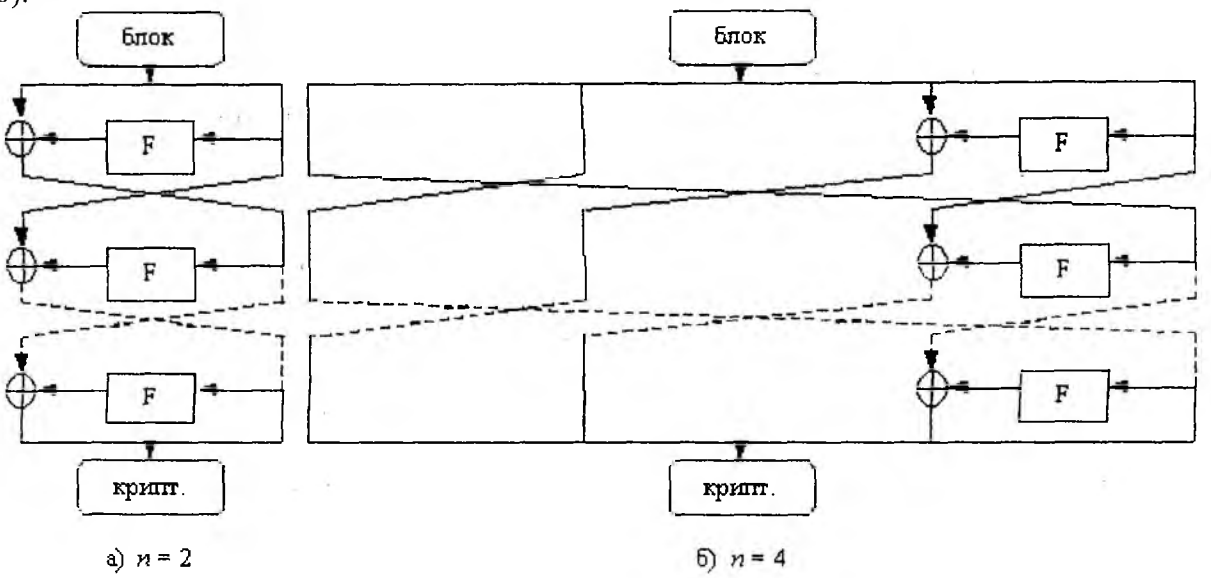
*И.В. ЛИСИЦКАЯ, канд. техн. наук, В.И. РУЖЕНЦЕВ*

# ЦЕПИ ФЕСТЕЛЯ И ДИФФЕРЕНЦИАЛЬНЫЙ КРИПТОАНАЛИЗ

На сегодняшний день не вызывает сомнения тот факт, что будущее информационных технологий неразрывно связано с совершенствованием методов и способов обеспечения конфиденциальности информации. Неоспорима также огромная роль симметричных блочных шифров в решении этого вопроса, так как данный класс шифров обладает наибольшей скоростью выполнения операций шифрования-дешифрования, чем и обусловлено широкое применение этих шифров. Однако, хороший симметричный блочный шифр должен отвечать ряду требований, среди которых одними из основных являются условия обеспечения стойкости шифра к различным видам криптоаналитических атак. В числе наиболее мощных криптоаналитических атак – дифференциальный криптоанализ (ДК). В этой связи актуальной представляется задача исследования вопроса о том, как модель шифра влияет на его стойкость к атакам дифференциального криптоанализа.

*Сегодня существует несколько моделей шифров.* В разных источниках приводятся различные классификации, но чаще всего выделяют шифры, построенные с использованием цепей Фестеля (DES, DEAL, E2, LOKI97, RC6, Twofish, MARS), и шифры, которые построены на основе чередования процедур перестановок и подстановок (SPN – substitution-permutation network) (Square, Rijndael, SAFER+, Serpent, CRYPTON).

Цепь Фестеля или конструкция Фестеля предполагает разбиение исходного информационного блока, в общем случае, на $n$ подблоков. На каждом цикле одна из частей подвергается преобразованию при помощи криптографического преобразования F. Результат операции суммируется по модулю 2 (операция XOR) с другой частью, и подблоки меняются местами. Классической схемой Фестеля считается вариант для $n = 2$, который представлен на рис. 1,а. Существуют также расширенные цепи Фестеля для $n > 2$ (см. пример для $n = 4$ на рис. 1,б).




а) $n = 2$ б) $n = 4$

Рис. 1

Преимущество конструкции Фестеля заключается, в первую очередь, в том, что прямое и обратное криптографические преобразования для блочного шифра имеют идентичную структуру, вследствие чего аппаратная реализация будет более компактной в сравнении с шифрами, для которых прямое и обратное преобразования отличаются. Кроме этого, конструкция Фестеля по сравнению с SPN-конструкцией позволяет использовать в шифрующей функции более широкий набор преобразований, так как к ним не предъявляется

требование обратимости. Наиболее известным примером фестелеобразного шифра, построенного на необратимой цикловой функции, является DES [1].

SPN-конструкция предполагает последовательное применение шифрующей функции, которая работает не с частями информационного блока, а сразу со всем блоком. Все преобразования в SPN-шифрах должны быть обратимыми, так как расшифрование производится путем выполнения обратных преобразований в обратном порядке.

При сравнении двух классов шифров будем привязываться к скорости выполнения криптографических преобразований. Поскольку при полном распараллеливании скорости выполнения одного цикла преобразований для SPN-структуры и структуры Фестеля примерно равны (хотя SPN-структура в этом случае требует большего распараллеливания), то будем сравнивать представителей этих классов с равным числом циклов.

В соответствии с классической работой Шеннона шифрующие функции состоят из операций перемешивания (нелинейные S-подстановки), рассеивания (битовые P-перестановки), введения секретности (сложение с ключом).

На рис. 2 представлены шифрующие функции для шифров с 16-битными информационными блоками, построенных с использованием классической цепи Фестеля (рис. 2,а – биективная шифрующая функция, рис. 2,б – с использованием сжимающе-расширяющих преобразований[1]) и по схеме SPN (рис. 2,в).

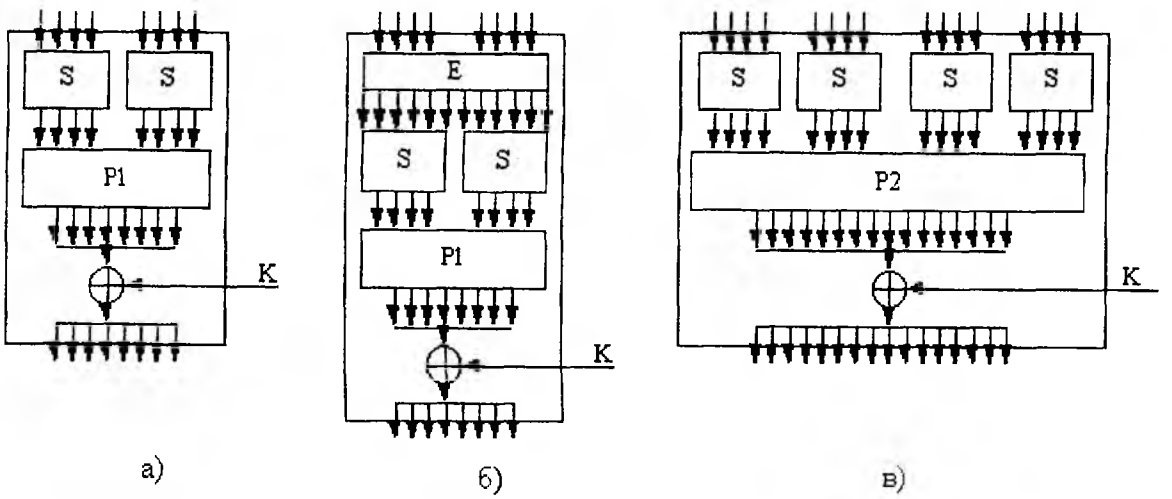



Рис. 2

Следует отметить, что сжимающие подстановки позволяют достигнуть более хороших дифференциальных показателей, чем, видимо, и обусловлено их применение. Но, вместе с тем, они менее удобны в реализации, так как требуют больше памяти для хранения.

Расширяющая перестановка E на рис. 2,б преобразует 8-битное значение в 12-битное путем дублирования определенных битов подобно аналогичной перестановке алгоритма DES [1].

Битовые перестановки обычно строятся, исходя из принципа, чтобы входное значение каждого S-блока зависело от максимального числа S-блоков предыдущего цикла. В соответствии с этим принципом перестановки P1 и P2 выходные биты каждого S-блока поровну распределяют между всеми S-блоками следующего цикла.

Теперь напомним основные принципы проведения дифференциального криптоанализа. Дифференциальная атака на SPN-шифр проводится аналогично дифференциальному криптоанализу фестелеобразного шифра, предложенного и детально описанного в работах [2, 3]. Целью дифференциального криптоанализа является определение битов подключа,

[1] Под сжимающе-расширяющими преобразованиями здесь и далее понимаются преобразования, подобные шифрующей функции алгоритма DES, когда сначала расширяющая E-перестановка в полтора раза увеличивает длину преобразуемого блока, а затем сжимающие подстановки (6 в 4 бита каждая) приводят блок к исходному размеру.

используемого в последнем цикле. При выполнении дифференциального криптоанализа DES-подобного шифра для определения битов подключа используются известные входные в последний цикл значения (непосредственные значения правых половин шифртекстов) вместе с вероятностным значением разности на выходе шифрующей функции последнего цикла. Аналогично, дифференциальный анализ SPN-шифра использует известные выходные значения (шифртексты) и вероятностное значение входной разности. Таким образом, основой построения и реализации дифференциальных атак является использование дифференциальных характеристик (ДХ), описывающих прохождение через циклы шифрования специфических пар открытых текстов. Если удается найти ДХ с высокой вероятностью, то можно ставить и решать задачи криптоанализа со сложностью меньшей, чем прямой перебор ключей (атака грубой силой).

Очевидно, что для 16-битового SPN-шифра, шифрующая функция которого представлена на рис. 2, в, граничное (максимальное) значение вероятности r-цикловой ДХ, при условии, что можно оставаться в рамках одного активного S-блока на каждом цикле, составляет

$$\left(\frac{\max NS_1(\alpha,\beta)}{2^4}\right)^r,\ \alpha,\beta\in\{0,1\}^4\ (\alpha\neq 0), \qquad (1)$$

где max $NS(\alpha, \beta)$ обозначает максимальное значение в таблице дифференциальной разности.

Существенным моментом при построении атак ДК на фестелеобразный шифр является возможность использования при построении ДХ тривиальных циклов. Тривиальные циклы хороши тем, что они не уменьшают общую вероятность ДХ (нулевая разность на входе цикла переходит в нулевую разность на выходе с вероятностью 1). Если обозначить разность на входе отдельно взятого цикла через Δ, то речь идет о характеристиках, которые используют в отдельных циклах переход входной разности Δ = 0 с вероятностью p = 1 в выходную разность F(Δ) = 0. Примеры таких характеристик для классической цепи Фестеля представлены на рис. 3. Более полный набор ДХ для фестелеобразных шифров приведен в работе [4].

| 1 | 3 | 4 | 5 |
|---|---|---|---|
| $0_x \leftarrow \Gamma$<br>$0_x \leftarrow 0_x$<br>2-х цикловая итеративная характеристика | $\Phi \leftarrow \Gamma$<br>$\Gamma \leftarrow \Phi$<br>$0_x \leftarrow 0_x$<br>$\Gamma \leftarrow \Phi$<br>$\Phi \leftarrow \Gamma$<br>$0_x \leftarrow 0_x$<br>6-ти цикловая итеративная характеристика | $\Phi \leftarrow \Gamma$<br>$\Psi \leftarrow \Phi$<br>$\Phi \leftarrow \Gamma \oplus \Psi$<br>$0_x \leftarrow 0_x$<br>$\Phi \leftarrow \Gamma \oplus \Psi$<br>$\Psi \leftarrow \Phi$<br>$\Phi \leftarrow \Gamma$<br>$0_x \leftarrow 0_x$<br>8-ми цикловая итеративная характеристика | $\Phi \leftarrow \Gamma$<br>$\Psi \leftarrow \Phi$<br>$\circledcirc \leftarrow \Gamma \oplus \Psi$<br>$\Gamma \oplus \Psi \leftarrow \Phi \oplus \circledcirc$<br>$0_x \leftarrow 0_x$<br>$\Gamma \oplus \Psi \leftarrow \Phi \oplus \circledcirc$<br>$\circledcirc \leftarrow \Gamma \oplus \Psi$<br>$\Psi \leftarrow \Phi$<br>$\Phi \leftarrow \Gamma$<br>$0_x \leftarrow 0_x$<br>10-ти цикловая итеративная характеристика |

Рис. 3

Важным моментом следует считать также обнуляющие циклы, использующиеся в характеристиках 1, 2 на рис. 3. Обнуление разности возможно, когда используются сжимающие S-блоки. Примером может служить шифр DES, S-блоки которого определяют закон отображения 6-битных значений в 4-битные. В этом случае нескольким входным значениям может соответствовать одно выходное, т.е., имея на входе S-блока ненулевую разность, можно получить нулевую разность на его выходе. Если обнулить таким образом все активные S-блоки то будет получен обнуляющий цикл. В результате чередования тривиального и обнуляющего циклов будет получена 2-цикловая итеративная дифференциальная характеристика (ИДХ) обнуляющего типа (рис. 3,1), которая может быть повторена нужное число раз для покрытия требуемого числа циклов.

Если в фестелеобразном шифре с разбиением на $n$ подблоков при прохождении через шифрующую функцию возможно обнуление разности, то ДХ, в общем случае, может содержать $n-1$ тривиальных циклов, прежде чем встретится обнуляющий цикл. На рис. 4 приведены ИДХ обнуляющего типа, представляющие, по мнению многих ученых, наибольшую опасность для шифров, построенных по схемам Фестеля.


Рис. 4

Для рассматриваемого фестелеобразного шифра с шифрующей функцией, изображенной на рис. 2,б, граничное значение вероятности r-цикловой ДХ, собранной из 2-цикловых характеристик обнуляющего типа, составит

$$\left(\frac{\max NS_2(\alpha,\beta)}{2^6}\right)^{\frac{r}{2}},\ \alpha \in \{0,1\}^6\ (\alpha \neq 0),\ \beta \in \{0,1\}^4. \qquad (2)$$

Дифференциальные характеристики обнуляющего типа опасны еще и потому, что число активных S-блоков в циклах не зависит от рассеивающих качеств линейных преобразований, так как на линейные преобразования всегда приходят нулевые значения разности. Поэтому возможность реализации атак на основе дифференциальных характеристик обнуляющего типа полностью определяется свойствами нелинейных подстановок и, как показано в работах [4,5], опасность со стороны таких ДХ может быть устранена путем предъявления дополнительных требований[2] к этим подстановкам. В случае предъявления дополнительных требований к подстановкам лучшей вероятностью будут обладать ДХ, общий вид которых приведен на рис. 3,3. Граничное значение вероятности r-цикловой ДХ, собранной из 6-цикловых итеративных характеристик, составит

$$\left(\frac{\max NS_3(\alpha,\beta)}{2^6}\right)^{r-\frac{r+2}{3}},\ \alpha \in \{0,1\}^6\ (\alpha \neq 0),\ \beta \in \{0,1\}^4. \qquad (3)$$

Для фестелеобразного шифра, шифрующая функция которого не содержит расширяюще-сжимающих преобразований (рис. 2,а), ДХ обнуляющего типа нереализуемы. В этом случае лучшей вероятностью опять будут обладать ДХ, общий вид которых приведен на рис. 3,3. Граничное значение вероятности r-цикловой ДХ, собранной из 6-цикловых итеративных характеристик, составит

$$\left(\frac{\max NS_4(\alpha,\beta)}{2^4}\right)^{r-\frac{r+2}{3}},\ \alpha,\beta \in \{0,1\}^4\ (\alpha \neq 0). \qquad (4)$$

Из формул (1)-(4) видно, что для получения окончательных значений необходимо учитывать дифференциальные свойства используемых S-блоков. Будем ориентироваться на использование подстановок с наименьшими максимальными значениями в дифференциальных таблицах[3]. Для биективных подстановок (в нашем случае для подстановок 4 в 4 бита) наиболее удачными в этом смысле считаются преобразования с

[2] Под дополнительными требованиями понимаются требования 3, 5, 6 разработчиков и условие 1 в редакции работы [4].

[3] Стойкость к ДК шифров с линейными битовыми перестановками, как показано в работе [6], зависит также от свойств рассеивания нелинейных подстановок (требование разработчиков DES к подстановкам), однако рассмотрение стойкости шифра в случае наложения этого требования существенно усложняется.

предельной нелинейностью (Almost Perfect Nonlinearity - APN), которые рассматриваются в [7,8]. Максимальное значение в таблице дифференциальной разности таких преобразований°– 4. Поэтому принимаем max $NS_1(\alpha,\beta)$ = max $NS_4(\alpha,\beta)$ = 4. В шифрующей функции на рис. 2,б используется небиективная подстановка 6 в 4 бита. Работы, в которых производилась бы оценка дифференциальных свойств таких подстановок, нам не встретились. В результате вычислительного эксперимента удалось построить подстановки, лучшая из которых имеет максимальное значение в дифференциальной таблице, равное 10. Поэтому предположим, что этот показатель может достичь значения 8 (max $NS_2(\alpha,\beta)$ = 8). Дополнительные требования, направленные на защиту от дифференциальных характеристик обнуляющего типа, как показывает эксперимент, не оказывают существенного влияния на минимаксное значение таблицы разностей. В ходе вычислительного эксперимента нам удалось построить подстановку, которая, как и в предыдущем случае, имеет максимальное значение в дифференциальной таблице, равное 10. Опять, предположим, что этот показатель может быть улучшен до значения 8 (max $NS_3(\alpha,\beta)$ = 8).

Зависимости граничных вероятностей ДХ от числа циклов $r$ для рассматриваемых шифров представлены на рис. 5.

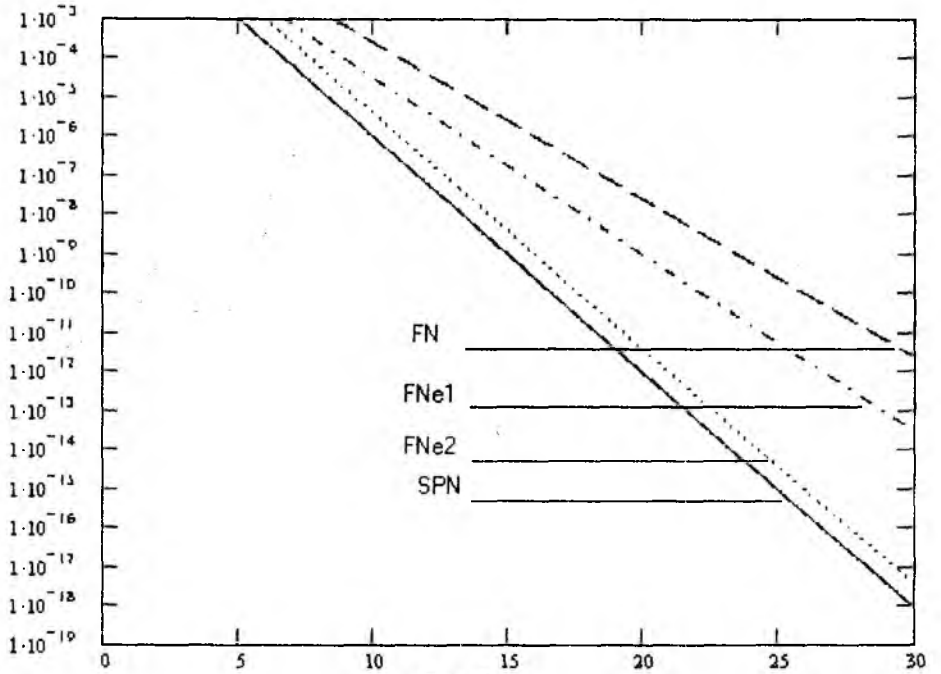



Рис. 5

Из графиков зависимостей видно, что если в качестве операции рассеивания используются битовые перестановки, то при использовании схемы Фестеля за равное число циклов может быть достигнут уровень стойкости, обеспечиваемый SPN-схемой. Для этого необходимо использовать расширяюще-сжимающую схему и накладывать на S-подстановки ограничения на обнуление разности (FNe2). В противном случае (FNe1 – подстановки без ограничений на обнуление разности, FN – биективная структура шифрующей функции) схема Фестеля уступает SPN-схеме по стойкости на одинаковом числе циклов.

В современных шифрах в качестве операций рассеивания вместо битовых перестановок используются линейные преобразования, преимущества которых обсуждаются, например, в работе [9]. Среди линейных преобразований широкое применение получили преобразования на основе МДР-кодов (коды с максимально-допустимым расстоянием). Впервые такие преобразования были использованы при построении блочного шифра Shark. Подобные преобразования также используются в шифрах Square, Rijndael, Khazad, Anubis. Главное преимущество линейных преобразований этого класса заключается в том, что сумма активных S-блоков в контексте дифференциального или линейного криптоанализа до и после

такого линейного преобразования будет максимально возможным, т.е. равным M+1, где M – число S-блоков покрываемых МДР-преобразованием.

Линейное преобразование на основе МДР-кодов обычно задается в виде таблицы (матрицы), а реализуется путем умножения информационного блока, представленного в виде вектора, на эту матрицу, причем умножение отдельных элементов в целях более высокой скорости обычно представляются в виде заранее просчитанных таблиц. Из этого следует, что основной недостаток линейных МДР-преобразований заключается в том, что по мере роста длины преобразуемого подблока растет время выполнения такого преобразования, растет число таблиц умножения, а размер матрицы увеличивается вдвое быстрее (2-кратное увеличение размера преобразуемого подблока влечет 4-кратное увеличение размера матрицы).

Рассмотрим, как в условиях использования в качестве линейного уровня МДР-преобразований будут зависеть верхние границы вероятностей дифференциальных характеристик от числа циклов для фестелеобразного и SPN-шифров с размерами информационных блоков 2M слов (одно слово соответствует одному S-блоку). В этом случае цикл фестель-подобного шифра будет содержать M S-блоков и умножение подблока размером M слов на матрицу размером M×M слов. Для фестель-подобного шифра, по-прежнему наибольшую опасность представляют характеристики обнуляющего типа, поскольку улучшение линейного уровня не снижает их вероятность. Следовательно, этот вид характеристик следует перекрыть в первую очередь. Как уже отмечалось ранее, это может быть выполнено либо путем использования биективных подстановок, либо путем предъявления дополнительных требований к сжимающим небиективным подстановкам. В этих случаях наиболее опасными будут являться дифференциальные характеристики, общий вид которых приведен на рис.3.3. В соответствии со свойствами линейных преобразований на основе МДР-кодов число активных S-блоков в каждых двух соседних нетривиальных циклах будет не менее, чем M+1, тогда граничное значение вероятности r-цикловой ДХ, собранной из 6-цикловых итеративных характеристик, составляет в первом случае:

$$\left(\frac{\max NS_4(\alpha,\beta)}{2^4}\right)^{\frac{r}{3}(M+1)}, \alpha,\beta \in \{0,1\}^4 (\alpha \neq 0), \quad (5)$$

а во втором:

$$\left(\frac{\max NS_3(\alpha,\beta)}{2^6}\right)^{\frac{r}{3}(M+1)}, \alpha,\beta \in \{0,1\}^4 (\alpha \neq 0). \quad (6)$$

Если в SPN-шифре использовать в качестве линейного преобразования умножение на матрицу 2M×2M слов, то такой шифр будет значительно медленнее, чем описанный выше фестель-подобный. Примерно одинаковые показатели быстродействия могут быть достигнуты, если линейное преобразование в SPN-шифре выполнить как параллельное умножение половинок (колонок) размером M слов каждая на матрицу M×M с последующей перестановкой слов между этими половинками (аналогично линейным преобразованиям, применяющимся в шифре Rijndael [10]). Исходя из того, что показатели рассеивания будут лучше, если при перестановке слова каждой колонки распределяются поровну между всеми колонками (в этом случае сложнее оставаться в рамках одной активной колонки) выбираем именно такую перестановку (см. рис. 6).

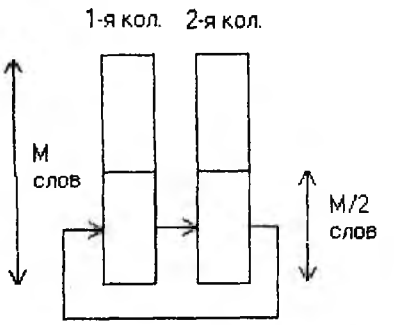



Рис. 6

Таким образом, шифрующая функция SPN-шифра будет содержать 2M S-блоков, перестановку слов, приведенную на рис. 6, умножение обоих колонок на матрицу размером M×M слов и, в заключение, сложение с подключом. Аналогично методике, предложенной в шифре Rijndael [10], покажем, что любая 4-цикловая дифференциальная или линейная характеристика для такого шифра будет содержать, по крайней мере, 3(M+1) активных S-блоков. Для этого введем следующие обозначения: для $i$-го цикла $a_{i-1}$ обозначает значение на входе перестановки слов между колонками, $b_{i-1}$ – значение после перестановки. Активной колонкой будем называть колонку, в которой есть хотя бы один активный S-блок (активное слово).

**Лемма 1.** Для рассматриваемого SPN-шифра число активных S-блоков в 2-цикловой линейной или дифференциальной характеристике равно, по крайней мере, (M+1)·Q, где Q – число активных колонок на входе 2-го цикла.

**Доказательство.** Если в $a_1$ какая-либо колонка активна, то аналогичная колонка активна и в $b_0$. Поскольку каждая колонка $b_0$ проходит через умножение на МДР-матрицу, то сумма активных S-блоков в каждой колонке $b_0$ и соответствующей колонке $a_1$ будет, по крайней мере, M+1. Таким образом, число активных S-блоков в $b_0$ и $a_1$ равно (M+1)·Q. Но, поскольку $b_0$ и $a_0$ содержат равное число активных S-блоков, то лемма доказана.

**Лемма 2.** Для рассматриваемого SPN-шифра сумма активных колонок на входе и выходе любой 2-цикловой характеристики всегда будет не менее 3.

**Доказательство.** Если на входе 2 активные колонки, то с учетом того, что дифференциальная разность всегда содержит хотя бы один активный S-блок, а, значит, и одну активную колонку, т.е., в этом случае лемма справедлива. Рассмотрим случай, когда на входе 1 активная колонка. Здесь возможно 2 варианта:

1. После операции перестановки слов между колонками остается активной одна колонка.
   В этом случае в $a_0$ и $b_0$ должны содержать менее чем M/2 активных слов в одной из колонок, тогда $a_1$ в этой же колонке будет содержать, по крайней мере, M/2+1 активных слов. Отсюда следует, что $b_1$, а значит, и $a_2$ будут содержать две активные колонки.
2. После операции перестановки слов между колонками активны две колонки.
   В этом случае $b_0$ будет содержать не более M/2 активных слов в каждой из двух колонок, а $a_1$, соответственно, – по крайней мере, M/2+1 активных слов в каждой колонке. Но тогда $b_1$ содержит, по крайней мере (M/2+1)+(M/2+1)=M+2 активных слова, следовательно, активны обе колонки.

Лемма доказана.

**Теорема 1.** Для рассматриваемого SPN-шифра каждая дифференциальная или линейная 4-цикловая характеристика содержит, по крайней мере, 3·(M+1) активных слова.

**Доказательство.** Применяя лемму 1 к первым двум циклам (1 и 2) и ко вторым двум (3 и 4) получаем, что минимальное число активных слов в четырех циклах равно сумме активных колонок в $a_1$ и $a_3$, умноженной на (M+1). А из леммы 2 следует, что эта сумма составит, по крайней мере, 3. Теорема доказана.

Учитывая теорему 1, граничное значение вероятности r-цикловой ДХ будет составлять

$$\left(\frac{\max NS_3(\alpha,\beta)}{2^4}\right)^{\frac{3r}{4}(M+1)},\ \alpha,\beta\in\{0,1\}^4\ (\alpha\neq 0). \qquad (7)$$

Для рассматриваемых ранее шифров M = 2. Зависимости граничных вероятностей ДХ от числа циклов $r$ для шифров, использующих МДР-преобразования, представлена на рис. 7.

Из приведенных графиков видно, что, при использовании в шифрах более хороших процедур рассеивания, чем битовые перестановки, схемы Фестеля (FNe и FN) уступают SPN-схеме по стойкости к атакам ДК. Так, схема Фестеля с расширяюще-сжимающими преобразованиями для достижения стойкости, обеспечиваемой SPN схемой, требует примерно в 1,5 раза больше циклов, а схема Фестеля с биективными преобразованиями – примерно в 2,2 раза.

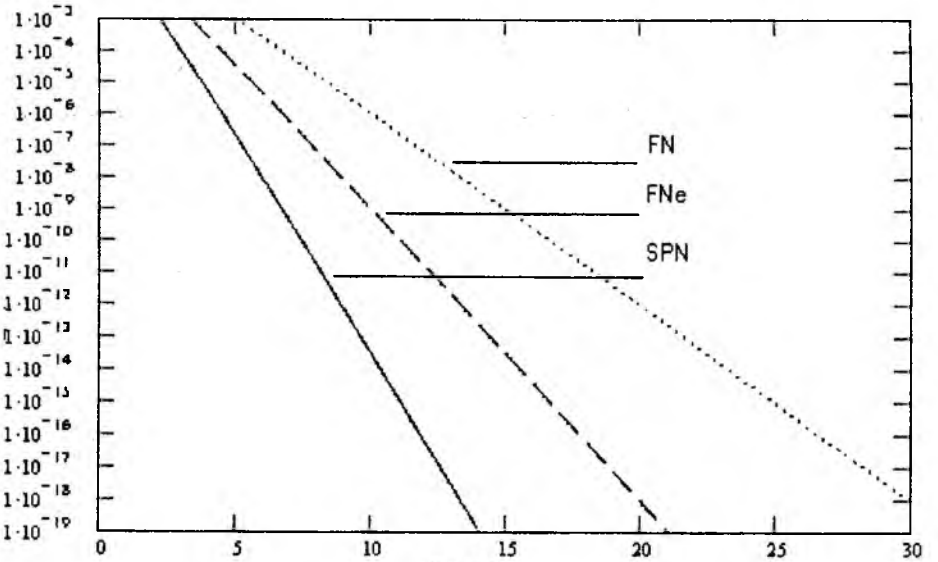


Рис. 7

Таким образом, в ходе исследований были предложены способы выполнения оценки стойкости различных шифров к дифференциальным атакам. Показано, что для фестель-подобных шифров, с точки зрения обеспечения стойкости к дифференциальному криптоанализу, эффективно использование расширяюще-сжимающих преобразований, хотя они менее удобны в реализации. Их применение, при использовании в качестве линейных преобразований битовых перестановок, позволяет фестель-подобному шифру достигнуть уровня стойкости, обеспечиваемого SPN-схемой с тем же числом циклов. Однако, при использовании в качестве операций рассеивания более сложных линейных преобразований, чем битовые перестановки, уровень стойкости, обеспечиваемый SPN-шифром, не достижим для фестель-подобного шифра с тем же числом циклов. Следовательно, вопросы защищенности фестель-подобного шифра от атак ДК в этом случае требуют большего внимания. Обычно эти вопросы решаются либо путем введения дополнительных операций (см. [11,12]), либо путем увеличения числа циклов шифрования, что в любом случае влечет за собой снижение скорости – важнейшей характеристики для симметричных блочных шифров.

**Литература:** 1. National Bureau of Standards, Data Encryption Standard, U.S. Department of Commerce, FIPS pub. 46, January 1977. 2. *E. Biham, A. Shamir.* Differential Cryptanalysis of the DES- like Cryptosystems, Journal of Cryptology. Vol. 4. P. 3-72. 1991. 3. *E. Biham, A. Shamir.* Differential Cryptoanalysis of the full 16-round DES // Technical Report. Computer Science Department. Technion. Israiel. 1993. 4. *Долгов В.И., Лисицкая И.В., Руженцев В.И.* Обеспечение стойкости шифра DES к атакам дифференциального криптоанализа. Перекрытие итеративных характеристик обнуляющего типа и четырехцикловых итеративных характеристик // Радиотехника. 2001. № 120. С. 192-198. 5. *Бондаренко М.Ф., Коряк А.С., Руженцев В.И.* Повышение устойчивости шифра DES к атакам дифференциального криптоанализа // Радиотехника: Всеукр. межвед. науч.-техн.сб. 2001. № 119. С. 172-176. 6. *Долгов В.И. Лисицкая И.В., Головашич С.А.* Принципы защиты алгоритма DES от атак дифференциального криптоанализа // Радиотехника: Всеукр. межвед. науч.-техн.сб. 2000. № 113. С. 148-157. 7. *T. Beth, C. Ding.* On Almost Perfect Nonlinear Permutation // Springer Verlag, Berlin. 1993. 8. *K. Nyberg.* Differentially uniform mappings for cryptography // Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765, T. Helleseth, Ed., Springer-Verlag, 1994, pp. 55-64. 9. *H.M. Heys, S.E. Tavares.* The design of product ciphers resistant to differential and linear cryptanalysis // Journal of cryptology, Vol. 9, no. 1, pp. 1-19, 1996. 10. *J. Daemen, V. Rijmen.* AES Proposal Rijndael, AES Round 1 Technical Evaluation CD-1: Documentation, National Institute of Standards and Technology, Aug 1998. See http://www.nist.gov/aes. 11. Пат. 42531A Украина, H04L9/06. Способ шифрования данных для систем обработки в ЭВМ / *Долгов В.И., Лисицкая И.В. и др.;* №2001032062; Заявл. 28.03.2001; Опубл. 15.10.2001. 12. *Долгов В.И., Руженцев В.И., Федотов М.А.* MDES-128 с таблицами подстановок случайного типа // Радиотехника: Всеукр. межвед. науч.-техн.сб. 2001. № 119. С. 166-171.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 25.04.2002*

УДК 681.3.06:519.248.681

*А.Д. ТЕВЯШЕВ, д-р техн. наук, Ю.И. ГОРБЕНКО*

# ОЦЕНКА ОПАСНОСТИ КРИПТОАНАЛИТИЧЕСКИХ АТАК МЕТОДОМ СОЗДАНИЯ КОЛЛИЗИЙ

## Введение

В системах криптографической защиты информации широкое применение находят криптографические преобразования типа криптографической хэш-функции, однонаправленной хэш-функции, цифровой подписи и др. [1-3]. К таким преобразованиям предъявляются жесткие требования по стойкости к созданию коллизий [1]. Под стойкостью к коллизиям понимают вычислительную невозможность нахождения двух сообщений $M_i$ и $M_j$, для которых

$$H(M_i)=H(M_j), \tag{1}$$

где $H$ есть соответствующее преобразование. В ряде источников [2-3] приводятся оценки вероятности создания коллизий, причем считается, что для этого необходимо выполнить не менее $\sqrt{n}$ экспериментов из общего числа $n$ возможных значений. Анализ состояния вопроса оценки опасности создания коллизии показывает, что актуальными являются по крайней мере две следующие задачи, требующие своего решения.

**Задача 1**. Пусть имеется некоторая функция преобразования $H$ информации $M$

$$h=H(M), \tag{2}$$

где $M$ есть информация произвольной длины $l_M$, причем h может принимать $n=2^m$ значений независимо от длины $l_M$. Необходимо определить число $k$ случайных сообщений $M_i$, которые необходимо подать на вход преобразователя $H$, чтобы с вероятностью $P_3$ состоялось хотя бы одно совпадение вида (1), т.е. состоялась коллизия.

**Задача 2.** Пусть на выходе преобразователя $H$ из полного множества значений $n=2^m$ формируются $k$ случайных значений функции преобразования (2), причем $k \le n$ и $k$ подчиняются равновероятному закону распределения. Пусть выполнено $z$ экспериментов, в каждом из которых получено $k$ значений $h$. Обозначим реализации двух экспериментов соответственно как $X$ и $Y$, причем $X=(x_1,x_2,\ldots,x_k)$ и $Y=(y_1,y_2,\ldots,y_k)$. Необходимо найти вероятность $P(n,k)$ того, что эти множества содержат в себе хотя бы по одному элементу $x_i$ и $y_j$, такие, что $x_i=y_j$.

Целью настоящей статьи и является решение задач 1 и 2 в общем виде и обсуждение полученных результатов.

## 1. Оценка вероятности появления коллизий (задача 1)

Проведенный анализ показал [1], что задача 1 может быть решена с использованием "парадокса" о дне рождения, но при подробном рассмотрении выясняется, что она носит более общий характер. В нашем случае имеется выборка из $k$ значений целочисленной случайной величины с равновероятным законом распределения, причем она может принимать значения от 1 до $n=2^m$, а $k \le n$. При этих условиях необходимо найти вероятность $P(n,k)$ того, что среди значений $H(M)$ выборки по крайней мере две совпадают, т.е.

$$H(M_i)=H(M_j).$$

Для решения задачи 1 найдем вероятность того, что в группе из $k$ событий не состоится коллизия, т.е. соотношение (1) не выполнится ни разу. Обозначим эту вероятность как $R(n,k)$. Ясно, что $P(n,k)$ и $R(n,k)$ составляют полную группу событий, т.е.

$$P(n,k)+R(n,k)=1$$

и

$$P(n,k)=1-R(n,k). \qquad (3)$$

Далее найдем общее число $N$ различных способов, которыми можно получить $k$ значений без повторений. Для первого элемента мы имеем $n$ значений без повторений, для второго $n$-1, третьего – $n$-2 и т.д., для $k$-го ($n$-$k$+1). Поэтому общее число способов, при которых совпадений вида (1) нет, равно

$$N = n\cdot(n-1)(n-2)..(n-k+1) = \frac{n!}{(n-k)!} \qquad (4)$$

Поскольку при каждом из событий с одинаковой вероятностью может происходить каждое из $n$ событий, то общее число событий $N_\Sigma$ можно оценить как

$$N_\Sigma = n^k. \qquad (5)$$

Далее, вероятность отсутствия совпадений можно оценить отношением числа вариантов без совпадений (4) к общему числу вариантов (5), т.е.

$$R(n,k) = \frac{n!}{(n-k)!}/n^k = \frac{n!}{(n-k)!\,n^k}. \qquad (6)$$

Используя соотношение (3), имеем

$$P(n,k) = 1-\frac{n!}{(n-k)!\,n^k}. \qquad (7)$$

Выражение (7) может быть использовано для оценки соответствующей вероятности, однако предпочтительнее получить общее решение уравнения (7), например, для значения $k$.

Для этого представим $P(n,k)$ в виде:

$$\begin{aligned} P(n,k) &= 1-\frac{n(n-1)...(n-k+1)}{n^k} = \\ &= 1-\left[\frac{n-1}{n}\cdot\frac{n-2}{n}\cdot...\cdot\frac{n-k+1}{n}\right] = \\ &= 1-\left[\left(1-\frac{1}{n}\right)\cdot\left(1-\frac{2}{n}\right)\cdot...\cdot\left(1-\frac{k-1}{n}\right)\right] \end{aligned} \qquad (8)$$

Далее воспользуемся тем, что для всех $x \ge 0$ [1]

$$(1-x) \le e^{-x}. \qquad (9)$$

При малых значениях $x$ (например, $x \le 0{,}1$) можно считать, что

$$(1-x) \approx e^{-x}. \qquad (10)$$

С учетом этого преобразуем выражение (8), подставив значение (10). В результате получим

$$P(n,k)=1-\left(e^{-1/n}\cdot e^{-2/n}\cdot\ldots\cdot e^{-\frac{k-1}{n}}\right)=1-e^{-\left(\frac{1}{n}+\frac{2}{n}+\ldots+\frac{k-1}{n}\right)}=1-e^{-k(k-1)\,2n}. \quad (11)$$

Обозначим $P(n,k)=P_з$, т.е. вероятностью, с которой должна быть коллизия. В результате имеем

$$P_з=1-e^{-k(k-1)/2n}$$

или

$$1-P_з=e^{-k(k-1)/2n}. \quad (12)$$

Прологарифмировав (12), имеем

$$\ln(1-P_з)=-k(k-1)/2n. \quad (13)$$

Преобразуя (13), имеем

$$\frac{k(k-1)}{2n}=-\ln(1-P_з).$$

или

$$k(k-1)=-2n\ln(1-P_з).$$

В конечном виде получаем

$$k^2-k+2n\ln(1-P_з)=0. \quad (14)$$

Таким образом, получено уравнение, в котором связаны три величины – число событий $k$, общее число событий и вероятность $P(n,k)$, с которой должна осуществляться коллизия. Задаваясь соответствующими значениями $P_з$ и $n$ можно получить точное решение.

Пусть $P_з$=0,5 , тогда из (14) получаем

$$k^2-k+2n\ln 0.5=k^2-k-2n\ln 2=0. \quad (15)$$

При $n=2^m$ уравнение (15) имеет вид

$$k^2-k-2^{m+1}\ln 2=0. \quad (16)$$

Дадим оценку значения $k$, учитывая, что $k^2>>k$. Из (14) получаем

$$k^2=-2n\ln(1-P_з). \quad (17)$$

При $P_З$=0,5 имеем

$$k^2=-2n\ln(1-0,5)=2n\ln 2$$

и

$$k=\sqrt{2n\ln 2}\approx 1,41\sqrt{n}. \quad (18)$$

При произвольном значении $P_з$ из (17) получим

$$k=\sqrt{2\ln\left(\frac{1}{1-P_з}\right)\cdot n}=1,41\sqrt{\ln\left(\frac{1}{1-P_з}\right)\cdot n}. \quad (19)$$

Соотношение (19) позволяет оценить число преобразований (экспериментов) $H(M)$, которые необходимо выполнить для осуществления коллизии с вероятностью $P_з$.

Соотношение (19) позволяет оценить число преобразований (экспериментов) $H(M)$, которые необходимо выполнить для осуществления коллизии с вероятностью $P_з$.

Сравнивая полученные для $k$ значения, например, (18) и (19) с оценкой, которая приводится в ряде источников [2-3]

$$k = \sqrt{n}, \tag{20}$$

можно оценить степень приближенности и возможность ее применения.

**Пример 1**. Пусть в качестве $H$ используется хэш-функция SHA-1, в которой $n=2^{160}$, и пусть $P_з^{'} = 0{,}5$ и $P_з^{''} = 0{,}99$. Используя выражение (19), имеем

$$k_{0,5} = 1{,}41\sqrt{n} = 1{,}41\sqrt{2^{160}} = 1{,}41 \cdot 2^{80} \approx 1{,}7 \cdot 10^{24};$$

$$k_{0,99} = 1{,}41\sqrt{\ln\frac{1}{1-0{,}99} \cdot 2^{160}} = 2^{80} \approx 3 \cdot 10^{24}.$$

В случае (20) получаем

$$k = \sqrt{n} = \sqrt{2^{160}} = 2^{80} \approx 1{,}2 \cdot 10^{24}.$$

Из примера видна погрешность, которую дает оценка (20), широко используемая для анализа коллизий.

**2. Решение задачи 2 анализа коллизий**

Рассмотрим решение задачи 2, опираясь на результаты задачи 1. Вторая задача возникает при рассмотрении и выборе способов создания коллизий [1]. Например, в нашей постановке эта задача имеет смысл, если рассмотрение сразу двух или большего числа множеств из $k$-реализаций позволяет ускорить процесс создания коллизий или извлечь полезную для криптоаналитика информацию.

В нашем случае событие $x_i=y_j$ может состояться с вероятностью $\frac{1}{n}$. Поэтому вероятность того, что $x_i \neq y_j$

$$Q(x_i \neq y_j) = 1 - \frac{1}{n}. \tag{21}$$

Если $Y$ включает в себя $k$ событий, то вероятность того, что все значения $y_1, y_2, \ldots, y_k$ не совпадут с $x_i$, может быть вычислена как

$$Q(x_i \neq Y) = \left(1 - \frac{1}{n}\right)^k. \tag{22}$$

Вероятность того, что хотя бы одно значение $Y$ совпадет с $x_i$, есть

$$R(x_i \in Y) = 1 - \left(1 - \frac{1}{n}\right)^k. \tag{23}$$

Пусть все элементы $X$ разные. Это справедливо, так как $k<<n$, например, $k=\sqrt{n}$. Тогда вероятность того, что

$$R(x_i \notin Y) = \left(1 - \frac{1}{n}\right)^k$$

и

Далее, вероятность того, что хотя бы одно событие из $X$ и $Y$ совпадет, есть

$$R(x_i = y_j) = 1 - \left(1 - \frac{1}{n}\right)^{k^2}. \quad (25)$$

Обозначим $x = \frac{1}{n} << 1$ и воспользуемся соотношением (10). В результате получим

$$R(n,k) = 1 - \left(1 - \frac{1}{n}\right)^{k^2} = \left(1 - \left(e^{-1/n}\right)^{k^2}\right) = 1 - e^{-\frac{k^2}{n}}. \quad (26)$$

Таким образом, вероятность того, что в двух множествах $X$ и $Y$ хотя бы по одному элементу совпадают,

$$P(n,k) = P_з = 1 - e^{-k^2/n}. \quad (27)$$

Преобразуя (27), получим

$$e^{-k^2/n} = 1 - P_з. \quad (28)$$

Логарифмируя (28), имеем

$$-\frac{k^2}{n} = \ln(1 - P_з)$$

и далее

$$k^2 = -n \ln\left(\frac{1}{1 - P_з}\right)^{-1} = n \ln\left(\frac{1}{1 - P_з}\right).$$

В заключение получим

$$k = \sqrt{n \ln\left(\frac{1}{1 - P_з}\right)}. \quad (29)$$

Рассмотрим частые случаи.
Пусть $P_з$=0,5, тогда

$$k = \sqrt{n \ln \frac{1}{1 - 0{,}5}} = \sqrt{n \ln 2} = 0{,}83\sqrt{n}. \quad (30)$$

При $P_з$=0,99 получим

$$k = \sqrt{n \ln \frac{1}{1 - 0{,}99}} = \sqrt{\ln 10^2 n} \approx 2{,}14\sqrt{n}. \quad (31)$$

**Пример 2**. Пусть в качестве хэш-функции используется хэш-функция SHA-1 и $n$=$2^{160}$.

Тогда

$$k_{0,5} = 0.83\sqrt{2^{160}} = 0{,}83 \cdot 2^{80} = 10^{24};$$

$$k_{0,99} = \sqrt{2^{160}} = 2{,}14 \cdot 2^{80} = 2{,}6 \cdot 10^{24}.$$

Полученные в результате решения второй задачи результаты позволяют сделать следующие выводы.

Если на вход преобразования $H$ (вычислителя значений хэш-функции) подаются $k$ случайных значений $M$, причем они образуют множество $X$, а затем еще раз $k$ случайных $M$, что образует множество $Y$, то для того чтобы значения $x_i$ и $y_j$ множеств $X$ и $Y$ совпали с вероятностью $P_з$, необходимо выполнить

$$k = \sqrt{n \ln\left(\frac{1}{1-P_з}\right)}$$

экспериментов.

Проведем оценки значений $k$ и $P_з$ для используемого в Украине алгоритма симметричного блокового шифрования ГОСТ 28147-89. В четвертом режиме длина криптографической контрольной сумы (имитовставки) равняется 64 битам. Поэтому $m$=64 и $n$=$2^m$=$2^{64}$. При $P_з$=0,99, используя (19), получим

$$k = 1{,}41\sqrt{\ln 10^2 n} = 1{,}41\sqrt{\ln 10^2 \cdot 2^{64}} \approx 1{,}3 \cdot 10^{10}.$$

При $P_з$=0,1, используя (19), получаем

$$k = 1{,}41\sqrt{\ln\frac{1}{0{,}9} n} = 1{,}41\sqrt{\ln 1{,}1 \cdot 2^{64}} \approx 1{,}8 \cdot 10^{9}.$$

При $P_з$=0,5, используя (18), имеем

$$k = 1{,}41\sqrt{n} = 1{,}41\sqrt{2^{64}} \approx 1{,}41 \cdot 2^{32} \approx 6{,}06 \cdot 10^{9}.$$

С учетом требований к криптографическим алгоритмам блокового шифрования, изложенных в [3], можно сделать вывод, что режим 4 выработки имитовставки ГОСТ 28147-89 существенно подвержен возможности создания коллизий.

**Заключение**

Полученное в наиболее общем виде уравнение (15) позволяет точно решить задачу определения количества экспериментов $k$, которые необходимо выполнить для создания коллизии с вероятностью $P_з$ на множестве объема $n$. Достаточно хорошим приближением оценки $k$ есть соотношение (19). Вместе с тем принимаемая в ряде источников без пояснения оценка $\sqrt{n}$ дает грубый результат. Наиболее точное значение можно получить из решения уравнения (15).

Выражение (29) позволяет оценить условия коллизий между двумя независимыми экспериментами в зависимости от размера $n$ множества выходов и вероятности, с которой коллизия должна состоятся. Соотношения (30) и (31) дают приближенные оценки.

Полученные в результате решения задач результаты позволяют получить как зависимость $k$ от $P_з$ и $n$, так и зависимость $P_з$ от $k$ и $n$. Указанное может использоваться при синтезе и анализе алгоритмов различных криптографических преобразований, например, блочных криптографических хэш-функций.

Алгоритм ГОСТ 28147-89 в 4 режиме не обеспечивает защиту от коллизий.

**Список литературы:** 1. *В. Столлингс.* Криптография и защита сетей. Принципы и практика. 2-е изд-е.. К.: Изд. дом "Вильямс". 2001, 669 с. 2. *А.П. Алферов.* Основы криптографии. М.: Гелиос АРВ, 2001. 478 с. 3. *А.А. Молдовян.* Криптография. С.-Петербург. 2001. 218 с.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 17.04.2002*

УДК 681.3.06:519.248.681

*В.Н. ВЕРВЕЙКО, А.И. ПУШКАРЕВ, Т.В. ЦЕПУРИТ*

# ФУНКЦИИ ХЭШИРОВАНИЯ: КЛАССИФИКАЦИЯ, ХАРАКТЕРИСТИКА И СРАВНИТЕЛЬНЫЙ АНАЛИЗ

## Введение

Для решения задач обеспечения целостности наблюдаемости и подлинности информации применяются криптографические контрольные суммы. Методы формирования криптографических контрольных сумм можно разделить на два класса: на базе симметричных криптографических преобразований (коды аутентификации сообщений (КАС)) и использующие несимметричные преобразования (цифровые подписи) с применением секретных ключей. Такие функции могут применяться как непосредственно в качестве криптографической контрольной суммы, так и в других преобразованиях. Например, для формирования цифровой подписи необходима эффективная функция отображения сообщения в образ небольшой фиксированной длины (хэш-значение, хэш-код или просто хэш). Эти функции называют функциями хэширования или хэш-функциями.

## 1. Классификация функций хэширования

Функцией хэширования (в широком смысле) называется функция $h$, удовлетворяющая минимум двум требованиям [1]:

1. *Сжатие* – функция $h$ отображает входное сообщение $x$ произвольной конечной длины в хэш-значение $y = h(x)$ небольшой фиксированной длины, при этом входное сообщение будем называть прообразом.
2. *Простота вычисления* – для заданной функции $h$ и сообщения $x$, $h(x)$ вычисляется не выше чем с полиномиальной сложностью.

Функции хэширования, используемые в криптографии, должны удовлетворять дополнительным требованиям, которые будут рассмотрены далее.

Все существующие функции хэширования можно разделить на два больших класса [1]: бесключевые хэш-функции, зависящие только от сообщения, и хэш-функции с секретным ключом, зависящие как от сообщения, так и от секретного ключа.

Подклассом бесключевых хэш-функций являются *коды обнаружения изменений* (modification detection codes, MDC-коды). В криптографии применяются специфические подклассы MDC-кодов, являющиеся однонаправленными и бесколлизионными хэш-функциями, которые получили широкое распространение в системах цифровой подписи.

*Функции выработки кодов аутентификации сообщений* (КАС) являются подклассом ключевых хэш-функций и обладают дополнительным свойством вычислительной стойкости.

По используемым внутренним преобразованиям функции хэширования можно разделить на:

- функции, использующие битовые логические преобразования. Эти функции применяют к входному сообщению побитовые нелинейные операции “И”, “ИЛИ”, “НЕ”, “ИСКЛЮЧАЮЩЕЕ ИЛИ”, различные сдвиги и, как правило, являются многоцикловыми;
- функции, использующие блочные симметричные шифры. Используются в основном для реализации функций выработки КАС;
- функции, использующие преобразования в группах, полях и кольцах с целочисленным или полиномиальным базисом;
- функции, использующие матричные преобразования.

## 2. Требования к применяемым в криптографии хэш-функциям

Анализ условий применения функций хэширования и практического их использования

позволил сформулировать требования, предъявляемые к применяемым в криптографии бесключевым хэш-функциям. Они состоят в следующем:

3. *Стойкость к вычислению прообраза* – невозможность нахождения неизвестного прообраза для любых предварительно заданных хэш-значений, т.е. для заданной хэш-функции $h$ вычислительно невозможно найти неизвестный прообраз $x$ при предварительно заданном хэш-значении $y = h(x)$ для любого значения $y$. Под термином "вычислительно невозможно" здесь и далее будем понимать, что алгоритм, выполняющий данное преобразование, обладает не менее чем экспоненциальной сложностью.
4. *Стойкость к вычислению второго прообраза* – невозможность нахождения любого другого прообраза, который давал бы такое же хэш-значение, как и заданный, т.е. для заданной хэш-функции $h$ и прообраза $x$ вычислительно невозможно найти другой прообраз $x' \neq x$, для которого выполнялось бы условие $h(x) = h(x')$.
5. *Стойкость к коллизиям* – невозможность нахождения двух прообразов для которых вырабатывалось бы одинаковое значение, т.е. для заданной хэш-функции $h$ вычислительно невозможно найти два прообраза $x$ и $x'$, $x \neq x'$, для которых выполнялось бы условие $h(x) = h(x')$.

Требование стойкости к коллизиям является более жестким, чем требование стойкости к вычислению второго прообраза, так как предполагает произвольный выбор двух прообразов.

*Однонаправленной хэш-функцией* называется функция $h$, удовлетворяющая требованиям сжатия, простоты вычисления, стойкости к вычислению прообраза и стойкости к вычислению второго прообраза.

*Бесколлизионной хэш-функцией* называется функция $h$, удовлетворяющая требованиям сжатия, простоты вычисления, стойкости к вычислению второго прообраза и стойкости к коллизиям.

На практике обычно используются хэш-функции, являющиеся одновременно бесколлизионными и однонаправленными.

Однонаправленные хэш-функции могут применяться для решения других задач, например, выработки ключей и псевдослучайных чисел. Для применения в таких задачах хэш-функция должна удовлетворять следующим требованиям [2]:

6. *Отсутствие корреляции* – входные и выходные биты не должны коррелировать, т.е. изменение любого входного бита приводит к большим непредсказуемым изменениям выходных бит.
7. *Стойкость к близким коллизиям* – для заданной однонаправленной функции $h$ вычислительно невозможно найти два прообраза $x$ и $x'$, для которых хэш-значения $h(x)$ и $h(x')$ отличались бы на малое количество бит.
8. *Стойкость к частичной однонаправленности* – вычислительно невозможно восстановить любую часть входного сообщения так же, как и все сообщение. Более того, по любой известной части входного сообщения вычислительно невозможно восстановить оставшуюся часть (восстановление $t$ неизвестных бит требует не менее чем $2^{t-1}$ операций).
9. *Возможность работы в режиме растягивания* – возможность вычисления хэш-функции при длине входного сообщения меньше чем длина хэш-значения.

Требование, предъявляемое к применяемым в криптографии хэш-функциям с секретным ключом, следующее:

*вычислительная стойкость* – невозможность нахождения хэш-значения для заданного сообщения без известного секретного ключа, т.е. для заданной ключевой хэш-функции $h$ и одной или более корректных пар прообразов и хэш-значений $(x_i, h(x_i, k))$ и неиз-

вестном секретном ключе $k$ вычислительно невозможно найти другую корректную пару $(x, h(x,k))$ для любого $x \neq x_i$.

Требование вычислительной стойкости предполагает выполнение требования стойкости ключа (по одной или более корректных пар прообразов и хэш-значения $(x_i, h(x_i,k))$ вычислительно невозможно восстановить секретный ключ $k$), однако, требование стойкости ключа не предполагает выполнение требования вычислительной стойкости.

Функция хэширования с секретным ключом $h$ является *функцией выработки КАС*, если выполняются требования сжатия, вычислительной простоты (при известном сеансовом ключе) и вычислительной стойкости.

Следует различать функции выработки КАС и однонаправленные хэш-функции с секретным ключом, являющимся частью сообщения, так как они обладают различными свойствами. В функциях выработки КАС секретный ключ применяется к каждому блоку сообщения, а в однонаправленных хэш-функциях ключ используется префиксным (в начале сообщения), постфиксным (в конце сообщения) или комбинированным методом, что снижает стойкость функции.

### 3. Возможные атаки на функции хэширования

Проведенный анализ показал, что на однонаправленные и бесколлизионные хэш-функции возможны следующие атаки:

1) нахождение прообраза $x$ по заданному значению $y = h(x)$. Такая атака особенно опасна для систем аутентификации, использующих хэш-значения паролей и секретных ключей;
2) нахождение прообраза $x'$ по заданному прообразу $x$, для которого выполняется условие $h(x) = h(x')$. Эта атака может быть использована для фальсификации сообщения, подписанного цифровой подписью;
3) нахождение двух прообразов $x$ и $x'$, $x \neq x'$, для которых выполнялось бы условие $h(x) = h(x')$.

На функции выработки КАС существует две атаки:

1) нахождение корректной пары прообраза и КАС $(x, h(x,k))$ по одной или более заданным корректным парам $(x_i, h(x_i,k))$ для любого $x \neq x_i$ при неизвестном секретном ключе $k$;
2) нахождение неизвестного сеансового ключа $k$ по одной или более заданным корректным парам прообразов и кодов аутентификации $(x_i, h(x_i,k))$.

Атаки на функции КАС могут выполняться при следующих условиях:

1) *атака с известным текстом* – атакующему заданы только одна или несколько корректных пар прообразов и кодов аутентификации $(x_i, h(x_i,k))$;
2) *атака с выбираемым текстом* – атакующий имеет возможность получить корректные пары $(x_i, h(x_i,k))$ для выбранных $x_i$ (атака на нахождение ключа);
3) *атака с адаптивным выбором текста* – атакующий может получить корректные пары $(x_i, h(x_i,k))$ для любых $x_i$, выбранных в зависимости от результатов предшествующих запросов (атака с целью нахождения ключа).

Все атаки на хэш-функции можно разделить на две группы: атаки, базирующиеся на уязвимости алгоритма преобразований (аналитические) и атаки, не зависящие от алгоритма.

Атаки, не зависящие от алгоритма: атака "грубой силой", атака методом "дня рождения", полный перебор ключей. К таким атакам уязвимы все алгоритмы, единственная возможность их избежать – увеличить длину хэш-значения, вырабатываемого однонаправленной или бесколлизионной хэш-функцией, и секретного ключа в функции выработки КАС.

Аналитические атаки: атака "встреча посередине", атака с коррекцией блока, атака с фиксированной точкой, атака на базовый алгоритм шифрования, дифференциальный анализ. Эти атаки основываются на недостатках внутренней структуры хэш-функций.

*Атака "грубой силой"* [3] может быть выполнена для нахождения прообраза по заданному хэш-значению или для нахождения прообраза, дающего заданное хэш-значение. Суть атаки заключается в последовательном или случайном переборе входных сообщений и сравнения результата выполнения хэш-функции с заданным. Сложность такой атаки оценивается $2^{l-1}$ операций вычисления хэш-значений, где $l$ - длина хэш-значения в битах.

*Атака методом "дня рождения"* [4] выполняется для нахождения двух различных сообщений с одинаковыми хэш-значениями. Эта атака основана на парадоксе "дня рождения" и заключается в том, что в двух сгенерированных множествах хэш-значений, содержащих $n_1$ и $n_2$ элементов соответственно, вероятность нахождения совпадающих элементов между этими множествами оценивается следующей формулой:

$$P \approx 1 - e^{-\frac{n_1 n_2}{2^l}}.$$

В частности, при $n_1 = n_2 = 2^{\frac{l}{2}}$ сложность атаки оценивается как $2^{\frac{l}{2}+1}$ операций вычисления хэш-значений, а вероятность успеха равна

$$P \approx 1 - \frac{1}{e} \approx 0{,}63.$$

*Атака полного перебора ключей* осуществляется для нахождения неизвестного секретного сеансового ключа функции выработки КАС. Для нахождения ключа атакующий, имеющий не менее одной пары (сообщение, КАС), последовательно перебирает ключи. Так как пространство сообщений неоднозначно отображается в пространство хэш-значений, то может быть обнаружено множество подходящих значений ключей. Чтобы точно найти правильный ключ, необходимо выполнить проверку найденных ключей на большом множестве различных пар (сообщение, КАС). Как показано в [5] максимальное число попыток точного определения ключа составляет

$$m + \frac{2^k - 1}{1 - 2^{-l}},$$

где $k$ - длина секретного ключа в битах, $m$ - количество различных пар (сообщение, КАС). В среднем достаточно $\frac{k}{l}$ различных пар (сообщение, КАС) чтобы точно определить секретный ключ.

*Атака "встреча посередине"* [5] является модификацией атаки методом "дня рождения" и используется для хэш-функций с циклической структурой, если цикловая функция $f()$ инвертируема по отношению к промежуточному значению $X$ или блоку сообщения $M_i$. Эта атака по сложности сопоставима с атакой методом "дня рождения".

*Атака с коррекцией блока* используется в случае, если атакующий обладает сообщением и хочет изменить в нем один или более блоков без изменения хэш-значения. Один цикл MD5 уязвим к этой атаке: атакующий берет блок сообщения $M_i$ (16 слов по 32 бита), оставляет 11 слов, модифицирует одно слово, и вычисляет оставшиеся 4. В результате получается блок $M_i'$, отображающийся в то же самое хэш-значение, что и $M_i$. Полная версия MD5 не уязвима к этой атаке.

*Атака с фиксированной точкой* [5] может применяться при условии, что цикловая функция $f$ имеет одну или несколько фиксированных точек. Фиксированной точкой называется блок сообщения $M_i$, для которого выполняется $f(X_i, M_i) = X_i$, т.е. существует блок сообщения $M_i$, не изменяющий промежуточный результат $X_i$. Таким образом, в сообщение $M$ можно добавлять или удалять блоки $M_i$ без изменения хэш-значения. Защитой от таких атак служит вычисление длины сообщения и добавления ее в конце сообщения.

*Атака на базовый алгоритм шифрования* [6] используется для атаки на хэш-функции, базирующиеся на блочных симметричных шифрах. Так как алгоритмы шифрования разрабатывались как двунаправленные (поддерживают обратное преобразование), то это может увеличить уязвимость в функцию сжатия с их применением.

*Дифференциальный анализ* исследует зависимости между входными и выходными значения цикловой функции или функции сжатия с целью определения статистических аномалий. Дифференциальный анализ применим к различным криптографическим системам, включая функции хэширования [7, 8].

Оценочные значения вычислительной стойкости хэш-функций приведены в табл. 1.

**Таблица 1**

| Тип хэш-функции | Цель атаки | Идеальная стойкость |
|---|---|---|
| однонаправленная хэш-функция | нахождение прообраза | $2^l$ |
| | нахождение 2-го прообраза | $2^l$ |
| бесколлизионная хэш-функция | нахождение любой коллизии | $2^{\frac{l}{2}}$ |
| функция выработки КАС | точное нахождение ключа | $\left\lceil \frac{k}{l} \right\rceil + \frac{2^k - 1}{1 - 2^{-l}}$ |
| | Подделка сообщения | $P_m = \max(2^{-k}, 2^{-l})$ |

В таблице используются следующие обозначения: $l$ - длина хэш-значения или кода аутентификации, $k$ - длина секретного ключа, $P_m$ - вероятность успешной подделки сообщения.

### 4. Анализ современных хэш-функций

Рассмотрим однонаправленные и бесколлизионные функции хэширования и функции выработки КАС, используемые в настоящее время в криптографических системах защиты информации, а также некоторые перспективные: ГОСТ 34311-95, HAVAL, SHA-1, RIPEMD-160, MD4, MD5, ГОСТ 28147-89 режим 4, Rijndael CBC–MAC, Whirlpool, SHA-2, UMAC. Все бесключевые хэш-функции, являются однонаправленными и бесколлизионными, поэтому дальнейшем будем их называть однонаправленными.

В качестве примера рассмотрим новую перспективную функцию хэширования SHA-2, в которой в качестве основных преобразований используются нелинейные битовые преобразования. Новые требования к стойкости криптографических алгоритмов, выдвинутые в ходе разработки проектов AES и NESSIE [9], предполагают три основных уровня стойкости: $2^{128}$, $2^{192}$ и $2^{256}$ (ключи длиной 128, 192 и 256 бит). Для обеспечения таких уровней защищенности минимальная длина хэш-значения должна составлять соответственно 256, 384 и 512 бит. Широко используемый в настоящее время алгоритм хэширования SHA-1 вырабатывает хэш-значение длиной 160 бит и обладает стойкостью к коллизиям $2^{80}$. В связи с тем, что на алгоритм SHA-1 на текущий момент не найдена ни одна аналитическая атака, было решено доработать его для возможности формирования хэш-значений длиной 256, 384 и 512 бит. Соот-

ветственно алгоритм SHA-2 подразделяется на три алгоритма: SHA-256, SHA-384 и SHA-512. Хэш-значение вычисляется по следующей итеративной формуле

$$H_i = H_{i-1} + C(M_i, H_{i-1}), \tag{1}$$

где $H_0$ - фиксированный начальный вектор хэширования, $C$ - цикловая функция сжатия, $H_n$ - хэш-значение сообщения $M$, $+$ - операция сложения 32-х битовых слов по модулю $2^{32}$ для SHA-256 или 64-х битовых слов по модулю $2^{64}$ для SHA-384 и SHA-512.

Функция сжатия алгоритма SHA-256 отображает входное 512-и битовое значение в промежуточное значение длиной 256 бит. Перед сжатием выполняется расширение сообщения до 64-х 32-х разрядных блоков. Алгоритм SHA-2 использует 6 нелинейных функций:

$$\begin{aligned}
&Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z),\\
&Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z),\\
&\sum_0(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x),\\
&\sum_1(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x),\\
&\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x),\\
&\sigma_1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x),
\end{aligned} \tag{2}$$

где $\oplus$ - сложение по модулю 2, $\wedge$ - логическое И, $\neg$ - логическое НЕ, $S^n$ - циклический сдвиг вправо на $n$ разрядов, $R^n$ - сдвиг вправо на $n$ разрядов. Промежуточное значение разбивается на восемь 32-х битовых слов, над которыми 64 раза выполняется преобразование, показанные на рис. 1.

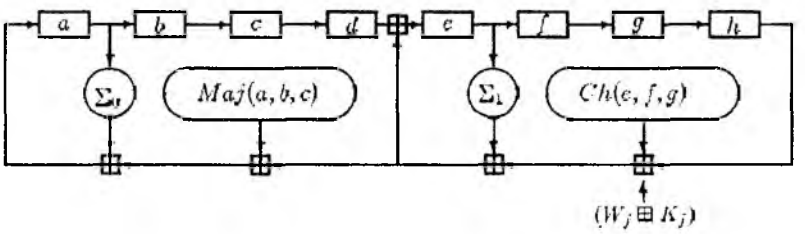



Рис. 1

На рис. 1 приняты следующие обозначения: + - сложение по модулю $2^{32}$, $K_j$ - цикловые константы $j$-го цикла, $W_j$ - 32-х битовый элемент расширенного сообщения $j$-го цикла. Расширенное сообщение $W$ получается из сообщения $M$ с использованием преобразования, изображенного на рис. 2. Начальным заполнением регистра является сообщение $M$.

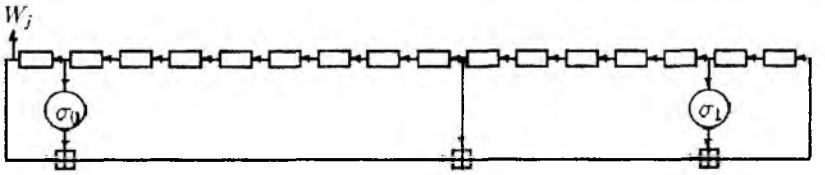



Рис. 2

Алгоритм SHA-512 аналогичен алгоритму SHA-256, но все операции в нем выполняются с 64-х битовыми блоками (сложение по модулю $2^{64}$), количество циклов и элементов расширенного сообщения – 80, на вход поступает 1024-х битовый блок, а на выходе формирует-

ся 512-и битовое промежуточное значение, а также используются следующие логические функции:

$$
\begin{aligned}
&Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z),\\
&Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z),\\
&\sum\nolimits_0(x) = S^{28}(x) \oplus S^{34}(x) \oplus S^{39}(x),\\
&\sum\nolimits_1(x) = S^{14}(x) \oplus S^{18}(x) \oplus S^{41}(x),\\
&\sigma_0(x) = S^{1}(x) \oplus S^{8}(x) \oplus R^{7}(x),\\
&\sigma_1(x) = S^{19}(x) \oplus S^{61}(x) \oplus R^{6}(x).
\end{aligned}
\qquad (3)
$$

Алгоритмы SHA-384 и SHA-512 одинаковы, за исключением того, что используются различные начальные заполнения, и используются только 384 бита полученного 512-и битового значения.

Некоторые характеристики хэш-функций представлена в табл. 2.

**Таблица 2**

| Функция хэширования | Класс функции | Базовые преобразования | Длина хэш-значения, бит |
|---|---|---|---|
| Whirlpool | однонаправленная | в конечных полях и матрицах | 512 |
| SHA-2 | однонаправленная | логические и арифметические | 256, 384, 512 |
| ГОСТ 34311-95 | однонаправленная | блочный симметричный шифр | 256 |
| HAVAL | однонаправленная | логические и арифметические | 128, 160, 192, 256 |
| SHA-1 | однонаправленная | логические и арифметические | 160 |
| RIPEMD-160 | однонаправленная | логические и арифметические | 160 |
| MD5 | однонаправленная | логические и арифметические | 128 |
| MD4 | однонаправленная | логические и арифметические | 128 |
| UMAC | однонаправленная и выработка КАС | в кольцах | 128, 64 |
| Rijndael CBC–MAC | выработка КАС | блочный симметричный шифр | 128 |
| ГОСТ 28147-89 (режим 4) | выработка КАС | блочный симметричный шифр | 64 |

Для оценки сложности реализации хэш-функций была измерена скорость работы официальных реализаций на языке C, а также для сравнения некоторые оптимизированные версии с использованием языка assembler. В качестве компилятора использовался Microsoft Visual C++ 6.0. Тестирование скорости работы проводилось на компьютерах Celeron 600 MHz, 128 MB RAM и Pentium III 1000 MHz, 256 MB RAM под управлением операционной системы Microsoft Windows 2000 Professional. Результаты тестирования скорости работы алгоритмов хэширования представлены в табл. 3.

**Таблица 3**

| Функция хэширования | Количество циклов | Язык реализации | Скорость работы на Celeron 600 MHz | Скорость работы на Pentium III 1000 MHz |
|---|---|---|---|---|
| Whirlpool | 10 | C | 28,013 Мбит/с | 46,961 Мбит/с |
| SHA-2 (512) | 80 | C | 41,159 Мбит/с | 68,701 Мбит/с |
| SHA-2 (256) | 64 | C | 81,308 Мбит/с | 135,557 Мбит/с |
| ГОСТ 34311-95 | - | C+Assembler | 49,408 Мбит/с | 83,056 Мбит/с |
| HAVAL | 96(128, 160) | C | 337,842 Мбит/с | 564,809 Мбит/с |
| SHA-1 | 80 | C<br>Assembler | 206,285 Мбит/с<br>361,581 Мбит/с | 344,433 Мбит/с<br>605,558 Мбит/с |
| RIPEMD-160 | 160 | C | 147,465 Мбит/с | 246,568 Мбит/с |
| MD5 | 64 | C | 278,715 Мбит/с | 574,635 Мбит/с |
| MD4 | 48 | C | 344,086 Мбит/с | 467,793 Мбит/с |
| UMAC | - | C<br>C+Assembler | 989,371 Мбит/с<br>3518,900 Мбит/с | 1648,953 Мбит/с<br>5885,057 Мбит/с |
| Rijndael CBC–MAC | 14 | C | 139,376 Мбит/с | 231,255 Мбит/с |
| ГОСТ 28147-89 (режим 4) | 16 | C+Assembler | 189,559 Мбит/с | 315,270 Мбит/с |

**Заключение**

В качестве основных критериев оценки функций хэширования можно использовать стойкость и вычислительную сложность (скорость) вычисления значения хэш-функции.

Минимальные требования по стойкости, определенные при разработке стандарта AES, соответствуют сложности атаки, равной $2^{128}$. Таким образом, для новых криптографических систем можно рекомендовать только однонаправленные хэш-функции, имеющие стойкость к коллизиям не менее $2^{128}$, т.е. функции с длиной хэш-значения не менее 256 бит. Однонаправленные функции хэширования с длиной хэш-значения, равного 128 битам, следует по возможности исключать из применения, по крайней мере, в перспективных системах. Для функций выработки КАС требования несколько другие. Для будущего применения длина ключа и хэш-значения должна составлять не менее 128 бит, однако, в настоящее время возможно использование функций с длиной ключа не менее 128 бит и длиной кода аутентификации не менее 64-х бит.

Исходя из вышесказанного, алгоритм SHA-2 (512) имеет большой запас стойкости и достаточное быстродействие, т.е. может применяться в системах, в которых наиболее важным требованием является стойкость в течение длительного времени. Алгоритм Whirlpool имеет стойкость, сравнимую с SHA-2 (512), но невысокое быстродействие и может применяться в системах, где стойкость намного важнее скорости.

В качестве рекомендуемого решения для большинства криптографических систем можно рекомендовать функции хэширования с хэш-значением длиной 256 бит. В системах, для которых необходима высокая скорость работы, рекомендуется применять HAVAL, в остальных - SHA-2 (256). В Украине в качестве государственного стандарта принят алгоритм ГОСТ 34311-95, который должен использоваться в системах, подлежащих сертификации. Следует отметить, что ГОСТ 34311-95 является одним из наиболее медленных алгоритмов.

Отдельно следует рассмотреть алгоритм UMAC. Этот алгоритм поддерживает два режима работы: однонаправленная хэш-функция и функция выработки КАС. Хотя для однонаправленной функции он обладает невысокой стойкостью, но зато имеет наивысшее быстродействие, приближающееся к 6 Гбит/с на процессоре Pentium III 1000 MHz. Эта функция может быть использована для систем, где требуется очень высокое быстродействие, например, для систем защиты сетевого трафика в высокоскоростных сетях.

**Список литературы:** 1. *A. Menezes, P. van Oorschot, and S. Vanstone.* Handbook of Applied Cryptography. Chapter 9. CRC Press, 1996. 2. *R. Anderson.* The classification of hash functions. Proc. of the IMA Conference on Cryptography and Coding, Cirencester, December 1993, Oxford University Press, 1995, pp. 83-95. 3. *ANSI X9.30 (PART 2)*, "American National Standard for Financial Services – Public key cryptography using irreversible algorithms for the financial services industry – Part 2: The secure hash algorithm (SHA)", ASC X9 Secretariat – American Bankers Association, 1993. 4. *K. Ohta and K. Koyama.* Meet-in-the-Middle Attack on Digital Signature Schemes. In Abstract of AUSCRYPT '90, pages 110-121, 1990. 5. *B. Preneel.* Analysis and Design of Cryptographic Hash Functions. PhD thesis, Katholieke University Leuven, January 1993. 6. *J.-J. Quisquater and J.-P. Delescaille.* How Easy is Collision Search. New results and applications to DES. In Advances in Cryptology, Proceedings of CRYPTO'89, pages 408-415, 1990. 7. *E. Biham.* On the Applicability of Differential Cryptoanalysis to Hash Functions. In E.I.S.S Workshop on Cryptographic Hash Functions, pages 25-27, March 1992. 8. *E. Biham and A. Shamir.* Differential cryptoanalysis of FEAL and N-Hash. In Advances in Cryptology – Eurocrypt '91, pages 1-16, 1991. 9. *http://www.cryptonessie.org.*

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 22.04.2002*

УДК 681.3.06

*А.А. КУЗНЕЦОВ, канд. техн. наук, А.В. ПОТИЙ, канд. техн. наук*

# ОБНАРУЖИВАЮЩАЯ СПОСОБНОСТЬ ПОМЕХОУСТОЙЧИВЫХ КОДОВ И СХЕМ АУТЕНТИФИКАЦИИ

## Введение

Техническая задача теории помехоустойчивого кодирования состоит в защите цифровых данных от появляющихся в процессе передачи по каналам связи ошибок. С этой целью в передаваемые данные по некоторому правилу (не секретному) вносится избыточная информация. На основе внутренней структуры передаваемых данных на приемной стороне обнаруживаются (возможно, исправляются) возникшие ошибки. Обнаружение, а соответственно и исправление возникших ошибок носит вероятностный характер и зависит от качества канала связи. Техническая задача теории аутентификации состоит в установлении подлинности информации после возможного на нее воздействия злоумышленником. На основе внутренней структуры передаваемой информации в нее вносят избыточность, сформированную по некоторому (секретному) правилу. На приемной стороне проверяют соответствие переданных данных внесенной избыточности. Установление подлинности также носит вероятностный характер и зависит от выбранной стратегии злоумышленника по навязыванию ложной информации.

Очевидно, что, несмотря на различные технические задачи в теории кодирования и теории аутентификации, используется один способ вероятностного достижения цели – внесение в передаваемые данные сформированной избыточности. При заданной вероятности достижения цели вносимая избыточность минимизируется. Задачей данной работы является исследование обнаруживающей способности помехоустойчивых кодов и схем аутентификации блоков данных, изучение их вероятностных характеристик.

## 1. Обнаруживающая и исправляющая способность алгебраических кодов, вероятность необнаружения ошибки и вероятность ошибочного декодирования

Рассмотрим алгебраический блоковый код $(n,k,d)$, где $n$-длина кода (блока), $k$-число информационных кодовых символов, $d$-минимальное кодовое расстояние.

Предположим, что ошибки в последовательно передаваемых кодовых символах происходят независимо с вероятностью $P_o$. Тогда вероятность ошибки кратности $i$ на длине блока $n$ будет

$$P_i = C_n^i P_o^i (1-P_o)^{n-i}.$$

Если код обнаруживает $d-1$ ошибок, то вероятность необнаружения ошибки в блоке

$$P_{но} \geq \sum_{i=d}^{n} P_i = \sum_{i=d}^{n} C_n^i P_o^i (1-P_o)^{n-i}. \qquad (1)$$

Равенство выполняется, если используется эквидистантный код (значения минимального и максимального кодового расстояния совпадают).

Если код правит $t$ ошибок, то вероятность ошибочного декодирования блока

$$P_{од} \geq \sum_{i=t+1}^{n} P_i = \sum_{i=t+1}^{n} C_n^i P_o^i (1-P_o)^{n-i}. \qquad (2)$$

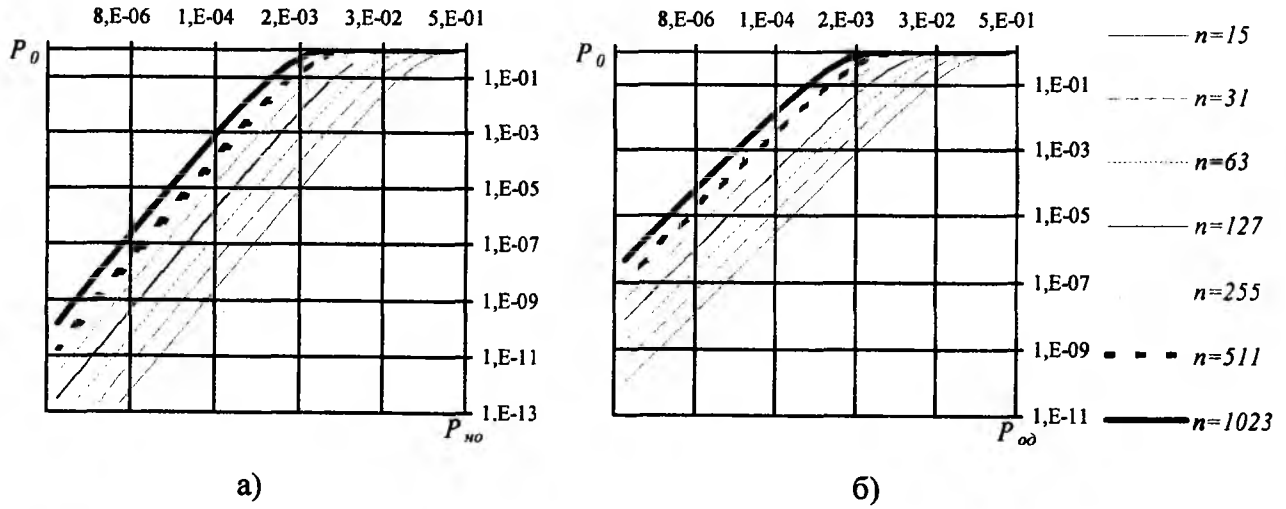
Равенство выполняется, если используется совершенный код. Для совершенного кода сферы одинакового радиуса вокруг кодовых слов, не пересекаясь, покрывают все пространство (радиус сферической упаковки кода совпадает с радиусом покрытия кода). Очевидно, что любой совершенный код является эквидистантным, однако не всякий эквидистантный код совершенен. Важным семейством совершенных кодов, которые легко кодировать и декодировать, являются коды Хемминга, исправляющие одну ошибку [1]. Для двоичного случая коды Хемминга удовлетворяют соотношению

$$\left(n = 2^r - 1, k = 2^r - 1 - r, d = 3\right). \quad (3)$$

При использовании двоичных кодов, лежащих на границе Хемминга, величина $t$ определяется как наименьшее целое, удовлетворяющее неравенству

$$2^{n-k} \geq \sum_{i=0}^{t} C_n^i .$$

На рис.1 представлены зависимости: а) вероятности необнаружения ошибки $P_{но}$; б) вероятности ошибочного декодирования $P_{оо}$ для кодов Хемминга от $P_0$ для различных $n$ с вносимой избыточностью $r = \overline{4,10}$




а) б)

Рис.1

Приведенные зависимости описывают идеальный случай – эквидистантный совершенный код с характеристиками (3), для которого существуют эффективные алгоритмы декодирования. Совершенные коды (если они существуют) обладают замечательными свойствами, однако они достаточно редки, вследствие чего имеют ограниченное практическое значение. Квазисовершенные коды (для которых все слова лежащие вне радиуса сферической упаковки кода находятся на расстоянии $t+1$ хотя бы от одного кодового слова) встречаются чаще, чем совершенные коды. Однако, они так же достаточно редки и не находят широкого применения. В практических схемах, как правило, используются каскадные конструкции кодов, характеристики которых отличны от (3). Выражения (1) и (2) в этом случае используют для оценки потенциальных обнаруживающих и исправляющих способностей алгебраических кодов – граничных вероятностей необнаружения ошибки и ошибочного декодирования.

**2. Обнаруживающая способность кодов аутентификации, вероятность коллизии**

Рассмотрим кодовую последовательность длины $k$. По некоторому правилу (секретному) сформируем и присоединим к ней аутентификатор длины $r$. Для двоичного случая мощность множества всех полученных таким образом блоков составит $2^k$. При передаче блока возможно его искажение (как умышленное, так и в результате воздействия помех). Если аутентификатор для кодовой последовательности в искаженном блоке не соответствует сформированному по фиксированному правилу, то ошибка (или преднамеренное искажение блока) будет обнаружена. Искаженный блок, кроме того, может соответствовать разрешенному блоку, т.е. блоку, аутентификатор которого соответствует сформированному по фиксированному правилу. В этом случае наблюдается коллизия – совпадение аутентификаторов для двух различных кодовых последовательностей. Вероятность такого события соответствует вероятности необнаружения ошибки для помехоустойчивых кодов. Если искажение блока преднамерено, то совпадение аутентификаторов приведет к навязыванию ложной кодовой последовательности. Мощность множества принимаемых блоков $2^{k+r}$. Вероятность коллизий

$$P_{кол} = \frac{2^k}{2^{k+r}} = 2^{-r} \qquad (4)$$

и зависит только от длины аутентификатора.

Это выражение дает нижнюю границу и соответствует идеальному правилу формирования кодов аутентификации. Физический смысл такого правила состоит в равновероятном распределении кодовых последовательностей по аутентификаторам. Другими словами, для всех аутентификаторов число соответствующих им кодовых последовательностей одинаково. В практических схемах формирования кодов аутентификации вероятность коллизий определяют, как правило, статистической проверкой гипотезы о предполагаемом значении вероятности. Исключение, в этом смысле, составляют схемы, эквивалентные строго универсальному классу хеширующих функций [2-3]. Значение вероятности коллизий для таких схем определяют не проверкой гипотезы, а постулированием свойств хеширирующего класса. Ортогональные таблицы [1] соответствуют строго универсальному классу, вероятность коллизий которого определяется (4). Практического распространения такие схемы не получили в виду того, что необходимый объем ключевых данных превышает объем передаваемых данных [4]. Композиционная схема ортогональных таблиц и алгебраических кодов [5] лишена подобных недостатков, она также соответствует строго универсальному классу, вероятность коллизий которого определяется выражением

$$P_{кол} = 2^{-r+1}.$$

Отметим, что в выражение для вероятности коллизий не входят вероятностные характеристики качества канала связи. Это объяснимо тем, что искажение блока данных может быть преднамеренным, а любые искажения вследствие воздействия помех можно интерпретировать как действия противника. Искажение кодовых символов определяется стратегией злоумышленника, которая, строго говоря, может заключаться в случайном искажении (или его замене) блока данных. В этом случае технические задачи аутентификации и помехоустойчивого кодирования совпадают, а физический смысл вероятности коллизий соответствует вероятности необнаружения ошибки.

### 3. Сравнительный анализ вероятностных характеристик помехоустойчивого кодирования и схем формирования аутентификаторов

Основное отличие теории алгебраических кодов от теории аутентификации состоит в распределении множества $2^k$ кодовых слов по множеству $2^n$ блоков. В первом случае распределение кодовых слов должно быть таким, чтобы наиболее вероятные изменения любого кодового слова лежали как можно ближе к нему. Наилучшим кодом в таком случае будет тот, который:

- для фиксированного множества $2^n$ блоков распределит $2^k$ кодовых слов таким образом, что мера различия между любыми двумя кодовыми словами будет одинакова для всего их множества (эквидистантность кода);
- для фиксированного множества $2^{n-k}$ блоков распределит их по каждому кодовому слову из множества $2^k$ так, что наименее отличающиеся от кодового слова блоки расположены ближе к нему и наоборот, блоки с большей мерой отличия расположены дальше (обнаружение наиболее вероятных ошибок).

В теории аутентификации распределение кодовых слов должно быть таким, чтобы минимизировать вероятность перехода из одного кодового слова в другое. Строго говоря, выбор распределения должен исходить из предполагаемой стратегии злоумышленника по навязыванию ложных блоков. Если учесть самую примитивную из стратегий – навязывание путем случайного изменения блока, то очевидно, распределение кодовых слов для целей аутентификации должно удовлетворять первому условию распределения кодовых слов для целей помехоустойчивого кодирования. Выполнение второго условия необязательно. Это означает, что для целей аутентификации сообщений необязательна группировка наиболее вероятных

изменений кодового слова вокруг него. Обязательно равновероятное распределение этих изменений по всему множеству $2^n$ слов. Такое распределение возможно при выполнении равенства числа кодовых последовательностей, соответствующих каждому аутентификатору.

Положим $r \le k$. Тогда если число кодовых слов, соответствующих фиксированному аутентификатору соответствует

$$\frac{2^k}{2^r} = 2^{2k-n}$$

для всего множества аутентификаторов, то имеем идеальное правило формирования аутентификаторов, вероятность коллизий при котором соответствует (4). Интерес представляет сравнение обнаруживающей способности помехоустойчивых алгебраических кодов и кодов аутентификации. На рис. 2 представлены зависимости вероятности необнаружения ошибки $P_{но}$, вероятности ошибочного декодирования $P_{оо}$ эквидистантных совершенных кодов и вероятности коллизий $P_{кол}$ кодов аутентификации от $P_0$ для различных $n$ при внесенной избыточности $r$: а) $r = 10$; б) $r = 20$; в) $r = 30$; г) $r = 40$.

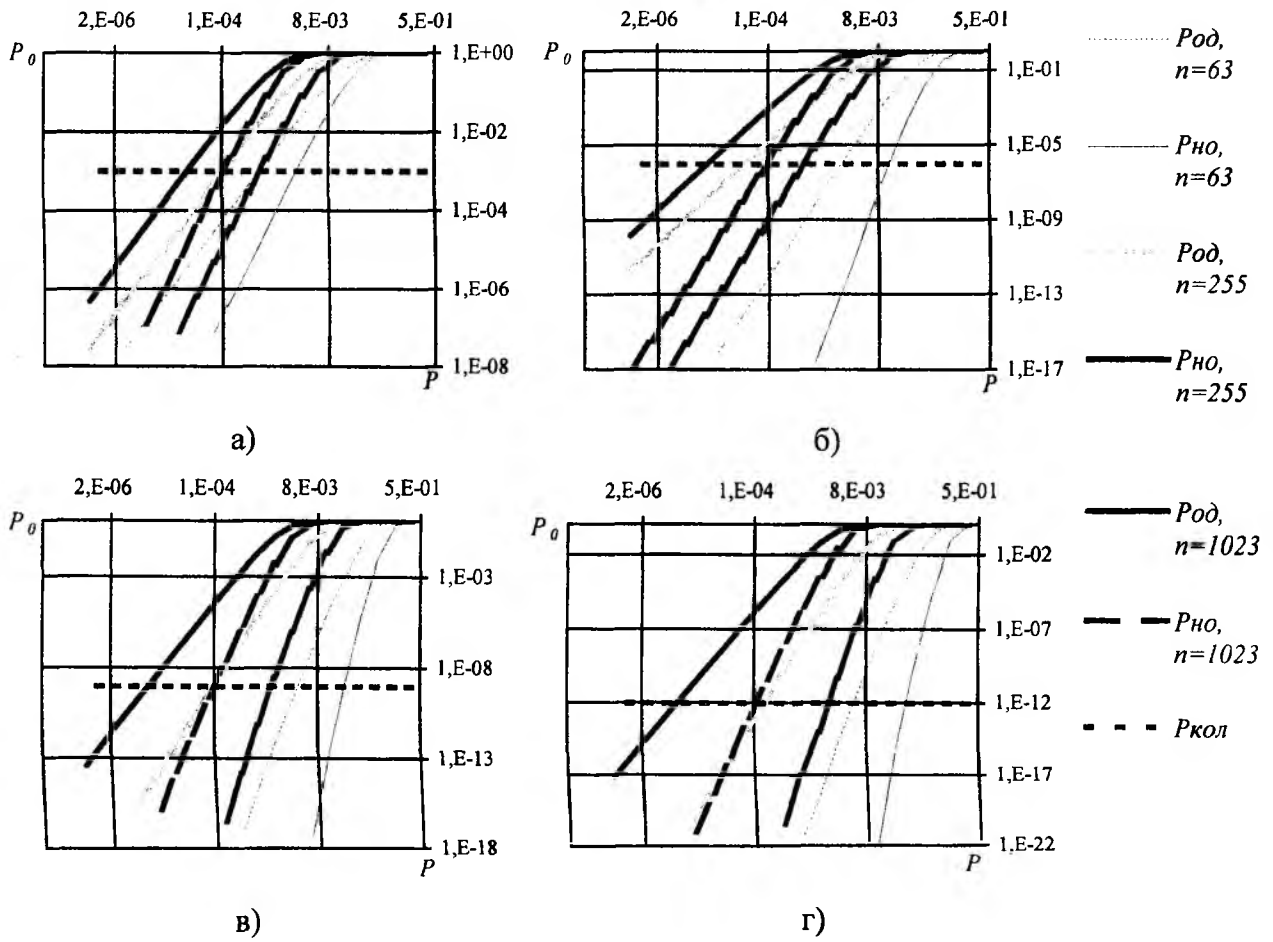



Рис.2

Интерес представляют схемы, решаюшие одновременно две задачи - защиту цифровых данных от ошибок в канале связи и задачу установления подлинности информации. Исправляющую способность такой конструкции предлагается использовать для обнаружения и исправления ошибок в канале связи, а обнаруживающую способность предлагается использовать для проверки подлинности передаваемых блоков. Если такая конструкция соответствует эквидистантному коду, то возможность навязывания ложного блока при исправлении ошибок минимизируется. Это объяснимо тем, что навязывание, соответствующее переходу одного кодового слова в другое, возможно в случае исправления декодером $d$ ошибок, что невозможно по определению. Если код совершенен, то значение вероятности ошибочного декоди-

рования будет лежать на границе (2). Обнаруживающая способность такой конструкции будет соответствовать (1) при обнаружении ошибок в канале связи, а для решения задачи аутентификации блоков – выражению (4). Значение вероятности ошибки декодирования такой конструкции ограничено $P_{од}$ на рис.2., а значение вероятности навязывания ложных блоков данных будет ограничена вероятностью коллизий $P_{кол}$.

**Список литературы:** **1.** *Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки: Пер. с англ. М.: Связь, 1979. 744 с. **2.** *J. L. Carter, M. N. Wegman.* Universal classes of hash functions. //J. Computer and System Sci. 18 (1979), 143-154. **3.** *D. R. Stinson.* Universal Hashing and Authentication Codes. //Designes, Codes and Criptography 4 (1994), 369-380. A preliminary version appeared in the Procecdins of CRYPTO 91, Lecture Notes in Computer Science 576 (1992), 74-85. **4.** *Г.З. Халимов, А.А.Кузнецов.* Аутентификация и универсальное хеширование // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С 88-94. **5.** *Г.З. Халимов, А.А.Кузнецов.* Аутентификация с применением алгеброгеометрических кодов // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С 81-87.

*Харьковский военный университет*

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 19.03.2002*

УДК 681.3.06:006.354

*Я.В. ЧУПРИЙ, Р.В. ОЛЕЙНИКОВ*

# ПЕРЕХВАТ СИСТЕМНЫХ ВЫЗОВОВ В ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS 2000

## Введение

Последние достижения в области информационных технологий непосредственно связаны с появлением персональных компьютеров и сетей передачи данных. Так, с появлением глобальной сети Internet значительно упростилась циркуляция информации, и решился целый круг вопросов информационного обмена.

Развитие Internet-технологий увеличило оперативность передачи информации и обусловило их широкое распространение в повседневной жизни и в важнейших процессах всего мира. Это в конечном итоге способствовало открытию новых направлений науки и техники, сокращению сроков научных разработок и изысканий.

Первоначально поставки первых ЭВМ осуществлялись без единой управляющей программы, и пользователи были вынуждены сами создавать программные средства, охватывающие все аспекты решаемой задачи. Так как программирование того времени осуществлялось на уровне машинных кодов, это приводило к тому, что квалификация пользователя должна была быть очень высокой, что весьма сужало их потенциальный круг и делало ЭВМ доступной только для программиста.

Со временем стали появляться программы, облегчающие труд программиста, затем программы-утилиты общего применения, которые могли использовать пользователи при решении различных задач. Однако эти компьютерные программы были неудобными и громоздкими. Простые пользователи в расчет не брались. Даже после того, как в программном обеспечении компьютеров появилась командная строка, пользователям необходимо было помнить множество команд и опций, которые не были представлены на экране. Создание хорошего пользовательского интерфейса ограничивалось нехваткой оперативной памяти и низкой производительностью процессора. В дальнейшем такие программы систематизировались и проектировались так, чтобы выполнять основные функции взаимодействия пользователя с аппаратными средствами ЭВМ. Это привело к появлению операционных систем (ОС). А появление новых технологий в области микроэлектроники (увеличение емкости оперативной памяти и снижение ее себестоимости) создало предпосылку для улучшения пользовательского интерфейса.

Программа, которая является дружелюбной к пользователю, требует больших затрат от программиста. Основой концепцией Microsoft является разработка программного обеспечения с максимально удобным интерфейсом. Именно благодаря этой концепции в мире около 90% персональных компьютеров работают под управлением ОС семейства Windows. ОС семейства Windows – это унифицированный интерфейс, позволяющий пользователю с минимальными знаниями среды выполнять операции различной сложности. Следовательно, если программа пишется для компьютеров, совместимых с IBM PC, то в первую очередь она пишется для ОС семейства Windows.

Программирование в Windows основано на использовании интерфейса программирования приложений (Application Programming Interface, API). Помимо API, существует целый ряд средств, облегчающих работу программиста (например, использование библиотек классов Microsoft Foundation Classes (MFC)), но в конечном итоге любая из этих оболочек использует все тот же API [1]. И хотя Microsoft выпускает операционные системы Windows с разными типами ядер (Win9x/Me, WinNT/2000/XP, WinCE), API у них практически полностью совпадает. Это означает, что код, написанный для одного ядра [2], может быть применен с небольшими изменениями для другого. Но, несмотря на то, что API предоставляет широкий круг возможностей для программиста, в

некоторых случаях его необходимо дополнить или изменить. Наиболее часто для этих целей используется перехват вызовов API. Целью настоящей статьи является анализ защищенности и разработка метода защиты от перехватов API-вызовов в ОС Windows.

**1. Анализ методов перехвата**

Круг применения перехватов этого вида весьма значителен. Это и изучение поведения исследуемого приложения (определение порядка вызываемых функций с возможностью модификации кода каждой из них), и тестирование приложения на наличие программных закладок, и даже попытки несанкционированного вторжения с целью нарушения целостности и корректного функционирования программного продукта; именно из этих соображений рассмотрение данного вопроса имеет большое значение.

Предположим, в системе запущен некий процесс Authentic.exe, который реализует услугу управления доступом. Его задачей является авторизация пользователя, т.е. подтверждение, что пользователь соответствует тому, за кого себя выдает. Эта задача выполняется с использованием парольной защиты. Каждый вновь входящий пользователь идентифицирует себя секретным паролем, известным только ему. После ввода пароля процесс Authentic.exe выполняет процедуру хеширования и сравнивает полученный хеш-код с соответствующим значением учетной записи. Все учетные записи хранятся в файле Password.prv, для которого средствами операционной системы установлено только монопольное использование (защита от несанкционированного считывания информации). Обычно такие действия реализуются с помощью передачи следующих аргументов функции CreateFile() [4]:

```
HANDLE CreateFile(
L“Password.prv”,                    // имя файла
GENERIC_READ | GENERIC_WRITE,       // режим доступа по чтению и по записи
0,                                  // запрещение совместного использования
                                    // (параметр dwShareMode)
NULL,                               // дескриптор защиты по умолчанию:
                                    // разрешение доступа владельца и
                                    // администратора и
                                    // запрещение всем остальным
OPEN_EXISTING,                      // открыть существующий
NULL,   NULL);                      // атрибуты файла по умолчанию
```

Наиболее значимой угрозой в данном случае является утечка информации из файла Password.prv, так как у злоумышленника появляется возможность получить хеш-код привилегированного пользователя и осуществить подбор пароля. Одним из возможных вариантов реализации данной атаки является использование перехвата API-вызова.

Злоумышленник запускает процесс Manager.exe, который внедряет в адресное пространство процесса Authentic.exe динамическую библиотеку CarrierDll.dll. После проецирования на адресное пространство Authentic.exe библиотека CarrierDll.dll загружает библиотеку злоумышленника HookFunction.dll и производит замену вызова функции CreateFile() на функцию HookCreateFile() из библиотеки HookFunction.dll. На рис. 1 представлена функциональная схема взаимодействия процессов Manager.exe и Authentic.exe. После этого библиотеку CarrierDll.dll необходимо выгрузить, так как необходимость в ней исчерпана. Рассмотрим более детально функционирование данной схемы.

Необходимо отметить, что при разработке приложений, ориентированных на ОС Windows 2000, актуальным является использование Unicode-строк вместо привычных ANSI-строк, которые применяются в Windows 9x. Во-первых, появляется возможность обмена данными на разных языках, во-вторых, это связано с тем, что ОС Windows 2000 полностью построена на Unicode и, соответственно, функции ОС ожидают передачу Unicode-строк в

качестве аргументов (название этих функций заканчиваются литерой ‘W’, например LoadLibraryW()) [2]. Реализация функций для работы с ANSI-строками основана на выделении буфера преобразования входной ANSI-строки в Unicode-строку и последующем вызове функции, предназначенной для Unicode (функции, принимающие ANSI-строки заканчиваются литерой ‘A’, например LoadLibraryA()), что определяет снижение производительности и увеличение необходимого объёма доступной памяти для такого преобразования.

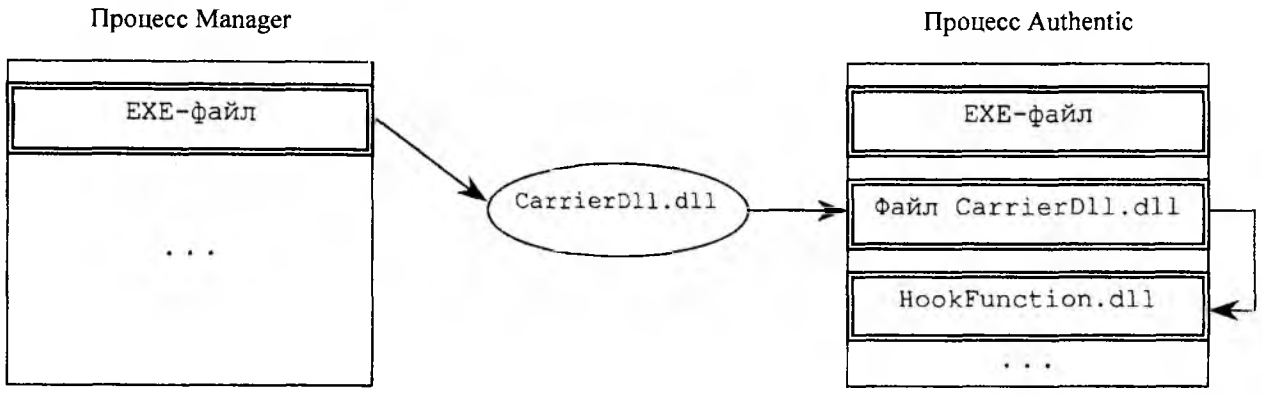



Рис. 1

Следовательно, применение Unicode оправдано с точки зрения производительности и экономии ресурсов; это можно сделать, объявив директивы:

```
#define UNICODE
#define _UNICODE
```

**2. Внедрение CarrierDll.dll в адресное пространство процесса Authentic.exe**

Средства операционной системы Windows2000 позволяют использовать эффективный метод внедрения DLL с помощью удаленных потоков.

Согласно концепции операционных систем семейства Windows NT, к которому относится и Windows 2000, каждый процесс имеет свое виртуальное адресное пространство. Доступ к адресному пространству процесса контролируется диспетчером виртуальной памяти и разрешается только потокам данного процесса, чтобы исключить возможность несанкционированной модификации или считывания адресного пространства одного процесса другим. Это возможно потому, что диспетчер виртуальной памяти во время трансляции виртуальных адресов в физические занимается вопросами разграничения доступа к адресным пространствам процессов. Однако в наборе API Win32 существуют функции ReadProcessMemory() и WriteProcessMemory(), которые позволяют считывать и модифицировать память процесса, они используются отладчиками для получения информации об отлаживаемом процессе и установки точек останова [3].

Принцип внедрения DLL в заданный процесс основан на вызове функции LoadLibrary() потоком этого процесса. Рассмотрим последовательность операций, необходимых для внедрения библиотеки CarrierDll.dll в адресное пространство процесса Authentic.exe:

а) Узнав идентификатор процесса Authentic.exe (ProcessId, с помощью Windows Task Manager), необходимо получить описатель этого процесса (ProcessHandle). Для этого целесообразно использовать функцию OpenProcess() [5] со следующим набором флагов доступа:

```
HANDLE ProcessHandle = OpenProcess(
PROCESS_CREATE_THREAD |      // разрешение на использование описателя процесса
```

```
                                    // для создания потока в этом процессе
                                    // CreateRemoteThread()
    PROCESS_VM_OPERATION |          // разрешение на использование описателя процесса
                                    // для операций с виртуальной памятью процесса
                                    // VirtualAllocEx()
    PROCESS_VM_WRITE,       // разрешение на использование описателя процесса
                            // для модификации виртуальной памяти процесса
    FALSE,                  // запретить наследование прав доступа
                            // порожденным процессам
    ProcessId);             // передаваемый идентификатор процесса
```

б) Выделить виртуальный блок памяти в адресном пространстве Authentic.exe для последующего размещения в этом блоке полного имени внедряемой библиотеки(CarrierDll.dll) [6]:

```
    PWSTR pszVirtualBaseAddress = (PWSTR)VirtualAllocEx(
    ProcessHandle,          // описатель процесса
    NULL,                   // место распределения памяти определяет система
    dwSize,                 // число байт, необходимых для строки, содержащей
                            // полное имя внедряемой библиотеки
    MEM_COMMIT,             // распределение памяти в оперативной памяти или
                            // в файле подкачки
    PAGE_READWRITE);        // обеспечить доступ для чтения и записи
                            // к выделенной области страниц
```

в) Записать строку в адресное пространство процесса Authentic.exe:

```
    WriteProcessMemory(
    ProcessHandle,          // описатель процесса в память которого будут
                            // записываться данные
    pszVirtualBaseAddress,  // указатель на адрес памяти, куда записываются данные
    (PVOID)pszDllName,      // полное имя внедряемой библиотеки CarrierDll.dll
    dwSize,                 // число байтов, необходимых для строки, содержащей
                            // полное имя внедряемой библиотеки
    NULL);                  // число байтов, записанных функцией, игнорируется
```

г) Определить точный адрес функции LoadLibraryW() в модуле Kernel32.dll:

```
    PTHREAD_START_ROUTINE lpStartAddress = (PTHREAD_START_ROUTINE)
    GetProcAddress(GetModuleHandle(
    TEXT("KERNEL32.DLL")),  // описатель модуля библиотеки Kernel32.dll
    "LoadLibraryW");        // имя определяемой функции
```

д) Создать удаленый поток в процессе Authentic.exe:

```
    HANDLE ThreadHandle = CreateRemoteThread(
    ProcessHandle,          // описатель процесса в который производиться внедрение
    NULL,                   // дескриптор защиты по умолчанию
    0,                      // размер стека потока по умолчанию, соответствует
                            // размеру стека основного потока процесса
    lpStartAddress,         // передаем адрес функции LoadLibrary()
    pszVirtualBaseAddress,  // в качестве аргумента функции LoadLibrary()
                            // передаем полное имя CarrierDll.dll
    0,                      // поток начинает немедленно выполняться после создания
    NULL);                  // идентификатор потока не возвращается
```

```
WaitForSingleObject(ThreadHandle, INFINITE);     // ожидаем завершения
                                                 // удаленного потока
```

е) Освободить выделенные ресурсы:

```
                                  // освобождение виртуальной памяти
VirtualFreeEx(hProcess, pszVirtualBaseAddress, 0, MEM_RELEASE);
CloseHandle(ThreadHandle);  // закрытие описателя удаленного потока
CloseHandle(ProcessHandle); // закрытие описателя процесса Authentic.exe
```

В результате выполнения этих операций мы получаем спроецированную библиотеку CarrierDll.dll на адресное пространство процесса Authentic.exe. Библиотека должна иметь функцию входа с использованием следующего механизма:

```
BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
                                  LPVOID lpReserved)
{
  switch(ul_reason_for_call)
  {
  case DLL_PROCESS_ATTACH:
          MyInit();
          break;
  default:        break;
  }
  return TRUE;
}
```

Вызов функции DllMain() с уведомлением DLL_PROCESS_ATTACH происходит после проецирования CarrierDll.dll на адресное пространство Authentic.exe. Функция MyInit() выполняет все задачи, возложенные на библиотеку CarrierDll.dll, а именно: загрузки библиотеки HookFunction.dll и подмены API-функции CreateFile() в таблице адресов импорта исполняемого модуля Authentic.exe на функцию HookCreateFile(), (с разрешением на совместное использование файла по чтению FILE_SHARE_READ, параметр dwShareMode) содержащуюся в модуле HookFunction.dll. Рассмотрим более детально процедуру перехвата.

### 3. Перехват функции CreateFile()

Перехват функции CreateFile() может быть реализован следующим образом:

а) Получить описатель модуля в котором нужно произвести перехват функции. Так как библиотека CarrierDll.dll находится в адресном пространстве Authentic.exe, целесообразно использовать функцию GetModuleHandle(NULL):

```
HMODULE hTargetModule = GetModuleHandle(NULL);
```

б) Определяем адрес функции, которую необходимо подменить:

```
PCSTR pszModule = "Kernel32.dll";
lpOriginFunc = GetProcAddress(GetModuleHandle(pszModule), "CreateFileW");
```

в) Находим раздел импорта в исполнительном модуле Authentic.exe:

```
PIMAGE_IMPORT_DESCRIPTOR pImportDescriptor =
(PIMAGE_IMPORT_DESCRIPTOR) ImageDirectoryEntryToData(
hTargetModule,                          //описатель исполнительного модуля
TRUE,                                   // файл спроецирован системой
IMAGE_DIRECTORY_ENTRY_IMPORT,  // определяет номер индекса необходимого
```

```
                                        // входа соответствующего раздела импорта
                                        // исполняемого модуля
    &Size);                             // размер переданного параметра
```

г) Организуем цикл для поиска описателя раздела импорта нужной DLL (Kernel32.dll)

```
    while(pImportDescriptor -> Name)
    {
    PSTR pszCurrentModule = (PSTR)((PBYTE)hTargetModule + pImportDescriptor -> Name);
    if(!lstrcmpiA(pszCurrentModule, pszModule)) break;
    pImportDescriptor++;
    }
```

д) Выделяем таблицу адресов импорта нашей библиотеки

```
    PIMAGE_THUNK_DATA pImportAddressTable = (PIMAGE_THUNK_DATA)((PBYTE)
    hTargetModule + pImportDescriptor -> FirstThunk);
```

е) Реализуем цикл поиска функции, требуемой для замены и в случае нахождения производим замену адреса в разделе импорта исполняемого модуля:

```
    while(pImportAddressTable -> u1.Function)
    {
      // определяем указатель на адрес текущей функции
      PROC *ppCurrentFunc = (PROC*)&pImportAddressTable -> u1.Function;

      // является ли текущая функция искомой?
      if(*ppCurrentFunc == lpOriginFunc)
      {
            WriteProcessMemory(
            GetCurrentProcess(),
            ppCurrentFunc,
            &lpReplFunc,              // адрес функции в библиотеке HookFunction.dll
            sizeof(lpReplFunc),
            NULL);
            return;                   // операция завершена успешно
      }
      pImportAddressTable++;          // перемещаемся внутри таблицы импорта
    }
```

Также необходимо повторить данную процедуру перехвата для функции CreateFileA(). Функция HookCreateFile() имеет вид:

```
    HANDLE WINAPI HookCreateFile(
    LPCTSTR        lpFileName
    DWORD          dwDesiredAccess,
    DWORD          dwShareMode,
    LPSECURITY_ATTRIBUTES lpSecurityAttributes,
    DWORD          dwCreationDisposition,
    DWORD          dwFlagsAndAttributes,
    HANDLE         hTemplateFile)
    {
      return CreateFile(
                   lpFileName,
```

```
dwDesiredAccess,
FILE_SHARE_READ,        // разрешение на совместное
                        // использование файла по чтению
lpSecurityAttributes,
dwCreationDisposition,
dwFlagsAndAttributes,
hTemplateFile)
```

Следует отметить, что функция HookCreateFile() производит только лишь модификацию значения dwShareMode с последующим вызовом CreateFile() из модуля Kernel32.dll. Так как мы модифицировали адрес вызова функции CreateFile() в таблице адресов импорта исполняемого модуля, то из модуля HookFunction.dll этот вызов осуществляется без изменения, из библиотеки Kernel32.dll. После того как модуль CarrierDll.dll загрузил HookFunction.dll и перенаправил вызов CreateFile() на HookCreateFile(), он успешно выполнил поставленные задачи и требует незамедлительной выгрузки для освобождения занимаемых ресурсов. Процедура выгрузки заключается в создании удаленного потока в целевом процессе Authentic.exe и передачи в качестве стартовой функции потока адреса функции FreeLibrary() из модуля Kernel32.dll.

**4. Выгрузка библиотеки CarrierDll.dll**

а) Получить снимок состояния процесса:

```
HANDLE SnapshotModuleListHandle = CreateToolhelp32Snapshot(
TH32CS_SNAPMODULE,          // снимок включает в себя список модулей
                            // процесса
ProcessId);                 // идентификатор процесса Authentic.exe
```

б) Определить описатель CarrierDll.dll

```
MODULEENTRY32W EntryListModuleProcess = { sizeof(EntryListModuleProcess) };
BOOL fCurrentModule = Module32FirstW(SnapshotModuleListHandle,
                                     &EntryListModuleProcess);
BOOL CheckFound = NULL;
While(fCurrentModule)
{
CheckFound = (lstrcmpiW(EntryListModuleProcess.szModule, pszDllName) == 0) ||
(lstrcmpiW(EntryListModuleProcess.szExePath, pszDllName) == 0);
if (CheckFound) break;
fCurrentModule = Module32NextW(SnapshotModuleListHandle, &EntryListModuleProcess)
}
```

в) Определить описатель Authentic.exe:

```
HANDLE ProcessHandle = OpenProcess(
PROCESS_CREATE_THREAD | PROCESS_VM_OPERATION, FALSE, ProcessId);
```

г) Получить действительный адрес FreeLibraryW() в Kernel32.dll

```
PTHREAD_START_ROUTINE lpStartAddress =
(PTHREAD_START_ROUTINE)GetProcAddress(GetModuleHandle(TEXT("Kernel32")),
                                      "FreeLibrary");
```

д) Создать удаленный поток в адресном пространстве Authentic.exe и передать в качестве стартовой функции истинный адрес функции FreeLibraryW() в Kernel32.dll:

```
HANDLE ThreadHandle = CreateRemoteThread(ProcessHandle, NULL, 0, lpStartAddress,
EntryListModuleProcess.modBaseAddr, 0, NULL);
```

е) Ожидаем завершения удаленного потока
WaitForSingleObject(ThreadHandle, INFINITE);

ж) Освободить выделенные ресурсы:
CloseHandle(SnapshotModuleListHandle);
CloseHandle(ThreadHandle);
CloseHandle(ProcessHandle);

В результате библиотека CarrierDll.dll, выполнив свою задачу, была выгружена из адресного пространства процесса Authentic.exe, перенаправив вызов функции CreateFile() в исполняемом модуле на функцию HookCreateFile() из модуля HookFunction.dll, которая разрешает совместное использование файла Password.prv по чтению.

**5. Метод защиты от перехватов API-вызовов**

Технология перехвата API-вызовов может быть использована не только для отладочных, диагностических целей, но и при написании программных закладок, нарушении корректного функционирования приложения, поэтому необходимо предусмотреть механизмы защиты от использования перехватов несанкционированным пользователем.

Одним из методов защиты от перехватов API-вызовов является включение в охраняемый программный продукт механизма предварительного перехвата API-вызовов, которые могут быть причастными к несанкционированному внедрению дополнительных библиотек и подмене API-функций в адресном пространстве защищаемого процесса. К таким функциям относятся: LoadLibraryA, LoadLibraryW, LoadLibraryExA, LoadLibraryExW, GetProcAddress и WriteProcessMemory.

Следует отметить, что при грамотном администрировании и применении приложений, созданных с учётом вопросов безопасности, Windows2000 является одной из наиболее защищенных ОС. Однако, несмотря на это, она содержит множество недокументированных функций и возможностей. Хотя ни одна из них не является специально направленной на снижение уровня безопасности, тем не менее, не исключается возможность наличия закладок, что делает нежелательным ее применение в организациях с высоким уровнем защищенности. Это подчеркивает необходимость в создании для Украины своей собственной операционной системы.

**Список литературы**: 1. *Петзолд Ч.* Программирование для Windows 95. Т. 1. СПб.: BHV, 1996. 495 с. 2. *Рихтер Дж.* Windows для профессионалов. СПб.: Питер, 2001. 722 с. 3. *Кастер Х.* Основы Windows NT и NTFS: Пер. с англ. М.: Изд. отдел "Русская редакция" ТОО "Channel Trading Ltd", 1996. 440 с. 4. *Рихтер Дж., Кларк Дж.* Программирование серверных приложений для Windows 2000. СПб.: Питер, 2001. 566 с. 5. Microsoft Platform SDK (Windows 2000) 6. *Саймон Р.* Microsoft Windows 2000 API. Энциклопедия программиста. 2001 1086 с.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 29.04.2002*

УДК 681.3.06

*И.Д. ГОРБЕНКО,д-р техн. наук, Е.Г. КАЧКО, канд. техн. наук, П.В. КОЛЕСНИКОВ*

# ГЕНЕРАЦИЯ ПАРАМЕТРОВ И КЛЮЧЕЙ ДЛЯ ЦИФРОВОЙ ПОДПИСИ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ ДЛЯ КОНЕЧНОГО ПРОСТОГО ПОЛЯ

## Введение

Криптостойкость цифровых подписей, которые используют модульное возведение в степень, основано на разложении произведения на простые сомножители (RSA) или вычислении дискретного логарифма (DSA, ГОСТ 34.310-95). В связи с достигнутыми успехами в области теории чисел и возрастанием мощности вычислительных систем, а также использованием параллельного программирования, для обеспечения требуемой стойкости таких алгоритмов вынуждены увеличивать длину ключа. Так, для RSA, в пакете PGP уже используются ключи длиной 4096 битов. Увеличение длины ключа не только уменьшает производительность системы при выполнении операций для выработки и проверки цифровой подписи, но и увеличивает размер цифровой подписи. Так как цифровая подпись обычно добавляется к самому документу, это уменьшает полезную пропускную способность канала связи. Поэтому сейчас повсеместно осуществляется переход на новые стандарты цифровой подписи на основе эллиптических кривых, которые позволяют без увеличения длины цифровой подписи достигнуть высокой криптографической стойкости.

В работе [1] на основании чернового стандарта X9.62-1998 [2] был предложен вариант цифровой подписи на эллиптических кривых. При построении этого алгоритма использовался механизм перехода от DSA к ECDSA и параметры действующего стандарта цифровой подписи ГОСТ 34.310–95. Алгоритм, аналогичный предложенному в [1], был принят Госстандартом России в 2001 г. (ГОСТ Р34.10-2001) [3]. В ГОСТ Р34.10-2001 не определена процедура формирования параметров цифровой подписи на эллиптических кривых. Данная работа посвящена описанию этой процедуры с учетом длин ключей, используемых в [1, 3]. Описанную процедуру можно будет использовать не только при выработке – проверке цифровой подписи, но и при реализации алгоритма Диффи – Хеллмана на эллиптических кривых.

## Параметры эллиптической кривой

К параметрам эллиптической кривой над простым полем GF(p) относятся следующие параметры[1]:

1. Простое число $p$ – модуль преобразования групп точек эллиптической кривой.
2. Эллиптическая кривая $E$, задаваемая коэффициентами $a, в \in F_p$[2].
3. Целое число $m$, определяющее количество точек эллиптической кривой (порядок группы точек).
4. Простое число $q$, определяющее порядок циклической подгруппы группы точек эллиптической кривой $E$.
5. Точка $P$ эллиптической кривой $E$ с координатами $(x_p, y_p)$, которая используется в качестве базисной для генерации других точек эллиптической кривой.

[1] При определении параметров использовались обозначения, принятые в. стандарте ГОСТ Р34.10-2001

[2] В стандарте ГОСТ Р34.10-2001 разрешается вместо $a, в$ использовать так называемый инвариант J(E), но так как инвариант легко вычисляется через $a, в$ и наоборот, в дальнейшем будем считать, что эллиптическая кривая задается своими коэффициентами уравнения

**Способы формирования параметров**

В стандарте **X9.62-1998** [2], в дальнейшем СТАНДАРТЕ, предлагается 3 способа получения параметров эллиптической кривой.

Первый способ состоит в использовании кривых, заданных в СТАНДАРТЕ (прил. J). Анализ готовых кривых показывает, что только одна (самая последняя) эллиптическая кривая удовлетворяет требуемому простому числу. Эта кривая может использоваться только в качестве тестового варианта, поэтому данный способ не пригоден.

Второй способ состоит в случайном выборе эллиптической кривой и проверке ее параметров. Это продолжается до тех пор, пока не будет найдена требуемая кривая. Экспериментальная проверка показала, что поиск эллиптических кривых с требуемым порядком в этом случае может продолжаться очень долго, процесс чисто вероятностный.

Третий способ заключается в выборе требуемых параметров и построении кривой по этим параметрам. Последний способ сразу дает одну или несколько эллиптических кривых с заданным порядком.

В данной работе рассматривается процедура генерации параметров с использованием третьего способа, приводятся рекомендации по ходу выполнения отдельных этапов процедуры.

**Выбор простого числа**

Простое число $p$ должно удовлетворять неравенству $p>2^{255}$. Верхняя граница этого числа определяется конкретными реализациями. Для получения цифровой подписи длиной 512 бит, как это было для стандарта ГОСТ 34.310-95, число $p$ должно удовлетворять неравенству $p<2^{256}$. В дальнейшем предполагается, что число $2^{255}<p<2^{256}$. Для генерации простого числа можно использовать:

1. Процедуру генерации числа $q$ в стандарте ГОСТ 34.310-95;
2. Генерацию случайного простого числа длиной 256 бит.
3. Генерацию сильного простого числа.

Первый способ не гарантирует 256-битное число. Длина числа может быть 254-256 битов. Наиболее надежным мы считаем третий способ, так как криптографические преобразования, в которых используются сильные простые числа, обладают большей стойкостью.

В контрольном примере к ГОСТ Р 34.1—2001 предлагается простое число $p$[3]:
57896044618658097711785492504343953926
634992332820282019728792003956564821041

**Вычисление параметров кривой**

Определяется минимальное значение порядка эллиптической кривой (число $m_{min}$). В соответствии с рекомендациями СТАНДАРТА, для обеспечения требуемого уровня безопасности по отношению к проблеме дискретного логарифма, $m_{min}$ должно удовлетворять неравенству:

$$m_{min} > 2^{200}. \qquad (1)$$

При заданном значении $p$

$$mmin \le (p+1-2\sqrt{p}) \qquad (2)$$

Проверим, что значение $m_{min}$, вычисляемое по формуле $m_{min} = (p-2\sqrt{p})$ удовлетворяет формулам (1) и (2). Так как $p>2^{255}$ то $m_{min}>2^{254}$, т.е. условие (1) выполняется с запасом. Выполнимость условия (2) очевидна.

---

[3] Все числа в данной статье представляются в 10-ой системе счисления, начиная со старшей цифры

Для контрольного примера к стандарту ГОСТ Р 34.1—2001 получаем значение $m_{min}$, равное

$m_{min}$=57896044618658097711785492504343953926

153760394484272996638724459001314707187.

Известно, что порядок эллиптической кривой $m$ зависит от размера поля, определяемого простым числом $p$, и удовлетворяет неравенству:

$$p+1-2\sqrt{p} \le m \le p+1+2\sqrt{p}. \qquad (3)$$

Из формулы (3) следует, что $4p-(p+1-m)^2 \ge 0$, целое. Обозначим это значение через $Z$. Любое положительное целое число может быть представлено как произведение сомножителей, которые встречаются четное и нечетное число раз. Тогда $Z = D * V^2$, где $D$ - произведение сомножителей, не содержащих квадратов. Значение $D$ называется дискриминантом эллиптической кривой.

С учетом (3) параметры эллиптической кривой связаны соотношениями:

$$\begin{aligned} m &= p+1 \pm W, \\ 4p &= W^2 + D*V^2, \end{aligned} \qquad (4)$$

где $W = \sqrt{p} + 1$.

Задача состоит в решении системы уравнений (4) относительно неизвестных $m$, $V$ и $D$ при заданном значении $p$. Так как в системе (4) два уравнения и три неизвестных, для ее решения выбирается значение дискриминанта $D$, требования к которому определены ниже. Как показывает практика, решения системы могут быть найдены при небольших значениях $D$.

**Требования к выбору дискриминанта *D***

Кроме того, что $D$ - положительное и не содержит квадратов, оно должно обладать следующими свойствами:

1.

$$D = \begin{cases} 2,3,7 \bmod 8, & \text{если } p \equiv 3 \bmod 8, \\ \text{нечетное, если } p \equiv 5 \bmod 8, \\ 3,6,7 \bmod 8, \text{если } p \equiv 7 \bmod 8, \\ 3 \bmod 8, \text{если } K = 1, \\ \neq 7 \bmod 8, \text{если } K = 2,3. \end{cases}, \qquad (5)$$

Здесь переменная $K$ определяет отношение между максимально и минимально возможным порядком кривой и определяется по формуле:

$$K = \frac{W^2}{m_{min}}. \qquad (6)$$

При рассмотренных выше значениях $W$, $m_{min}$ значение $K$ всегда равно 1 и формула (5) для проверки дискриминанта превращается в:

$$D = 3 \bmod 8, \qquad (7)$$

т.е. в качестве дискриминанта можно использовать значения: 3,11,19,27,...

2. Должен существовать корень:

$$\left(\overline{(-D)}\right) mod\ P, \qquad (8)$$

т.е. символ Якоби для *P-D* и *P* должен быть равен 1.

Для контрольного примера, минимальное значение D, удовлетворяющее (7) и (8) равно 915.

Чем меньше число *D*, используемое для построения эллиптической кривой, тем быстрее эта кривая может быть построена. В работе выполнено исследование возможности построения эллиптических кривых для первых 1000 простых чисел длиной 256 бит. Если для данного простого числа не удавалось построить эллиптическую кривую при значении дискриминанта меньше 1000, такое простое число отвергалось. Как показали исследования, при максимальном значении дискриминанта, равном 995, найдено 114 эллиптических кривых, из них 15 эллиптических кривых вырождено, т.к. имеют коэффициент $a = 0$. Проверялись различные выборки простых чисел, характер зависимости количества эллиптических кривых от значения дискриминанта в этом случае не изменяется. Авторы считают, что достаточно использовать дискриминанты до 1000.

Рассмотрим построение кривой по заданным значениям размера поля *p*, порядка кривой *u*, ее дискриминанта *D* и значений констант *V, W*.

Значению дискриминанта D соответствует матрица: $\begin{pmatrix} A & B \\ B & C \end{pmatrix}$, для которой *D* - определитель.

Элементы матрицы *A, B, C* должны обладать свойствам:

*A, B, C* - целые;

$$íîä\ (A,\ 2*B,\ C) = 1. \qquad (9)$$

Если $A=|2*B|$ или $A = C$, то $B \geq 0$.

**Таблица 1**

| Диапазон значений дискриминанта | Значение дискриминанта | | Количество матриц | |
|---|---|---|---|---|
| | Минимум | Максимум | Минимум | Максимум |
| 0..99 | 3 | 59 | 3 | 9 |
| 100..199 | 163 | 131 | 3 | 15 |
| 200..299 | 235 | 299 | 6 | 24 |
| 300..399 | 307 | 371 | 9 | 24 |
| 400..499 | 403 | 419 | 6 | 27 |
| 500..599 | 547 | 563 | 9 | 27 |
| 600..699 | 643 | 659 | 9 | 33 |
| 700..799 | 715 | 731 | 12 | 36 |
| 800..899 | 883 | 899 | 9 | 42 |
| 900..999 | 907 | 971 | 9 | 45 |
| 1000..1099 | 1003 | 1091 | 12 | 51 |
| 1100..1199 | 1123 | 1139 | 15 | 48 |
| 1200..1299 | 1227 | 1259 | 12 | 45 |
| 1300..1399 | 1387 | 1379 | 12 | 48 |
| 1400..1499 | 1411 | 1427 | 12 | 45 |
| 1500..1599 | 1507 | 1571 | 12 | 51 |
| 1600..1699 | 1603 | 1691 | 18 | 54 |
| 1700..1799 | 1723 | 1739 | 15 | 60 |
| 1800..1899 | 1867 | 1811 | 15 | 69 |
| 1900..1999 | 1915 | 1979 | 18 | 69 |

Эти свойства обеспечивают положительность *D* и отсутствие в его разложении сомножителей с четным показателем.

Одному значению определителя соответствует группа матриц, причем, если ему соответствует матрица *(A, B, C)* при $B \neq 0$, то определителю соответствует матрица *(A, -B, C)*. Все матрицы, соответствующие данному значению определителя, образуют группу, которой

может быть поставлен в соответствие полином, степень полинома равна количеству матриц, а коэффициенты определяются по самим матрицам и являются целыми числами по модулю *D*. Зависимость количества матриц от значения дискриминанта D, удовлетворяющего (7), представлено в таблице 1. Количество матриц и сами матрицы определяются по алгоритму Е2.2 СТАНДАРТА.

Как следует из таблицы, количество матриц растет при увеличении значения дискриминанта. В соответствии с таблицей можно подобрать значение дискриминанта из любого из приведенных диапазонов, при котором количество матриц минимально. Почему необходимо стремиться к небольшому числу матриц? Чем меньше количество матриц, тем меньше порядок образующего полинома, тем быстрее его можно разложить на множители для выделения уравнения третьего порядка для эллиптической кривой, т.е. далее полином должен быть разложен на полиномы третьей степени.

Значение дискриминанта D используется для определения порядка базовой точки эллиптической кривой. Для определения порядка m используется алгоритм Е.3.2.с СТАНДАРТА. Для контрольного примера получаем значение порядка m, удовлетворяющего $m > m_{min}$:

m= 57896044618658097711785492504343953927082934583725450622380973592137631069619.

**Построение и разложение полинома**

Для построения полинома по заданным коэффициентам матриц $(Ai, Bi, Ci)$ используется алгоритм Е2.3 СТАНДАРТА.

Для контрольного примера получаем полином порядка 24:

x^24 - 72114*x^23 + 2126408*x^22 - 5895642*x^21 - 5532022*x^20 - 60792018*x^19 - 70125360*x^18 + 97965798*x^17 + 35668303*x^16 - 72288692*x^15 + 129495040*x^14 - 78718468*x^13 + 30977772*x^12 + 72917852*x^11 - 15938320*x^10 - 29367412*x^9 + 51609743*x^8 - 23991322*x^7 + 10115960*x^6 - 3523618*x^5 + 2550698*x^4 - 669962*x^3 + 73088*x^2 - 114*x + 1.

Для получения параметров уравнения эллиптической кривой строятся их начальные значения, которые определяются из разложения полинома, полученного на предыдущем шаге на множители, в их числе множитель с полиномом степени 3. Один и тот же полином может иметь несколько сомножителей требуемого порядка, например, если количество матриц 6, то может быть 2 полинома, при числе матриц 24 может быть до 8 полиномов третьего порядка, т.е по одному полиному можно построить сразу несколько эллиптических кривых, имеющих одинаковые модули и порядки базовой точки. Для разложения полинома на множители используется постепенное выделение сомножителя третьего порядка по алгоритму Е1.4 СТАНДАРТА.

В результате факторизации для контрольного примера получаем такой трехчлен:

x^3 + 28122443055414125650524061560182863399889457803909307045456633293834119814058*x^2 - 1115785347824070460927661595348732319109250450282595459175303778348205830367 3*x – 1.

Для полученного полинома третьей степени строится эллиптическая кривая, которая имеет заданное значение дискриминанта D, но пока не гарантируется ее порядок. Значения коэффициентов $a_0$, $b_0$ такой кривой определяется по алгоритму Е.3.4.1. Для контрольного примера

A0=40418041243272845574322042644730969642132756176440874349302870804884135919179,
B0=34699716189756471699896117326078024952181449703850381273077498683297306383262.

По полученным начальным значениям $a_0$, $b_0$ формируются значения коэффициентов эллиптической кривой $a$, $b$. При формировании коэффициентов используется алгоритм, описанный в Е3.4.2 СТАНДАРТА. На этом же шаге получаем базовую точку эллиптической кривой с заданным значением порядка.

Для контрольного примера коэффициенты эллиптической кривой равны:

$$a = 7,$$

$$b = 43308876546767276905765904595650931995$$

$$942111794451039583252968842033849580414.$$

Порядок циклической подгруппы группы точек эллиптической кривой равен

$$q = 57896044618658097711785492504343953927082934583725450622380973592137631069619.$$

**Заключение**

В работе приведен практический алгоритм формирования параметров эллиптической кривой для схемы электронной цифровой подписи в соответствии с [1, 2]. Для иллюстрации достоверности приведенного алгоритма используется контрольный пример из [2]. В статье приведены промежуточные данные при выполнении каждого шага алгоритма, что существенно упрощает практическую реализацию вычислений параметров.

**Список литературы:** 1. *ГОСТ 34.310 – 95* Стандарт цифровой подписи на эллиптических кривых. 2. X9.62-1998 [2]. Public Key Cryptography For The Financial Services Industry. Public Key Cryptography For The Financial Services Industry 3. *ГОСТ Р34.10-2001.* Информационная технология. Криптографическая защита информации. Процессы формирования и проверки цифровой подписи// Госстандарт России, М.: 2001.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 30.04.2002*

УДК 681.3.06:519.248.681

*Г.Н. ГУЛАК, Е.Г.КАЧКО, канд. техн. наук., С.С. ЛАВРИНЕНКО*

# ЗАЩИЩЕННОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ: ТРЕБОВАНИЯ, ОСНОВНЫЕ АТАКИ, СРАВНИТЕЛЬНЫЙ АНАЛИЗ

## Введение

По данным годового отчета «2001 Computer Crime and Security Survey» [1] Института компьютерной безопасности в Сан-Франциско и ФБР, финансовые потери от компьютерных преступлений в США за минувший год выросли на 43% с 265,6 млн. долл. до 377,8 млн. При этом 85% респондентов из 538, в основном из промышленных и государственных структур, заявили о фактах нарушения компьютерной безопасности, причем не только из-за атак злоумышленников. Почти 64% были озабочены понесенными убытками, но лишь 35% смогли оценить их в денежном выражении. Около 70% респондентов заявили, что чаще всего атакам подвергались Internet-каналы, а 31% показали, что атакам подвергались внутрикорпоративные системы. Случаи вторжения извне подтверждали 40% респондентов (в 2000 г. — 25%), а 38% фиксировали отказ в обслуживании (27% в 2000 г.). На нарушение привилегий из-за злоупотребления сотрудниками работой в Сети жаловались 91% респондентов, а 94% обнаружили в своих системах вирусы (в 2000 г. это отмечали 85%).

Даже из этих скупых цифр видна явно негативная тенденция — Internet не только возводит мосты между странами и континентами, но и приближает преступника к жертве. Если оставить в стороне извечные вопросы разведки и промышленного шпионажа и сосредоточиться только на «бытовой» стороне дела, то одними из ведущих проблем в области информационной безопасности в минувшем году стали атаки на платежные системы, дискредитация компаний (отказ в обслуживании), производственный саботаж, вскрытие корпоративных секретов, нарушение прав интеллектуальной собственности. По оценкам отдела по науке и технологиям при президенте США, ежегодный урон, наносимый американскому бизнесу компьютерными злоумышленниками в последние годы, достигал 100 млрд. долл. Потери от несанкционированного доступа к информации, связанной с деятельностью финансовых институтов США, составляли не менее 1 млрд. долл. в год. Таким образом, американский бизнес вплотную подошел к тому рубежу, когда своевременное и адекватное решение вопросов безопасности для него становится экономически целесообразным.

Еще более критическая ситуация в части уровня защищенности ОС возникла в большинстве стран мира, в том числе в Украине. Дело в том, что поставляемые в Украину ОС могут содержать ненадежное системное ПО, выполнять незадекларированные функции, иметь ограничение в использовании, например использовании механизмов и средств защиты. Выходом из этой ситуации мог бы быть подробный анализ исходных кодов ОС, однако они недоступны потребителю по ряду причин.

Целью настоящей статьи является анализ основных угроз безопасности ОС, классификация и анализ основных атак на ОС, а также определение основных методов защиты от них. При изложении будем ориентироваться на методологию, изложенную в ISO-15408 [2].

## 1. Требования к защищенной ОС

### 1.1. Классификация угроз безопасности операционной системы

Применительно к ОС Windows NT, Windows 2000 и Linux угрозы можно классифицировать по цели реализации, принципам воздействия на ОС, характеру воздействия на ОС, типу используемой слабости, по способам воздействия на объект атаки, по способу действия злоумышленника, по объекту атаки, используемым средствам атаки, по состоянию атакуемого объекта ОС на момент атаки.

**По цели реализации угрозы:**

- несанкционированное чтение информации;

- несанкционированное изменение информации;
- несанкционированное уничтожение информации;
- полное или частичное разрушение операционной системы. Под разрушением операционной системы понимается целый комплекс разрушающих воздействий от кратковременного вывода из строя отдельных программных модулей системы до физического стирания с диска системных файлов;
- несанкционированное использование ресурсов ОС;
- несанкционированная модификация отдельных программных модулей, библиотек и др.

**По принципу воздействия на операционную систему:**

- используя известные (легальные) каналы получения информации. К этому классу относится, например, угроза несанкционированного чтения файла, доступ пользователей к которому, согласно адекватной политике безопасности, должен быть запрещен;
- используя скрытые каналы получения информации. Например, угроза использования злоумышленником недокументированных возможностей операционной системы;
- создавая новые каналы получения информации с помощью программных закладок.

**По характеру воздействия на операционную систему:**

- активное воздействие - несанкционированные действия злоумышленника в системе, приводящее к изменению состояния ОС;
- пассивное воздействие - несанкционированное наблюдение злоумышленника за процессами, происходящими в системе;
- адаптивное состояние – когда злоумышленник может, в зависимости от состояния ОС, воздействовать то активно то пассивно.

**По типу используемой слабости защиты:**

- используя неадекватную политику безопасности, в том числе и ошибки администратора системы;
- используя ошибки и недокументированные возможности программного обеспечения операционной системы, в том числе и так называемые люки - случайно или преднамеренно встроенные в систему "служебные входы", позволяющие обходить систему защиты, используя ранее внедренную программную закладку.

**По способу воздействия на объект атаки:**

- непосредственное воздействие, в том числе дистанционное;
- превышение пользователем своих полномочий;
- работа от имени другого пользователя;
- использование результатов работы другого пользователя (например, несанкционированный перехват информационных пакетов, инициированных другим пользователем).

**По способу действий злоумышленника:**

- в интерактивном режиме (например, вручную);
- в пакетном режиме (с помощью специально написанной программы, которая выполняет негативные воздействия на операционную систему без непосредственного участия пользователя-злоумышленника).

**По объекту атаки:**

- операционная система в целом;
- объекты операционной системы (файлы, устройства и т. д.);
- библиотеки, драйверы, протоколы взаимодействия;
- субъекты операционной системы (пользователи, системные процессы и т. д.);

- информационно-телекоммуникационные каналы.

**По используемым средствам атаки:**

- штатными средствами операционной системы без использования дополнительного программного обеспечения;
- программным обеспечением третьих фирм. К этому классу программного обеспечения относятся как компьютерные вирусы и другие вредоносные программы (exploits), которые можно легко найти в Internet, так и программное обеспечение, изначально разработанное для других целей (отладчики, сетевые мониторы и сканеры и т. д.);
- специально разработанным программным обеспечением;
- средствами, заложенными в систему при ее проектировании и изготовлении.

**По состоянию атакуемого объекта операционной системы на момент атаки:**

- хранение информации;
- обработка информации;
- передача информации;
- решение задач;
- активация критической информации, например, ввод ключей или паролей и др.

### 1.2. Понятие защищенной операционной системы

#### Основные определения

Операционная система является защищенной, если она предусматривает средства защиты от основных классов угроз, описанных выше. Защищенная операционная система обязательно должна содержать средства разграничения доступа пользователей к своим ресурсам, а также средства проверки подлинности пользователя, начинающего работу с операционной системой. Кроме того, защищенная операционная система должна содержать средства противодействия случайному или преднамеренному выводу операционной системы из строя.
Если операционная система предусматривает защиту не от всех основных классов угроз, а только от некоторых, то такая операционная система называется частично защищенной. Например, операционная система MS-DOS с установленным антивирусным пакетом является частично защищенной системой - она защищена от компьютерных вирусов.

Политикой безопасности называется набор норм, правил и практических приемов, регулирующих порядок хранения, обработки и использования ценной информации. В отношении операционной системы политика безопасности определяет то, какие пользователи могут работать с операционной системой, какие пользователи имеют доступ к каким объектам операционной системы, какие события должны регистрироваться в системных журналах и т. д.

Адекватной политикой безопасности называется такая политика безопасности, которая обеспечивает достаточный уровень защищенности операционной системы. Следует особо отметить, что адекватная политика безопасности - это не обязательно та политика безопасности, при которой достигается максимально возможная защищенность системы. По сути это политика, которая минимизирует потери в ОС до допустимых значений.

### 1.3. Комплексная система защиты

Существуют два основных подхода к созданию защищенных операционных систем - фрагментарный и комплексный. При фрагментарном подходе вначале организуется защита от одной угрозы, затем от другой и т. д. Примером фрагментарного подхода может служить ситуация, когда за основу берется незащищенная операционная система (например, Windows 95), на нее устанавливаются антивирусный пакет, система шифрования, система регистрации действий пользователей и т. д.

Основной недостаток фрагментарного подхода очевиден - при применении этого подхода подсистема защиты операционной системы представляет собой набор разрозненных про-

граммных продуктов, как правило, от разных производителей. Эти программные средства работают независимо друг от друга, организовать их тесное взаимодействие практически невозможно. Кроме того, отдельные элементы такой подсистемы защиты могут некорректно работать в присутствии друг друга, что приводит к резкому снижению надежности системы. Поскольку подсистема защиты, созданная на основе фрагментарного подхода, не является неотъемлемой компонентой операционной системы, при отключении отдельных защитных функций в результате несанкционированных действий пользователя-злоумышленника остальные элементы операционной системы продолжают нормально работать, что еще более снижает надежность защиты.

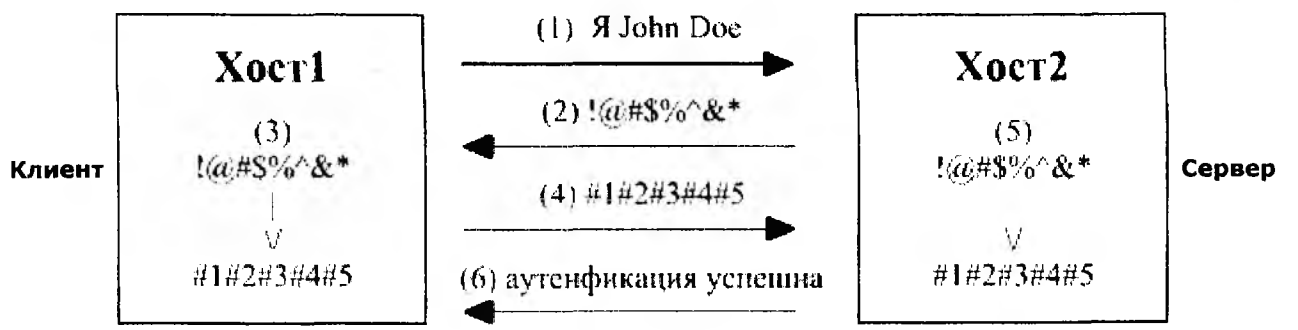
При комплексном подходе защитные функции внедряются в операционную систему на этапе проектирования архитектуры операционной системы и являются ее неотъемлемой частью. Комплексный подход предусматривает применение организационных и физических мер, организационно-технических мер и мероприятий, юридических и законодательных норм и др. Отдельные элементы подсистемы защиты, созданной на основе комплексного подхода, тесно взаимодействуют друг с другом при решении различных задач, связанных с организацией защиты информации. Поскольку вся подсистема защиты разрабатывается и тестируется в совокупности, конфликты между ее отдельными компонентами практически невозможны. Подсистема защиты, созданная на основе комплексного подхода, может быть устроена так, что при фатальных сбоях в функционировании ее ключевых элементов она вызывает крах операционной системы, что не позволяет злоумышленнику отключать защитные функции системы. При использовании фрагментарного подхода такая организация подсистемы защиты невозможна.

Как правило, подсистему защиты операционной системы, созданную на основе комплексного подхода, проектируют так, что отдельные ее элементы являются заменяемыми и соответствующие программные модули могут быть заменены другими модулями, реализующими предусмотренный интерфейс взаимодействия соответствующего программного модуля с другими элементами подсистемы защиты.

## 2. Основные атаки

### 2.1. Локальные атаки

Рассмотрим, как устроена процедура входа в домен или локальный вход в Windows NT и где хранятся пароли. Вход в систему реализован по алгоритму *CHAP* (*Challenge Handshake Autenfication Protocol*). Схема передачи пароля:




Рассмотрим этапы подробнее:

1. Клиент передает серверу запрос об аутенфикации пользователя (John Doe).
2. Сервер генерирует случайную последовательность данных (challenge) и передает клиенту.
3. Клиент, получив данные, с помощью хэш-функции генерирует хэш (от английского "hash" - мешанина), где входными данными являются пароль и полученные данные.
4. Передача полученного хэша серверу.

5. Сервер генерирует на своей стороне хеш, используя те же входные данные (пароль и случайные данные).
6. Сверив два хэша, сообщается результат аутенфикации.

Хэш-функция необратима, т.е. нельзя получить пароль, имея только хэш (не перебирая все варианты). Как видно - при данной схеме избегается передача пароля в незашифрованном виде. В Windows NT, Windows2000 пароли, а вернее хэш-значения паролей, для локального и удаленного входа в систему хранятся в файле *%systemroot%\system32\config\SAM*. Однако просмотреть этот файл, даже имея права администратора, не удастся - система блокирует обращения к этому файлу. В файле *SAM* (Security Account Manager) хранятся хеш-значения паролей для каждого пользователя в структуре, называемой *V-блок*. Он имеет размер 32 байта и содержит в себе хеш пароля для локального входа (NT-hash - 16 байт), а также хэш, используемый при аутенфикации при попытке использовать общие ресурсы других хостов (LanMan-hash - 16 байт).

Алгоритм формирования NT-hash:

1. Введенный пароль перекодируются в юникод.
2. На основе полученной строки генерируется хеш (MD4).
3. Полученный хэш шифруется алгоритмом DES. В качестве ключа используется RID (младшая часть SID - ID пользователя). Этот шаг используется для того, чтобы два пользователя с одинаковыми паролями имели разные хеш-значения.

Алгоритм формирования LanMan-hash:

1. Веденный пароль переводится в верхний регистр.
2. Затем константная строка шифруется алгоритмом DES, используя в качестве ключа 7 первых байтов пароля (пароль может быть максимум 14 символов, если он короче, то добавляется нулями). Другая постоянная строка шифруется байтами 7-14 пароля.
3. Затем с полученной строкой производится манипуляция как и в шаге 3 для NT-hash.

Основными целями атак являются [3]:

- получение прав привилегированного пользователя или администратора;
- нарушение доступности сервера;
- нарушение функциональности сервера.

Анализ показал, что существует несколько методов локального получения прав привилегированного пользователя или администратора. Рассмотрим основные из них:

- подмена пароля - этот метод базируется на подмене данных авторизации пользователя, которые хранятся в файле SAM (а именно в V-блоке). На основе известного пароля генерируются NT-hash и LM-hash и записываются в V-блок пользователя (встроенной учетной записи администратора). После этого можно будет спокойно локально войти в систему, используя логин встроенной учетной записи администратора и уже известный пароль;
- подбор пароля - имея на руках файл SAM, в котором хранится хэш-значение пароля, можно найти пароль, используя полный перебор;
- модификация исполняемого кода - очень эффективный метод, который состоит в том, чтобы обойти исполняемый код проверки пароля. Данная проверка осуществляется в библиотеке MSV1_0.DLL;
- подмена системных файлов - суть этого метода состоит в замене системного сервиса. Суть этого метода состоит в том, чтобы найти сервис (service), запускающий от имени system и расположенный в каталоге, куда обычный пользователь имеет полный доступ, и заменить его на свой . Это может быть брандмауэр или какие-нибудь другие дополнительно установленные сервисы. Недостаток данного метода в том, что далеко не на всех системах можно будет найти нужные сервисы

Резюмируя все предложенные методы локального получения администраторских прав, хочется отметить следующее: самый быстрый способ получения прав администратора для

систем, не использующих утилиту syskey (Windows NT 4.0) - это прямая запись хеша пароля в SAM или подмена MSV1_0.DLL, а для Windows 2000 - это только подмена MSV1_0.DLL. Для этих методов достаточно иметь DOS-загрузочную дискету плюс утилиты для записи файлов в NTFS-раздел, а также пропатченные версии MSV1_0.DLL. Умещается это все всего лишь на одну дискету (с ее помощью можно локально взломать почти любую NT-систему).

Если система, к которой необходимо получить пароли, находится в локальной сети, можно попробовать анализировать трафик, с целью перехвата хэш-значений паролей при авторизации в домен (*NT-hash* или *LanMan-hash*). Это можно осуществить как с помощью просто программ-анализаторов, так и программой *L0phtCrack* (*LC3*) - что намного удобнее. В ней имеется функции прослушивания сети на предмет передачи *LanMan-* и *NT*-хэшей. Если локальная сеть коммутирована свитчами, то хэш-значения паролей будет получить весьма сложно. Что делать в этом случае? Можно поступить следующим образом - послать человеку, работающему на хосте с правами администратора, письмо в HTML-виде. В нем должна присутствовать ссылка на какой-нибудь рисунок (можно даже пустой), находящийся на общих ресурсах атакующей машины. После открытия письма почтовым клиентом будет запрошен файл с общих ресурсов. В этот момент можно будет "поймать" LanMan-хэш (т.к. будет проведена процедура аутенфикации). Для того чтобы "поймать" *LanMan*-хэш (при попытке подключения общего ресурса), можно воспользоваться утилитой *smbrelay*.

Как же защитить свою систему от локального/удаленного взлома?

Можно предложить несколько основных рекомендаций по защите системы на базе NT от атак, связанных с локальным доступом:

- Windows NT должна быть единственной операционной системой на вашем компьютере;
- у пользователя не должно быть возможности загрузиться с дискеты;
- системный блок должен находиться в недоступном для пользователей месте;
- в качестве файловой системы следует использовать только NTFS;
- число пользователей с правом локальной регистрации должно быть максимально ограничено (для этого в программе UserManager в меню Policies выберите пункт UsersRights, в списке Rights укажите log on locally и отредактируйте список разрешенных пользователей);
- при появлении новых версий сервисных пакетов и "заплат" на сервере http://www.microsoft.com их по возможности следует тут же устанавливать.
- Пароль должен быть не менее 8 символов (лучше 10-14), состоять из символов в верхнем и нижнем регистре, а также содержать в себе цифры и, желательно, неалфавитные символы. В этом случае можно будет точно быть уверенным, что взломать пароль перебором без применения распределенного вычисления нельзя.
- Используйте утилиту *syskey*. Она имеет три режима хранения дополнительного ключа, без которого не может быть осуществлен вход в систему: хранение ключа на диске, дискете или ввод его непосредственно пользователем. Нежелательно хранение ключа на дискете - в случае порчи дискеты доступ к системе может быть осуществлен только с помощью ее взлома.
- Также никогда не оставляйте машину без присмотра незаблокированной.

### 2.2. Атака PipeBomb

Системные сервисы в Windows NT не ограничивают максимальное количество создаваемых экземпляров канала, а каждый канал, как правило, обрабатывается отдельным потоком (т. е. происходит классическое, популярное со времен UNIX, расщепление процесса-обработчика при запросе на очередное подключение). Все потоки и каждый экземпляр канала требуют некоторого количества оперативной памяти, и, если злоумышленник вздумает в бесконечном цикле устанавливать все новые и новые соединения, оперативной памяти может попросту не хватить!

При создании канала система размещает входящий и исходящий буферы в неоткачиваемой памяти (non-paged pool). Поэтому максимальное количество экземпляров канала определяется объемом неоткачиваемой памяти, выделенной процессу. Таким образом, существует возможность как заблокировать создание новых экземпляров канала, так и замедлить работу системы, отобрав у системных процессов всю свободную оперативную память и заставляя их за каждой страницей обращаться к диску. Такая атака получила название PipeBomb [4].

### 2.3. Атака AdminTrap

В программном интерфейсе Win32 существует функция ImpersonateNamedPipeClient, выполняющая олицетворение (impersonation) клиента канала. Олицетворение клиента канала заключается в том, что потоку, вызвавшему данную функцию, назначается маркер доступа (access token) клиента экземпляра канала, handle серверного конца которого указан в качестве параметра функции. При этом поток процесса-сервера, обслуживающий данный экземпляр канала, получает полномочия пользователя, который подключился к этому экземпляру канала в качестве клиента.

Если прикладная программа, выполняющаяся с правами обычного пользователя, создаст экземпляр канала, дождется подключения клиента и выполнит олицетворение, эта программа получит полномочия клиента, которые могут превышать изначальные полномочия программы. Таким образом можно получить права привилегированного клиента, в том числе и администратора. Эта идея была положена в основу атаки AdminTrap [5].

Таким образом, используя атаку PipeBomb, можно заблокировать создание системных каналов и, создав свой, дождавшись удаленного входа администратора, захватить его права.

Эта атака удаленного перехвата не может быть устранена даже при помощи правильного администрирования.

Как же защититься от этих атак?

Необходимо написать драйвер, который должен перехватывать запросы к драйверу npfs.sys, который ответственен за создание новых экземпляров каналов. После выполнения запроса драйвер должен с помощью вызова NtSetSecurityObject установить только что созданному экземпляру канала корректные атрибуты защиты.

Однако создание такого драйвера, перехватывающего те или иные системные вызовы, является весьма трудоемкой задачей. Кроме того, драйверы, перехватывающие системные вызовы kernel mode, как правило, жестко привязаны к конкретной версии ядра Windows NT. Установка нового сервис-пака чаще всего делает такой драйвер неработоспособным. А изготовлять для каждого сервис-пака свою версию драйвера явно нецелесообразно.

Таким образом, полная ликвидация вышеописанных слабостей NPFS ведет к тому, что должен быть полностью переписан драйвер npfs.sys, что является весьма непростой задачей, особенно если учесть необходимость обеспечения обратной совместимости со старыми спецификациями NPFS.

### 2.4. Атака LKM (Loadable kernel module)

Драйвер это фактически часть ядра ОС со всеми вытекающими отсюда последствиями. Возможностью динамически загружать драйверы в процессе работы системы обладают многие современные ОС (Solaris,FreeBSD,Linux,Win9x,WinNT,Win2000 и т.д.)

Таким образом, загрузив свой драйвер и перехватив, к примеру, операции файловой системы, мы сможем:

- перехватывать операции создания, открытия, чтения, записи в любой файл (у нас привилегии ядра системы);
- перехватывать операции сканирования каталогов;
- перенаправлять файловые операции (например, перенаправить вывод в log-файл).

Такая атака получила название LKM-атаки.

Основным отличием данного перехватчика от остальных является то, что он не существует для системы. То есть при загрузке драйвера он копирует себя в новое место, а системе говорит, что его загрузка завершилась неудачей.

### 2.5. Подмена MAC

Как может выглядеть атака с подменой MAC- и IP-адреса? Рассматривается случай с провайдером.

Первый этап выполняется, когда "жертва" в сети. С помощью программы-анализатора проверяется, чтобы "жертва" находилась в том же сегменте сети. Сегмент сети в данном случае область, внутри которой все машины равнозначны, т.е. получают одну и ту же информацию. Например, все машины, соединенные хабом (хабами), находятся в одном сегменте, так как хаб просто рассылает получаемую от одной машины информацию по всем остальным.

Критерием того, что ты находишься в одном сегменте сети с X, является то, что ты видишь пакеты, отправленные X, но не предназначенные тебе. Информация по сети передается в виде пакетов - кусков информации определенного размера.

Пакеты бывают двух типов: адресованные кому-то конкретно либо "всем". Последние выделяются тем, что у них MAC-адрес получателя равен FF-FF-FF-FF-FF-FF. Если вы получили пакет, в котором MAC-адрес получателя не равен вашему MAC-адресу, то можете быть уверены, что его отправитель находится в том же сегменте сети. Адрес можно узнать при помощи команды arp -a - она выводит список MAC-адресов, хранящихся в кэше компьютера. Если адреса в кэше не оказалось, стоит попробовать сделать ping на этот адрес и одновременно arp -a, если же его там так и не оказалось - скорее всего, жертва не в ваше сегменте сети.

Будьте внимательны, в том же сегменте сети находится машина, имеющая полученный MAC-адрес, а не IP-адрес. IP-адрес может принадлежать системе, находящейся на другом континенте, дело в том, что при передаче через маршрутизаторы MAC-адрес отправителя заменяется MAC-адресом маршрутизатора, и увидеть вы сможете только его. Узнать MAC-адрес реального отправителя в общем случае невозможно (да и нужно ли?).

После проверки, что вы в одном сегменте с жертвой, нужно узнать ее настройки маршрутизации, конкретнее - шлюз по умолчанию. Шлюзом называется маршрутизатор, обеспечивающий связь с другой сетью, например, с Internet. Для нахождения маршрутизаторов используется их вышеуказанная особенность: от одного MAC-адреса приходят пакеты со многих IP-адресов.

Можно воспользоваться пассивным методом - анализировать (с помощью программы-анализатора), с кем жертва обменивается пакетами, и таким образом вычислить его. Можно воспользоваться активным: назначить у себя в настройках жертву шлюзом и послать пакет, допустим, на www.microsoft.com с помощью команды ping. Жертва, получив пакет, перешлет его маршрутизатору. Но такой способ уже более опасен, чем предыдущий, т.к. теоретически жертва может это обнаружить.

Можно вообще не искать маршрутизатор, а оставить тот, что выдан вам (предполагается, что вы подключены к тому же провайдеру). Это еще более опасный способ, но в большинстве случаев это срабатывает.

Выяснив все детали о жертве (IP, MAC, шлюз по умолчанию), можно приступать ко второму этапу. Нужно дождаться момента, когда жертва не находится в сети (выключена), установить параметры идентичными ей и спокойно работать. Когда жертва вернется в сеть, работа нарушится, кроме того, она может получить предупреждение о конфликтующих IP-адресах, но это ничем особым не грозит, т.к. определить, кто именно из вашего сегмента сети пытается притвориться ею, невозможно.

Как можно достоверно идентифицировать машину в сети? Необходимо использовать авторизацию и криптографические системы защиты. Доверять компьютеру просто на основании его IP и MAC - небезопасно.

Полностью надежной защитой от использования чужого трафика было бы использование для связи с маршрутизатором протокола, поддерживающего безопасную авторизацию и защиту от перехвата.

Ниже приводятся два метода защиты. Оба - достаточно надежные, но не защищают от одновременной работы нескольких машин под одним IP. Впрочем, это можно заметить.

1. Microsoft Winsock proxy client-server. Программа, заменяющая стандартный winsock в win9x, NT и win2k у клиента и MS Proxy server. Авторизуется через netbios входом в домен NT.
2. Нужно самому написать "клиент", который бы периодически посылал запрос "откройте доступ такому-то IP" на маршрутизатор. Для отсылки запроса нужен пароль. Если запроса нет, то доступ закрывается. В результате только знающий пароль откроет себе доступ, злоумышленник, изменивший настройки, ничего сделать не сможет.

## 3. Команды для удаленной атаки в Linux

### 3.1. Поиск файлов с паролями

Команда find позволяет найти файлы по какому-то схожему признаку (дата создания, атрибут, имя файла и т.д.).

Формат команды find: find <директория начала поиска> <шаблон значение> Шаблоны бывают разные, вот некоторые из них:

- name - поиск по имени
- user - поиск на принадлежность к какому-либо пользователю
- group - аналогично, только с группой.
- perm - поиск по атрибуту
- type - поиск по типу (l - ссылка, f - файл, d - директория)

Ниже приведена команда для поиска файла shadow [6] – это файл теневых паролей. Набираем команду либо через cgi-скрипт, либо в telnet`e:

http://yourprov.com/path/to/script/test.pl?find -type f -name shadow

Если вы нашли его, то вам осталось только запустить программу John the Ripper и подождать, пока не будет взломан пароль.

### 3.2. Авторизация для сервисов

Большинство сервисов в Linux требуют авторизации, и их passwd-файлы также могут лежать на сервере с account-ми пользователей (порой даже незашифрованных). Пример: uucp, webmin, sockd и многие другие сервисы.

Чтобы посмотреть, а не запущены ли они на сервере, надо просто выполнить команду "ps ax" и искать нечто похожее на программу-сервис.

Допустим вы обнаружили, что на взламываемой машине работает uucp (unix to unix connect protocol). Смотрим в каталог /etc/uucp на предмет файла passwd (в большинстве случаев он там есть). Остается только взломать пароли при помощи программы John the Ripper.

### 3.3. Причины существования уязвимостей в UNIX-системах

На рис.1 перечислены причины, по которым происходит до 90% всех случаев вскрытия UNIX-хостов:

- Наличие демонов.

- Механизм *SUID/SGID*-процессов. Эти механизмы, являющиеся неотъемлемой частью идеологии UNIX, были и будут лакомым кусочком для хакеров, т. к. в этом случае пользователь всегда взаимодействует с процессом, имеющим большие привилегии, чем у него самого, и поэтому любая ошибка или недоработка в нем автоматически ведет к возможности использования этих привилегий.
- Излишнее доверие. Об этом уже достаточно говорилось выше. Повторим, что в UNIX достаточно много служб, использующих доверие, и они могут тем или иным способом быть обмануты.
- Человеческий фактор с весьма разнообразными способами его проявления - от легко вскрываемых паролей у обычных пользователей до ошибок у квалифицированных системных администраторов, многие из которых как раз и открывают путь для использования механизмов доверия.

Рассмотрим теперь более подробно причины, по которым оказываются уязвимы демоны и *SUID/SGID*-процессы:

- возможность возникновения непредусмотренных ситуаций, связанных с ошибками или недоработками в программировании;
- наличие скрытых путей взаимодействия с программой, называемых "люками" ;
- возможность подмены субъектов и объектов различным образом.

К первым можно отнести классическую ситуацию с переполнением буфера или размерности массива, ведущую к затиранию области стека и записи туда специальных команд, которые будут затем исполнены. Этот способ, несмотря на свою популярность, всегда будет системозависимым и ориентирован только на конкретную платформу и версию UNIX.

Хорошим примером непредусмотренной ситуации в многозадачной операционной системе является неправильная обработка некоторого специального сигнала или прерывания. Часто хакер имеет возможность смоделировать ситуацию, в которой этот сигнал или прерывание будет послано (в UNIX'е посылка сигнала решается очень просто: командой kill).

Наконец, одна из самых распространенных программистских ошибок является неправильная обработка входных данных (это является некоторым обобщением случая переполнения буфера.) Так в 1990 и 1995 годах были подвергнуты автоматизированному тестированию около 80 программ на 9 различных платформах UNIX . Специальная программа подавала на вход строки длиной до 100000 символов. Результатом явилось то, что 25- 33% в 1990 г. и 18- 23% в 1995 г. работали некорректно: зависали, сбрасывали аварийный дамп и т. п. (Интересно, что в коммерческих версиях UNIX этот процент доходил до 43, тогда как в свободно распространяемых он был меньше 10.) Впрочем, справедливости ради надо отметить, что только 2 программы-демона вели себя таким образом в 1990 г., а через 5 лет эти ошибки были исправлены. Ну, а если программа неправильно обрабатывает случайные входные данные, то очевидно, что можно подобрать такой набор специфических входных данных, которые приведут к желаемым для хакера последствиям. Примером этого может служить innd.

Люком или "черным входом" (*backdoor*) часто называют оставленную разработчиком недокументированную возможность взаимодействия (чаще всего входа в систему), например, известный только разработчику универсальный "пароль" . Люки оставляют в конечных программах вследствие ошибки, не убрав отладочный код или вследствие необходимости продолжения отладки уже в реальной системе в связи с ее высокой сложностью, или же их корыстных интересов. Люки - это любимый путь входа в удаленную систему не только у хакеров, но и у журналистов и режиссеров вкупе с подбором "главного" пароля перебором за минуту до взрыва, но в отличие от последнего способа люки реально существуют. Классический пример люка - это, конечно, отладочный режим в sendmail.


Рис.1

Наконец, вследствие многих особенностей UNIX, таких как асинхронное выполнение процессов, развитый командный язык и файловая система, злоумышленниками могут быть использованы механизмы подмены одного субъекта или объекта другим. Например, часто применяется замена имени файла, имени получателя и т. п. именем программы.

Аналогично может быть выполнена подмена некоторых специальных переменных. Так, для некоторых версий UNIX существует атака, связанная с подменой символа разделителя команд или опций "¦" на символ "/" . Это приводит к тому, что когда программа вызывает /bin/sh, вместо него вызывается файл bin с параметром sh в текущем каталоге. Наконец, очень популярным в UNIX видом подмены является создание ссылки (*link*) на критичный файл. После этого файл-ссылка некоторым образом получает дополнительные права доступа и тем самым осуществляется несанкционированный доступ к исходному файлу. Аналогичная ситуация с подменой файла возникает, если путь к файлу определен не как абсолютный (/bin/sh), а относительный (../bin/sh или $(BIN)/sh).

И последнее - нельзя приуменьшать роль человека при обеспечении безопасности любой системы. Возможно, он даже является слабейшим звеном. О необходимости выбора надежных паролей уже говорилось. Неправильное администрирование - такая же актуальная проблема, а для UNIX она особенно остра, т. к. сложность администрирования UNIX-систем давно уже стала притчей во языцех и часто именно на это упирают конкуренты. Но за все надо платить, и это обратная сторона переносимости и гибкости UNIX. Более того, если говорить о слабости человека, защищенные системы обычно отказываются и еще от одной из основных идей UNIX - наличия суперпользователя, имеющего доступ ко всей информации и никому не подконтрольного. Его права могут быть распределены среди нескольких людей:

администратора персонала, администратора безопасности, администратора сети и т. п., а сам он может быть удален из системы после ее инсталляции. В результате вербовка одного из администраторов не приводит к таким катастрофическим последствиям, как вербовка суперпользователя.

Настройки некоторых приложений, того же sendmail, настолько сложны, что для поддержания работоспособности системы требуется специальный человек - системный администратор, - но даже он не всегда знает о всех возможностях того или иного приложения и о том, как правильно их настроить. И если хакеры смогли проникнуть в систему, то это не всегда говорит о халатности администратора, а, зачастую, о его ограниченном знании того или иного продукта.

### 3.4. Windows NT, Windows 2000 или Linux

Согласно исследованию, проведенному компанией Attrition.org, 52 % из всех серверов, которые были взломаны в течение 2000 г., работали именно под Windows NT. На втором месте Win2000 – 29.55 % взломанных серверов. А вот Linux показала себя весьма надежной системой - серверы под Linux пострадали всего в 3.96 % случаев от общего количества успешных хакерских атак. Следующая таблица иллюстрирует общее количество и процентное соотношение взломанных операционных систем.

**Таблица 1**

Май 2001 г.

| Операционная система | Количество | Проценты |
|---|---|---|
| Win-NT | 594 | 52,24 |
| Windows 2000 | 336 | 29,55 |
| Linux (unknown distro) | 49 | 4,31 |
| Linux (RedHat) | 45 | 3,96 |
| Irix | 37 | 3,25 |
| Solaris | 36 | 3,17 |
| FreeBSD | 15 | 1,32 |
| BSDI | 6 | 0,53 |
| SCO | 3 | 0,26 |
| Generic Unix | 2 | 0,18 |
| Linux (Cobalt) | 2 | 0,18 |
| Linux (Debian) | 2 | 0,18 |
| Linux (SuSE) | 2 | 0,18 |
| Unknown | 2 | 0,18 |
| AIX | 1 | 0,09 |
| Compaq True64 Unix | 1 | 0,09 |
| Linux (Conectiva) | 1 | 0,09 |
| Linux (Mandrake) | 1 | 0,09 |
| MacOS | 1 | 0,09 |
| PowerBSD | 1 | 0,09 |
| Общее количество взломов на май 2001 г. | 1137 | 100% |

На рис.2 представлено количество взломанных операционных систем, а на рис.3 – их процентное соотношение.

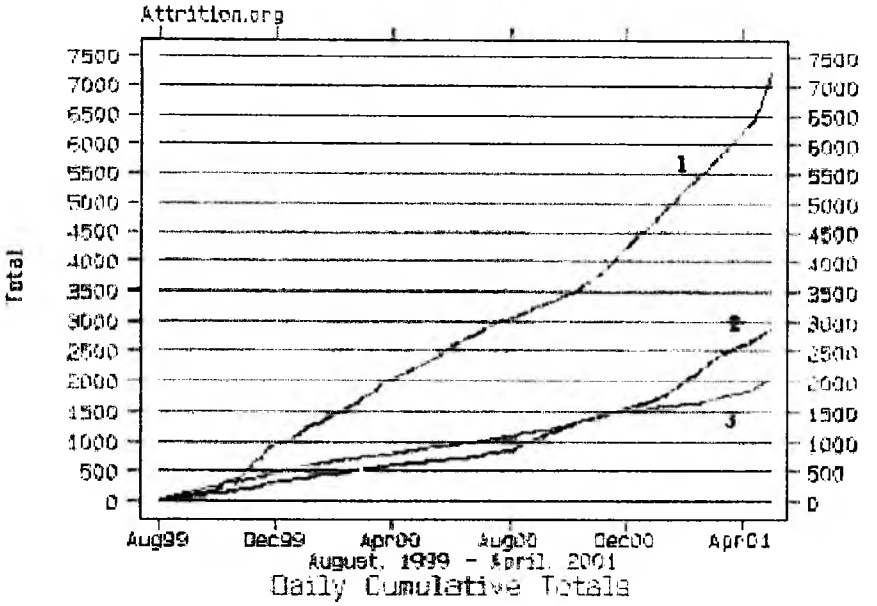



Рис. 2

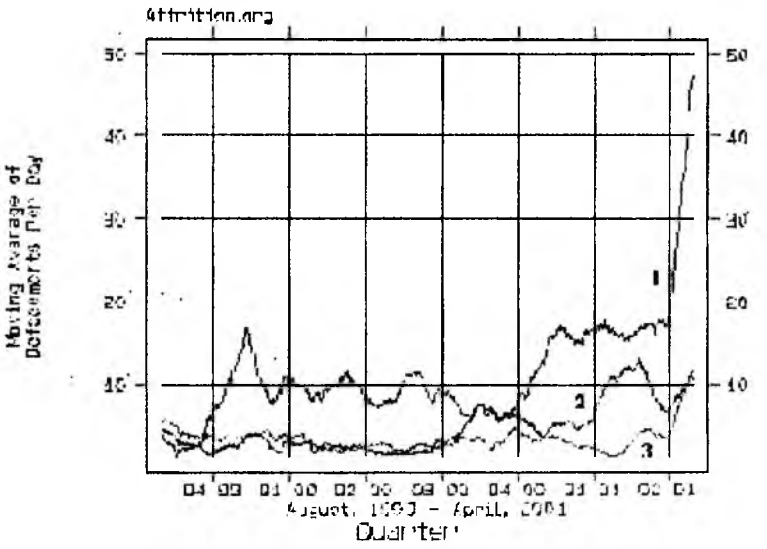



Рис. 3

Обозначение:

1 – WinNT, Win2000;

2 – Linux (все дистрибутивы);

3 – все остальные.

**Заключение**

Проведенный анализ возможностей и производительности Windows- и Unix-систем показал, что ОС Linux является несомненным лидером. Она предлагается большим числом поставщиков (нет угрозы монополии), хорошо масштабируется, более эффективна в использовании системных ресурсов, содержит средства удаленного администрирования, удаленного вычисления, многопользовательские возможности, полную палитру (профессионального) ПО, независимые стандарты разработчиков (POSIX). Далеко немаловажным фактором является цена: если Linux вместе со всем ПО Вы можете приобрести не более чем за 50$, то при приобретении Windows NT или Windows 2000 Вам придется платить за все программные

продукты отдельно и по баснословным ценам. Но самое важное при выборе между Windows NT и многими ОС Linux это то, что:

- ОС Linux дает Вам право выбора: любой тип "железа", CLI или GUI, коммерческое или GNU, разнообразное число поставщиков. Она динамична, т.е. можно собрать свое ядро, под свои конкретные нужды.
- Windows NT дает Вам сплошные ограничения: только Intel или Alpha; нет CLI, только GUI и только один GUI (нет того разнообразия оконных систем, которые есть под X-Window); только коммерческие MTA, только Microsoft и т.д. Windows NT статична, т.е. никогда не будет возможности собрать ядро на заказ. Одно ядро на все случаи.

Таким образом, на сегодняшний день Linux намного выгоднее чем Windows.

**Список литературы:** 1. 2001 Computer Crime and Security Survey // Computer Security Institute, San Francisco, March 12, 2001; 2. Common Criteria for Information Technology Security Evaluation (CCITSE) V2.1 // 1998; 3. *Зегжда Д. П., Ивашко А. М.* Как построить защищенную информационную систему/ Под науч. ред. Д. П. Зегжды и В. В. Платонова. СПб: Мир и семья 95, 1997. 312 с.: ил. 4. *Соболев. К.* Исследование системы безопасности в Windows NT, http://www.hackzone.ru/articles/ntadmin.html, 1998. 5. *Проскурин В. Г.* Проблемы защиты сетевых соединений в Windows NT, http://www.hackzone.rui/articles/ntadmintrap.html, 1999. 6. *Христов П. В.* Безопасность данных в ОС Unix / Открытые системы // М.: НИИСИ РАН, 1993.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 13.04.2002*

УДК 681.3.06

*А.И. ТИМОЧКО, канд. техн. наук, Я.В. ЕФИМЕНКО*

# К СОЗДАНИЮ ИНФОРМАЦИОННОГО ПРОСТРАНСТВА БОЯ В ПЕРСПЕКТИВНОЙ СИСТЕМЕ НАВЕДЕНИЯ ИСТРЕБИТЕЛЕЙ

Конфликт в Персидском заливе (1991 г.) дал первый опыт по изменению технологии ведения современных войн, а именно решительным наступательным действиям сухопутных войск предшествовала не многочасовая артиллерийская и авиационная подготовка, а многосуточная воздушная кампания. Ее главным результатом стало завоевание превосходства в воздухе, что позволило войскам многонациональных сил всего за сто часов очистить территорию Кувейта от агрессора.

Господство в воздухе может быть достигнуто нанесением решительного поражения основным группировкам авиации противника на земле (аэродромах), в воздухе и на море (многоцелевых авианосцах); группировке кораблей – носителей высокоточных крылатых ракет морского базирования большой дальности; основным группировкам ПВО; дезорганизацией управления авиацией и силами ПВО; поражением предприятий авиационной промышленности; центров подготовки специалистов для авиации, в особенности центров подготовки летного состава; запасов авиационного топлива и основных центров его добычи и производства.

Решение этих задач может быть осуществлено только при решающей роли военно – воздушных сил [1]. Вооруженные силы любого государства, не готовые к ведению войн нового поколения, вынуждены будут лишь наблюдать, как идет со всех направлений огромный по масштабам и длительности массированный удар высокоточных средств противника, проводимый одновременно с его операцией радиоэлектронной борьбы (РЭБ). Для противодействия подобному удару вооруженные силы должны создаваться не на базе крупных сухопутных группировок войск, а прежде всего на базе эффективной стратегической системы воздушно-космической обороны, способной отражать длительные массированные удары высокоточных средств противника, и на базе достаточного количества собственных высокоточных средств поражения различной дальности действия и средств на новых физических принципах.

Вследствие вышесказанного особую актуальность приобретает необходимость обеспечения надежного и эффективного управления войсками. Для повышения эффективности применения военно – воздушных сил широко используются автоматизированные системы управления авиацией и соответствующее программное обеспечение. В работе [3] рассматривается вариант построения перспективной системы для наведения истребителей на воздушные цели. Однако для принятия эффективного решения офицер боевого управления должен владеть единым информационным пространством поля боя. В настоящее время от средств радиолокации получается и обрабатывается достаточно большой объем данных, но имеющиеся индикаторы позволяют отображать их только на плоскости, что существенно затрудняет восприятие информации.

Вариантом решения информационных и расчетных задач в перспективной АСУ авиацией является построение трехмерного отображения информационного пространства боя с использованием современных ПЭВМ, что позволит достичь следующих преимуществ:

- Поле боя для офицера боевого управления преобразуется в пространство боя.
- При помощи изменения точки зрения - передвижения и наклона "камеры" и фокусного расстояния есть возможность просматривать поле боя в разных плоскостях и с разных позиций, что позволит быстрее, эффективнее и точнее оценить обстановку.
- В результате отображения информации с применением цветовой гаммы повысится наглядность изображения, что позволит улучшить читаемость информации и снизить степень утомляемости офицеров боевого управления при длительной работе.

Всю отображаемую информацию можно разделить на несколько составляющих:

1. Наземные неподвижные объекты.
2. Наземные подвижные объекты.
3. Воздушные объекты.
4. Воздушные зоны различного назначения.

Так как информация, касающаяся подвижных объектов, как наземных, так и воздушных, является динамической и постоянно обновляется с течением времени, наиболее разумно соответствующие программные объекты разместить в оперативной памяти. В этом случае обеспечивается быстрый доступ к необходимым данным, так как скорость обмена при обращении к ОЗУ намного превышает скорость при обращении к долговременной памяти. Данные по наземным неподвижным объектам и воздушным зонам обновляются намного реже, но их значение больше, и поэтому к ним предъявляются более высокие требования по сохраняемости. Поэтому наиболее целесообразно для их хранения создать специализированные базы данных, а при загрузке комплекса на основе данных, получаемых с сервера баз данных и контроллеров устройств сопряжения с аппаратурой передачи данных, построить пространство поля боя.

При разработке программного комплекса для решения поставленной задачи появляется ряд вопросов, связанных с наличием различных библиотек графического API (Application Programming Interface) и серверов баз данных, различными вариантами структуры баз данных и технологий разработки.

Для реализации подобного трехмерного терминала можно использовать библиотеку API Direct3D компании Microsoft, библиотеку PHIGS, VRML, Java-3D или OpenGL [4]. Что касается возможностей по реализации графического вывода, эти библиотеки предоставляют примерно одинаковые возможности, но библиотека Direct3D разрабатывалась для использования в разработке программ, предназначенных для работы с операционной системой Windows. Библиотека OpenGL поддерживается большинством существующих операционных систем, в том числе и операционной системой Linux, и большинством производителей рабочих графических станций и превосходит по удобству использования и возможностям остальные библиотеки, а независимые разработчики создали средства ее поддержки для большинства существующих программных и аппаратных платформ. Поэтому для обеспечения многоплатформенности разрабатываемого программного комплекса при разработке использовалась библиотека API OpenGL. Благодаря заложенной в библиотеку парадигме модели синтезированной камеры (Synthetic-Camera Model) сравнительно легко обеспечивается реализация трехмерного изображения и движение камеры.

В целях обеспечения быстрого доступа к данным, хранящимся в базе, из серверов баз данных был выбран MySQL Server and Clients 3.23 [5]. Так как этот сервер баз данных является многопоточным, его использование позволяет достичь высокой скорости получения данных, а при установке комплекса на многопроцессорной ЭВМ – выполнения отдельных потоков на различных процессорах, что также приводит к увеличению скорости обработки данных, что является важным фактором при построении систем реального времени. Кроме того, использование MySQL позволяет обеспечить быструю переносимость программного комплекса на операционную систему Linux, так как имеются версии сервера под операционные системы Windows и Linux. Согласно лицензии, которую предоставляет компания TcX, использование сервера баз данных MySQL в программных комплексах, предназначенных для некоммерческого использования, является бесплатным. Также преимуществом MySQL является наличие исходного кода сервера, который может быть при необходимости проверен и самостоятельно скомпилирован пользователем.

В настоящее время объектно-ориентированный подход к разработке программного обеспечения для многих прикладных задач является преобладающим [6,7]. Это дает возможность использовать современные CASE-средства (Computer Aided Software Engineering), такие как Rational Rose, для моделирования структуры и поведения системы с помощью UML (Unified Model Language) и объектно-ориентированные интегрированные среды разработки (Integrated Development Environment) C++ Builder и Delphi.

Для отображения наземных неподвижных объектов и воздушных зон используются объекты классов TAirZone, TAirBase, TCommandCenter и TRLS, задачей которых является хранение информации, характеризующей соответствующий реальный объект, и формирование дисплейного списка при создании объекта для реализации вывода объекта на экран. Диаграмма наследования классов данных объектов приведена на рис.1, причем в нее включены соответствующие классы для реализации задачи наведения и задачи отображения воздушных объектов. Для контроля вывода изображения на экран и контроля списков отображаемых объектов может использоваться соответствующий менеджер, который соответствует объекту Display Manager [3]. Эти функции в данном случае возлагаются на компонент MainForm (рис. 2). В настоящее время хранение указателей на объекты реализовано с помощью списков стандартного класса TList, включенных в класс компонента. Подобное решение позволяет ограничивать максимально возможное количество объектов в оперативной памяти только ее размерами. Так как емкость современных ОЗУ очень велика, а размеры объектов, напротив, малы, число отображаемых обьектов значительно превышает максимальное количество объектов в существующих системах управления. А так как каждый объект, соответствующий истребителю, имеет возможность осуществлять его наведение, количество каналов наведения в подобной системе ограничивается только количеством каналов радиоуправления.

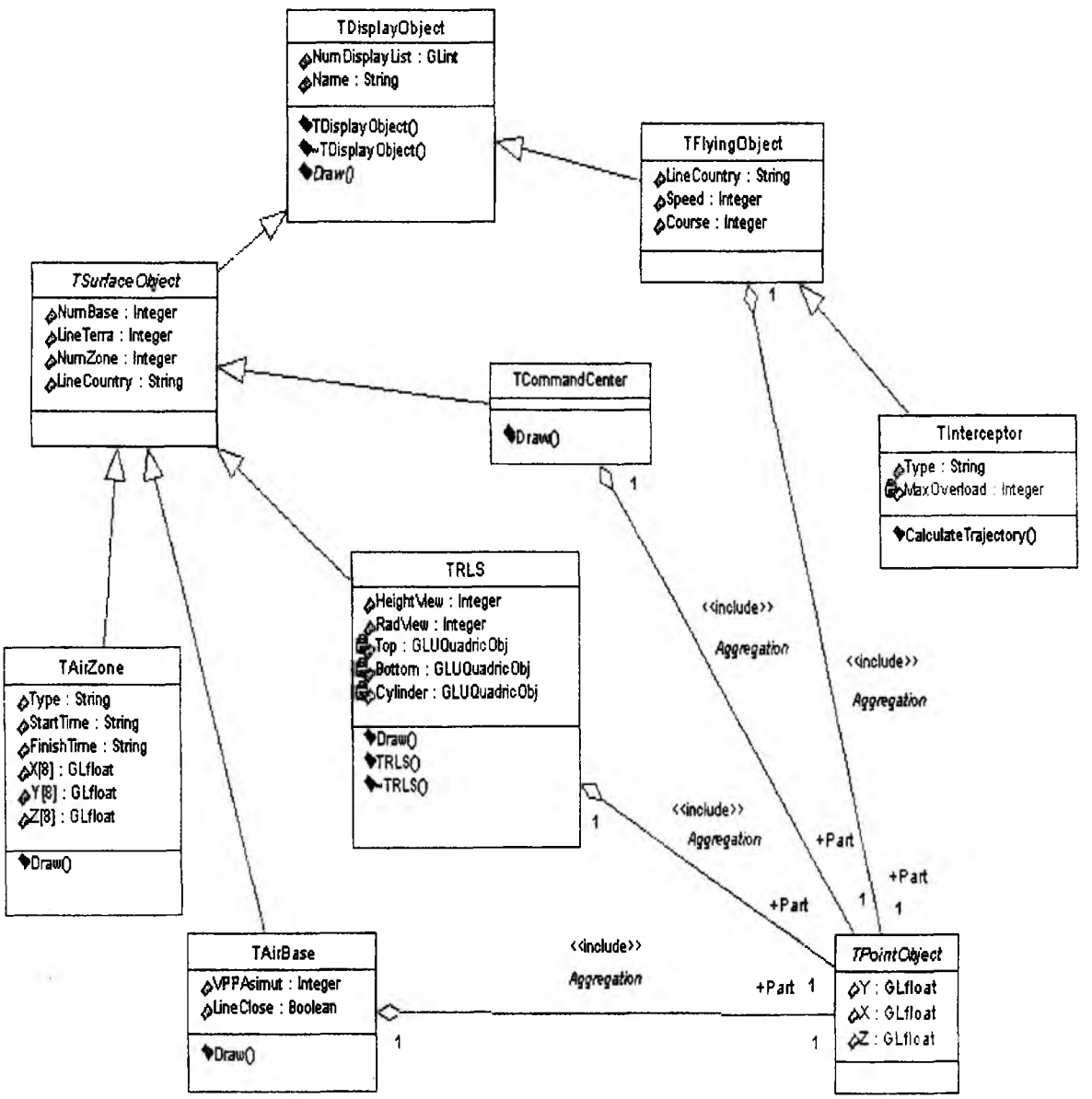



Рис.1 Диаграмма классов отображения информации

Создание, копирование, переименование, удаление и изменение свойств объектов осуществляются с помощью специализированных менеджеров (рис. 3), которые обращаются к

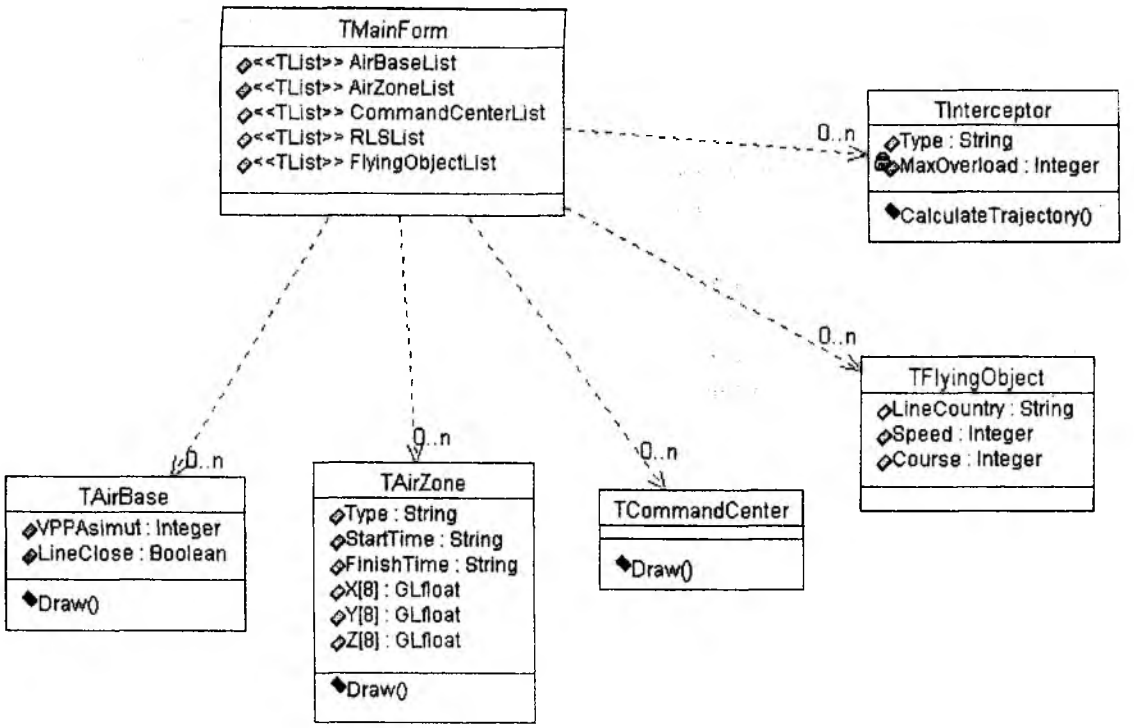



Рис. 2. Диаграмма взаимодействия классов

соответствующим спискам объектов компонента MainForm. В результате совершения какого-либо действия из перечисленных выше обновляется информация в базе данных программы и изменяются параметры соответствующих объектов в оперативной памяти или создаются новые объекты, а указатели их добавляются в список объектов данного типа, который входит в объект MainForm. Таким образом, информация в оперативной памяти и в базе данных дублируется, что позволяет обеспечить малую вероятность потери информации. Каждый менеджер состоит из двух форм: информационной формы и формы, предназначенной для ввода и изменения параметров объектов, необходимых для работы программного комплекса. Так как при выключении компьютера информация об объектах сохраняется в специализированных таблицах базы данных программы, и построение информационного пространства боя осуществляется на их основе, при помощи информационной формы пользователю предоставляется информация, которая находится именно в базе данных. Дружественное построение интерфейса менеджеров и применение соответствующих форм позволяет инкапсулировать особенности работы с базами данных и дублирование информации и упростить операции по изменению параметров объектов до уровня, доступного рядовому пользователю.

Хотя графическая библиотека API OpenGL не является объектно-ориентированной, ее использование позволяет сформировать изображение соответствующего объекта в виде дисплейного списка, в котором находится последовательность команд создания необходимых графических примитивов. Использование управления прозрачностью объектов позволяет отображать воздушные зоны и области видимости РЛС в виде сплошных полупрозрачных областей, благодаря чему резко повышается наглядность отображаемой информации. В результате применения заложенной в библиотеку парадигмы модели синтезированной камеры

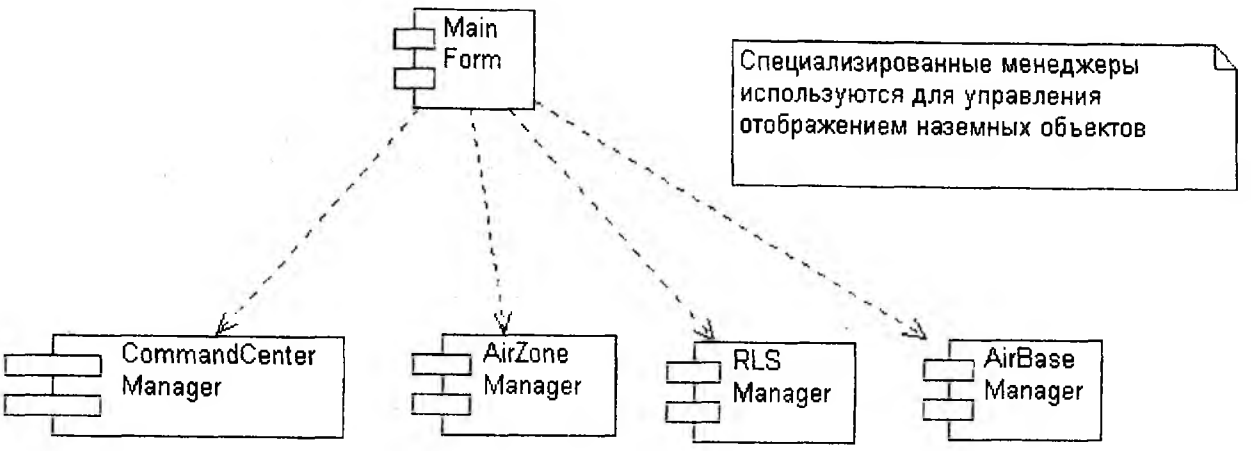



Рис. 3. Диаграмма компонентов

легко реализуется перемещение точки зрения по всем трем осям координат, повороты вектора зрения и изменение фокусировки.

**Выводы**

В результате проведения моделирования разработана структура программного комплекса, предназначенного для отображения информационного пространства боя в трехмерном виде, и опробованы на практике проектные решения. Преимуществами предложенной структуры программной системы являются:

1. Доступность и наглядность отображаемой информации и простота решения информационных задач.

2. Многоплатформенность программного комплекса, что достигается путем выбора соответствующих программных компонентов и принятия проектных решений.

3. Построение комплекса на основе стабильных промежуточных описаний при помощи объектно-ориентированного подхода.

4. Инкапсуляция дублирования информации и особенностей работы с сервером баз данных благодаря объектно-ориентированному построению баз данных и реализации клиентской части в виде специализированных менеджеров.

В результате подобного подхода к разработке программного комплекса обеспечивается возможность его эволюционного развития и дополнения возможностей новыми элементами без повторения процесса разработки в полном объеме.

**Список литературы:** 1. *Рог В.Г.* Господство в воздухе: прошлое и будущее. / Независимое военное обозрение. №7 (227), 2002 г. 2. *В. Слипченко.* Война будущего. / Военная мысль. №1, 1999 г. 3. *Тимочко А.И., Шапошникова Е.П., Ефименко Я.В.* Объектно-ориентированный подход для реализации задачи наведения в специализированной АСУ. / Авіаційно-космічна техніка і технологія: Зб. наук. праць. вип. 22 – Харків: Нац. Аерокосмічний ун-т "Харк. авіац. ін-т", 2001. – с.239-241. 4. *Эйнджел Э.* Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд.: Пер. с англ. – М.: Издательский дом "Вильямс", 2001. – 592 с.: ил. – Парал. тит. англ. 5. *Яргер Р., Риз Дж., Кинг Т.* MySQL и mSQL. Базы данных для небольших предприятий и Интернета. – СПб: Символ-Плюс, 2000 – 560 с., ил. 6. *Гради Буч.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд./Пер. с англ. - М.: "Издательство Бином", СПб.: "Невский диалект", 2000. - 560 с. 7. *Буч Г., Рамбо Д., Джекобсон А.* Язык UML. Руководство пользователя: Пер. с англ. – М.: ДМК, 2000. – 432 с.: ил. (Серия "Для программистов"). 8. *Архангельский А.Я.* Программирование в C++ Builder 5. - М.: ЗАО "Издательство Бином", 2000. - 1152 с.

*Харьковский институт ВВС*
*им. Ивана Кожедуба*

*Поступила в редколлегию 15.04.2002*

УДК 681.324.067

*А.А. ТОРБА, канд. техн. наук, В.А. БОБУХ, С.Г. ЕЛАКОВ, А.З. СТЕПЧЕНКО*

# ПОВЫШЕНИЕ БЫСТРОДЕЙСТВИЯ ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА ОСНОВЕ ФИЗИЧЕСКИХ ДАТЧИКОВ

Ключевой проблемой технических средств защиты информации является генерация случайных равновероятных последовательностей на основе физических датчиков. Важным параметром таких генераторов является быстродействие, измеряемое количеством генерируемых случайных битов в секунду (бит/с).

Рассмотренные в статье [1] методы генерации случайных последовательностей на основе физических датчиков шума обладают ограниченным быстродействием, потому что скорость генерации случайных битов, определяемая частотой заполнения сдвигающего регистра **Fo** (см. рис.1), должна быть в 3...5 раз меньше средней частоты шумовых импульсов **Fш** на выходе датчика. Повышение быстродействия возможно за счет применения более широкополосных квантовых генераторов шума. Однако это значительно удорожает генератор, а также приводит к увеличению его габаритов и веса.

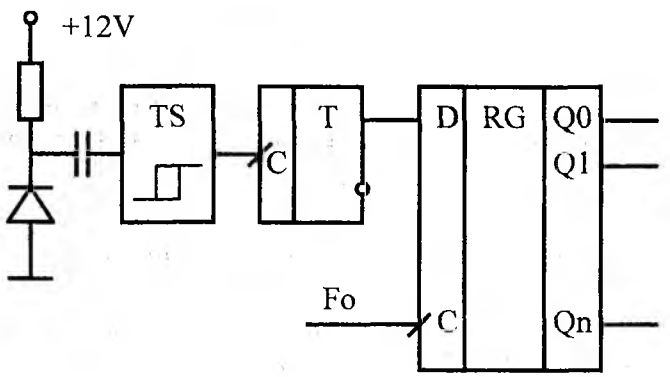



Рис. 1

Если нарушить приведенное в статье [1] ограничение, то есть увеличить частоту заполнения сдвигающего регистра **Fo** в несколько раз, то это приведет к ухудшению статистических свойств генерируемой случайной последовательности. Экспериментально проверено, что нарушение указанных ограничений приводит к увеличению количества серий из двух и трех последовательных нулей и единиц.

В той же статье [1] приведены и методы улучшения статистических свойств генерируемых случайных двоичных последовательностей, в частности, метод «дельта квадрат», то есть объединение элементом «ИСКЛЮЧАЮЩЕЕ ИЛИ» двух генерируемых битов. На рис. 2 приведена функциональная схема генератора, в которой схемой «ИСКЛЮЧАЮЩЕЕ ИЛИ» объединены: генерируемый шумовым датчиком случайный бит и другой бит, сгенерированный ранее. Для того, чтобы эти биты были независимы, необходимо выбирать длину (m) сдвигающего регистра в два-три раза больше, чем разрядность (n) случайных слов, считываемых в компьютер.

Эту схему (рис. 2) можно рассматривать и как сдвигающий регистр, замкнутый в кольцо (кольцевой счетчик), у которого элемент «ИСКЛЮЧАЮЩЕЕ ИЛИ» используется для инверсии в случайные равновероятные моменты времени сигнала обратной связи. Известно, что у обычного кольцевого счетчика последовательности будут повторяться с периодичностью, не превышающей длину (m) этого счетчика. Поэтому разрядность (длину) регистра желательно увеличивать.

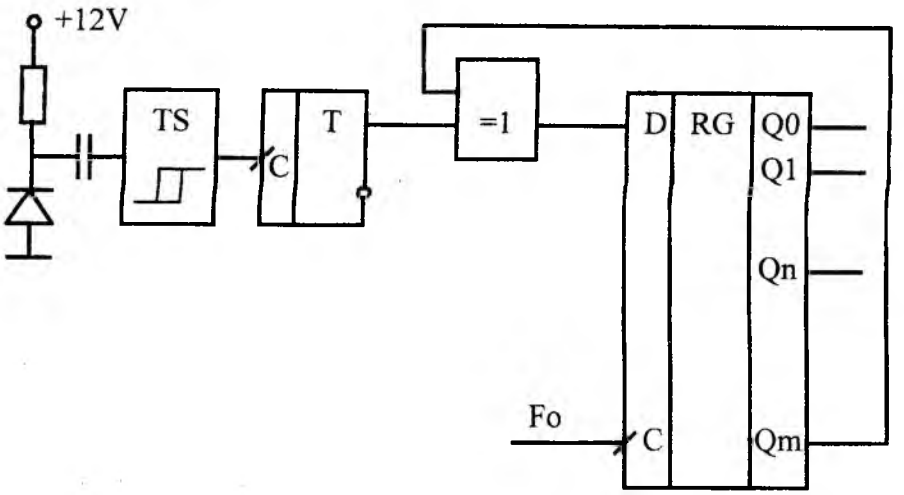



Рис. 2

Но можно значительно увеличить периодичность сдвигающего регистра [2], если реализовать обратную связь с дополнительным элементом «ИСКЛЮЧАЮЩЕЕ ИЛИ» (см. рис. 3). Максимальная периодичность такого линейного рекуррентного регистра (ЛРР) равна:

$$K = 2^m - 1$$

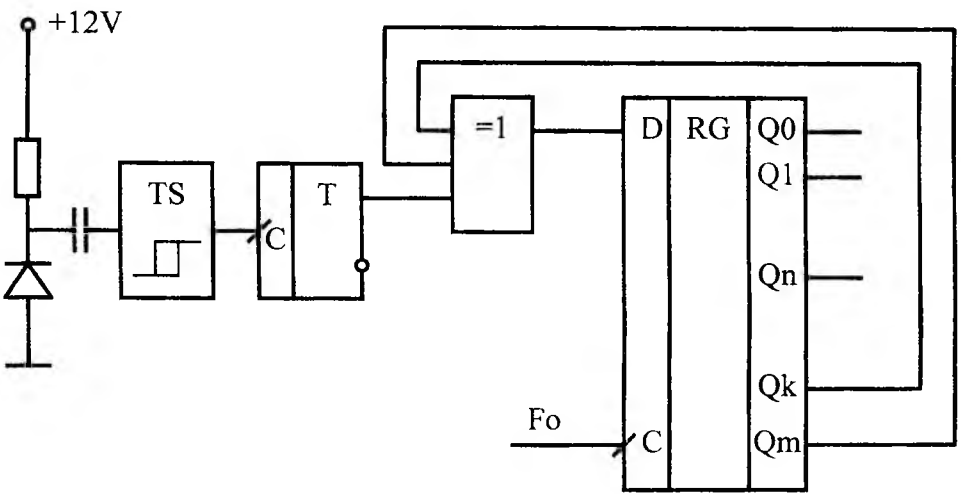



Рис. 3

Так, 60-ти разрядный регистр при тактовой частоте 10 МГц имеет период повторения несколько миллионов лет. А 100-разрядный регистр имеет период повторения $10^{18}$ лет, то есть в миллион раз превышающий возраст Вселенной.

Известно, что ЛРР является идеальным генератором *псевдослучайных равновероятных последовательностей*. Каждый отрезок такой последовательности можно рассматривать как *случайный*, если не известна предыстория его формирования. Для этого необходимо, чтобы каждое слово этой последовательности считывалось в случайные моменты времени и длина каждой считываемой последовательности была значительно меньше, чем пропущенное количество битов между считываниями.

Схему на рис. 3 можно рассматривать как сдвигающий регистр, в который вводятся случайные биты от источника с физическим датчиком шума, а цепь обратной связи с элементом

«ИСКЛЮЧАЮЩЕЕ ИЛИ» используется для улучшения статистических свойств случайной последовательности по методу «дельта квадрат».

Эту же схему (рис. 3) можно рассматривать как генератор псевдослучайных последовательностей на основе ЛРР, в котором в случайные равновероятные моменты времени «разрушается рекуррента» за счет инверсии сигнала обратной связи элементом «ИСКЛЮЧАЮЩЕЕ ИЛИ», что делает такие последовательности непредсказуемыми, то есть случайными.

Какой из этих подходов преобладает, определяется отношением частоты сдвига **Fo** к средней частоте шума физического датчика **Fш**. Если частота шумового датчика **Fш** соизмерима или больше, чем тактовая частота сдвигающего регистра **Fo**, то схему на рис. 3 необходимо рассматривать, как обычный генератор случайных сигналов с улучшенными параметрами по методу «дельта квадрат». Если частота **Fo** значительно превышает среднюю частоту датчика шума **Fш** (в 100 и более раз), то эту схему необходимо рассматривать, как генератор на основе ЛРР с «разрушением рекурренты» в равновероятные случайные моменты времени. Причем, если длина рекурренты превышает *миллионы лет*, а разрушается она *миллионы раз в секунду*, то говорить о возможности восстановления ее предыстории бессмысленно.

Для повышения надежности генератора случайных равновероятных последовательностей, а также для повышения устойчивости генерируемых случайных последовательностей к алгоритмам криптоанализа предлагается *многоканальная схема* с горячим резервированием каналов физических датчиков шума (на рис. 4 приведена двухканальная схема).

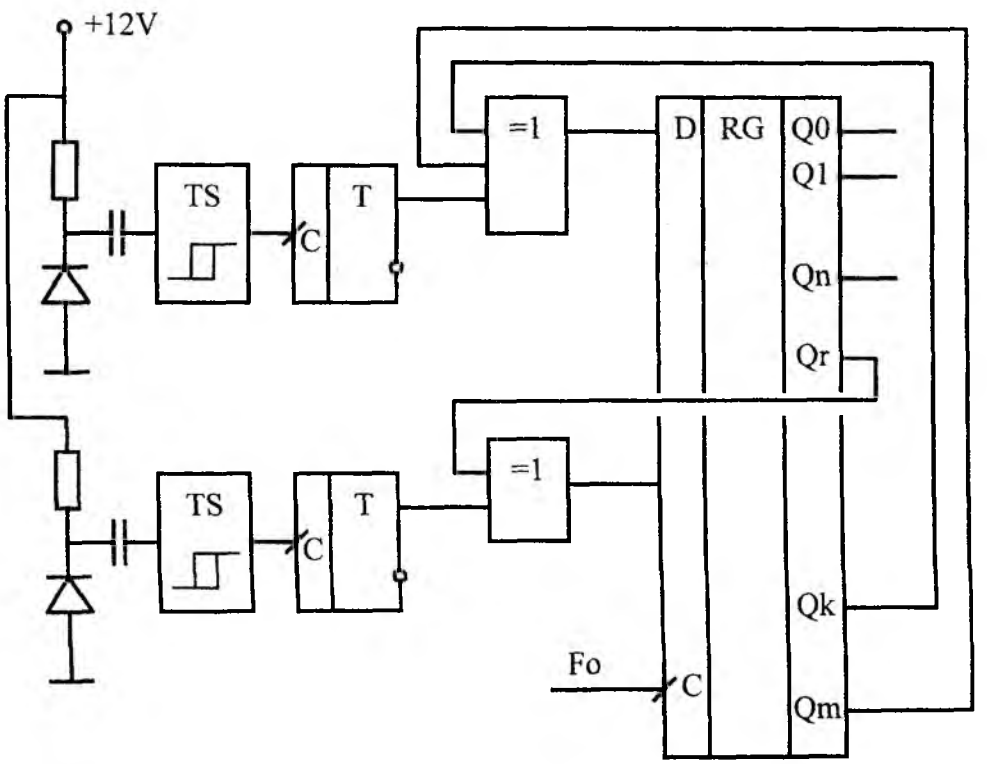



Рис. 4

Эта схема позволяет многократно «разрушать рекурренту» при прохождении случайного сигнала по сдвигающему регистру, что делает генерируемые последовательности еще более непредсказуемыми и затрудняет их криптоанализ. Это позволяет также повысить скорость генерации случайных последовательностей, то есть увеличить частоту **Fo** по сравнению с частотой **Fш** для каждого физического датчика.

Случайный характер сигнала сохраняется даже при неработоспособности всех каналов физических датчиков шума, кроме одного. Этим достигается горячее резервирование.

На основе вышеизложенного сформулируем требования к генераторам случайных последовательностей на основе физических датчиков шума с высоким быстродействием (на примере схемы на рис.4):

1. Длина сдвигающего регистра (m) должна значительно превышать разрядность (n) считываемых в компьютер случайных слов. Это позволяет также реализовать период рекурренты в несколько миллионов лет и более.
2. Количество пропущенных случайных битов между соседними операциями считывания должно значительно превышать разрядность считываемых слов.
3. Случайные временные интервалы между операциями считывания определяются спецификой работы компьютера, например, за счет многозадачной работы операционной системы.
4. Многоканальные схемы датчиков физического шума позволяют повысить надежность системы и затрудняют криптоанализ генерируемых случайных последовательностей.

Авторами реализованы генераторы случайных последовательностей со скоростью генерации от 5 до 15 Мбит/с. Максимальные скорости определяются быстродействием примененной элементной базы и скоростью ввода информации в компьютер через слот PCI-32/33МГц. Использовалась двухканальная схема резервирования физических датчиков шума. Результаты тестирования подтвердили правильность принятых решений.

**Список литературы**: 1. *А.А. Торба, С.Г. Елаков, А.З. Степченко* Генерация равновероятных случайных последовательностей на основе физических датчиков // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С.108-113. 2. Деклар. пат. 36108 України, МКІ 6 G06F7/58, G07C15/00. Спосіб генерації випадкових чисел та пристрій для його здійснення/ О.О. Торба (Україна). – 4 с. іл.; Опубл. 16.04.2001, Бюл. № 3.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 23.04.2002*

УДК 681.3.067+621.396.626:537.87

*В.І. ЗАБОЛОТНИЙ, канд. техн. наук, Г.А. ЛЕБЕДЕВ, О.П. МЕТЕЛЕВ*

# ЗАБЕЗПЕЧЕННЯ ДОСТОВІРНОСТІ ОЦІНКИ ДАЛЕКОСТІ ВИЯВЛЕННЯ ВИПРОМІНЮВАНЬ ТЕХНІЧНИХ ЗАСОБІВ ПЕРЕДАЧІ ІНФОРМАЦІЇ

## Вступ

Державні стандарти [1], нормативно-правові документи [2,3,4] передбачають виявлення загроз витоку інформації з обмеженим доступом за рахунок побічних електромагнітних випромінювань (ПЕМВ) технічних засобів зберігання, передачі, обробки інформації (ТЗПІ). Одним із можливих критеріїв наявності технічного каналу витоку інформації (ТКВІ) є співвідношення між далекістю можливого виявлення випромінювань (ДМВВ) апаратурою розвідки (АР) та контрольованою територією навколо об'єкту ТЗПІ. Таким чином, достовірна оцінка далекості можливої розвідки $R_p(\varphi)$ ПЕМВ навколо об'єкту ТЗПІ ($\varphi$ - азимут на АР) являє собою важливу практичну задачу. Поставлена задача може вирішуватись аналітичним, експериментальним та аналітично-експериментальним способами.

Оцінка ДМВВ останнім способом має переваги перед іншими. По-перше, у реальній обстановці проявляються поглинання випромінювань стінами будинків і споруд, вплив місцевих предметів, рослинності і таке інше, що ускладнює проведення відповідних розрахунків. По-друге, технічні характеристики апаратури контролю (АК), як правило, відрізняються від характеристик АР, або буває неможливо помістити апаратуру контролю у точку ймовірного розташування АР. Розрахунково-експериментальний спосіб не має згаданих недоліків і полягає у перерахунку виміряного у точці контролю $R_k$, рівня сигналу $P_k$ на сигнал $P_{op}$, який відповідає передбаченій чутливості АР, розташованої на відстані, що оцінюється $R_p$.

## 1. Модель оцінки далекості виявлення випромінювань

В реальних умовах виникає можливість одночасного впливу на приймач не тільки прямої хвилі, але й хвилі, що відбита від земної поверхні. Механізм такого впливу був досліджений академіком Б. О. Введенським [5] і реалізований у вигляді формули:

$$E_i = \frac{\sqrt{30PF(\varphi_i)L(\varphi_i)}\ HH_i}{R_i^2 \qquad \lambda}, \tag{1}$$

де $i = p,k$ – індекс приналежності величини до АР або АК, відповідно;

$E_i$ - напруженість поля у точці розташування антени $i$-го приймача;

$P$ - потужність сигналу, що випромінюється ТЗПІ;

$L(\varphi)$– послаблення сигналу спорудами між випромінювачем та приймачем;

$F(\varphi)$ – значення діаграми спрямованості випромінювання в напрямок на приймач;

$Ri$ – відстань між випромінювачем та апаратурою прийому сигналу;

$H$- висота антени випромінювача;

$Hi$- висота антени апаратури прийому сигналу;

$\lambda$- довжина хвилі випромінювання;

$\varphi$- азимут на апаратуру прийому сигналу.

Застосування формули (1) обмежено умовою яка звичайно має місце на практиці: $4HH_i / R\lambda << 1$.

Взаємний зв`язок між потужністю прийнятого апаратурой сигналу $P_i$ та напруженістю електричного поля $E_i$ можна знайти за формулою: $P_i=(E_i^2/z)(G_i \lambda^2/4\pi)$, де $z$ – хвильовий опір середовища, що дорівнює $z = 120\pi$; $G_i$ - коефіцієнт спрямованої дії антени апаратури прийому сигналу.

Для того, щоб можна було вести контроль (розвідку), необхідно, щоб виконувались умови: $P_p \geq P_{op}$; $P_k \geq P_{ok}$. Тоді для максимальних далекостей розвідки та контролю при рівнях сигналу, який дорівнює чутливості апаратури:

$$P_{op} = \frac{30PF(\varphi_p)L(\varphi_p)}{R_p^4 z}\frac{G_p\lambda^2}{4\pi}\left(\frac{4\pi H \cdot H_p}{\lambda}\right)^2,$$

$$P_{ok} = \frac{30PF(\varphi_k)L(\varphi_k)}{R_k^4 z}\frac{G_k\lambda^2}{4\pi}\left(\frac{4\pi H \cdot H_k}{\lambda}\right)^2,$$

де $P_{ok}$ $(P_{op})$- чутливість АК (АР);

$G_k$ $(G_p)$- коефіцієнт спрямованої дії антени АК (АР);

$H_k$ $(H_p)$- висота антени АК (АР).

При збіганні напрямку контролю та розвідки $\varphi_k=\varphi_p$ дає $F(\varphi_k)=F(\varphi_p)$ та, відповідно,

$L(\varphi_k) = L(\varphi_p)$.

Звичайно висоти $H_p$ та $H_k$ не дуже відрізняються, тоді не дуже відрізняються у вертикальних напрямках $F(\varphi_k)$ від $F(\varphi_p)$ та $L(\varphi_k)$ від $L(\varphi_p)$. Надалі запис азимуту $\varphi$ буде опущено.

Якщо виразити з цих формул $R_p$ та $R_k$, а потім визначити $R_p$, то можна отримати вираз для знаходження далекості розвідки випромінювань ТЗПІ по результатах контролю

$$R_p = R_k \sqrt[4]{\frac{P_{ok}}{P_{op}}\frac{G_p}{G_k}\left(\frac{H_p}{H_k}\right)^2}. \qquad (2)$$

**2. Похибки оцінки далекості виявлення випромінювань**

Одержаний вираз (2) перерахунків далекості контролю $R_k$ у далекість розвідки $R_p$ потребує аналізу з точки зору можливих похибок, які виникають при оцінці всіх складових формули. Природа таких похибок обумовлена наступним.

Вихідна величина $R_k$, як і завжди при проведенні вимірювань відстаней, являє собою випадкову величину з середньоквадратичною похибкою виміру $\sigma_k = \delta R_k$, де $\delta$ - відносна похибка виміру, яка залежить від способу проведення виміру відстані: рулеткою, спідометром автомобіля, кроками і таке ін. Звичайно $\delta << 1$.

Величина $P_{ok}$ оцінюється за результатами вимірювань за допомогою робочого генератора високої частоти. Шкала потужності сигналу генератора оцифрована у абсолютних одиницях або у децибелах відносно певного рівня, наприклад, 1Вт. В інструкції з експлуатації такого генератора зазначені можливості похибки при встановленні певного сигналу генератора

у відносних одиницях, як правило у дБ. Вважається, що похибка встановлення відносно певного рівня сигналу по шкалі у децибелах, розподіляється за нормальним законом.

Величина $G_k$, задана у інструкції з експлуатації апаратури контролю. Часто величина $G_k$ задається у вигляді $G_k=G_{kmin}\ldots G_{kmax}$ для певної антени у певному діапазоні хвиль. Цю величину теж можна вважати випадковою.

Величина $H_k$ має випадковий характер, бо при проведенні вимірів висоти виникають похибки інструментального характеру, як у випадку виміру відстаней.

Величини $P_{op}$, $G_p$, $H_p$ вважаються точними, бо їх значення задані у моделях [6].

Постановка задачі оцінки помилки ДМВВ наступна. Дана система $m$=4 випадкових величин $R_k, P_{ok}, G_k, H_k$ та величина $R_p$, яка являє собою однозначну аналітичну функцію від цих величин (2).

Відома сумісна густина ймовірності чотирьох вихідних величин $\omega_4(R_k, P_{ok}, G_k, H_k)$. Треба знайти одномірну густину ймовірності $\omega_1(R_p)$ для подальшої оцінки такого $R_p^*$, що:

$$\int_0^{R_p^*} \omega_1(R_p)\,dR_p = P_{зад},$$

де $P_{\sim w}$ – вірогідність достовірної оцінки далекості виявлення випромінювань ТЗПІ;

$R_p^*$ - далекість виявлення виявлення випромінювань ТЗПІ АР у заданому напрямку з вірогідністю не менш ніж $P_{\sim w}$.

Шлях одержання одномірного розподілу $\omega_1(R_p)$ з чотирьохмірного $\omega_4(R_k, P_{ok}, G_k, H_k)$ математично визначений, наприклад, у [7] і пов'язаний з одержанням якобіану від відповідних величин.

Інтегрування по області визначення незалежних величин $P_{ok}$, $G_k$, $H_k$ дає потрібний вираз:

$$\omega_1(R_p)= \int_{P_{ok}}\int_{G_k}\int_{H_k} \omega_1[R_k(R_p,P_{ok},G_k,H_k)]\omega_1(P_{ok})\omega_1(G_k)\times$$

$$\times\,\omega_1(H_k)\left(\frac{P_{ok}}{P_{op}}\,\frac{G_p}{G_k}\left(\frac{H_p}{H_k}\right)^2\right)^{-1/4} dP_{ok}\,dG_k\,dH_k.$$

Труднощі аналітичного обчислення величини $R_p$ обумовлені, перш за все, мультиплікативною залежністю кінцевої величини від вихідних ( 2 ), що як правило, не дає змогу застосовувати операцію інтегрування. Виходом із цієї ситуації може бути перетворення мультиплікативної формули в адитивну, і знаходженню густини розподілу $R_p$ композицією законів розподілів вихідних величин $R_k$, $P_{ok}$, $G_k$, $H_k$.

### 3. Модернізований шлях враховування похибок

Перетворити формулу (2) в адитивну можна, використовуючи операцію логарифмування або, як прийнято у радіотехніці, перевід у систему одиниць, що виражена у децибелах відносно опорних одиниць (метр, Ват). Для визначеності, модифіковані одиниці вихідних величин будуть записуватись рядковими буквами.

Отже, враховуючи, що одиниці потужності знаходяться під знаком радикалу, логарифмувати (2) потрібно з коефіцієнтом "40" перед знаками логарифму

Підставляючи після логарифмування згадані вище рядкові букви, можна одержати:

$$r_p = -m + r_k - p_{ok} + q_k + h_p - h_k,$$

де $r_p = 40\ lg\ R_p$ – відстань розташування (АР) відносно 1м ;

$m = 10\ell g\frac{G_p}{P_{op}} = -10\ell g\frac{P_{op}}{G_p}$ - еквівалентна чутливість апаратури розвідки відносно 1Вт, надалі m – масштабна складова;

$r_k = 40\ell gR_k$ - відстань розташування АК відносно 1м;

$p_{ok} = 10\ell gP_{ok}$ - чутливість приймача АК відносно 1Вт;

$q_k = 10\ell gG_k$ - коефіцієнт спрямованої дії антени АК у дБ;

$h_k = 20\ell gH_k$ -висота антени АК відносно 1м;

$h_p = 20\ell gH_p$ висота антени АР відносно 1м.

Приведені вище величини $r_k$ , $p_{ok}$, $q_k$, $h_k$ і похідна від них $r_p$ – випадкової природи. В більшості випадків названі величини розподілені за нормальним законом.

Розподіл $r_P$ може бути знайдений досить легко, якщо вихідні величини $r_k$, $p_{ok}$, $q_k$, $h_k$ являють собою незалежні випадкові величини і кожна з них відноситься до одного зі стійких розподілів [8].

Застосовуючи властивості стійких розподілів випадкових величин для композиції нормальних розподілів, можна вказати ,що:

$$W_1^1(r_p) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{Rk}^2 + \sigma^2 p_k + \sigma_{q_k}^2 + {}^2_{h_k}}} \times \exp\left[ -\frac{1}{2} \frac{(r_p - r_k - p_{ok} + q_k - m)^2}{\sigma_{Rk}^2 + \sigma_{p_k}^2 + \sigma_{q_k}^2 + \sigma_{h_k}^2} \right],$$

де $\sigma_{Rk}$ -середньоквадратична похибка виміру далекості відстані АК;

$\sigma_{p_k}$ -середньоквадратична похибка чутливості АК;

$\sigma_{q_k}$ -середньоквадратична похибка виміру коефіцієнту спрямованої дії антени АК;

$\sigma_{h_k}$ -середньоквадратична похибка виміру висоти антени АК.

Узагальнюючи останній вираз можна записати, що значення математичного очікування питомої відстані у логарифмічних одиницях $\hat{r}_p$ буде мати вигляд:

$$\hat{r}_p = 40\ell gR_k + m + p_{ok} - q_k + h_p - h_k,$$

значення середньоквадратичного відхилення величини: $r_p$:

$$\sigma_p = \sqrt{\sigma_{Rk}^2 + \sigma_{pk}^2 + \sigma_{qk}^2 + {}^2_{h_k}}.$$

Наведене вище дає змогу розрахувати $r_p^*$ таке, що

$$\int_{-\infty}^{r_p^*} W_1^1(r_p)\,dr_p = P_{зад}.$$

На практиці $P_{зад}$ може, наприклад, мати значення, як прийнято у статистичних розрахунках, від 0,9 - для об'єктів, захист яких має не дуже велике значення, до 0,995 – для об'єктів із дуже високим, ступенем захисту.

**Висновки**

З наведеного вище витікає наступна послідовність оцінки границі ЗМВВ ТЗПІ зарезультатами вимірів.

1. Для частоти ПЕМВ ТЗПІ, що оцінюють, вимірюють значення чутливості приймача АК $p_{ok}$ у відносних одиницях [дБВт] і розраховують помилку $\sigma_{pk}$ [дБ].

2. Встановлюють величину $q_k$ [дБ] і її помилки $\sigma_{qk}$ [дБ].

3. Вимірюють висоту антени АК $H_k$, обчислюють $h_k$ і розраховують помилки $\sigma_{h_k}$.

4. Розраховують еквівалентну чутливість $m$ за даними довідкової літератури чи відповідної моделі. Висота антени АР $h_P$ визначається за даними довідкової літератури.

5. Здійснюються виміри відстаней $R_k$ у кожному з потрібних напрямків і оцінюється їх похибка.

6. Обчислюється величина $r_k = 40 \lg R_k$ і її похибка.

7. Встановлюється величина $r_p = -m + r_k - p_{ok} + q_k + h_p - h_k$,

8. Для заданого $P_{зад}$ знаходиться величина $t$ така, що:

$$P\sim w\{=\frac{1}{2}+\frac{1}{\sqrt{2\pi\sigma_p}}\int_0^t \exp\left[-\frac{\left(r_p-\hat{r}_p\right)^2}{2\sigma_p^2}dr_p\right];$$

9. За одержаним $t$ знаходять $r_p^* = t\sigma_p + \hat{r}_p$, після чого встановлюється величина

$$R_p^* = 10^{\frac{r^*}{40}} = 10^{\frac{t\sigma_p}{40}} 10^{\frac{\hat{r}_p}{40}} = 10^{\frac{t\sigma_p}{40}} R_k \sqrt[4]{\frac{P_{ok}}{P_{op}}\frac{G_p}{G_k}\left(\frac{h_p}{h_k}\right)^2}.$$

Отримана формула дозволяє врахувати помилки, які виникають при оцінці ДМВВ ТЗПІ і дозволяє одержати оцінку далекості з заданою достовірністю.

**Список літератури:** 1.*ДСТУ 3396.0 –96*. 2.*"Інструкція щодо умов і правил здійснення діяльності у галузі ТЗІ та контролю за їх дотриманням"*. 3. *"Временные рекомендации по технической защите информации в средствах вычислительной техники, автоматизированных системах и сетях от утечки по каналам побочных електромагнитных излучений и наводок"* //Безопасность информации//. 1995. №2. 4. " *Временные рекомендации по технической защите информации от утечки по каналам побочных електромагнитных излучений и наводок* ".//Безопасность информации//. 1995.№2. 5. *А.И.Калинин, Е.Л.Черенкова* " Распространение радиоволн и работа радиолиний". М.: Изд. "Связь", 1971. 546с. 6. *"Положение по технической защите информации в Украине"*//Безопасность информации//, 1996. №2. 7. *Левин Б.Р.* Теоретические основы статистической радиотехники. Книга первая. Изд. 2-е, перераб. и доп., М., «Сов. Радио», 1974. 552с. 8. *Справочник по вероятностным расчётам*". М.:Воениздат.1970. 536 с.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 29.04.2002*

УДК 389.2

*Ю.И. ЕВДОКИМЕНКО к.ф.-м.н., А.П. НАРЕЖНИЙ*

# КРИТЕРИЙ ВЫДЕЛЕНИЯ КВАЗИПЕРИОДИЧЕСКИХ СОСТАВЛЯЮЩИХ, ОБУСЛОВЛЕННЫХ ВЗАИМОДЕЙСТВИЕМ МЕР В ГРУППОВОМ ЭТАЛОНЕ ЧАСТОТЫ

В [1] был предложен подход к рассмотрению группового эталона (ГЭ) частоты и времени как к многочастотной автоколебательной системе, что позволило разработать математическую модель флуктуаций фазы каждой меры из состава ГЭ [2]. Идентификацию параметров ГЭ предлагается осуществлять стохастической моделью системы связанных осцилляторов по результатам парных сличений, в режиме амплитудных биений для каждой пары мер [2]. Наличие биений в системе связанных осцилляторов с учетом их шумов приводит к появлению в фазе составляющих, имеющих квазипериодический и стохастический коррелированный тип проявления, и, соответственно, к появлению «ярких» линий в спектральных плотностях мощности фазовых флуктуаций (СПМФФ) данных мер [3].

Выявление данных линий по оценкам энергетических спектров требует разработки статистического критерия выделения значимых периодичностей. Для разработки критерия можно воспользоваться предложенным в работах [3,4] подходом к выделению скрытых периодических составляющих в прецизионных мерах частоты по оценкам СПМФФ. На первом этапе идентифицируется $\widehat{S}_{\phi_{ij}}(f(k))$ - оценка фоновой части СПМФФ разности фаз между $i$ - той и $j$ - той мерой и $S_{\phi_{ij}}(f(k))$ - оценка с максимальным разрешением по частоте анализа разности фаз данных мер (см. рис.1). Поведение остатка СПМФФ разности фаз $\Delta S(f_k) = S_{\phi_{ij}}(f(k)) - \widehat{S}_{\phi_{ij}}(f(k))$ между первой и второй мерами группы, состоящей из трех мер, представлено на рис. 2.

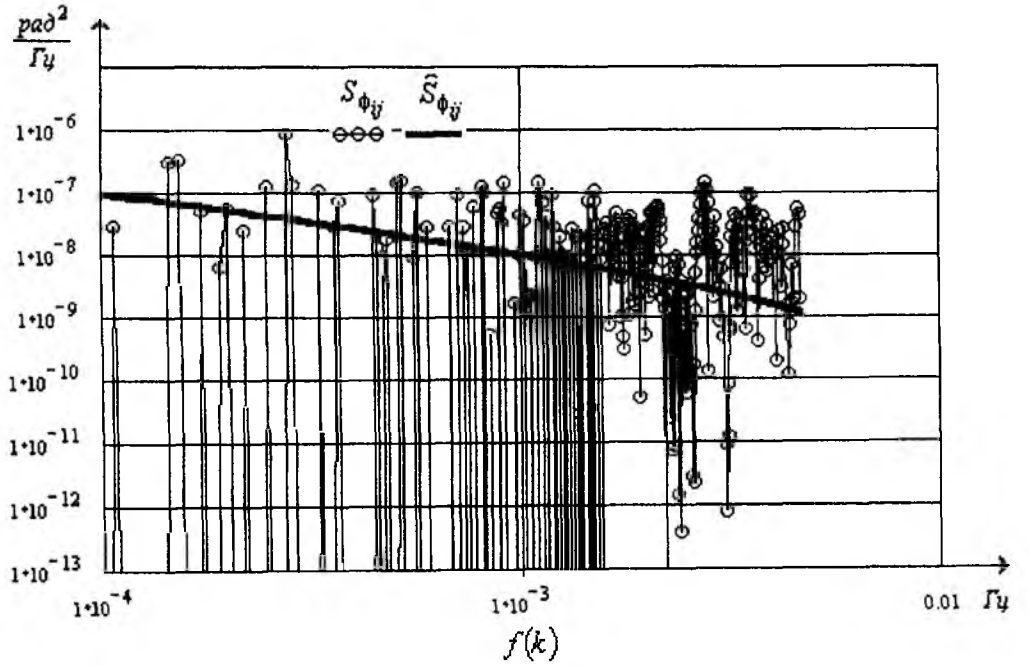



Рис. 1

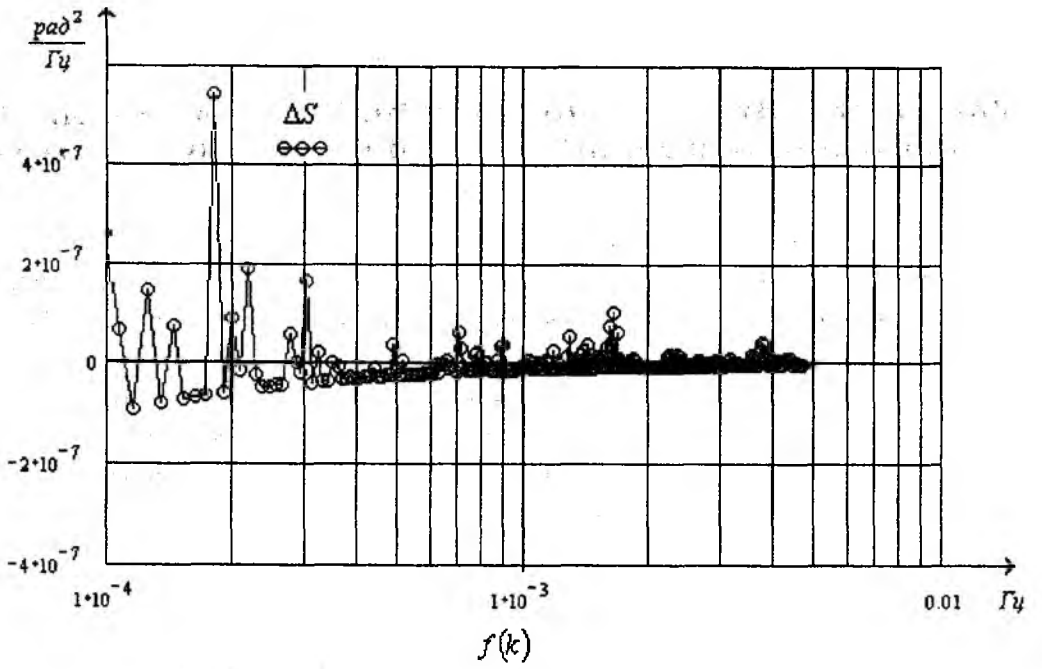



Рис. 2

На основании анализа гистограммы остатков (рис. 3) можно предположить, что плотность вероятности остатка $p(\omega,\Delta S)$ энергетического спектра $\Delta S(f_k)$ близка к нормальному закону распределения. Поэтому данную плотность вероятности предлагается аппроксимировать нормальным законом. В качестве критерия близости плотности вероятности $p(\omega,\Delta S)$ к нормальному закону берется минимум некоторого интегрального расстояния [5]. Наиболее удобным является критерий Кульбака [5], обеспечивающий минимум потери информации за счет аппроксимации. По критерию Кульбака, наилучшая интегральная аппроксимация произвольного распределения $p(\omega,\Delta S)$ нормальным законом достигается приравниванием их математических ожиданий и дисперсий [5]. Анализ погрешностей оценки фоновой части СПМФФ $\hat{S}_{\phi_{ij}}(f(k))$ и оценки с максимальным разрешением по частоте $S_{\phi_{ij}}(f(k))$ позволяет утверждать, что математическое ожидание остатка

$$\hat{E}\{\Delta S(f(k))\}=\frac{1}{M}\sum_{k=1}^{M}\left(S_{\phi_{ij}}(f(k))-\hat{S}_{\phi_{ij}}(f(k))\right)\approx 0,$$

где $M$ - максимальное количество спектральных линий СПМФФ, определяется из условия $M=enter\left[(\tau\cdot\omega_c/2\pi+1)/2\right]$, где $\omega_c$ - частота среза фильтра устройства измерения (компаратора); $\tau$ - интервал измерения.

В качестве оценки дисперсии для интегральной аппроксимации распределения остатка $p(\omega,\Delta S)$ используется

$$\hat{D}\{\Delta S((f(k)))\}=\frac{1}{M-1}\sum_{k=1}^{M}\left(\left(S_{\phi_{ij}}(f(k))-\hat{S}_{\phi_{ij}}(f(k))\right)-\hat{E}\{\Delta S(f(k))\}\right)^2.$$

На рис.3 представлен результат интегральной аппроксимации гистограммы по критерию Кульбака (-∘-∘-∘-).

Порог обнаружения определяется исходя из задаваемой доверительной вероятности для нормального закона распределения. Так, например, (см. рис. 4) спектральные составляющие, превосходящие $\hat{S}_{\phi_{ij}}(f(k))$ на величину, равную $3\cdot\sqrt{\hat{D}\{\Delta S(f(k))\}}$, с доверительной вероятностью 0.99 можно идентифицировать как спектральные линии регулярных перио-

дических составляющих девиации фазы. Критерий «три сигма», определяющий энергетику спектра, является необходимым, но не достаточным.

Строго говоря, никакие статистические методы анализа не позволяют с полной уверенностью доказать наличие периодических компонент в конечной реализации случайного процесса. Поэтому статистические тесты целесообразно сочетать с априорными данными и физическими соображениями [6].

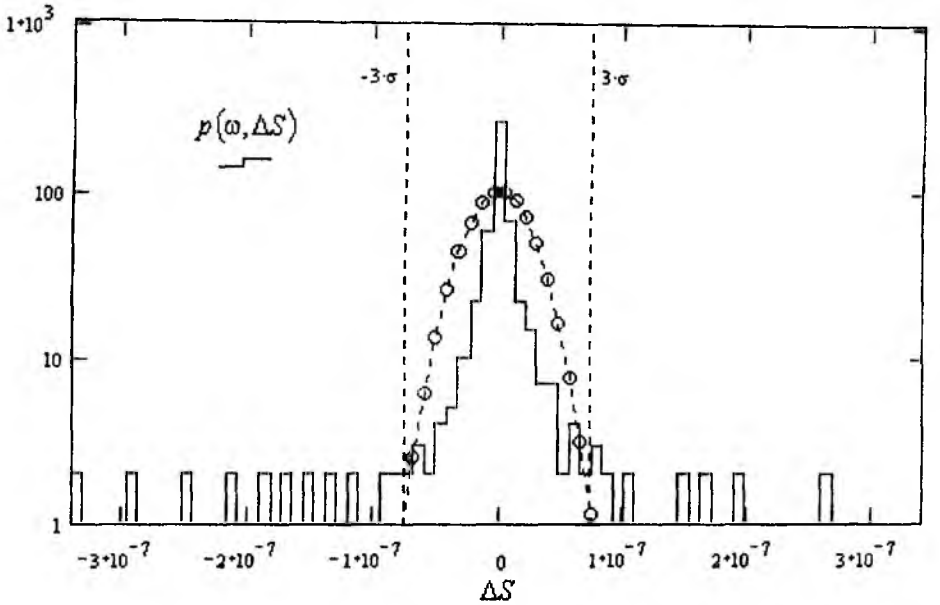



Рис. 3

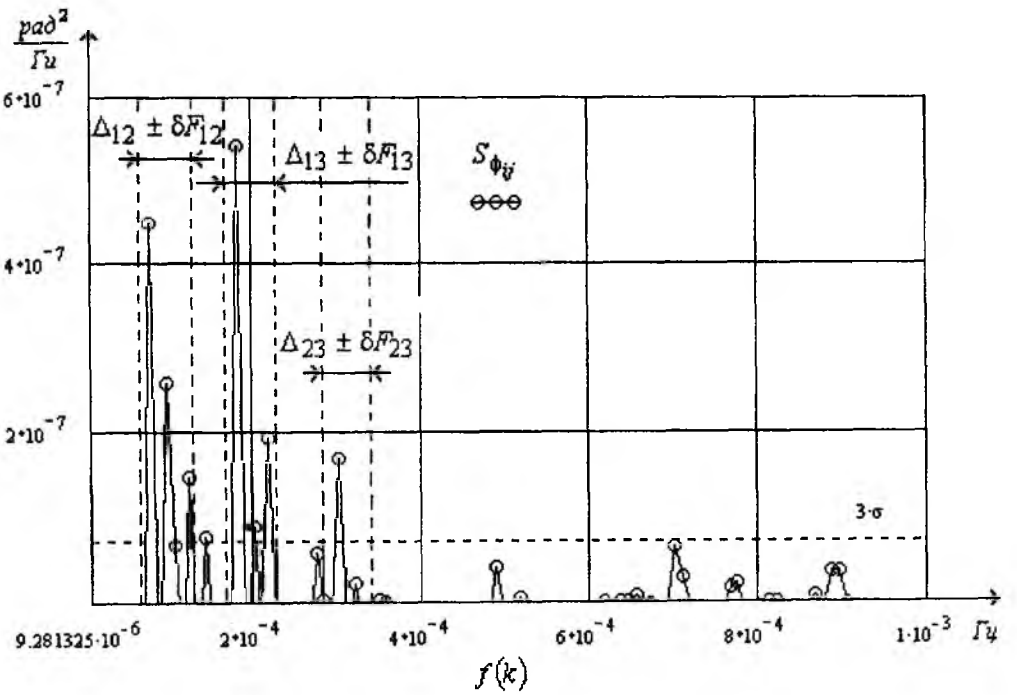



Рис. 4

Известно [7], что частоты модуляции фаз выходных сигналов мер частоты, обусловленные биениями, в первом приближений равны разностным частотам $\Delta_{ij} = |F_i - F_j|$, где $F_i$ - собственная частота $i$ - той меры. Поэтому частоты, выделенные по критерию "три

сигма" элементов вектора остатков, сравниваются со значениями $\Delta_{ij}$ и оставляются только те, для которых выполняются условия $\Delta_{ij} \approx f(k)$, где $k$ - индекс выделенного элемента. Однако определение частоты $\Delta_{ij}$ на конечном интервале наблюдения возможно лишь с некоторой погрешностью $\delta F_{ij}$, зависящей от длительности данного интервала [7]. Поэтому необходимо задавать диапазон частот $\Delta_{ij} \pm \delta F_{ij}$, в котором надлежит искать «яркую» спектральную линию, обусловленную частотным биением. В работе [7] показано, как связаны между собой погрешность определения центральной частоты $\delta F_{ij}$ с относительным временем наблюдения $\Delta_{ij} N\tau$, где $N$ - количество измерений в выборке. Данная погрешность для нашего случая будет определять необходимый диапазон частот $\Delta_{ij} \pm \delta F_{ij}$.

Выделенные с помощью данного критерия "яркие" спектральные линии являются исходными данными для определения амплитуд модулирующих сигналов, обусловленных биениями мер частоты. При этом можно утверждать, что амплитуда модулирующего сигнала равна корню квадратному из $\Delta S(\nu_k)$, где $\nu_k$ - значение частоты из диапазона $\Delta_{ij} \pm \delta F_{ij}$, при котором функция $\Delta S(f_k)$ имеет максимальное значение. При этом погрешность определения значения $\Delta S(\nu_k)$ будет определятся погрешностью оценки СПМФФ с максимальным разрешением по частоте. При использовании дисперсии Адамара с различным числом выборок для определения СПМФФ с максимальным разрешением по частоте [4] основными видами погрешностей определения энергетического спектра с помощью изложенной процедуры будут: погрешность квантования при измерении частотных флуктуаций $\bar{y}_{ij}$, погрешность, обусловленная неидеальностью АЧХ перестраиваемых полосовых фильтров, ограничивающих полосу анализа энергетического спектра фазовых флуктуаций, и погрешность, обусловленная неидентичностью полос пропускания эквивалентных фильтров Адамара для разных значений времени $\tau_k$ измерения $\bar{y}_{ij}$.

Таким образом, получен критерий выявления значимых квазипериодических составляющих, позволяющий получить оценки амплитуды и частоты модулирующего сигнала на основе анализа СПМФФ, полученной по результатам парных сличений мер частоты, входящих в состав группы. Данный критерий может найти применение в разрабатываемых алгоритмах формирования групповых шкал времени и частоты, учитывающих квазипериодические составляющие, которые существенно влияют на точность формирования шкалы времени.

**Список литературы**: 1. *Евдокименко Ю. И., Нарежний А. П.* Анализ стохастических характеристик группы мер частоты и времени при их взаимных сличениях // Радиоэлектроника и информатика. 1998. №3(04). С.44-46. 2. *Евдокименко Ю.И., Нарежний А.П.* Идентификация группового эталона частоты стохастической моделью системы связанных осцилляторов / Системи обробки інформації. Зб. наук. праць. Вип. 2(18). Харків: НАНУ, ПАНМ, ХВУ. 2002. С. 126-131. 3. *Нарежний А.П.* Выделение скрытых периодичностей в фазовых флуктуациях прецизионных мер частоты / Системи обробки інформації. Зб. наук. праць. Вип. 1(7). Харків: НАНУ, ПАНМ, ХВУ. 2000. С. 29-32. 4. *Евдокименко Ю. И.* Измерение энергетического спектра фазовых флуктуаций мер частоты вблизи несущей // Український метрологічний журнал. 2000. №4. С.41-43. 5. *Тихонов В. И, Харисов В. Н.* Статистический анализ и синтез радиотехнических устройств и систем. М.: Радио и связь, 1991. 608 с. 6. *Бендат Дж., Пирсол А.* Прикладной анализ случайных данных: Пер. с англ. М.: Мир, 1989. 540 с. 7. *Євдокименко Ю.І., Хижняк В.В.* Про систематичну складову похибки визначення дійсного значення частоти методом звірень за допомогою компаратора / Праці інституту електродинаміки НАНУ. Енергоефективність: Зб. наук. пр. Київ: ІЕД НАН України, 2000. С.213-216.

*Научный метрологический центр*
*Военных Эталонов*

*Поступила в редколлегию* 25.04.2002

УДК 681.04

*В.А.КРАСНОБАЕВ, д-р техн. наук*

# МЕТОД И АЛГОРИТМЫ КОРРЕКЦИИ ОШИБОК В СИСТЕМАХ ЦИФРОВОЙ ОБРАБОТКИ ИНФОРМАЦИИ

Известно [1-5], что использование непозиционной системы счисления в системе остаточных классов ( СОК ) позволяет улучшить такие характеристики систем цифровой обработки информации (СЦОИ), как пользовательская производительность, надежность и отказоустойчивость при решении определенного класса задач (для определенного типа операций). Как правило, в известных работах [6-7] рассматриваются коды в СОК со взаимно попарно простыми основаниями (модулями). Однако, если ставится задача минимизации времени коррекции ошибок, то в этом случае целесообразно рассмотреть коды в СОК со взаимно непростыми основаниями.

В предлагаемой статье рассматривается метод и алгоритмы коррекции ошибок в СЦОИ посредством применения кодов в СОК со взаимно непростыми модулями.

В СОК произвольный операнд А представляется в виде набора остатков $\{a_i\}$ от последовательного деления его на совокупность $\{m_i\}$ натуральных чисел, т.е. число А представится как $A = [A(\mathrm{mod}\ m_1), A(\mathrm{mod}\ m_2),\ldots, A(\mathrm{mod}\ m_n)]$ или $A = (a_1, a_2,..., a_n)$, где $a_i \equiv A(\mathrm{mod}\ m_i)$. Диапазон представимых чисел представится в виде ( 0, M ], где $M = [\ m_1, m_2,\ldots m_n\ ]$ – НОК оснований СОК.

Для последующего доказательства теорем, на основании которых базируется метод коррекции ошибок, рассмотрим следующую лемму [8].

**Лемма.** Для любого целого числа А, заданного в СОК с основаниями $\{m_i\}$, $i = \overline{1,n}$ и для любой пары оснований $m_i$, $m_j$ ( $i, j = \overline{1,n}$; $i \neq j$) должно выполнятся следующее условие:
$(a_i - a_j) \bmod d_{ij} = 0$, где $d_{ij} = (m_i, m_j)$ – НОД оснований $m_i$ и $m_j$.

Используя результаты леммы сформулируем и докажем следующую теорему, которая определяет необходимые условия обнаружения ошибок в СОК.

**Теорема 1.** Для обнаружения ошибки в остатке по произвольному основанию $m_i$ числа А, заданного в СОК совокупностью оснований $\{m_i\}$, $i = \overline{1,n}$, необходимо, чтобы произвольное основание $m_i$ имело хотя бы один, отличный от единицы, общий делитель с каждым из остальных оснований $m_j$ ( $j = \overline{1,n}$; $j \neq i$).

Доказательство. Обозначим НОД оснований $m_i$ и $m_j$ как $d_{ij} = (m_i, m_j)$ для $i, j = \overline{1,n}$ и $j \neq i$. Пусть ошибка произошла по модулю $m_i$, т.е. $\tilde{a}_i = (a_i + \Delta a_i) \bmod m_i$.

Вначале покажем, что выражение $(\tilde{a}_i - \Delta a_i) \bmod d_{ij}$ (1) эквивалентно выражению $\Delta a_j \pmod{d_{ij}}$. Действительно, в соответствии с результатами вышеприведенной леммы имеем $(a_i + a_j) \bmod d_{ij} \equiv 0$. Запишем выражение $\tilde{a}_i = (a_i + \Delta a_i) \bmod m_i$ в следующем виде:

$$a_i + \Delta a_i = m \cdot m + \tilde{a}_i, \qquad (2)$$

где m – целое число. Из выражения (2) определим значение искаженного остатка $\tilde{a}_i = a_i + \Delta a_i - m \cdot m_i$ и, подставив его в выражение (1), получим:

$$(\widetilde{a}_i - a_j) \bmod d_{ij} = [(a_i + \Delta a_i - m \cdot m_i) - a_j] \bmod d_{ij} =$$
$$= [(a_i - a_j) + (-m \cdot m_i + \Delta a_i)] \bmod d_{ij} = \quad (3)$$
$$= [(a_i - a_j) + (-m \cdot k \cdot d_{ij} + \Delta a_i)] \bmod d_{ij},$$

где: $m_i = k \cdot d_{ij}$ (по условию леммы); k – натуральное число.

Рассматривая каждое из слагаемых выражений (3), получим: $(a_i - a_j) \bmod d_{ij} \equiv 0$ (согласно лемме) и также $(m \cdot k \cdot d_{ij}) \bmod d_{ij} \equiv 0$. В этом случае

$$\Delta a \equiv (\widetilde{a}_i - a_j) \bmod d_{ij}. \quad (4)$$

Анализируя формулу (4) очевидно, что при отсутствии для модулей $m_i$ и $m_j$ общих делителей больше единицы выполняется условие $\Delta a_i \pmod{d_{ij}} \equiv 0$. Таким образом необходимым условием обнаружения ошибки по одному из оснований является выполнение условия

$$d_{ij} = (m_i, m_j) \neq 1. \quad (5)$$

Необходимое условие (5) теоремы 1 является и достаточным, если величина $\Delta a_i \equiv (\widetilde{a}_i - a) \bmod d_{ij}$ не кратна одновременно двум делителям $d_{i-1,i}$ и $d_{i,i+1}$, т.е.

$$\text{НОД } d_{\Delta a_i}^{(i-1)} = (d_{i-1,i}, \Delta a_i) = 1 \text{ и НОД } d_{\Delta a_i}^{(i+1)} = (d_{i,i+1}, \Delta a_i) = 1. \quad (6)$$

Для исправления ошибок в СОК воспользуемся доказательством следующей теоремы [9].

**Теорема 2.** Для исправления ошибки в остатке по произвольному основанию $m_i$ $(i = \overline{1,n})$ числа A, заданного в СОК основаниями $\{m_i\}$, необходимо выполнение следующего условия:

$$(d_{ik} - 1) \cdot (d_{ij} - 1) \geq \delta(\Delta a_i), \quad (7)$$

где $\delta(\Delta a_i) = m_i - 1 - (K_{d_{ik}} + K_{d_{ij}} - K_{[d_{ik}, d_{ij}]})$.

При этом :

$K_{d_{ik}}$ - число возможных делителей значения ошибки $\Delta a_i$ по основанию $m_i$, кратных значению $d_{ik}$;

$K_{d_{ij}}$ - число возможных делителей значения ошибки $\Delta a_i$ по основанию $m_i$, кратных значению $d_{ij}$;

$K_{[d_{ik}, d_{ij}]}$ - число возможных делителей ошибки $\Delta a_i$ по основанию $m_i$, кратных значению НОК чисел $d_{ik}$, $d_{ij}$.

Условие (7) теоремы 2 является и достаточным, если различным возможным значениям $\delta(\Delta a_i)$ ошибок соответствуют различные пары величин $a_{ik}$, $a_{ij}$.

На основании вышеприведенных теорем построим алгоритмы обнаружения и исправления однократных (по одному из оснований СОК) ошибок.

### Алгоритмы обнаружения ошибок

***Алгоритм 1.*** Пусть необходимо проверить факт наличия либо отсутствия ошибок в операнде $A = (a_1, a_2, \ldots, a_n)$.

1.Определим следующую совокупность значений:

$$\begin{cases} a_{12} = (a_1 - a_2) \bmod d_{12}, \\ a_{13} = (a_1 - a_3) \bmod d_{13}, \\ \quad \cdot \quad \cdot \quad \cdot \\ a_{1n} = (a_1 - a_n) \bmod d_{1n}. \end{cases} \tag{8}$$

Если вся совокупность (8) значений $a_{1i} = 0$ $(i = \overline{1,n}; i \neq 1)$ равна нулю, то далее вычисляется и проверяется совокупность значений $a_{2i} = (a_2 - a_1) \bmod d_{2i} (i \neq 2)$ и т.д.

2.При получении всех возможных значений $a_{ij}$ $(j \neq j)$ составляем матрицу G вида

$$G = \begin{vmatrix} a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{23} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}.$$

При составлении матрицы G нет необходимости указывать истинное числовое значение элементов $a_{ij}$, а достаточно представить элементы матрицы в виде единицы или нуля, т.е.

$$a_{ij} = \begin{cases} 0, \text{при } a_{ij} = 0, \\ 1, \text{при } a_{ij} \neq 0. \end{cases}$$

3.Если определитель $|G| = 0$, то А считается неискаженным, а если $|G| \neq 0$ - число А искажено.

Если воспользоваться соотношением

$$a_i - a_j \equiv [d_{ij} - (a_j - a_i)] \bmod d_{ij},$$

то для определения правильности или неправильности операнда А достаточно определить лишь следующую совокупность чисел:

$$a_{12}, a_{23}, a_{34}, \dots, a_{n-1n}, a_{n1}.$$

Исходя из вышеизложенного очевидно, что посредством предлагаемого метода кодирования информации в СОК исключительно просто реализуется процесс обнаружения ошибок. При этом время обнаружения ошибок, сравнительно с обнаружением ошибок в позиционной двоичной системе счисления, достаточно мало и постоянно для любого числа оснований СОК [10].

***Алгоритм 2.*** Приведем некоторые соображения, позволяющие упростить вышеприведенный алгоритм обнаружения ошибок.

Покажем, что

$$[(a_1 + \overline{a_j}) + (\overline{a_1} + a_j)] \equiv 0 \pmod{d_{1j}},$$

где: $\bar{a}_j = m_j - a_j$; $\bar{a}_1 = m_1 - a_1$.

Пусть в операнде $A = (a_1, a_2, ..., a_j, ..., a_n)$ искажен остаток $a_j$ по основанию $m_j$, т.е. $\tilde{a}_j = (a_j + \Delta a_j) \bmod m_j$. Запишем систему следующих равенств:

$$\begin{cases} K_1 = a_i - \tilde{a}_j = a_i + (m_j - \tilde{a}_j) = a_i + m_j - a_j - \Delta a_j; \\ K_2 = \tilde{a}_j - a_i = a_j + \Delta a_j - a_i = a_j - a_i + \Delta a_j. \end{cases}$$

Сложив эти два равенства, получим:

$$K_1 + K_2 = m_j \text{ или } K_1 + K_2 \equiv 0 \pmod{m_j}.$$

Таким образом очевидно, что

$$a_1 + \bar{a}_j = d_{1j} - (\bar{a}_1 + a_j),$$

т.е. для определения факта наличия или отсутствия ошибок нет необходимости вычисления точного значения величины $(a_1 + \bar{a}_j) \bmod d_{1j}$, а достаточно знать факт равенства или неравенства этого значения нулю. Это в свою очередь позволит в техническом устройстве для обнаружения ошибок в СОК [11] вместо (n – 1) сумматоров по модулям $m_j$ ($j = \overline{2, n}$), которые определяют совокупность значений $\bar{a}_j = m_j - a_j$, использовать всего один сумматор по модулю $m_1$, определяющий значение $\bar{a}_1 = m_1 - a_1$.

**Алгоритм обнаружения и исправления ошибок**

1. Определим все возможные значения вида $a_{i,i+1} = (a_1 - a_{i+1}) \bmod d_{i,i+1}$ :

$$\begin{cases} a_{12} = (a_1 - a_2) \bmod d_{12}, \\ a_{23} = (a_2 - a_3) \bmod d_{23}, \\ \quad \cdots \\ a_{n-1,n} = (a_{n-1} - a_n) \bmod d_{n-1,n}, \\ a_{n1} = (a_n - a_1) \bmod d_{n1}. \end{cases} \qquad (9)$$

2. Если вся совокупность (9) значений принимает нулевое значение, то ошибка отсутствует либо она кратна одному из делителей $d_{i-1,i}, d_{i,i+1}$, что противоречит условию ограничения класса возможных корректируемых ошибок. Таким образом считается, что ошибка отсутствует.

3. Если одновременно выполняются условия $a_{i-1,i} \neq 0$ и $a_{i,i+1} \neq 0$, а все остальные значения в совокупности (9) принимают значения ноль, то считается, что ошибка имеется в остатке по модулю $m_i$, т.е.

$$\tilde{a}_i = (a_i + \Delta a_i) \bmod m_i \, (0 < \Delta a_i \leq m_{i-1}).$$

4. В соответствии с известным алгоритмом производится коррекция ошибок по i–му основанию СОК.

В соответствии с данным алгоритмом разработано устройство для его реализации [12]. Кратко рассмотрим алгоритм его функционирования (см. рис. 1).

Пусть задана СОК основаниями $m_1 = 4$, $m_2 = 6$, $m_3 = 12$. Возможная совокупность кодовых слов представлена в табл. 1 (т.е. НОК значений 4,6 и 12 равно $M = [4, 6, 12] = 12$). Определим следующее значение НОД: $d_{12} = (4, 6) = 2$; $d_{23} = (6, 12) = 6$; $d_{31} = (4, 12) = = 4$; $\delta(\Delta a_1) = 2$ (табл. 2); $\delta(\Delta a_2) = 3$ (табл. 3); $\delta(\Delta a_3) = 8$ (табл. 4).

Пусть искажено правильное число A = (11, 001, 0111) по основанию $m_2$, т.е. на вход устройства (см. рис. 1) подается операнд вида $\widetilde{A}$ = (11, 100, 0111). Необходимо определить правильность числа $\widetilde{A}$ и при необходимости провести его коррекцию.

Операнд $\widetilde{A}$ записывается в первый $P_г1$ и второй $P_г2$ входные регистры. На выходах соответствующих сумматоров $1_1 \div 1_3$ по модулям $m_1 \div m_3$ получим инвертированные значения соответствующих остатков $\bar{a}_i = m_i - a_i (i = \overline{1,3})$. На выходах сумматоров $2_1 \div 2_3$ получим такие значения: $2_1 - (a_3 + \bar{a}_1) \bmod d_{31} = 0000$; $2_2 - (a_1 + \bar{a}_2) \bmod d_{12} = 001$; $2_3 - (a_2 + \bar{a}_3) \bmod d_{23} = 0011$, которые через соответствующие дешифраторы $ДШ_1 - ДШ_3$ в унитарном коде поступают на соответствующие входы коммутаторов $К_1 \div К_3$, которые реализуют значения $\delta(\Delta a_i)$ соответствующих таблиц.

**Таблица 1**

| A в ПСС | Кодовые числа | | |
|---|---|---|---|
| | A в СОК | | |
| | $m_1$ | $m_2$ | $m_3$ |
| 0000 | 00 | 000 | 0000 |
| 0001 | 01 | 001 | 0001 |
| 0010 | 10 | 010 | 0010 |
| 0011 | 11 | 011 | 0011 |
| 0100 | 00 | 100 | 0100 |
| 0101 | 01 | 101 | 0101 |
| 0110 | 10 | 000 | 0110 |
| 0111 | 11 | 001 | 0111 |
| 1000 | 00 | 010 | 1000 |
| 1001 | 01 | 011 | 1001 |
| 1010 | 10 | 100 | 1010 |
| 1011 | 11 | 101 | 1011 |

**Таблица 2**

| $a_{31}$ | $a_{12} = 1$ |
|---|---|
| 1 | $\Delta\bar{a}_1 = 1$ |
| 2 | — |
| 3 | $\Delta\bar{a}_1 = 3$ |

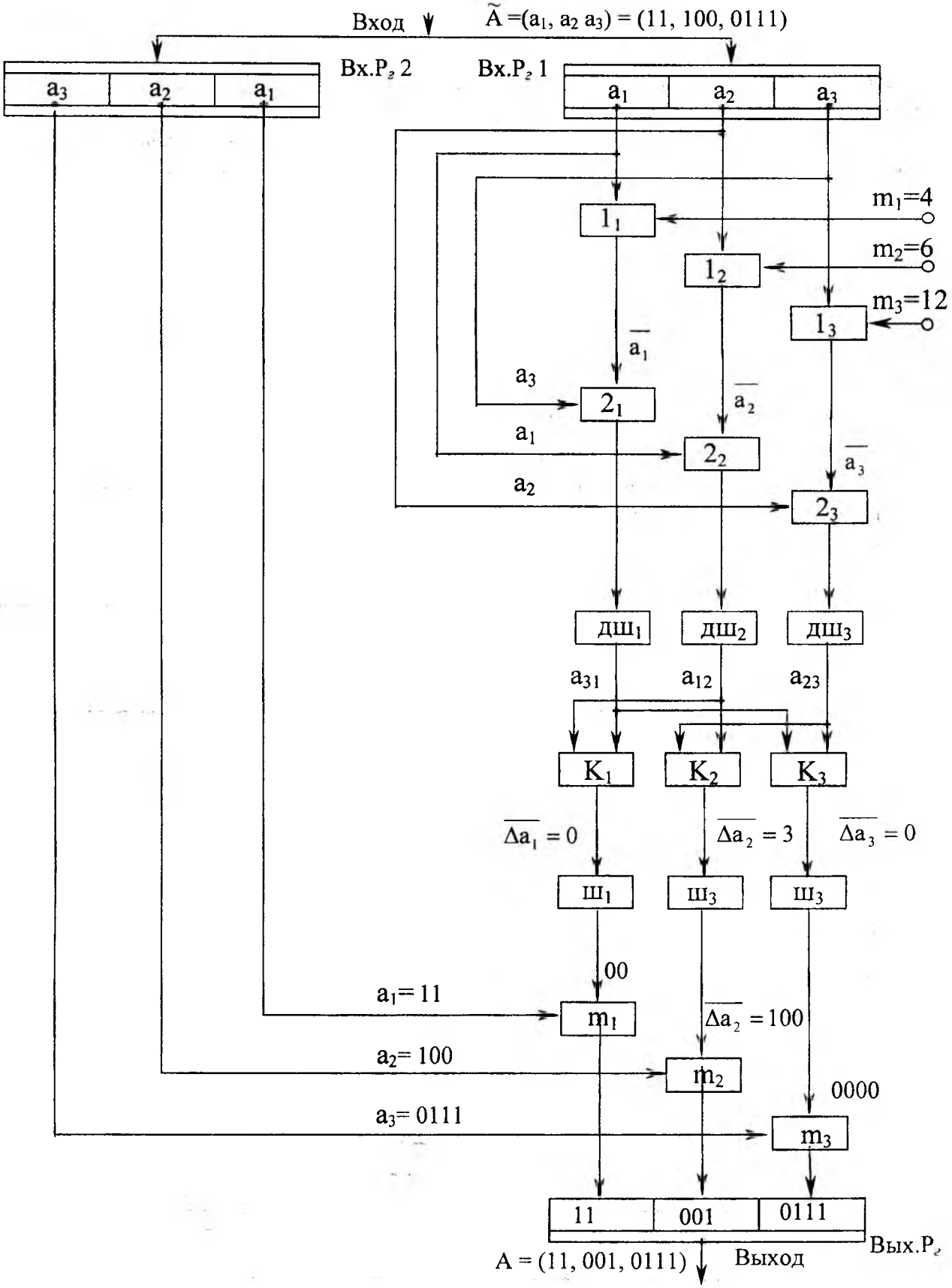

Вход
$\tilde{A}=(a_1, a_2\, a_3)=(11, 100, 0111)$
Вх.$P_г$ 2
Вх.$P_г$ 1
$a_3$ $a_2$ $a_1$
$a_1$ $a_2$ $a_3$
$1_1$
$1_2$
$1_3$
$m_1=4$
$m_2=6$
$m_3=12$
$\overline{a_1}$
$\overline{a_2}$
$\overline{a_3}$
$a_3$
$a_1$
$a_2$
$2_1$
$2_2$
$2_3$
дш$_1$ дш$_2$ дш$_3$
$a_{31}$ $a_{12}$ $a_{23}$
$K_1$ $K_2$ $K_3$
$\overline{\Delta a_1}=0$
$\overline{\Delta a_2}=3$
$\overline{\Delta a_3}=0$
ш$_1$ ш$_3$ ш$_3$
00
$a_1$= 11
$a_2$= 100
$a_3$= 0111
$m_1$
$m_2$
$m_3$
$\overline{\Delta a_2}=100$
0000
11 001 0111
Вых.$P_г$
A = (11, 001, 0111)
Выход

Рис.1

**Таблица 3**

| $a_{23}$ | $a_{12}=1$ |
|---|---|
| 1 | $\Delta\bar{a}_2=5$ |
| 2 | — |
| 3 | $\Delta\bar{a}_2=3$ |
| 4 | — |
| 5 | $\Delta\bar{a}_2=1$ |

**Таблица 4**

| $a_{31}$ | $a_{23}$ | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1 | $\Delta\bar{a}_3=7$ | — | $\Delta\bar{a}_3=3$ | — | $\Delta\bar{a}_3=11$ |
| 2 | — | $\Delta\bar{a}_3=2$ | — | $\Delta\bar{a}_3=10$ | — |
| 3 | $\Delta\bar{a}_3=1$ | — | $\Delta\bar{a}_3=9$ | — | $\Delta\bar{a}_3=5$ |

(табл. 2 - табл. 4) значений ошибок. В этом случае только на выходе второго (табл. 3) коммутатора $К_2$ имеется ненулевое значение (это факт наличия ошибки в операнде $\tilde{A}$), которое поступает через шифратор $Ш_2$ (который преобразует его в двоичный код) на первый вход сумматора по модулю $m_2$. На второй вход этого сумматора с выхода второго регистра Вх. $Р_г2$ поступает значение искаженного остатка $\bar{a}_2=100$. С выхода сумматора по модулю $m_2$ в выходной регистр (Вых. $Р_г2$) поступит значение исправленного остатка

$$(\tilde{a}_2+\overline{\Delta a_2})\bmod m_2=(a_2+\Delta a_2)+(m_2-\Delta a_2)=a_2(\bmod m_2)=001.$$

В этом случае в выходном регистре устройства содержится исправленный операнд $A=(11,001,0111)$.

Таким образом, в статье предложен метод коррекции ошибок, основанный на использовании непозиционных кодовых структур в СОК. В соответствии с рассмотренным методом разработаны алгоритмы обнаружения и исправления ошибок в СЦОИ, на основании которых разработаны патентоспособные технические устройства. Использование этих устройств в СЦОИ показали высокую эффективность применения непозиционного кодирование информации в СОК.

**Список литературы:** 1. *Акушский И.Я., Юдицкий Д.И.* Машинная арифметика в остаточных классах. М.: Сов. радио, 968. 440 с. 2. *Блейхут Р.* Быстрые алгоритмы цифровой обработки сигналов: Пер. с англ. М.: Мир, 1989. 448 с. 3. *Кравченко В.Ф., Крот А.М.* Методы и микроэлектронные средства цифровой фильтрации сигналов и изображений на основе теоретико – числовых преобразований // Зарубежная радиоэлектроника. Успехи современной радиоэлектроники. 1997. №6. С. 3-31. 4. *Червяков Н.И., Тынчеров К.Т., Велигоша А.В.* Высокоскоростная цифровая обработка сигналов с использованием непозиционной арифметики // Радиотехника. 1997. №10. С. 23-27. 5. *Лавриненко Д.И.* Применение быстрого преобразования Фурье в криптографических преобразователях // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2000. Вып. 114. с. 75-79. 6. *Краснобаев В.А.* Основы создания вычислителей на основе остаточных классов // Системи обробки інформації. Харків:НАНУ, ПАНМ, ХВУ. 2001. Вип. 1(11). с. 3-7. 7. *Краснобаев В.А.* Методы реализации модульных операций в системах цифровой обработки информации // Радиотехника: Всеукр. межвед. сб. 2001. Вып. 119. с. 130-134. 8. *Торгашев В.А.* Система остаточных классов и надежность ЦВМ. М.: Сов. радио, 1973. 118 с. 9. *Краснобаев В.А.* Техническая реализация метода коррекции ошибок в системе остаточных классов // АСУ и приборы автоматики. 1987. Вып. 81. С. 97 – 101. 10. А.с. 964645 СССР. Устройство для обнаружения одиночных ошибок кода в системе остаточных классов / *В.А. Краснобаев, А.И. Бецков, Г.И. Бороденко и др.* // Бюл. изобрет. 1982. №37. С. 184. 11. А. с. 1013957 СССР. Устройство для обнаружения ошибок в системе остаточных классов / *В.А. Краснобаев, И.Б. Давыдов* // Бюл. изобрет. 1983. №15. С. 210. 12. А. с. 1166117 СССР. Устройство для контроля информации в системе остаточных классов / *В.А. Краснобаев* // Бюл. изобрет. 1985. №25. С. 124.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 25.04.2002*

УДК 681.3.07

*І.Д. ГОРБЕНКО, д-р техн. наук, А.В. ПОТІЙ, канд. техн. наук,*
*О.А. МЕЛЬНИКОВА, канд.техн. наук*

# ДОСВІД ПІДГОТОВКИ ТА РЕЗУЛЬТАТИ ПРОВЕДЕННЯ IV ВСЕУКРАЇНСЬКОЇ ОЛІМПІАДИ СЕРЕД БАКАЛАВРІВ, СТУДЕНТІВ ТА МАГІСТРІВ З НАПРЯМКУ «ІНФОРМАЦІЙНА БЕЗПЕКА»

Згідно з наказом Міністерства освіти і науки України № 800 від 13.12.2001 р. "Про проведення Всеукраїнської студентської олімпіади 2001 - 2002" на базі Харківського Національного університету радіоелектроніки у квітні 2002 року проводився другий етап всеукраїнської студентської олімпіади з напрямку "Інформаційна безпека" серед студентів III - V курсів університетів та технічних ВУЗів України. Олімпіада проводилася по двом рівням складності. Перший рівень — для студентів III - V курсів та магістрів зі спеціальностей напрямку "Інформаційна безпека", другий рівень — для студентів III - V курсів та магістрів, що вивчають загальноосвітні та прикладні дисципліни з захисту інформації.

Олімпіада проводилась в два тури. Перший тур був пов'язаний з розв'язком задач різної складності та реалізації однієї задачі в вигляді програмного продукту. Всього для розв'язку було запропоновано 7 теоретичних задач, 3 з яких по складності відповідають першому рівню, а 4 — другому рівню. За правильний розв'язок кожної задачі, в залежності від складності, призначалося від 5 до 12 балів. А також були запропоновані варіанти практичних завдань для програмної реалізації. При оцінці розробленої програми враховувалася як правильність її функціонування так і якість реалізації (наприклад, характеристики швидкості, пам'яті, повнота проведеного тестування і т.і.). Додатково на Олімпіаді проводився другий тур, який мав вигляд презентації та захисту власних проектів та розробок по напрямку "Інформаційна безпека", в тому числі статей, патентів, програмних продуктів, апаратних засобів тощо. Цей тур здійснювався в вигляді стендової комп'ютерної доповіді.

Цього року в олімпіаді приймали участь наступні ВУЗи: Національний технічний університет України "КПІ" (м. Київ) — 4 чол., Українська академія банківської справи Харківської філії (м. Харків) — 1 чол., Харківський національний політехнічний університет "ХПІ" — 2 чол., Сумський Державний університет (м. Суми) — 2 чол., Харківський інститут Військово - Повітряних сил — 5 чол., Волинський державний університет ім. Лесі Українки — 1 чол., Харківський національний університет радіоелектроніки — 9 чол., Національна юридична академія України ім. Ярослава Мудрого — 1 чол., Дніпропетровський національний горний університет України — 2 чол., Дніпропетровський національний університет фізико технічний інститут — 2 чол.

## 1. Конкурсні завдання

Цього року було змінено підхід до формування завдань та повністю оновлено набори конкурсних завдань. Для прикладу нижче наведемо використаний набір конкурсних завдань. Завдання 1 - 4 були призначені для напрямку "Інформаційна безпека", завдання 4 – 7 — для спеціальності "Захист інформації в системах та мережах".

*Завдання № 1* (12 балів). В системі захисту інформації для забезпечення цілісності використовується однонаправлена хеш - функція $H(M) = h$, де $M$ — інформація довільної довжини, а $h$ може приймати $n = 2^m$ значень. Скільки випадкових повідомлень $k$ необхідно подати на вхід перетворювача $H$, щоб з імовірністю $P_з$ відбулося хоча б одне співпадання вигляду $H(M_i) = H(M_j)$, тобто відбулася колізія. Визначити конкретне значення $k$ для відомої Вам хеш - функції, якщо $P_з = 0{,}5 + i \cdot 0{,}02$, де $i$ — номер реєстрації учасника олімпіади. При $i > 25$ $i := i \pmod{25}$.

*Завдання № 2* (12 балів). В стандарті США X9.30 цифровий підпис $< r, S >$ формується згідно наступних правил

$$\begin{cases} r = g^k \pmod P \bmod q \\ S = \left[k^{-1} \cdot \left(h(M) + x \cdot r\right)\right] \bmod q \end{cases} \quad (1)$$

Для повідомлення $[M', < r', S' >]$ перевірка підпису здійснюється в наступній послідовності

$$W = (S')^{-1} \bmod q \quad (2)$$

$$U_1 = [H(M') \cdot W] \bmod q \quad (3)$$

$$U_2 = (r') \cdot W \bmod q \quad (4)$$

$$V = \left[\left(g^{U_1} \cdot y^{U_2}\right) \bmod p\right] \bmod q \quad (5)$$

де $x$ — особистий ключ цифрового підпису, $g$ — первісний елемент, $P$, $q$ — великі прості числа, причому $q$ входить в канонічний розклад $P$ - 1 як співмножник, $H(M)$ — функція хешування, $y$ — відкритий ключ. Доведіть, що повідомлення $M'$ буде цілісним та справжнім, якщо $V = r$.

Визначте складність криптоаналізу методом розв'язку дискретного логарифмічного порівняння, якщо $P_i = 2^{512+i\cdot 64}$, $i := i \pmod 8$, де $i$ — номер реєстрації учасника олімпіади.

Як та на яку величину зміниться стійкість, якщо замість DSA використати ECDSA (підпис на еліптичних кривих) з порядком базової точки $G$ дорівнюючим $n_i = P_i$.

*Завдання № 3* (12 балів). Монохроматичний монітор ПЕОМ з діагоналлю 17′ підтримує (забезпечує) 1024·168 елементів зображення. Частота кадрів 80 Гц. Амплітуда імпульсів вихідного каскаду, що подаються на катод електронної променевої трубки (ЕПТ) 200 В, 10 мА. Зворотний хід рядкової та кадрової розгортки займає 18% та 8% відповідних періодів. Необхідно виконати наступні підпункти завдання.

– Намалювати (якісно) вигляд спектру амплітуд сигналу у колі монітора.

– Як буде змінюватись спектр амплітуд сигналу у колі при зміні фронтів імпульсу від прямокутного до косинус - квадратичного з різною протяжністю ? Навести на малюнку якісні зміни. Як це буде впливати на фактори ТЗІ ?

– Як зміниться спектр при застосуванні монітора з діагоналлю 21′ ?

– Які елементи конструкції монітора впливають на інтенсивність випромінювання сигналу монітора.

*Завдання № 4* (12 балів). У федеральному стандарті симетричного шифрування США 21 століття FIPS - 197 для перемішування та розсіювання використовуються два табличні перетворення: заміна байтів та зсув байтів рядків.

Заміна байтів здійснюється спочатку послідовною заміною кожного $a_{ij}$ байту стану на зворотній $a_{ij}^{-1}$ згідно порівняння

$$a_{ij} \cdot a_{ij}^{-1} \equiv 1 \left(\bmod x^8 + x^4 + x^3 + x + 1\right). \quad (6)$$

Потім кожен $a_{ij}^{-1} = x$ байт замінюється на $Y$ згідно ниженаведенного афінного перетворення

$$Y = \begin{vmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{vmatrix} \oplus \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix} \quad (7)$$

Зсув рядків байтів плинного стану $A_V$

$$A_V = \begin{vmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{vmatrix} \quad (8)$$

здійснюється циклічно зліва направо згідно наступних правил: перший рядок без зсуву, другий на один байт, третій на 2 байти, четвертий на 3 байти. Необхідно виконати наступні пункти завдання.

– Довести, що перетворення типу заміна байтів та зсув рядків можна представити у вигляді підстановки.

– Знайти, на який елемент - байт в результаті перетворень буде змінено $a_{ij}$ байт, якщо $a_{ij} = 89 + 3 \cdot i + 1$, $i$ — номер реєстрації.

*Завдання № 5* (6 балів). Служба технічної розвідки перехопила випромінювання шифруючого засобу та визначила $2 \cdot m$ бітів двійкової гами зашифрування, зробивши при цьому одну помилку. Визначте по $2 \cdot m$ символам закон формування гами зашифрування та сформуйте наступні $2 \cdot m$ символів, якщо $2 \cdot m$ перехоплених з помилкою символів наведені.

*Примітка.* Проаналізувати $2 \cdot m$ символів, зробити виправлення помилки, скласти та розв'язати систему лінійних рівнянь $m$ - го порядку. Номер варіанту $R = i \pmod 8 + 1$, $i$ — номер реєстрації.

*Завдання № 6* (6 балів). У системі захисту інформації використовується направлене шифрування в групах точок еліптичної кривої (ЕК). Порядок базової точки $G$ дорівнює $n_i = 2^{256+i \cdot 32}$, де $i$ — номер реєстрації. Визначте необхідну довжину модуля RSA направленого шифрування при якій забезпечувалася б така ж стійкість, як і в групах точок ЕК, якщо потужність криптоаналітичної системи складає $10^9$ операцій додавання на ЕС, що становить 0,1% від потужності RSA криптоаналітичної системи.

*Завдання № 7* (6 балів). У виданому конкурсанту алгоритмі вірусу є дефект. Визначити в чому його суть та як його виправити.

Нижче наведені приклади варіантів завдань для програмної реалізації (від 1 до 12 балів в залежності від рівня та якості реалізації).

— Для усіх поліномів ступеню $m < 8$ програмно побудувати таблицю відповідних зворотних поліномів $h(x) = (g(x))^{-1} \bmod f(x)$ по заданому модулю $f(x)$ (наприклад, $f(x) = x^8 + x^4 + x^3 + x + 1$), використовуючи відповідні математичні методи.

— Для заданого модулю $f(x)$ (наприклад, $f(x)=x^8+x^4+x^3+x+1$, $m=8$) програмно знайти усі первісні елементи $g(x)$ поля $F(2^m)$, використовуючи необхідні та достатні умови первісності.

— Використовуючи заданий первісний елемент $g(x)$ (наприклад, $g(x)=x^4+1$) програмно побудувати усі елементи поля $F(2^m)$ при заданому модулю $f(x)$ (наприклад, $f(x)=x^8+x^4+x^3+x+1$, $m=8$).

**2. Результати олімпіади та аналіз рівня підготовки конкурсантів**

Всього в олімпіаді прийняла участь 31 особа із 10 вузів України (КПІ, ДнНУ, ХПУ, УАБС, СДУ, ХІВПС, ВДУ, ХНУРЕ, Національна юридична академія, НГУ). За результатами олімпіади місця розподілилися таким чином.

З напрямку "Інформаційна безпека" (командний залік):
1 місце – команда ХНУРЕ в складі: Балагура Д.С., Лавриненко С.С., Пінчук М.В. (каф. БІТ);
2 місце – команда НТУ "КПІ" в складі: Руденко О.В., Коваль А.О., Афонін О.О. (каф. ММЗІ);
3 місце – команда НГУ в складі: Калініченко В.П., Іванько А.М.;
4 місце – команда ХІВПС в складі: Сухов В.В., Бойко М.М., Єфіменко Я.В.

З напрямку "Інформаційна безпека" (особистий залік):
1 місце – Балагура Дмитро Сергійович (ХНУРЕ) – 52 бали;
2 місце – Руденко Олексій Володимирович (НТУ "КПІ") – 48 балів;
2 місце – Коваль Андрій Олегович (НТУ "КПІ") – 48 балів;
2 місце – Лавриненко Станіслав Сергійович (ХНУРЕ) – 48 балів.

З дисципліни "Захист інформації" (особистий залік):
1 місце – Вітріщенко Вікторія Анатоліївна (ХНУРЕ) – 29 балів;
2 місце – Перепияка Олена Миколаївна (УАБС, харківська філія) – 28 балів;
3 місце – Горбунов Олексій Вікторович (ХНУРЕ) – 24 бали;
3 місце – Серьга Михайло Олександрович (ДнНУ, ФТІ) – 24 бали.

З дисципліни "Програмне забезпечення захисту інформації":
1 місце – Афонін Олександр Олександрович (НТУ "КПІ") – 14 балів;
2 місце – Колибельников Олександр Іванович (ХНУРЕ) – 12 балів;
3 місце – Ілібман Володимир Григорович (НТУ "КПІ") – 12 балів.

За представленням особистих програмних засобів захисту інформації:
1 місце – Горбенко Юрій Іванович, Вервейко Віталій Миколайович за роботу "Система захисту інформації в корпоративній мережі" (ХНУРЕ);
2 місце - Єфіменко Ярослав Валерійович за роботу "Структура програмного забезпечення спеціалізованої АСУ" (ХІВПС);
3 місце – Іванько Артем Миколайович за роботу "Програмно – апаратний комплекс виявлення закладних пристроїв" (НГУ).

Крім того, за оригінальність розв'язку задач були нагороджені наступні учасники олімпіади: Горбунов О.В., Пінчук М.В., Рогова Л.С. (ХНУРЕ), Коваль А.О., Руденко О.В. (НТУ "КПІ"), Єфименко Я.В. (ХІВПС).

До основних недоліків у підготовці конкурсантів слід віднести наступне.

— При вивченні тем, наприклад цифрових підписів, студенти поверхово засвоюють алгоритми, не володіють уміннями доводити коректність алгоритмів цифрового підпису.

— Не всі студенти знають критерії та показники, що можуть бути застосовані при порівнянні різних алгоритмів цифрового підпису (наприклад в полях та групах точок еліптичних кривих).

— Студенти, що навчаються за іншими спеціальностями, не знають перспективні алгоритми криптозахисту (наприклад, алгоритм 21 століття RIJNDAEL, який затверджено в 2001 р. в США в якості державного стандарту).

— Недостатня, а то й слабка підготовки студентів з питань таких спеціальних розділів математики як теорія чисел, теорія груп взагалі та теорія еліптичних груп частково.
— Значна кількість студентів не може розробити модель розвідки та криптоаналізу, з помилками розв'язують системи рівнянь над полями GF(2).
— Деякі студенти не знають навіть проблемних питань випромінювання ПЕОМ взагалі та монітору частково.

Проведена олімпіада продемонструвала зросший рівень підготовленості магістрів та студентів з проблемних питань захисту інформації в напрямку "Інформаційна безпека", конкурсанти продемонстрували розуміння основних протиріч та орієнтацію в напрямках їх розв'язку. Добре підготовлені, в тому числі математично, магістри та студенти НТТУ "КПІ" та ХНУРЕ. Зросла підготовленість студентів Національної гірничої академії. Крім того, студенти цих вузів здатні вільно складати програми для вирішення складних криптологічних задач з використанням ПЕОМ.

В номінації "Захист інформації в системах та мережах", в якій виступали студенти інших спеціальностей (спеціальностей, що вивчали лише одну або дві загально освітні дисципліни з захисту інформації) студенти також показали зрослий рівень знань та умінь. Вони продемонстрували ознайомленість з проблемними питаннями та новітніми алгоритмами криптографічного захисту інформації. Але, на наш погляд, математичний рівень підготовки цих студентів у криптології недостатній.

В цілому слід відмітити суттєво зросший рівень знань, обізнаність студентів та магістрів з проблемними питаннями інформаційної безпеки, а також велику зацікавленість у вивченні сучасних методів та засобів захисту інформації.

*Харьковский национальный университет радиоэлектроники*

*Поступила в редколлегию 15.04.2002*

## СОДЕРЖАНИЕ

## МЕТОДЫ И СРЕДСТВА АУТЕНТИФИКАЦИИ, ФУНКЦИИ ХЭШИРОВАНИЯ



## СИСТЕМЫ И СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

РЕФЕРАТЫ РЕФЕРАТИ ABSTRACTS

УДК 681.322

**Инфраструктура открытых ключей как основа обеспечения информационной безопасности национальных, ведомственных и коммерческих систем информационных технологий** / М.Ф. Бондаренко, И.Д. Горбенко, С.П. Черных, А.В. Потий // Радиотехника. Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С 5-17

Предлагается и анализируется инфраструктура открытых ключей национальных ведомственных и коммерческих систем информационных технологий.

Ил. 6. Библиогр.: 12

УДК 681.322

**Інфраструктура відкритих ключів як основа забезпечення інформаційної безпеки національних, відомчих та комерційних систем інформаційних технологій** / М.Ф. Бондаренко, І.Д. Горбенко, С.П. Черних, О.В. Потій// Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С.5-17.

Пропонується та аналізується інфраструктура відкритих ключів національних відомчих та комерційних систем інформаційних технологій.

Іл. 6. Бібліогр.: 12 назв.

UDC 681.322

**Public key infrastructure as basis of information security support of national, departmental and commercial information system**./ M.F. Bondarenko, I.D. Gorbenko, S.P. Chernyh, A.V. Potiy // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N 126. P.5-7.

The public key infrastructure (PKI) of national, department and commercial information system are suggested.

6 fig. Ref.: 12 items.

УДК 681.3.06:519.248.681

**Методологические аспекты германского стандарта «Руководство по базовому уровню защиты информационных технологий»**/ Л. В.Скрипник, М. Ф. Бондаренко, И. Д. Горбенко, А. А. Ткач, А.В. Потий // Радиотехника. Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С 18-30..

Рассматриваются основные понятия, структура и рекомендации по методологии работ германского стандарта «Руководство по базовой защите информационных технологий».

Табл. 7. Ил.2. Библиогр. 7 назв.

УДК 681.3.06:519.248.681

**Методологічні аспекти германського стандарту «Керівництво щодо базового рівня захисту інформаційних технологій»**/ Л. В.Скрипник, М. Ф. Бондаренко, І. Д. Горбенко, О. О. Ткач, О.В. Потій // Радіотехніка. Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С 18-30.

Розглядаються основні поняття, структура та рекомендації щодо методології робіт германського стандарту "Керівництво щодо базового захисту інформаційних технологій".

Табл. 7. Іл.2. Бібліогр. 7 назв.

UDC 681.3.06:519.248.681

**Methodological aspects of German standard "IT Baseline Protection Manual".** L.V. Skrypnik, M.F. Bondarenko, I.D. Gorbenko, O.O. Tkach, A.V. Poty // Radiotekhnika: All-Ukr. Interdep. Mag. 2002. N 126..P.18-30

The basic principles, structure and recommendations of German standard "IT Baseline Protection Manual" methodology works are considered in the article.

7 tab. 2 fig. Ref.: 7 items.

УДК 681.3.06

**Оценка эффективности действий злоумышленника при реализации угроз информации в распределенных компьютерных системах с открытой архитектурой** / А.Н. Новиков, А.А. Тимошенко // Радиотехника. Всеукр. Межвед. научн.-техн. сб. 2002.Вып.126. С.31-41.

Предлагается методика оценки эффективности действий злоумышленника, определенной как показатель оценки риска, связанного с реализацией угрозы информации, основанная на использовании метода анализа иерархий и результатов анализа статистических данных о реализации угроз информации в распределенных вычислительных сетях с открытой архитектурой.

Табл.2. Ил.6. Библиогр: 13 назв.

УДК 681.3.06

**Оцінка ефективності дій зловмисника при реалізації загроз інформації в розподілених комп'ютерних системах з відкритою архітектурою** / О.М. Новиков, А.О. Тимошенко // Радіотехніка. Всеукр. міжвід. наук.-техн. зб.2002.Вип.126.С.31-41.

Пропонується методика оцінки ефективності дій зловмисника, визначеної як показник оцінки ризику, пов'язаного з реалізацією загрози інформації, яка грунтується на використанні методу аналізу ієрархій і результатів аналізу статистичних даних про реалізацію загроз інформації в розподілених обчислювальних мережах з відкритою архітектурою.

Табл.2. Іл.6. Бібліогр: 13 назв.

UDC 681.3.06

**Evaluation of violator's operations efficiency during information's threats realization in the distributed computer systems with open architecture** / A.N. Novikov, A.A. Timoshenko // Radiotekhnika. All-Ukr. Sci.Interdep.Mag.2002.№126.P.31-41.

Technique of an evaluation of violator's operations efficiency, defined as a risk evaluation parameter, based on use of Analitic Hierarchy Process and outcomes of the analysis of statistical data about information's threats realization in the distributed computer networks with open architecture is offered.

2 tab. 6 fig. Ref.: 13 items.

УДК 621.391: 519.7

**Регулярные конгруэнции и строение алгебраических моделей симметричных криптосистем** / А.Н. Алексейчук, А.І Романов // Радиотехника: Всеукр. межвед. науч.-техн. сб. Вып.126. С. 42-58.

Исследуются строение и общие структурные свойства алгебраических моделей симметричных криптосистем (шифров Вводится понятие группы вполне эндоморфного шифра, с использованием которого с точностью до изоморфизма описываютс решетки подшифров и регулярных конгруэнций данного шифра. Доказывается теорема о представлении шифра импримитивной группой в виде сплетения двух неразложимых шифров и описывается строение их групп. Обсуждаютс практические применения полученных результатов, связанные с возможностью сокращения трудоемкости ряда алгоритмс криптоанализа шифров с импримитивными группами на основе "метода гомоморфизмов".

Библиогр.: 19 назв.

УДК 621.391: 519.7

**Регулярні конгруенції та будова алгебраїчних моделей симетричних криптосистем** / А.М. Олексійчук, О.І. Романс // Радіотехніка: Всеукр. міжвід. наук-техн. зб. Вип. 126. С. 42-58.

Досліджуються будова та загальні структурні властивості алгебраїчних моделей симетричних криптосистем (шифрів Вводиться поняття групи цілком ендоморфного шифру, з використанням якого з точністю до ізоморфізму описуються грат підшифрів та регулярних конгруенцій цього шифру. Доводиться теорема про представлення шифру з імпримітивною групою вигляді сплетіння двох нерозкладних шифрів та описується будова їх груп. Обговорюються практичні застосування отримани результатів, що пов'язані з можливістю зменшення трудомісткості ряду алгоритмів криптоаналізу шифрів з імпримітивним групами, на основі "методу гомоморфізмів".

Бібліогр.: 19 назв.

UDC 621.391: 519.7

**Regular congruences and design of algebraic models of the conventional cryptosystems** / A.N. Alexeychuk, A.I. Romanov Radiotekhnika: All.-Ukr. Sci. Interdep. Mag. 2002. N 126. P. 42-58.

The design and the common structural properties of the conventional cryptosystems (ciphers) algebraic models are studied. Tl notion of the perfectly endomorphic cipher group is introduced. The use of this notion up to the isomorphism precisely allows to descril the lattices of the subciphers and regular congruences of this cipher. The theorem on presentation of a cipher with the imprimitive grou as a junction of two non-exspandable ciphers is proved, and design of their groups is described. The practical applications of the receive results connected with the possibility of reducing the complexity of certain algorithms, based on "the metod of homomorphisms", f cryptoanalysis of the ciphers with imprimitive groups, are discussed.

Ref. 19 items.

УДК 681.3.06: 519.248.681

**Возможность применения современных методов сжатия информации для сжатия речи** / А.А. Замула, Ю.С. Павленко // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 59-64.

Рассматриваются вопросы, связанные со сжатием информации. Проанализированы существующие стандарты сжати речевой информации. Проведен анализ новых методов сжатия информации. Рассмотрена возможность применения их дл сжатия речи.

Табл. 1.Ил. 2. Библиогр.: 6 назв.

УДК 681.3.06: 519.248.681

**Можливість застосування сучасних методів стискання інформації для стискання мови** /О.А.. Замула, Ю.С. Павленк // Радітехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 59-64.

Розглядаються питання, пов'язані із стисканням інформації. Проаналізовані існуючі стандарти стискання мовної інформації. Проведено аналіз нових методів стискання інформації. Розглянута можливість застосування їх для стискання мови.

Табл. 1. Іл. 2. Бібліогр.: 6 назв.

UDC 681.3.06: 519.248.681

**The possibility for modern data compression method application speech compression** / A.A. Zamula, Y.S. Pavlenko // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N 126. P. 59-64.

The problems involving data compression are considered. Existing standards of speech compression were analyzed. The analysis of new methods of data compression was performed. The possibility to apply them for speech compression was considered.

1 tab. 2 fig. Ref.: 6 items.

УДК 681.3.06

**Методология анализа рисков и управления рисками** / А.А. Замула // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 65-70.

В статье рассматривается модель процесса обеспечения информационной безопасности, учитывающая воздействие на информацию различных дестабилизируюших факторов (угроз). Приводиться анализ содержания этапов процесса управления рисками.

Табл. 1.Ил. 1. Библиогр.: 4 назв.

УДК 681.3.06

**Методологія аналізу ризиків та управління ризиками** ./О.А. Замула// Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 65 – 70.

У статті розглядаються модель процесу забезпечення інформаційної безпеки, яка враховує вплив на інформацію різних погроз. Наводиться аналіз змісту етапів процесу управління ризиками.

Табл. 1. Іл. 1. Бібліогр.: 4 назв.

UDC 681.3.06: 519.248.681

**Methodology of risk analysis and risk control** / A.A. Zamula // Radiotekhnika. All-Ukr. Sci. Interdep. Mag. 2002. N 126. P. 65—70.

The model of informational security supplying process, which takes into account different threats influence on information, is considered. The analysis of phases of risk control process is given.

1 tab. 1 fig. Ref.: 4 items.

УДК 681.3.06:519.248.681

**Сравнительный анализ ЦП в группах точек эллиптических кривых** / И.Д. Горбенко, С.И. Збитнев, А.А. Поляков // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С.71-84.

Приводятся требования к цифровым подписям (ЦП). Показаны наиболее известные ЦП. Проводится анализ стойкости ЦП к различным типам атак. Рассматриваются перспективные направления развития преобразований в группах точек эллиптической кривой.

Табл. 1. Библиогр.: 33 назв.

УДК 681.3.06:519.248.681

**Порівняльний аналіз ЦП у групах точок еліптичної кривої** / І.Д. Горбенко, С.І. Збітнєв, А.О. Поляков // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 71-84.

Наводяться вимоги щодо цифрових підписів (ЦП). Наведені найбільш відомі ЦП. Проводиться аналіз стійкості ЦП до різноманітних типів атак. Розглядаються перспективні напрямки розвитку перетворень у групах точок еліптичної кривої.

Табл. 1. Бібліогр.: 33 назви.

UDC 681.3.06:519.248.681

**Comparative analysis of DS in group of elliptic curve points** / I.D. Gorbenko, S.I. Zbitnev, A.A. Polaykov // Radiotekhnika: all-Ukr. Sci. Interdep. Mag. 2002. N 126. C. 71-84.

Requirement to digital signatures (DS) are presented. The most familiar DS are shown. The analysis of DS resistance to various attack types is carried out. The perspective directions of transformation evolution in elliptic curve group points are considered.

1 tab., Ref.: 33 items

УДК 681.3.06:519.248.681

**Протоколы – примитивы управления ключами в группах точек эллиптических кривых** / И.Д. Горбенко, А.А. Поляков, С.И. Збитнев // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып.126. С. 85-96

Приводятся требования и дается классификация протоколов. Рассматриваются простые протоколы установления ключей, основанные на схемах Диффи-Хелмана и MQV. Приводятся основные критерии состоятельности протоколов. Дана полная схема протокола «точка–точка».

Табл. 1. Ил.: 7, Библиогр.: 33 назв.

УДК 681.3.06:519.248.681

**Протоколи – примітиви управління ключами у групах точок еліптичній кривій** / І.Д. Горбенко, А.О. Поляков, С.І.

Збітнєв // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 85-96.

Наводяться вимоги та дається класифікація протоколів. Розглядаються прості протоколи управління ключами, грунтовані на схемах Диффи-Хелмана та MQV. Надаються основні критерії слушності протоколів. Наведена повна схема протоколу "точка – точка".

Табл. 1. Іл.: 7, Бібліогр.: 33 назви.

UDC 681.3.06:519.248.681

**Protocols–primitives of key control in the group of elliptic curve points** / I.D. Gorbenko, A.A. Polaykov, S.I. Zbitnev // Radiotekhnika: au-Ukr. Sci. Interdep. Mag. 2002. N 126.C. 85-96.

Requirements are presented, and protocols classification is gave. Basic protocols key agreement scheme based on Diffi - Hellman and MQV schemes are considered. Primary criteria for well-grounded protocols are presented. The full scheme of protocol "point-to-point" is given.

1 tab., 7 Fig., Ref.: 11 items

УДК 681.3.06:519.248.681

**Схемы направленного шифрования в группах точек на эллиптических кривых** / И.Д. Горбенко, Д.С. Балагура // Радиотехника: Всеукр. межвед. научн.-техн. сб. 2002. Вып. 126 . С. 97-105.

Рассматривается состояние вопроса в области направленного шифрования. Проводится обоснование возможности реализации направленного шифрования в группах точек на эллиптических кривых и разработка концептуальных вопросов их реализации и оценки стойкости.

Табл. 2. Ил. 2. Библиогр.: 8 назв.

УДК 681.3.06:519.248.681

**Схеми направленого шифрування в групах точок на еліптичних кривих** / І.Д. Горбенко, Д.С. Балагура // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 97-105.

Розглядається стан питання в області направленого шифрування. Проводиться обґрунтування можливості реалізації направленого шифрування в групах точок на еліптичних кривих і розробка концептуальних питань їхньої реалізації та оцінки стійкості.

Табл. 2. Іл.2. Бібліогр.: 8 назв.

UDC 681.3.06:519.248.681

**Public key encryption schemes in Elliptic Curves** / I.D. Gorbenko, D.S. Balagura // Radiotekhnika. All-Ukr. Sci. Interdep. Mag. 2002. N126..P. 97-105.

Current state at public key encryption field is considered, possibility ground of achievement of public key encryption in elliptic curves and conceptual questions elaboration of their implementation and strength estimation are given in the article.

Tab. 2. Fig. 2.Ref.: 8 items.

УДК 681.3.06:519.248.681

**Схемы разделения секрета: основные методы и средства реализации** / И.Д. Горбенко, А.А. Поляков, Е.В. Попович, Е.С. Коробельников // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 106-114.

Приведены основные методы и схемы протоколов в области разделения секрета. Проведена классификация и сравнительный анализ схем разделения секрета, а также разработаны рекомендации по их использованию.

Табл. 1. Ил. 2. Библиогр.: 8 назв.

УДК 681.3.06:519.248.681

**Схеми розподілу таємниці: основні методи та засоби реалізації** / І.Д. Горбенко, А.О. Поляков, Л.В. Скрипник, Є.С. Коробельников // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 106-114.

Наводяться основні методи та схеми протоколів у галузі розподілу таємниці. Наведена класифікація та порівняльний аналіз схем розподілу таємниці, а також розроблені рекомендації за їхнім використанням.

Табл. 1. Іл. 2. Бібліогр.: 8 назв.

UDC 681.3.06:519.248.681

**Secret shares shames: basic methods and means of implementation** /I.D. Gorbenko, A.A. Polaykov, E.V.Popovich, Y.S.Korobelnikov//Radiotekhnika: All-Ukr.Sci.Interdep.Mag.2002.N126.C.106-114.

Basic methods and schemes protocols in the field secret share are presented. Classification and comparative analyses schemes of secret share are presented, and recommendations on their use are developed.

Tab. 1. Fig. 2. Ref.: 8 items

УДК 681.3.06:519.248.681

**Методы формирования псевдослучайных последовательностей в группах точек эллиптических кривых** /Т.А.Гриненко,С.И.Збитнев,Д.В.Мялковский// Радиотехника:Всеукр. межвед.науч.-техн.сб.2002.Вып.126.С.115-122.

Дается сравнительный анализ свойств генератора псевдослучайных последовательностей на эллиптических кривых со свойствами псевдослучайных последовательностей генератора псевдослучайных чисел BBS. Приведены результаты статистического тестирования псевдослучайных последовательностей с использованием методики NIST STS. Обосновывается возможность применения псевдослучайных последовательностей на эллиптических кривых в криптографии.

Табл. 3. Ил. 7. Библиогр.: 6 назв.

УДК 681.3.06:519.248.681

**Методи формування псевдовипадкових послідовностей в групах точок еліптичних кривих** /Т.О.Гріненко, С.І.Збітнев, Д.В. Мялковський//Радіотехніка:Всеукр.міжвід.наук.-техн. зб. 2002. Вип. 126. С.115-122.

Подається порівняльний аналіз властивостей генератора псевдовипадкових послідовностей на еліптичних кривих з властивостями псевдовипадкових послідовностей генератора псевдовипадкових чисел BBS. Наведені результати статистичного тестування псевдовипадкових послідовностей з використанням методики NIST STS. Обґрунтовується можливість використання псевдовипадкових послідовностей на еліптичних кривих у криптографії.

Табл. 3. Іл. 7. Бібліогр.: 6 назв.

UDC 681.3.06:519.248.681

**Methods of the pseudo-random sequences building on the elliptic curves** / T.A. Grinenko, S.I. Zbitnev, D.V. Mialkovskiy // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. № 126 C. 115-122.

Comparative analysis of pseudo-random sequences generator properties on elliptic curves with pseudo-random sequences generator properties of pseudo-random numbers BBS are given. Results of pseudo-random sequences statistic testing with NIST STS methods are presented. The possibility to use pseudo-random sequences on elliptic curves in cryptography are substantiated.

3 tab., 7 fig., Ref.: 6 items.

УДК 681.3.06:519.248.681

**Проективная геометрия - не все так гладко** / С.И. Збитнев // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып.126. С.123-131.

Приводятся всевозможные базисы представления точек на эллиптической кривой. Показывается эффективность использования смешанных координат. Приводятся дополнительные требования и ограничения к использованию проективных координат.

Табл. 2. Библиогр.: 21 назв.

УДК 681.3.06:519.248.681

**Проективна геометрія - не все так гладко** / С.І. Збітнєв // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип.126. С.123 - 131.

Наводяться всілякі базиси представлення точок на еліптичній кривій. Показується ефективність використання змішаних координат. Надаються додаткові вимоги й обмеження до використання проективних координат.

Табл. 2. Бібліогр.: 21 назви.

UDC 681.3.06:519.248.681

**Projective geometry – not everything is so good** / S.I. Zbitnev // Radiotekhnika: all-Ukr. Sci. Interdep. Mag. 2002. N 126. C. 123 - 131.

All possible bases of points representation on the elliptic curve are presented. Effectiveness of using mixed coordinates is shown. The additional requirements and restrictions to the projective coordinates application are cited.

2 tab., Ref.: 21 items

УДК 681.3.06

**Обоснование выбора метода построения криптографически стойких булевых функций** /, А.В. Потий, Ю.А. Избенко // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 132-137.

Проводится анализ основных показателей стойкости булевых функций. Обосновывается выбор класса методов для построения криптографически стойких булевых функций.

Табл. 1. Ил. 1. Библиогр.: 7 назв.

УДК 681.3.06

**Обґрунтування вибору методу побудови криптографічно стійких булевих функцій** / О.В. Потій, Ю.А. Ізбенко, // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 132-137.

Проводиться аналіз основних показників стійкості булевих функцій. Обґрунтовується вибір класу методів для побудови криптографічно стійких булевих функцій.

Табл. 1. Іл. 1. Бібліогр.: 7 назв.

UDC 681.3.06

**Substantiation of choose of method design cryptographically strength Boolean functions** / A.V. Poty, Y.A. Izbenko, // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. №. 126. P. 132-137.

Analysis of the main criteria of Boolean functions strength is discussed. The choice of the methods class for cryptographically strength Boolean functions constructions is justified.

1 tab. 1 fig. Ref.: 7 items.

УДК 681.3.06: 519.248.681

**Обеспечение стойкости шифра DES к атакам линейного криптоанализа .Требования к отбору S-блоков, защищенных от атак на десятицикловые линейные итеративные линейные аппроксимации** / И.В. Лисицкая, А.С.Бондаренко, Т.В.Цепурит А.И.Колыбельников // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 138-146..

Рассматриваются возможности повышения стойкости шифра DES к атакам линейного криптоанализа. Предлагаются и обосновываются новые требования к отбору S-блоков, защищенных от атак на восьмицикловые линейные итеративные характеристики.

Ил. 8. Библиогр.: 2 назв.

УДК 681.3.06: 519.248.681

**Забезпечення стійкості шифру DES до атак лінійного криптоаналізу. Вимоги до відбору S-блоків, захищенних від атак на десятициклові лінійні ітеративні апроксимації** / І.В. Лисицька, А.С.Бондаренко, Т.В.Цепуріт, О.І.Колибельніков // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 138-146.

Розглядаються можливості підвищення стійкості шифру DES до атак лінійного криптоаналізу. Пропонуються та обгрунтовуються нові вимоги до відбору S-блоків, захищених від атак на восьмициклові лінійні ітеративні характеристики.

Іл.8. Бібліогр.: 2 назв.

UDC 681.3.06: 519.248.681

**Provision of the cipher DES resistance to linear cryptoanalysis attacks. Requirements to selection of S-bloks protected against the attacks on 10-cyclic linear iterate approximations** / I.V. Lysytska, A.S. Bondarenko,T.V.Czepurit, O. I. Kolibelnikov // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N 126. P. 138-146.

The possibilities of the cipher DES resistance to linear cryptoanalysis attacks are considered. New requirements to the selection of S-blocks protected against the attack on 10-cyclic linear iterate approximations are proposed and justified.

8 fig. Ref.: 2 items.

УДК 681.3.06:519.248.681

**Оценка путей использования дискретного преобразования Хартли в криптографических системах** / И.В. Рубан, С.В. Дуденко // Радиотехника: Всеукр. межвед. науч. – техн. сб. 2002. Вып. 126 С. 147-152.

Рассматривается возможность использования дискретного преобразования Хартли для криптографического преобразования информации и его математические свойства. Оцениваются параметры криптограммы, корреляционная зависимость открытого текста и криптограммы.

Табл.2. Ил.2. Библиогр.: 9.

УДК 681.3.06:519.248.681

**Оцінка шляхів використання дискретного перетворення Хартлі в криптографічних системах** / І.В. Рубан, С.В. Дуденко // Радіотехніка: Всеукр. міжвід. наук.- техн. зб. 2002. Вип. 126 С.147-152.

Розглядається можливість використання дискретного перетворення Хартлі для криптографічного перетворення інформації та його математичні властивості. Оцінюються параметри криптограми, кореляційна залежність відкритого тексту та криптограми.

Табл.2. Іл.2. Бібліогр.: 9.

UDC 681.3.06:519.248.681

**Estimation of ways of the Hartley transform usage in cryptography systems** / I.V.Ruban, S.V.Dudenko // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. № 126 P. 147-152.

The possibility to use the Hartley transform for the information cryptography transformation and its mathematical properties are considered. The parameters of the cryptograms, correlation relation of a plain text and cryptogram are estimated.

2 tab. 2 fig. Ref.: 9 items.

УДК 681.3.06: 519.248.681

**Алгоритм порождения противоречивых подстановок**/ И.В. Лисицкая, Т.В. Цепурит, С.В. Супрунюк // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С.153-157.

Предлагается алгоритм построения противоречивых подстановок, которые могут использоваться в качестве таблиц подстановок в алгоритмах симметричного шифрования. Используются и обосновываются как случайные, так и детерминированные принципы выбора значений для таблиц подстановок.

Ил. 4. Библиогр.: 3 назв.

УДК 681.3.06: 519.248.681

**Алгоритм породження суперечливих підстановок**/ І.В. Лисицька, Т.В. Цепуріт, С.В. Супрунюк // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С.153-157.

Пропонується алгоритм будування суперечних підстановок, які можна використовувати як таблиці підстановок у алгоритмах симетричного шифрування. Використовуються та обґрунтовуються як випадкові, так і детерміновані принципи вибору значень для таблиць підстановок.

Іл. 4. Бібліогр.: 3 назв.

UDC 681.3.06: 519.248.681

**The algorithm of the contradictory substitutions generation**/ I.V. Lysytska, T.V. Tsepurit, S.V. Suprunyuk// Radoitekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N. 126. P.153-157.

The algorithm for contradictory of substitutions generation, which can be used as the substitution tables in the symmetric encryption algorithms, is proposed. Both random and deterministic principles of the choice of values for the substitution tables are proposed and justified.

4 fig. Ref.: 3.

УДК 681.3.06: 519.248.681

**Цепи Фестеля и дифференциальный криптоанализ** / И.В. Лисицкая, В.И. Руженцев // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 158-165.

Внимание сосредотачивается на сравнении стойкости к дифференциальным атакам различных видов шифров, при этом предлагаются способы выполнения оценки стойкости этих шифров.

Ил. 7. Библиогр.: 12 назв.

УДК 681.3.06: 519.248.681

**Ланцюги Фестеля та диференційний криптоаналіз** / І.В. Лисицька, В.І. Руженцев // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 158-165.

Увага зосереджується на порівнянні стійкості до диференціальних атак різних видів шифрів, при цьому пропонуються способи виконання оцінки стійкості цих шифрів.

Іл. 7. Бібліогр.: 12 назв.

UDC 681.3.06: 519.248.681

**Feistel networks and differential cryptanalysis** / I.V. Lisitskaya, V.I. Ruzhentsev // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N 126. P. 158-165.

The attention is focused on comparison of various ciphers resistance to differential attacks, thus the ways to perform resistance estimation for these ciphers are proposed.

7 fig. Ref.: 12 items.

УДК 681.3.06

**Оценка опасности криптоаиалитических атак методом создания коллизий** / А.Д. Тевяшев, Ю.И. Горбенко // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 166-171..

Рассматривается степень опасности реализации криптоаналитических атак методом создания коллизий. Приводятся количественные оценки вероятностей реализации атак.

Библиогр.: 3 назв.

УДК 681.3.06

**Оцінка небезпеки криптоаналітичних атак методом створення колізій** / А.Д. Тевяшев, Ю.І. Горбенко // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 166-171.

Розглядається рівень небезпеки реалізації криптоаналітичних атак методом створення колізій. Наводяться кількісні оцінки ймовірностей реалізації атак.

Бібліогр.: 3 назв.

UDC 681.3.06: 519.248.681

**Estimation of crypto-analytic attacks danger using the collision method** / A.D. Tevyashev, U.I. Gorbenko // Radiotekhnika: All-Urk. Sci. Interdep. Mag. 2002. N 126. P. 166-171.

The degree of crypto-analytic attacks realization danger using the collision creation method is considered. The quantitative estimates of the attacks possibilities are cited.

Ref.: 3 items.

УДК 681.3.06:519.248.681

**Функции хэширования: классификация, характеристика и сравнительный анализ** / В.Н. Вервейко, А.И. Пушкарев, Т.В. Цепурит // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 172-179.

В статье рассматриваются существующие и перспективные функции хэширования, их классификация и требования, предъявляемые к ним. Приводятся возможные атаки на хэш-функции. Даются рекомендации по применению функций хеширования.

Табл. 3. Ил. 2. Библиогр.: 9 назв.

УДК 681.3.06:519.248.681

**Функції хешування: класифікація, характеристика та порівняльний аналіз** / В.М. Вервейко, А.І. Пушкарьов, Т.В. Цепуріт // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 172-179.

У статті розглядаються існуючі функції та перспективні функції хешування, їх класифікація та вимоги щодо них. Наводяться можливі атаки на хеш-функції. Надаються рекомендації щодо використання функій хешування.

Табл. 3. Іл.2. Бібліогр.: 9 назв.

UDC 681.3.06:519.248.681

**Hashing functions: classification, characteristics and comparative analysis** / V.N. Verveyko, A.I. Pushkarev, T.V. Tsepurit //Radoitekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N. 126. P.172-179.

Existent and perspective hashing functions, their classification and requirements to them are considered. Possible attacks on hash-functions are presented. The recommendations for hashing functions application are given.

3 tab. 2 fig. Ref.: 9 items.

УДК 681.3.06

**Обнаруживающая способность помехоустойчивых кодов и схем аутентификации** / А.А. Кузнецов, А.В. Потий // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 180-184.

Исследуется обнаруживающая способность помехоустойчивых кодов и схем формирования кодов аутентификации, их вероятностные характеристики. Обосновывается возможность построения комбинированных схем помехоустойчивого кодирования и проверки подлинности.

Ил. 2. Библиогр.: 5 назв.

УДК 681.3.06

**Викриваюча здібность похибкостійких кодів та схем автентифікації** / О.О. Кузнецов, О.В. Потій // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 180-184.

Досліджується викриваюча здібность похибкостійких кодів та схем автентифікації, їх ймовірносгні характеристики. Обгрунтовується можливість побудови комбінованих схем похибкостійкого кодування та перевірки дійсності.

Іл. 2. Бібліогр.: 5 назв.

UDC 681.3.06

**Detection ability of noise-resistant codes and authentification circuits** / A.A. Kuznetsov, A.V. Potiy // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. №. 126. P. 180-184.

The detection ability of noise-resistant codes and authentification codes shaping circuits, their probability performances is studied. The possibility to build up the combined circuits of noise-resistant encoding and authenticity check is justified.

2 fig. Ref.: 5 items.

УДК 681.3.06:006.354

**Перехват системных вызовов в операционной системе Windows 2000** / Я.В. Чуприй, Р.В. Олейников // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 185-192.

Рассматриваются особенности внутренних структур данных Windows 2000 с точки зрения обеспечения безопасности. Показано, что после внедрения специализированной динамически загружаемой библиотеки, создания удалённого потока в целевом процессе и замены адресов в таблице импорта становится возможным получение полного контроля над приложением. Приведен пример реализации перехвата, и предложено несколько методов предотвращения таких атак.

Ил. 1. Библиогр.: 6 назв.

УДК 681.3.06:006.354

**Перехоплення системних викликів в операційній системі Windows 2000** / Я.В. Чупрій, Р.В. Олійников // Радіотехніка: Всеукр. межвід. наук.-техн. зб. 2002. Вип. 126. С. 185-192.

Розглядаються особливості внутрішніх структур даних Windows 2000 з точки зору забезпечення безпеки. Показано, що після впровадження спеціалізованої динамічної бібліотеки, що завантажується, створення віддаленого потоку в цільовому процесі та заміни адрес у таблиці імпорту стає можливим одержання повного контролю над застосуванням. Наведено приклад реалізації перехоплення, і запропоновано декілька методів щодо запобігання таких атак.

Іл. 1. Бібліогр.: 6 назв.

UDC 681.3.06:006.354

**System call hooking in Windows 2000 operating system** / Y.V. Chupriy, R.V. Oliynykov. Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. № 126. P. 185-192.

The features of internal Windows 2000 data structures are considered at the security point of view. It is shown, that after introducing a specialized dynamically loaded library, creating a remote thread in the target process and substituting addresses in the import table one can get the complete control over the application. The example of the hook implementation is given, and several methods for preventing such kind of attacks are affered.

1 fig. Ref.: 6 items.

УДК 681.3.06

**Генерация параметров и ключей для цифровой подписи на эллиптических кривых для конечного простого поля/** И.Д. Горбенко, Е.Г. Качко, П.В. Колесников // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 193-198.

Приведен практический алгоритм формирования параметров эллиптической кривой для схемы электронной цифровой подписи в соответствии с X9.62-1998. Для иллюстрации достоверности приведенного алгоритма используется контрольный пример. Приведены также промежуточные данные при выполнении каждого шага алгоритма, что существенно упрощает практическую реализацию вычислений параметров.

Табл. 1. Библиогр.: 3 назв.

УДК 681.3.06

**Генерація парамерів та ключів для цифрового підпису на еліптичних кривих для скінченого простого поля/** І.Д. Горбенко, О.Г.Качко, П.В. Колесніков // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 193-198.

Приведено практичний алгоритм формування параметрів еліптичної кривої для схеми електронного цифрового підпису у відповіді з X9.62-1998. Для ілюстрації достовірності наведеного алгоритму використається контрольний приклад. Приведені також проміжні данні після виконання кожного шага алгоритму, що суттєво полегшує практичну реалізацію обчислювань параметрів.

Табл. 1. Бібліогр.: 3 назв.

UDC 681.3.06

**Parameters and keys generation for digital sign on elliptic curves for finite simple field** / I.D. Gorbenko, E.G. Kachko, P.V. Kolesnikov // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N 126. P. 193-198.

An algorithm of parameters generation for elliptic curves for digital sign standard X9.62-1998 is considered. A control example is used to illustrate the given algorithm reliability. Intermediate results when executing each algorithm step are also cited, they significantly simplify practical implementation of parameters computation.

1 tab. Ref.: 3 items.

УДК 681.3.06:519.248.681

**Защищенность операционных систем: требования, основные атаки, сравнительный анализ** / Г.Н. Гулак, Е.Г. Качко, С.С. Лавриненко // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 199-212.

Рассматриваются проблемы защиты операционных систем. Проводится анализ основных атак на операционные системы, и предлагается ряд мер по их защите.

Табл. 1. Ил. 3. Библиогр.: 6 назв.

УДК 681.3.06:519.248.681

**Захищеність операційних систем: вимоги, основні атаки, порівняльний аналіз** / Г.М. Гулак, Є.Г. Качко, С.С. Лавриненко // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С. 199-212.

Розглядаються проблеми захисту операційних систем. Проводиться аналіз основних атак на операційні системи та пропонується ряд заходів щодо їх захисту.

Табл. 1. Іл.3. Бібліогр.: 6 назв.

UDC 681.3.06:519.248.681

**Operation systems security: requirements, main attacks, comparative analysis** / G.N. Gulak, H.G. Kachko, S.S. Lavrinenko // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N 126. P. 199-212.

Problems of the operation system's protection are considered. Main attacks on operation systems are analyzed and the measures for their protection are proposed.

Tab.1. Fig.3. Ref.: 6 items.

УДК 681.3.06

**К созданию информационного пространства боя в перспективной системе наведения истребителей** / А.И. Тимочко, Я.В. Ефименко// Радиотехника. 2002. Вып. 126. С. 213-217.

Для построения автоматизированных систем управления авиацией предлагается использовать концепцию объектно-ориентированного проектирования. Проведен объектно-ориентированный анализ построения систем отображения информации и проектирование структуры перспективной системы трехмерного отображения информации.

Ил. 3. Библиогр.: 8 назв.

УДК 681.3.06

**До створення інформаційного простору бою у перспективній системі наведення винищувачів/** О.І. Тимочко, Я.В. Єфіменко// Радіотехніка. 2002. Вип. 126. С. 213-217.

Для побудови автоматизованих систем управління авіацією пропонується використовувати концепцію об'єктно-орієнтованого проектування. Проведений об'єктно-орієнтований аналіз побудови систем відображення інформації та проектування структури перспективної системи тримірного відображення інформації.

Іл. 3. Бібліогр.: 8 назв.

UDC 681.3.06

**For creation the information area of battle at the perspective fighter guidance system/** A.I.Timochko, I.V.Efimenko // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. № 126. P.213-217.

It is offered to use the concept of object-oriented designing for construction of automated control systems of aircraft. The fighters prompting systems construction object-oriented analysis and designing of structure perspective system of three-dimensional display of information.

УДК 681.324.067

**Повышение быстродействия генераторов случайных последовательностей на основе физических датчиков** / А.А. Торба, В.А. Бобух, С.Г. Елаков, А.З. Степченко // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып.126.С.218-221..

Предложены схемные решения повышения быстродействия генераторов случайных последовательностей на основе физических датчиков шума. Проведен анализ причин нарушения равновероятного закона распределения генерируемых случайных чисел при повышении частоты генерации. Описаны алгоритмы выравнивания вероятностей случайных чисел и повышения надежности работы генераторов.

Ил.3. Библиогр.: 2 назв.

УДК 681.324.067

**Підвищення швидкодії генераторів випадкових послідовностей на основі фізичних давачів/** О.О. Торба, В.А. Бобух, С.Г. Єлаков, О.З. Степченко // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип.126.С.218-221.

Запропоновано схемні рішення підвищення швидкодії генераторів випадкових послідовностей на основі фізичних давачів шуму. Проведено аналіз причин порушення рівноімовірного закону розподілу випадкових чисел, що генеруються, при підвищенні частоти генерації. Описано алгоритми вирівнювання імовірностей випадкових чисел і підвищення надійності роботи генераторів.

Іл.3. Бібліогр.: 2 назв.

UDC 681.324.067

**Rise in speed of random sequences generators based on physical sensors/** A.A. Torba, V.A. Bobuh, S.G. Yelakov, A.Z. Stepchenko // Radiotekhnika: All-Ukr. Sci. Interdep. Mag.2002 N 126 P.218-221.

System decisions of rise in speed of random sequences generators based on the noise physical sensors are offered. The causes for violation of the generated random numbers distribution equiprobable law, when rising the generation frequency, are analysed. Algorithms of random numbers probabilities alignment and the generator operation reliability are described.

3 fig. Ref.: 2 items.

УДК 681.3.067+621.396.626:537.87

**Обеспечение достоверности оценки дальности выявления излучений ТСПИ /**
В.И. Заболотный, А.П. Метелев // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып.126. С.222-226.

Рассматривается порядок учёта инструментальных ошибок при оценке дальности побочных электромагнитных излучений технических средств обработки информации (ТСПИ).

Библиогр.: 7 назв.

УДК 681.3.067+621.396.626:537.87

**Забезпечення достовірності оцінки далекості виявлення випромінювань ТЗПІ /**
В.І. Заболотний, О.П. Метелев // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип 126. С.222-226.

Розглядається порядок враховування інструментальних похибок при оцінці далекості побічних випромінювань технічних засобів обробки інформації (ТЗПИ).

Бібліогр.: 7 назв.

UDC 681.3.067+621.396.626:537.87

**Support of IPTT radiation detection range estimation reliability** / V.I. Zabolotny, A.P. Metelev // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2001. №126. P. 222-226..

The sequence of instrumental error registration when estimating the information processing technical tools (IPTT) leakage electromagnetic radiation range is considered.

Ref.: 8 items.

УДК 389.2

**Критерий выделения квазипериодических составляющих, обусловленных взаимодействием мер в групповом эталоне частоты** / Ю.И. Евдокименко, А.П. Нарежний // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 227-230.

Предлагается критерий выявления значимых флуктуаций частоты и фазы мер в групповом эталоне, имеющих квазипериодический тип проявления с позиций поведения группы как многочастотной автоколебательной системы.

Ил.4. Библиогр.: 7 назв.

УДК 389.2

**Критерій виділення квазіперіодичних складових, обумовлених взаємодією мір у груповому еталоні частоти** / Ю.І. Євдокименко, О.П. Нарєжній // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С.227-230.

Пропонується критерій виявлення значущих флуктуацій частоти і фази мір в груповому еталоні, що мають квазіперіодичний тип прояву з позицій поведінки групи як багаточастотної автоколивальної системи.

Іл.4. Бібліогр.: 7 назв.

UDC 389.2

**Criterion of quasi-periodic component detection, caused by interaction of measures in the frequency group standard** / Yu. I. Evdokimenko, A.P. Narezshny // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2002. № 126. P. 227-230.

The criterion for significant frequency fluctuation and phases of group standard measures detection is offered; these fluctuations have quasi-periodic type of manifestation in the context of group behavior as multi-frequency auto-fluctuation system.

Fig.4. Ref.: 7 items.

УДК 681.04

**Методы и алгоритмы коррекции ошибок в системах цифровой обработки информации**/ В.А. Краснобаев// Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С.231-237.

Рассматривается метод коррекции ошибок в системах цифровой обработки информации, основанный на применении непозиционной системы счисления в остаточных классах (СОК). На основании данного метода разработаны алгоритмы обнаружения и исправления ошибок в СОК, в соответствии с которыми предложены устройства для их реализации.

Табл. 4. Ил. 1. Библиогр.:12 назв.

УДК 681.04

**Методы и алгоритмы коррекции ошибок в системах цифровой обработки информации**/ В.А. Краснобаев// Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С.231-237.

Розглядається метод корекції помилок у системах цифрової обробки іінформації, який заснований на застосуванні непозиційної системи числення у залишкових класах (СЗК). На основі цього методу розроблені алгоритми виявлення та виправлення помилок в СЗК, у відповідності з котрими запропоновані пристрої для їх реалізації.

Табл. 4. Іл. 1. Бібліогр.: 12 назв.

UDC 681.04

**Methods and algorithms of error corection in the system of information digital processing** / V.A Krasnobaev// Radoitekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N. 126. P.231-237.

A methods of error correction in the system of information digital processing based on application of nonpositional number system in residual classes (SRC) is considered. Algorithms based on given method for finding and correcting errors in SRC are developed, according to them the devices for their realization are offered..

Tabl. 4. Fig. 1. Ref.: 12 titles.

УДК 681.3.07

**Опыт подготовки и результаты проведения IV Всеукраинской олимпиады среди бакалавров, студентов и магистров по направлению "Информационная безопасность"** /И.Д.Горбенко, А.В.Потий, О.А.Мельникова // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С.238-242.

Обсуждаются вопросы подготовки и проведения олимпиад по направлению "Информационная безопасность". Приводятся примеры наборов конкурсных теоретических и практических заданий. Анализируется уровень подготовки участников последней олимпиады и указываются типичные недостатки при решении конкурсных заданий.

УДК 681.3.07

**Досвід підготовки та результати проведення IV Всеукраїнської олімпіади серед бакалаврів, студентів та магістрів з напрямку "Інформаційна безпека"** / І.Д. Горбенко, О.В. Потій, О.А. Мельникова // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2002. Вип. 126. С.238-242.

Обговорюються питання підготовки та проведення олімпіад з напрямку "Інформаційна безпека". Наводяться приклади наборів конкурсних теоретичних та практичних завдань. Аналізується рівень підготовки учасників останньої олімпіади та вказуються типові недоліки при розв'язанні конкурсних завдань.

UDC 681.3.07

**The experience in preparation and results of carrying out the IV all-Ukrainian competitions among bachelors, students and masters in the field of information security** / I.D. Gorbenko, A.V. Poty, O.A. Melnikova // Radoitekhnika: All-Ukr. Sci. Interdep. Mag. 2002. N. 126. P.238-242.

The problems of preparation and carrying out competitions in the field of information security are discussed. The sets of competition theoretical and practical tasks are given as example. The competitives level of training at the latest competitions is analyzed and typical shortages in solving the competitive tasks are pointed out.

# БАГАТОРІВНЕВА СИСТЕМА ЗАХИСТУ ІНФОРМАЦІЇ

## ОПИС АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ – БАГАТОРІВНЕВИЙ ЦЕНТР СЕРТИФІКАЦІЇ І УПРАВЛІННЯ КЛЮЧІВ

### 1 Загальна характеристика ключової системи

У БЦСУК для захисту критичної інформації застосовується цифровий підпис, симетричне і спрямоване шифрування. Багаторівневість апаратно-програмного комплексу полягає в тім, що центри сертифікації і управління ключів (ЦСУК) організовані в ієрархічну деревоподібну структуру. На будь-якому рівні ієрархії кожен ЦСУК здійснює сертифікацію і управління ключами у своїй зоні управління. До складу зони управління кожного ЦСУК n-го рівня можуть входити такі елементи: ЦСУК (n+1)-го рівня, робочі місця (станції) користувачів n-го рівня, іменні генератори ключів n-го рівня. Кількість рівнів і конфігурація (структура і кількість елементів) зони управління ЦСУК будь-якого рівня задається при розгортанні конкретної системи. У БЦСУК застосовуються такі типи ключів: головні ключі, конфіденційні сеансові ключі і робочі ключі (відкриті і закриті).

Відкриті робочі і конфіденційні сеансові ключі передаються по каналах зв'язку. При цьому відкриті і робочі ключі передаються як повідомлення особливого виду, для їхнього захисту використовуються головні ключі. Робочі ключі використовуються для цифрового підпису і шифрування сеансових ключів. Сеансовий ключ використовується для шифрування основного обсягу інформації, що захищається, при цьому кожен документ або повідомлення шифрується на новому ключі.

Головні ключі використовуються для шифрування й автентифікації тільки відкритих робочих ключів при їхній передачі по каналах зв'язку, причому по каналах зв'язку передаються тільки відкриті загальносистемні параметри, відкриті ключі і бази відкритих робочих ключів. Основним призначенням головних ключів є забезпечення цілісності і справжності відкритих робочих ключів на всіх етапах їхнього життєвого циклу. Генерація головних ключів здійснюється в центрі сертифікації і управління ключами зони (ЦСУК n-го рівня). Головні ключі є спрямовуючими, тобто для кожного елемента зони управління ЦСУК n-го рівня використовуються індивідуальні ключі.

Цифровий підпис (ЦП) виконується за стандартом ДСТ 34.310-95, довжина підпису 512 біт. У стандарті реалізована криптосистема з відкритими ключами і відкритим поширенням ключів. Її застосування дозволяє забезпечити захист від обману і контроль цілісності для моделі взаємної недовіри взаємодіючих сторін.

Шифрування основного обсягу документа або повідомлення здійснюється за ДСТ 28147-89 на сеансовому ключі. Шифрування здійснюється в режимі потокового гамування.

Шифрування сеансовых ключів здійснюється з використанням семантично захищеного RSA алгоритму (Діффі – Хеллмана).

Головні ключі мають ту ж структуру і склад, що і робочі ключі. Їхня відмінність полягає в тім, що вони використовуються для вироблення підпису і шифрування відкритих робочих ключів при пересиланні їх по каналах зв'язку.

Цифровий підпис у системі захисту файлів для їхньої передачі по електронній пошті здійснюється за ДСТ 34.310-95. Цифровим підписом є 512 бітне число, що складається з двох відкритих 256-бітних чисел **r** і **s**, які передаються або зберігаються разом з підписаним файлом. Загальномережними відкритими ключовими параметрами є 1024-бітне просте число **p**, просте 256 бітне число **q**, що входять у розкладання числа **p**-1, і ціле число **a (1<a<p-1)**, таке, що $\mathbf{a^q (mod\ p)=1}$. Загальномережні параметри цифрового підпису для робочих і головних ключів різні. Конфіденційними ключами цифрового підпису є довгостроковий ключ підпису **X** (0<**X**<q) і сеансовый параметр **k** (0<**k**<**q**). Ключі **X** і **k** є 256-бітними числами і формуються датчиком випадкових або псевдовипадкових чисел і використовуються при виробленні підпису. Відкритими ключами цифрового підпису (перевірки) є довгостроковий відкритий ключ **Y** (0<**Y**<**p**) і сеансовий відкритий ключ **r** (0<**r**<**q**). Конфіденційні ключі повинні зберігатися в таємниці, генеруватися для кожного користувача і не повинні виходити з його володіння.

Для симетричного шифрування основного обсягу інформації застосовується потоковий режим ГОСТ 28147-89. У цьому режимі використовується два ключі (довгостроковий і сеансовий). Довгостроковим ключем є блок заміни – вісім випадкових підстановок 4 біти в 4 біти. Загальна довжина ключа дорівнює 512 бітів. Довгостроковий ключ змінюється рідко.

Сеансовий ключ **w** – являє собою 256-бітну випадкову послідовність, що задовольняє ряду спеціальних вимог. Формується сеансовий ключ з використанням датчиків псевдовипадкових або випадкових чисел. Для шифрування повідомлення (блоку даних) у системі щоразу генерується новий сеансовий ключ. Одночасно із сеансовим ключем за псевдовипадковим законом формується щоразу новий синхромаркер (64 бітна псевдовипадкова послідовність). Сеансовий ключ і синхромаркер передаються або зберігаються разом із зашифрованим повідомленням або даними. Сеансовий ключ і синхромаркер разом з іншими службовими

параметрами шифрується з використанням семантично захищеного алгоритму RSA (Діффі – Хеллмана) на відкритих ключах одержувача.

Семантично-захищений алгоритм спрямованого шифрування RSA використовується для спрямованого шифрування службової частини повідомлення (даних). Під спрямованим шифруванням розуміється шифрування на відкритих ключах одержувача. Наприклад, user1 використовує відкритий ключ user2 $E_2$ , а user2– відкритий ключ $E_1$ user1. Розшифрування документів кожнен з користувачів робить на своєму конфіденційному ключі $D_2$ ($D_1$), що забезпечує спрямованість передачі. В алгоритмі RSA використовуються такі ключі: відкритий ключ user1 $E_{u1}$; відкритий ключ user2 $E_{u2}$; конфіденційний ключ user1 $D_{u1}$; конфіденційний ключ user2 $D_{u2}$.

СЗИ оперує з наступними базами відкритих ключів:

1) відкритих робочих ключів;
2) відкритих застарілих робочих ключів;
3) відкритих головних ключів;
4) відкритих застарілих головних ключів.

У базу відкритих ключів записуються відкриті ключі користувачів. Конфіденційні ключі записуються на дискету.

## 2 Склад і структура системи управління ключовими структурами

Центр сертифікації і управління ключами n-го рівня призначений для:

- генерації, розподілу і видачі головних ключів, контролю за їхнім застосуванням, планової зміни та зміни у випадку компрометації;
- генерації робочих ключів користувачів центра;
- розподілу відкритих робочих ключів центра всім користувачам;
- прийому і сертифікації відкритих робочих ключів користувачів;
- формування і передачі користувачам бази відкритих ключів;
- включення нових користувачів у СЗИ;
- віддалення користувачів зі СЗИ.

ЦСУК n-го рівня реалізується, як правило, на виділеному персональному комп'ютері. Інсталяція центра здійснюється відповідно до посібників оператора ЦСУК.

Ключ сертифікації генерується за письмовим наказом керівництва Замовника. У розпорядженні вказуються ідентифікатори користувачів, посади і прізвища користувачів у СЗІ, а також термін і умови підключення користувача до обслуговування. На підставі розпорядження оператор ЦСУК генерує транспортний ключ для організації ключового напрямку між центром і користувачом. Відкриті ключі записуються в базу відкритих транспортних ключів центра. Факт генерації ключів для абонента і ключовий носій реєструються в журналі обліку транспортних ключів центра з вказівкою ідентифікатора, прізвища оператора ЦСУК, часу і дати генерації транспортного ключа. Запис у журналі завіряється оператором. Видача генератора ключів, носія транспортного ключа і загальномережних параметрів представнику користувача виробляється під підпис у журналі обліку транспортних ключів із указівкою вихідного номера розпорядження представника. Транспортні ключі реєструються в журналі обліку і видачі ключів під підпис представника користувача.

Генерація робочих ключів користувача виробляється після інсталяції робочої станції, а також при плановій і аварійній зміні ключів.Робочі ключі можуть генеруватись усім або деяким користувачам на ЦСУК n-го рівня. Загальномережні параметри цифрового підпису генеруються в ЦСУК n-го рівня і є загальними для всіх користувачів його зони управління. Передача загальномережних параметрів, як правило, виробляється разом з генератором ключів і інсталяційним пакетом. Для їхнього захисту використовуються транспортні ключі.

Після одержання загальномережних параметрів користувач, який входить у зону управління ЦСУК n-го рівня, генерує свої робочі ключі, використовуючи вилучений генератор ключів (УГК) n-го рівня. Конфіденційні ключі цифрового підпису і розшифрування записуються в захищеному виді на дискету. Відкриті складові робочих ключів передаються на ЦСУК n-го рівня для їхньої сертифікації і запису в базу відкритих ключів. Про виконані дії користувач робить запис у журнал обліку і видачі ключових носіїв. Носій з конфіденційними робочими ключами опечатується в щільний конверт і зберігається в сейфі або спеціальному сховищі. При відсутності сейфа ключовий носій здається по закінченні робочого часу в опечатаному виді на збереження.

Носій з робочими ключами видається на початку робочого дня або у разі необхідності. Робочий ключ центра генерується аналогічно. У випадку відмови в сертифікації ключів відповідні користувачі повинні здійснити повторну генерацію робочих ключів і відправлення відкритих ключів для сертифікації в ЦСУК свого рівня.

## 3 Поводження з ключами і носіями ключових документів

Робочі ключові дані і ключові параметри генеруються на ЦСУК або за допомогою іменних генераторів ключів користувача відповідно до інструкції. В останньому випадку конфіденційні робочі ключі генеруються особисто кожним користувачем, записуються на зареєстрований носій, обліковуються і зберігаються як конфіденційні документи. Усі ключові носії з записаними ключовими даними реєструються в журналі обліку

видачі ключів. У журналі враховується дата генерації ключа, ідентифікатор або код користувача, прізвище і підпис особи (осіб), що виконували генерацію, а також обліковий номер носія і номер сейфа, де цей носій зберігається. Система дозволяє визначити терміни дії транспортних і робочих ключів при інсталяції ЦСУК n-го рівня. Рекомендується встановлювати такі терміни дії:

- ключа сертифікації і транспортних ключів – не більш 365 днів;
- робочих ключів– не більш 30 днів.

Загальномережні параметри і відкриті ключі записуються і зберігаються в захищеній базі відкритих ключів. Допускається запис бази відкритих ключів на спеціальний обліковий носій у виді копії. Носій бази відкритих ключів повинен зберігатися в сейфі, в окремому щільному конверті в опечатаному виді. На конверті вказується термін дії і початок застосування цієї бази в якості діючої.

Носії з конфіденційними особистими ключами повинні зберігатися в сейфі або залізних шухлядах в опечатаному виді окремо від інших документів і носіїв. Допускається спільне зберігання конвертів із транспортними і робочими ключами. Конверти і носії позначаються умовними фразами "Головний - Z", "Робочий - Z", де Z є строк вводу та термін дії ключів.

Кожен факт використання конфіденційних ключів реєструється в спеціальному журналі автоматично. Конфіденційні ключові дані, термін яких минув, стираються з носіїв, про що робиться запис у журналі обліку з підписом двох осіб, одна із яких – користувач, а друга – визначена наказом або розпорядженням. Повторне використання ключових дискет допускається тільки після їхнього форматування. Ключові дані можуть мінятися планово або аварійно. Планова зміна виробляється у визначений час, по закінченню терміну дії ключів. Аварійна (раптова) зміна ключових даних здійснюється у випадку компрометації діючих.

Під компрометацією конфіденційних ключів розуміється факт розголошення або підозри на розголошення цих ключів. Компрометація може бути явною і неявною. Ознаками явної компрометації є:

- установлення факту створення "зловмисником" документа і формування цифрового підпису від імені санкціонованого користувача;
- установлення факту розшифрування "зловмисником" не призначеного йому повідомлення;
- передача носія з конфіденційними ключами сторонній особі;
- розкрадання або втрата носіїв з діючими ключами;
- проникнення в сейф і порушення цілісності упакування з конфіденційними ключами;
- компрометація паролів доступу до ключових носіїв.

При виявленні ознак явної компрометації користувач і представник відділу безпеки Замовника приймають рішення про вилучення скомпрометованих ключів з дії і припинення роботи у встановленому порядку. Робота на скомпрометованих ключах категорично заборонена. Після цього приймаються заходи для перекриття погроз компрометації, генеруються нові ключі, на яких відновлюється система. При неявній компрометації, тобто в інших випадках, які обговорені вище, виконується розслідування і установлюється факт наявності компрометації і приймаються рішення про перекриття всіх можливих погроз.

Усі ключові дані і ключові параметри перед використанням обов'язково контролюються на цілісність і справжність автоматично. При виявленні факту порушення цілісності ключів і ключових параметрів робота з ними неможлива. Для відновлення ключів використовується їхня копія, якщо така мається. Якщо копії ключів відсутні, за взаємним узгодженням між користувачем і центром відбувається перехід на чергові ключі. При порушенні цілісності відкритих ключів виробляється запит бази відкритих ключів від ЦСУК, і вона передається з використанням головних ключів. При порушенні цілісності транспортних ключів відновлення здійснюється за допомогою генерації й одержання на ЦСУК нових транспортних ключів представником користувача. Після відновлення ключів виробляється аналіз причин порушення цілісності і виробляються пропозиції щодо їхнього усунення.

Організація і проведення контролю за виконанням вимог при експлуатації БЦСУК "Цитадель-НС" покладається на підрозділи інформаційної безпеки Замовника. Контроль повинен включати планові і позапланові перевірки відповідно до чинного законодавства.

## КРИПТОМАРШРУТИЗАТОР МІЖМЕРЕЖНОГО ПОТОКУ ДАНИХ СИСТЕМИ ЗАХИСТУ "ЦЕНТАВР"

### 1 Забезпечення безпеки мережі з використанням шлюзів безпеки

Для забезпечення безпеки внутрішніх і зовнішніх мереж і мережних ресурсів організації існує цілий ряд засобів. Одними з основних компонентів таких засобів є брандмауери – шлюзи безпеки.

Шлюз безпеки дозволяє захищати потоки даних, що проходять через нього, між локальними мережами або між локальною мережею і віддаленим користувачем. Безпечну передачу даних між двома системами через зовнішню (небезпечну мережу) забезпечує тунель безпеки. Процес передачі даних через тунель припускає використання правил безпеки, прийнятих у системах, між якими ці дані передаються. Ці правила містять у собі адреси сторін, що з'єднуються, методи форматування (відповідно до яких інформація буде вміщуватися в блоки

даних іншого протоколу), криптографічні алгоритми і параметри цих алгоритмів (ключі). Дані, що проходять через шлюз безпеки, форматуються відповідно до встановленої політики безпеки.

Настроювання політики безпеки для шлюзу забезпечується настроюванням таблиць тунелювання і таблиць маршрутизації (для шлюзу безпеки) чи таблиць трансляції мережних адрес (на хості віддаленого користувача). Таблиці тунелювання містять адреси шлюзів безпеки (чи хостів віддалених користувачів), між якими створюється захищений тунель, контекст безпеки і тип форматування потоку даних. Таблиці маршрутизації (чи трансляції мережних адрес) містять дані, необхідні для приховання мережних адрес хостів, що працюють через шлюз безпеки, і забезпечення їхнього проходження між шлюзами або хостами в зовнішніх мережах.

## 2. Застосування криптомаршрутизатора (шлюзу безпеки)

Криптомаршрутизатор (шлюз безпеки), розроблений для створення тунелів безпеки між локальними мережами. Для захисту потоку даних між двома локальними мережами використовуються два шлюзи безпеки, установлені на виході кожної з мереж. Вихідний потік даних локальної мережі обробляється, інкапсулюється і маршрутизується на другий шлюз безпеки відповідно до встановлених правил безпеки й обмеження доступу. Шлюз безпеки віддаленої мережі обробляє вхідний потік даних з першого шлюзу – відкриває дані і передає їх у свою локальну мережу. При включенні шлюзів і завантаженні тунелів безпеки здійснюється їхня взаємна ідентифікація й установка контекстів безпеки і форматування потоку даних.

Криптомаршрутизатор (шлюз безпеки), реалізований у виді додаткової робочої станції, установлюваної на виході локальної мережі. При установці на шлюз інсталюється необхідне програмне забезпечення "Криптомаршрутизатора". Управління криптомаршрутизатором (шлюзом безпеки), може здійснюватися локально або віддалено. При віддаленому керуванні адміністратор безпеки мережі виконує настроювання шлюзу безпеки через свого програмного агента встановленого на шлюзі. Для віддаленого управління на робочу станцію адміністратора безпеки встановлюється програмне забезпечення "Віддаленого адміністратора шлюзів безпеки "Криптомаршрутизатор".

## 3. Реалізація шлюзу безпеки "Криптомаршрутизатор"

Шлюз безпеки "Криптомаршрутизатор" являє собою маршрутизатор або робочу станцію і забезпечує захист мережного трафіка, що проходить через нього, між локальними мережами або між локальною мережею і віддаленим користувачем. Шлюз безпеки реалізує такі основні функції:

- тунелювання (захист та інкапсуляцію) мережного трафіка між шлюзами;
- встановлення і тунелювання мережних з'єднань з віддаленими користувачами;
- диспетчеризацію даних віддаленого адміністратора безпеки мережі для:
- віддаленого завантаження ключових даних;
- віддаленого настроювання параметрів тунелів і маршрутів безпеки;
- віддаленого конфігорування мережного вузла шлюзу;
- моніторинг стану шлюзу, мережної підсистеми і з'єднань.

Шлюз безпеки реалізований у виді робочої станції (системного блоку), яка встановлюється на виході локальної мережі, і може поставлятся у виді системного блоку (без монітора) чи пакета установки програмного забезпечення. Технічні вимоги до апаратних засобів шлюзу наведені в таблиці 5.1.

Таблиця 5.1.Технічні вимоги до апаратних засобів шлюзу

| Тип конфігурації | Частота процесора, МГц | Тип мережного адаптера** |
|---|---|---|
| Мінімальна | 166 | Ethernet 10 |
| Середня | 300 – 600 | Ethernet 10/100 |
| Максимальна* | вище 600 | Ethernet 10/100 |

Примітка. * – найбільш ефективна;

** – при якій шлюз з даною конфігурацією функціонує найбільш ефективно.

Шлюз безпеки виконує обробку мережного трафика (мережний протокол IPv4) і підтримує локальні мережі стандарту IEEE 802.3 і ANSI X3T9.5. Шлюз безпеки може мати декілька мережних адаптерів для підключення декількох мережних сегментів Програмною платформою шлюзу безпеки є операційна система Windows 2000 Professional. Програмне ядро шлюзу безпеки використовує частина вбудованих мережних компонентів (зокрема програмний маршрутизатор) операційної системи. Програмне ядро (оброблювач трафіка) побудовано на основі проміжного драйвера NDIS, що вбудовується в підсистему мережних драйверів. При завантаженні тунелю безпеки встановлюється захищене з'єднання (взаємна ідентифікація й автентифікація) між шлюзами безпеки чи шлюзом безпеки й агентом віддаленого користувача. Створення захищеного тунелю між шлюзами безпеки або шлюзом безпеки й агентом вилученого користувача забезпечується за рахунок інкапсуляції, шифрування і імітозахисту пакетів з використанням пакетних чи тунельних ключів. У процесі інкапсуляції пакетів вихідний IP-заголовок заміняється IP-заголовком тунелю безпеки, забезпечуючи передачу

інкапсульованого пакета на шлюз безпеки або вузол віддаленого користувача. Пакетні ключі генеруються при захисті пакета. Тунельні ключі генеруються на робочій станції адміністратора безпеки мережі і встановлюються віддалено. Контекст тунелю безпеки містить лічильник пакетів, що забезпечує захист від повторів інкапсульованих пакетів. Шлюз безпеки може автоматично відключати "ушкоджені" тунелі.

Криптографічний захист у тунелі безпеки здійснюється з використанням потокового шифрування кожного IP-пакета й обчислення імітовставки на пакетних чи тунельних ключах. У якості базових використовуються алгоритми ГОСТ 28147-89 або Х-ГОСТ 28147-89.

Управління шлюзом безпеки може здійснюватися локально або віддалено. Віддалене управління шлюзом безпеки здійснюється з робочої станції адміністратора безпеки мережі з використанням програмного агента віддаленого адміністратора. Віддалений адміністратор може набудовувати таблиці тунелювання і маршрутизації, з'єднань і встановлювати згенеровані тунельні і транспортні ключі. При віддаленому управлінні шлюз безпеки функціонує в автоматичному режимі. При підключенні віддаленого адміністратора до шлюзу безпеки виконується їхня взаємна ідентифікація. Для захисту даних управління використовуються багатоетапні спроможні протоколи, що базуються на криптографічних алгоритмах ДСТ 28147-89, ДСТ 34310-95, ДСТ 34311-95, а також на протоколі X9.42.

Програмне ядро криптомаршрутизатора складається з таких компонентів:

1) КМ Відлагоджувальна Консоль – забезпечує графічний інтерфейс локального управління криптомаршрутизатором. Реалізована у виді оснащення MMC (Microsoft Management Console);
2) КМ Управляюча Служба – призначена для автоматичного настроювання криптомаршрутизатора при завантаженні операційної системи. Реалізована у виді служби користувальницького режиму (User-mode Own-process Service);
3) КМ Програмний Інтерфейс – являє собою оболонку основних програмних компонентів ядра криптомаршрутизатора. Реалізований у виді розширення Програмного Інтерфейсу (API) і містить у собі такі складені компоненти:
   - КМ Криптографічний Програмний Інтерфейс (КПІ) – надає функції управління транспортними ключами, а також функції управління контейнерами даних. Реалізований у виді бібліотеки динамічного компонування.
   - КМ Криптографічна Бібліотека (КБ) – надає криптографічні функції для інтерфейсу управління транспортними ключами. Реалізований у виді бібліотеки динамічного компонування CSP с інтерфейсом CryptoSPI.
   - КМ УА Криптографічний Програмний Інтерфейс (КПІ) – надає функції управління ключами для віддаленого адміністрування, а також функції захисту даних у каналі віддаленого управління. Реалізований у виді бібліотеки динамічного компонування.
   - КМ УА Криптографічна Бібліотека (КБ) – надає криптографічні функції для інтерфейсу управління ключами для віддаленого адміністрування і захисту даних у каналі віддаленого управління. Реалізований у виді бібліотеки динамічного компонування CSP з інтерфейсом CryptoSPI.
   - КМ Агент Віддаленого Адміністратора (АВА) – забезпечує віддаленого управління, диспетчеризуючи дані віддаленого адміністратора. Реалізований у виді бібліотеки динамічного компонування серверного розширення WSP.
   - КМ Консоль Віддаленого Управління (КВУ) – призначена для віддаленого відображення стану з'єднання в каналі управління. Використовується в КМ АВА. Реалізована у виді бібліотеки динамічного компонування.
   - КМ Конфігуратор Мережного Вузла (КМВ) – призначений для динамічного настроювання мережного вузла криптомаршрутизатора. Набудовує таблиці мережних адрес, таблиці маршрутизації і управляє мережними інтерфейсами. Реалізований у виді бібліотеки динамічного компонування розширення інтерфейсу IPHelpAPI.
   - КМ Програмний Інтерфейс Драйвера (ПІД) – забезпечує управління драйвером криптомаршрутизатора (КМ Драйвер – КМ Д). Реалізований у виді бібліотеки динамічного компонування.
   - КМ Управляючий Драйвер (УД) – забезпечує передачу управляючої інформації в КМ Д. Реалізований у виді мережного драйвера-протоколу NDIS (NDIS Protocol-driver).
   - КМ Драйвер (Д) – забезпечує обробку мережного потоку даних. Реалізований у виді проміжного драйвера NDIS (NDIS Intermediate Driver).

## 4 Віддалений адміністратор шлюзів безпеки "Криптомаршрутизатор"

Програмне забезпечення віддаленого адміністратора шлюзів безпеки "Криптомаршрутизатор" установлюється на робочу станцію адміністратора безпеки мережі і забезпечує віддалене управління шлюзами й агентами віддалених користувачів. Віддалений адміністратор реалізує такі основні функції:

- настроювання таблиць шлюзів безпеки й агентів віддалених користувачів;

- створення ключових даних для захисту каналу управління і експорт їх на зовнішній носій для передачі на встановлюваний шлюз безпеки або агент віддаленого користувача;
- створення ключових даних для захисту каналу встановлення з'єднання і експорт їх на зовнішній носій для передачі на встановлюваний шлюз безпеки або агент віддаленого користувача (для тунелів "шлюз-віддалений користувач" або "віддалений користувач-віддалений користувач");
- настроювання таблиць тунелів безпеки між шлюзами безпеки;
- створення тунельних ключів і їхню віддалену встановку в шлюз безпеки або агент віддаленого користувача (для тунелів "шлюз-шлюз");
- настроювання таблиць маршрутизації шлюзів;
- збір і резервне копіювання конфігурації шлюзів безпеки;
- моніторинг стану шлюзів і мережних з'єднань.

Віддалений адміністратор реалізується у виді робочої станції адміністратора безпеки мережі з установленим відповідним програмним забезпеченням і може постачатися у виді робочої станції або пакета установки програмного забезпечення. Програмною платформою віддаленого адміністратора є операційні системи Windows (9x, ME, NT, 2000 чи XP).

Віддалене управління шлюзом безпеки або віддаленим користувачем здійснюється з робочої станції віддаленого адміністратора з використанням програмного агента шлюзу безпеки або віддаленого користувача. Віддалений адміністратор може набудовувати таблиці тунелювання, маршрутизації та з'єднань і встановлювати згенеровані тунельні і транспортні ключі на шлюзах безпеки та віддалених користувачів. При підключенні віддаленого адміністратора до шлюзу безпеки виконується їхня взаємна ідентифікація. Для захисту даних управління використовуються багатоетапні спроможні протоколи, що базуються на криптографічних алгоритмах ДСТ 28147-89, ДСТ 34310-95, ДСТ 34311-95, а також на протоколі X9.42. Ключові дані представлені у відповідності до стандартів X.509.

Програмне забезпечення віддаленого адміністратора складається з таких компонентів:

1) ВА Управляюча Консоль – призначена для представлення графічного інтерфейсу віддаленого управління криптомаршрутизаторами. Реалізована у виді оснащення MMC (Microsoft Management Console);
2) ВА Програмний Інтерфейс – являє собою оболонку основних програмних компонентів віддаленого управління. Реалізований у виді розширення Програмного Інтерфейсу (API) і містить у собі такі складені компоненти:
   –ВА Диспетчер З'єднання (ДЗ) – забезпечує віддалене підключення до криптомаршрутизатора і передачу управляючої інформації. Реалізований у виді бібліотеки динамічного компонування користувачського розширення WSP.

## КОРОТКИЙ ОПИС АПАРАТНО-ПРОГРАМНИХ ЗАСОБІВ ЗАХИСТУ ІНФОРМАЦІЇ СЕРІЇ "ГРЯДА"

Ряд апаратно-програмних засобів серії "Гряда" містить у собі процесори (модулі) "Гряда-1", "Гряда-1М", "Гряда-11", "Гряда-31", "Гряда-31М" та «Гряда-41» (знаходиться на стадії виготовлення). Коротка характеристика апаратних засобів "Гряда" наведена в таблиці 1.

| Найменування | Призначення | Інтерфейс | Процесор |
|---|---|---|---|
| «Гряда-1» | Генератор випадкових послідовностей (чисел) | ISA | Немає |
| «Гряда-1М» | Генератор випадкових послідовностей (чисел) | RS-232 | Немає |
| «Гряда-11» | Процесор КЗІ | ISA, RS-232 | ADSP 21062 |
| «Гряда-31» | Процесор КЗІ | PCI-32, RS-232 | ADSP 21062 |
| «Гряда-31М» | Процесор КЗІ | PCI-32, RS-232 | ADSP 21061 |
| «Гряда-41М» | Процесор КЗІ | PCI-32, RS-232 | 2*ADSP 21065L |

Основними функціями криптографічного захисту, що реалізовані в "Гряді-11", "Гряді-31", "Гряді-31М" і "Гряді-41М"на апаратному рівні, є:

- криптографічне перетворення інформації з використанням стандартних криптографічних алгоритмів (стандартів);
- генерація ключів цифрового підпису, симетричного і спрямованого шифрування;
- збереження конфіденційних ключів і сертифікатів;
- завантаження ключів із зовнішнього, незалежного від ОС, носія (електронна картка touch memory і ін.);
- запис згенерованих ключів на зовнішній носій або передача ПЗ за слушним протоколом поза процесором;

- завантаження і збереження спеціальних даних, параметрів ОС і ін.;
- генерація випадкових послідовностей (ключів, підписів, параметрів і ін.);
- збереження в зашифрованому виді ПЗ самого процесора і критичного ПЗ ОС;
- виконання різних перетворень з метою забезпечення захисту від НСД;
- забезпечення цілісності і конфіденційності критичного ПЗ зовнішніх ОС;
- виконання різних фіскальних функцій.

Підвищення рівня захисту інформації і захисту від НСД здійснюється на апаратному рівні за рахунок застосування таких методів інженерної криптографії:

- забезпечення мінімально можливої взаємодії прикладного ПЗ й ОС ПЕОМ з апаратними засобами (АЗ "ГРЯДА") за рахунок:
- використання спеціального командного (процедурного) інтерфейсу взаємодії прикладного ПЗ з АЗ "ГРЯДА", що забезпечує неможливість прямого доступу прикладного ПЗ й ОС до ресурсів АЗ "ГРЯДА" (пам'яті, процесору і т.п.) і обов'язкову початкову ініціалізацію АЗ "ГРЯДА" з застосуванням ключів (паролів) доступу заданої довжини (не менш 8 байт);
- взаємодію прикладного ПЗ (ОС) з АЗ "ГРЯДА" через «прозору» для користувача «поштову скриньку» (буфери обміну – прийомний і передавальний, які фізично розташовані в АЗ "ГРЯДА" і недоступні для прямого читання/запису прикладному ПЗ й ОС);
- збереження криптографічних алгоритмів (модулів, що виконуються,) в енергонезалежній пам'яті (ЕНП) АЗ "ГРЯДА" у зашифрованому виді (таким чином відпадає необхідність завантаження алгоритмів з зовшішнього середовища);
- застосування резервного (дубльованого) генератора випадкових чисел для формування фізичних ключів заданої довжини;
- завантаження криптографічних алгоритмів, що виконуються, з ЕНП в убудоване ОЗП (RAM) мікропроцесора в шифрованому виді;
- виконання криптографічних алгоритмів тільки з убудованого ОЗП мікропроцесора (з попереднім розшифруванням безпосередньо в даному ОЗП);
- збереження шифрованих даних у ЕНП у структурованому виді з застосуванням різних методів контролю (наприклад, контрольне підсумовування програмних модулів, що завантажуються, для забезпечення контролю вірогідності і цілісності інформації);
- передачі даних (ключової інформації, що завантажується, криптографічних алгоритмів, результатів проміжних обчислень і т.д.) по внутрішніх шинах АЗ "ГРЯДА" тільки в зашифрованому виді;
- «прошивання» окремих ключів (наприклад, ключа, який використовується при ініціалізації АЗ "ГРЯДА") у «залізі» із забезпеченням гарантованої неможливості «добування»;
- застосування «схемотехніки, що оперативно завантажується» в апаратних прискорювачах;
- використання зовнішніх ключів, що завантажуються користувачем по окремому інтерфейсу зі спеціальних зовнішніх накопичувачів (наприклад, електронна картка рідера);
- застосування в АЗ "ГРЯДА" спеціального захисного екрану, що забезпечує захист від НСД самого АЗ "ГРЯДА" (при необхідності);
- включення до складу АЗ "ГРЯДА" апаратного розширення BIOS ПК (64Кбайт), що дозволяє контролювати (до завантаження ОС) цілісність ОС і прикладного ПЗ (наприклад, фізично розташованого на жорсткому диску ПК).

Підвищення продуктивності криптографічних операцій розглядається тільки в контексті з підвищенням рівня захисту інформації і здійснюється за рахунок:

- застосування в якості ядра в АЗ "ГРЯДА" максимально адаптованих для реалізації більшості криптографічних алгоритмів \ операцій (наприклад, RIJNDAEL) сучасних високопродуктивних мікропроцесорів (сигнальних процесорів DSP) на основі RISC архітектури;
- застосування «схемотехніки, що оперативно завантажується» апаратних прискорювачів (необхідних у ряді випадків), що на апаратному рівні реалізують окремі елементи криптографічних алгоритмів (наприклад, таблицю підстановок у ДСТ-28147) або цілком криптографічні алгоритми (наприклад,RIJNDAEL), що не зовсім оптимально «накладаються» на ядро АЗ "ГРЯДА" (мікропроцесор) з погляду тимчасових витрат на їхнє виконання;
- застосування в окремих обгрунтованих випадках багатопроцесорних структур (2-6 процесорів) на базі одного уніфікованого мікропроцесора (наприклад, процесор КЗІ «Гряда-41М» на базі 2-х сигнальних процесорів ADSP 21065L).

«Гнучка» адаптація до вимог замовника припускає реалізацію широкого кола задач потенційних замовників без зміни «заліза» АЗ "ГРЯДА". Функціональна універсальність припускає можливість використання того самого АЗ "ГРЯДА" без будь яких змін (доробок) у різних функціональних режимах, наприклад, у режимі робочої станції (основний режим) або в режимі центру управління і сертифікації ключів.

До складу АЗ "ГРЯДА" входять тести самоперевірки (ТСП), які:

- забезпечують повну перевірку правильності функціонування АЗ "ГРЯДА" як на етапі виготовлення, так і при експлуатації її користувачем;
- прошиваються в BIOS АЗ "ГРЯДА" на етапі виготовлення, тобто є убудованими;
- виконуються щораз після апаратного (по вмиканню живлення) або програмного скидання АЗ "ГРЯДА".

Передбачається також і оперативне тестування кожної команди (режиму функціонування) АЗ "ГРЯДА" у виді повернення в прикладну програму статусу завершення даної команди (режиму функціонування) АЗ "ГРЯДА".

Крім того, спеціальні вимоги накладають обмеження на конструктивне виконання АЗ "ГРЯДА" (топологію друкованої плати і т.д.) у частині:

- припустимого рівня електромагнітних перешкод, випромінюваних АЗ "ГРЯДА";
- перешкодозахищеності АЗ "ГРЯДА";
- захисту від електростатичного розряду;
- захисту від зовнішніх електромагнітних полів;
- захисту від електричних перевантажень;
- захисту від теплових перевантажень (перегріву).

Вимоги до мікропроцесора (сигнального процесору):

- архітектура RISC + Harvard Architecture (не менше);
- робоча частота не менше 40 Мгц;
- убудовані: помножитель, баррельний зсувач, регістри загального призначення (Рони), КЕШ пам'ять (Cache memory) команд, ОЗП (RAM), високошвидкісні канали DMA;
- обсяг убудованого ОЗП не менш 64 Кбайта;
- розрядність арифметико-логічного пристрою, помножителя, зсувача не менш 32 розрядів;
- час виконання команди не більш 25 нс;
- апаратна і програмна підтримка багатопроцесорних структур (від 2 до 6 процесорів);
- апаратна і програмна підтримка оверлеїв.

Вимоги до ЕНП:

- обсяг накопичувача не менш 1Мбайта (з урахуванням розміщення: BIOS, тестів самоперевірки, розширення BIOS ПК і зашифрованих криптографічних алгоритмів);
- розрядність шини даних 8, 16 розрядів;
- цикл читання не більш 120 нс;
- кількість циклів стирання/запис не менш 1000000.

Апаратний прискорювач (якщо є необхідність у такому) може бути реалізований на БІСА FPGA типу FLEX (термінологія Intel, Altera), що забезпечує:

- можливість розміщення на кристалі досить великого обсягу апаратних засобів, необхідних для реалізації того або іншого криптографічного алгоритму (або окремих його частин), при порівняно невеликій вартості і малих розмірах корпусу даних БІС;
- «динамічне завантаження» схемотехніки апаратного прискорювача щораз при подачі живлення на АЗ (у не запитанному АЗ БІС апаратного прискорювача «порожня»), що підвищує загальний захист АЗ від НСД;
- можливість розміщення досить складної, а значить «об'ємної» схемотехніки на обмеженому за геометричними розмірами конструктиві;
- можливість сполучення схемотехніки із сучасними високошвидкісними мікропроцесорами (сигнальними процесорами);
- продуктивність операцій защифрування та розшифрування даних ПЗ алгоритмах: ГОСТ 28147-89 - 22 Мбіт/сек , DES - 48 Мбіт/сек, TDES – 16 Мбіт/сек (у перспективі можливе збільшення продуктивності ПЗ DES – до 200 Мбіт/сек), RIJNDAEL-25 мбіт/сек.

Забезпечення принципу мінімально можливої взаємодії прикладного ПЗ й ОС ПК з АЗ висуває такі вимоги до системної шини, що забезпечує зв'язок АЗ із ПК:

- розрядність шини даних не менш 32 розрядів;
- обмін блоками зі швидкістю не менш 132 Мб/сек;
- обмін з квитуванням;
- шина повинна бути сучасноюі розповсюдженою у ПК.

Найбільш повно відповідає даним вимогам шина PCI-32/33 МГц. Контролер інтерфейсу шини PCI-32/33 МГц реалізує протокол шини PCI-32 з частотою 33 МГц відповідно до специфікації PCI V2.1 PCISIG (target).

З урахуванням вимог по захисту від НСД, а також забезпечення принципу мінімально можливої взаємодії прикладного ПЗ й ОС з АЗ, останні конструктивно виконані у виді плати для ПК IBM/PC із системною шиною PCI-32/33 Мгц, що займає одне стандартне посадкове місце в ПК. Конструктивно плата являє собою багатошарову друковану плату розміром не більш 110 х 190 мм із ламельним рознімання по довгій стороні плати для шини PCI-32/33 Мгц.

Для реалізації зв'язку з рідером можливі два варіанти:

- розміщення зовнішнього рознімання (DB9M під RS-232C, series "A"/"B" connectors під USB) на короткій стороні плати, якщо рідер (під Smard Card, MiniKey, Touch Memory) розташований поза ПК;
- розміщення на платі мініраз'єму для зв'язку з рідером по окремому інтерфейсу, якщо рідер розташований усередині ПК.

В останньому випадку інтерфейс плати з рідером може бути реалізований на базі спрощеного RS-232C (три лінії зв'язку: прийняті дані, передані дані і "загальний") по чисто програмному протоколу.

В технічно обгрунтованих випадках можливе підвищення надійності АЗ "ГРЯДА" за рахунок "гарячого" резервування (дублювання/троірування) АЗ "ГРЯДА". Прикладом реалізації 2-х процесорної системи в одному конструктиве плати з шиною PCI 32 є процесор КЗІ «Гряда-41». У платі реалізована 2-х процесорна модель на базі сучасних високошвидкісних сигнальних процесорів ADSP 21065L Analog Devices Inc. с тактовою частотою 60 Мгц із загальною пам'яттю ємкістю 16-32 К х 32р. Плата може працювати у двох режимах, що задаються зовнішньою прикладною програмою:

- режим максимальної продуктивності (рівнобіжна 64 бітова обробка криптографічних операцій (наприклад, цифровий підпис)) для «швидких» додатків;
- режим максимальної надійності (кожен сигнальний процесор працює як незалежний канал резервування дубльованої системи) для особливо важливих додатків.

Поверхня плати покрита захисним покриттям. На платі нанесене відповідне маркірування. На ламель плати нанесене покриття з золота. Для усунення можливості НСД додатково на плату (з боку розташування електронних компонентів) установлюється спеціальний захисний екран. Екран установлюється на етапі інсталяції плати. Порушення цілісності захисного екрана при експлуатації користувачем приводить до "зависання" ПЕОМ або одержанню статусу завершення "Недійсна команда" при першому ж звертанні з боку комп'ютера до плати. Після чого плата повинна повторно пройти етап інсталяції на спеціальному технологічному стенді.

Експлуатація дослідних зразків плати "ГРЯДА-31" показала наступні особливості застосування АЗ "ГРЯДА" – BIOS ПК повинен підтримувати технологію Plug&Play, а саме специфікації: BIOS Boot Specification ver. 1.00; Plug&Play BIOS Specification ver. 1.0A; PCI Specification V2.1 (2.2) PCISIG.

Для взаємодії прикладного ПЗ з АЗ "ГРЯДА" розроблені спеціальні драйвери під операційні системи Window'95, Window'98, Windows NT. Для забезпечення режиму багатозавданнєвості ОС, дані драйвери мають "модульну" структуру, тобто "розбиті" на ряд функцій, що послідовно викликаються прикладним ПЗ.

## ОПИС ПРОГРАМНОГО КОМПЛЕКСУ – "БІБЛІОТЕКА ПРОГРАМ КРИПТОГРАФІЧНИХ ПЕРЕТВОРЕНЬ"

### 1 Загальні відомості про програму

Бібліотека призначена для виконання основних криптографічних операцій, таких як установка і перевірка цифрового підпису, шифрування і розшифрування, а також сервісних функцій, таких як робота з журналом (файл протоколу), читання ключів, обслуговування криптографічного контейнера. Для роботи бібліотеки потрібен процесор типу PENTIUM і вище та операційна система типу WINDOWS - 95 і вище. Бібліотека поставляється як DLL, яку можна використовувати в 32-бітних програмних середовищах типу VISUAL C++, BORLAND C++ BUILDER, DELPHI.

Бібліотека містить у собі сертифіковані модулі для цифрового підпису (ГОСТ 34.310-95, ГОСТ 34.311-95) і шифрування (ГОСТ 28147-89).

### 2 Структура програми

Бібліотека має ім'я **shld32a.dll.** Для використання бібліотеки необхідні додаткові файли:

- errors. h - заголовний файл для визначення діагностичних повідомлень.
- Shld32.h - заголовний файл для визначення функцій бібліотеки.

2.1 Робота з журналом. Для протоколювання виконуваних криптографічних операцій використовується файл протокола (журнал). Система щодня створює новий журнал, забезпечує збереження попередніх журналів у захищеному виді. Робота системи повинна починатися з функції ініціалізації журналу. Процедура перевіряє наявність і цілісність файла протоколу. Цю процедуру необхідно викликати найпершою. Можливі наступні варіанти роботи процедури:

1) є сьогоднішній протокол і його цілісність не порушена - відбувається вихід із процедури.

2) немає сьогоднішнього протоколу, але є вчорашній і його цілісність не порушена - створюється сьогоднішній протокол, учорашній протокол записується під ім'ям LOG\<дата>.LOG і підписується.

3) порушена цілісність вчорашнього або сьогоднішнього протоколу - протокол записується під ім'ям LOG\<дата>.E<nn> і підписується.

У випадках 2 і 3 у дисководі повинна знаходитись робоча дискета. Пароль повинний збігатися з паролем, обраним для робочої дискети. Шлях до файлу конфігурації фактично збігається з каталогом, де встановлене робоче місце користувача. При завданні ідентифікатора користувача локальне ім'я, з якого починається ідентифікатор, повинно бути доповнено пробілами до 8 символів включно. Глобальне ім'я повинно записуватись за локальним.

2.2 Створення контейнера. Всі криптографічні операції виконуються в спеціальному контейнері, що містить ключі, вихідні дані і результати виконання операцій. Контейнер повинний бути створений до виконання першої криптографічної операції, але після формування журналу. Для кожного користувача повинний бути створений свій контейнер. Для створення контейнера використовується функція, яка повертає дескриптор контейнера (далі він позначається як h). Якщо для контейнера не вистачає пам'яті, повертає -1.

2.3 Знищення контейнера. Після завершення криптографічних операцій з даним контейнером контейнер повинний бути знищений. Для знищення контейнера використовується функція: CRYPTAPI WORD. Функція завершується з помилкою при спробі знищення контейнера, що не створювався, або якщо об'єкти контейнера зайняті виконанням криптографічної операції

2.4 Створення копії контейнера. Функція використовується, якщо необхідно виконувати операції для того самого користувача, наприклад використовувати один контейнер різними потоками. Створення копії контейнера здійсьнюється з усіма завантаженими в нього ключами. Повертає дескриптор на новий контейнер.

2.5 Завантаження ключових даних у контейнер. Функція повинна бути виконана до першої операції – підпису/ шифрування. У контейнер завантажуються всі ключі, крім сеансового, тому можна виконати одну операцію читання для всіх криптографічних операцій.

2.6 Функції для автентифікации. Функція використовується, якщо в системі потрібна автентифікація користувачів перед використанням захищеної передачі даних

2.7 Запис даних, що підписуються, у контейнер. Функція повинна бути виконана до операції підпису даних. Можна використовувати функцію багаторазово для запису різних ділянок пам'яті. Ці ділянки записуються в контейнер послідовно.

2.8 Підпис даних. Функція використовується для формування цифрового підпису для даних, записаних у контейнер (див. попередню функцію). Результат виконання функції - цифровий підпис знаходиться в контейнері.

2.9 Експортування підпису. Для запису підпису в пам'ять з метою подальшого використання (наприклад, передача по каналу зв'язку) використовується функція експортування підпису:

2.10 Перевірка підпису. Використовується функція CRYPTAPI WORD cr_verify (CR_HANDLE h, PBYTE p_sign, LPCTSTR user, BOOL bWriteLog).

2.11 Генерація сеансового ключа. Ключ повинен бути сформований до шифрування даних шляхом використання функції. Формує в раніше створеному контейнері **h** сеансовий ключ.

2.12 Експортування сеансового ключа. Для передачі разом із шифрованими даними використовується функція, яка експортує сеансовий ключ з контейнера **h** користувачу **user**. Дані, що містять експортований ключ, записуються за адресою **p_data** (розмір блока дорівнює PARAM_LEN.OffLnKeyXch). Якщо bWriteLog == TRUE, то виконується запис у файл протоколу.

2.13 Імпортування сеансового ключа. Для запису заданого ключа в контейнер використовується функція, яка імпортує сеансовый ключ у контейнер **h** із області пам'яті з адресою **p_data** (розмір блока дорівнює **PARAM_LEN. OffLnKeyXch**). Якщо bWriteLog == TRUE, то виконується запис у файл протоколу.

2.14 Шифрування даних. Використовується така функція:

CRYPTAPI WORD __stdcall cr_encrypt (CR_HANDLE h, PBYTE p_data, DWORD p_datalen);

До шифрування необхідно сформувати сеансовый ключ. Результат шифрування записується замість відкритих даних.

2.15 Розшифрування даних. Використовується така функція:

CRYPTAPI WORD __stdcall cr_decrypt (CR_HANDLE h, PBYTE p_data, DWORD p_datalen)

До дешифрування необхідно імпортувати сеансовий ключ, що використовувався при шифруванні даних.

2.16 Допоміжні функції. Використовуються для визначення довжин ключових даних і цифрового підпису. Для сумісності з майбутніми версіями рекомендується замість фіксованих довжин даних використовувати функції для визначення цих довжин. Функції записують структуру опису довжин ключових даних контейнера (PARAM_LEN) за адресою **ptr**, якщо **ptr** не дорівнює NULL, а також повертають розмір необхідної для структури PARAM_LEN пам'яті в байтах.

## ЗАЩИТА ЭЛЕКТРОННОЙ ПОЧТЫ

Актуальность проблемы защиты электронной почты связана с использованием ее не только для обмена новостями рядовыми пользователями, но и для деловой переписки организаций разного уровня. Эта проблема является составной в общей задаче защиты Интернета, в том числе его сайтов, решение этой проблемы может использоваться при решении задач защиты сетевых технологий, в том числе для локальных и корпоративных сетей.

Определены функциональные требования к системе защиты информации для электронной почты, требования к интерфейсу и среде, в которой она функционирует. При определении функциональных требований учтены стандартные услуги, которые должны поставляться системами защиты с учетом требований использования национальных Стандартов на криптографию. При определении требований к пользовательскому интерфейсу учтено, что электронную почту могут использовать неподготовленные пользователи, поэтому, чем интерфейс проще, тем шире сможет использоваться система, тем меньше вероятность ошибок. При определении требований к окружению мы рассмотрели основные почтовые программы и наиболее распространенные почтовые серверы и в требованиях заложена возможность использования этих программ с наиболее полным покрытием их режимов функционирования. Для поддержки интернет - технологий используются различные операционные системы, в том числе Windows, различные версии Unix. Окончательная версия программного продукта должна работать для всех этих систем.

Для уменьшения затрат, связанных с обучением непосредственных пользователей работе с системой и администраторов в требованиях задается необходимость создания справочной системы. Определены требования к инсталляционному пакету, который с одной стороны должен быть стандартным, с другой обеспечивать возможность автоматической настройки параметров почтовых программ и администрирования пользователей системы.

Выполненный этап планирования позволил создать сценарий использования системы, тестирование которого показало необходимость рассмотрения дополнительного требования по работе с чужой почтой

Опыт работы с украинскими предприятиями показывает, что система должна поддерживать многоязыковый интерфейс (украинский, русский, английский). Для обеспечения этой услуги все ресурсы системы реализованы в виде DLL, подключение требуемой DLL на этапе инсталляции фактически определяет язык общения. В дальнейших версиях предполагается возможность смены языка «на лету».

На этапе реализации для максимального учета требования много платформенности не использовались специальные компоненты конкретных быстрых сред разработки. Не использовались также специальные технологии Microsoft типа Active X. Применялась технология сокетов, которая поддерживается для всех современных ОС, используемых для работы с Интернет. Это позволило не только упростить проблему перехода на другие операционные системы, но и уменьшило размер кода, загруженного в памяти. Это важно, так как программа должна быть загружена во все время использования защищенной почты.

Использование технологии сокетов делает разработанное программное обеспечение универсальным с точки зрения возможности использования как при построении систем защиты в глобальной сети, так и в локальных сетях. Отличие состоит лишь в том, что серверный сокет должен «слушать» другой порт (порты). Действительно, для непосредственной работы с Internet Explorer достаточно перехватывать информацию с 80 порта, аналогично работают системы с портами локальной сети.

Система защиты выполняет следующие функции при отсылке письма:

- подпись всех компонентов письма, в том числе предмета, тела письма и прикрепленных файлов;
- шифрование всех компонентов письма, в том числе предмета, тела письма и прикрепленных файлов;
- запись фамилии подписавшего почту в тело письма в зашифрованном виде;
- экспортирование ключа шифрования и запись его в теле письма в защищенном виде;
- отметка письма как зашифрованного;
- регистрацию выполняемых криптографических операций.

Система защиты выполняет следующие функции при получении письма:

- импортирование ключа шифрования;
- расшифровку всех компонентов письма, перечисленных выше;
- проверку подписи с учетом всех компонентов письма;
- вывода информации о полученном письме с фамилией подписавшего;
- архивирование полученной почты с цифровой подписью.
- регистрацию выполняемых криптографических операций.

Особое внимание уделено проблеме управления ключевыми данными. Фактически используется 3 модели управления ключевыми данными. Первая модель применяется, если пользователи доверяют центру генерацию и распространение ключей. В этом случае в системе предполагается наличие пользователей центра, всем пользователям центр генерирует ключи и передает вместе с личным ключом базу открытых ключей всех пользователей. Это одноуровневый центр, который обеспечивает взаимодействие каждого пользователя с каждым. При смене ключевых данных одного из пользователей выполняется распространение его ключей с использованием защищенной почты. На клиентском месте выполняется импортирование этих ключей с помощью дополнительной утилиты.

Вторая модель предполагает наличие двух уровневого центра. Эта модель используется, если пользователи не доверяют центру генерацию своих ключей. Центр выполняет сертификацию и распространение этих ключей для обеспечения взаимодействия каждого с каждым.

Последняя модель учитывает многоуровневую систему, в которой используются центры разных уровней, которые позволяют при минимальном количестве ключей обеспечить гибкую систему передачи их в соответствии с заданной конфигурацией системы. Ключи каждый пользователь генерирует сам. Оперативно можно изменять конфигурацию системы. Вопросам построения многоуровневых центров посвящены специальные доклады.

Как и всякая система защиты, данная система регистрирует все события, которые происходят, а именно использование ключевых данных, выполнение криптографических операций. Так как использование системы может быть достаточно интенсивным, протокол создается в начале каждого дня, предыдущие протоколы архивируются с датой их создания и цифровой подписью. В настоящей версии полученные сообщения после расшифровки могут быть записаны в файл, определенный пользователем вместе с цифровой подписью. Так как для функционирования системы требуется программное обеспечение для защищенной почты и стандартные почтовые программы, а также личный ключ, почта может быть отправлена в защищенном виде с любого компьютера, достаточно иметь необходимое программное обеспечение и личный ключ.

Разработан тестовый вариант системы, который позволяет для одного пользователя с использованием фиксированного ключа выполнять все операции защищенной почты. Тестовый ключ «зашит» в программе, поэтому не требуется создание личного ключевого носителя.

Исследованы вопросы криптографической стойкости системы. В данной системе применяется режим использования ключей максимальной длины для цифровой подписи (1024 битный вариант). Для передачи ключевых данных используется состоятельный протокол.

## ВТОРОЕ ВЫСШЕЕ ОБРАЗОВАНИЕ

**ХНУРЭ объявляет набор в группы по переподготовке специалистов по специальности 7.160102**
**«Защита информации с ограниченным доступом и автоматизация её обработки» на базе высшего образования**

Вы получаете диплом государственного образца, квалификацию
***администратор доступа.***

Форма обучения: очная, заочная.
Срок обучения: **1 год**

**Документы**: - 4 фото для документов;
копии дипломов, приложений к диплому

**Обращайтесь**: ХНУРЭ, каф. «Безопасность информационных технологий», к. 37-з, тел. 40-94-25.

**Адрес почтовый**: Харьков-61166, пр. Ленина, 14, ХНУРЭ, к. № 37-з
e-mail: iit@kture.kharkov.ua

***Перечень основных изучаемых дисциплин:***

Основы криптологии;
Защищённые операционные системы и базы данных;
Специальные разделы программирования;
Защита информации в открытых системах и сетях;
Высокопроизводительные компьютерные системы;
Методы технической защиты информации;
Криптографические системы и протоколы;
Информационные банковские технологии;
Технологии создания систем защиты информации;
Защита информации в банковских и коммерческих системах;
Комплексные системы обеспечения информационной безопасности;
Стандартизация и сертификация систем и средств защиты информации;
Организация и обеспечение безопасности информации;
Правовые основы обеспечения информационной безопасности;
Аппаратные средства защиты информации;