

221.396(06)
Р 15

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

*К 75-летию Харьковского национального
университета радиоэлектроники*

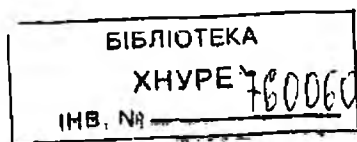
РАДИОТЕХНИКА

**Всеукраинский межведомственный
научно-технический сборник**

Тематический выпуск
«ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

Основан в 1965 г.

В Ы П У С К 1 4 1



Харків
Харківський національний
університет радіоелектроніки
2005

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

Регистрационное свидетельство КВ № 4486 от 22. 08. 2000.

В тематическом выпуске «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ» представлены статьи по четырем направлениям:

Перспективные симметричные шифры и их свойства;
Криптографические преобразования в группах точек эллиптических и гиперэллиптических кривых;
Методы и средства защиты информации;
Критерии, показатели и методики оценки в криптографии.

Для специалистов в области защиты информации. Материал может быть использован в учебном процессе по специальностям направления «Информационная безопасность»

Ответственность за содержание статей несут авторы.

Редакционная коллегия: главный редактор *В.М. Шокало*, д-р техн. наук, проф.; зам. главного редактора *А.И. Лучанинов*, д-р физ.-мат. наук, проф.; ответственный секретарь *Ж.Ф. Пащенко*, канд. техн. наук, проф.; *В.М. Ажажа*, чл.-кор. НАН Украины; *Б.М. Булгаков*, д-р физ.-мат. наук, проф.; *И.Д. Горбенко*, д-р техн. наук, проф.; *Ю.Е. Гордиенко*, д-р физ.-мат. наук, проф.; *А.И. Довбня*, д-р физ.-мат. наук, проф.; *В.В. Конин*, д-р техн. наук, проф.; *А.А. Коноваленко*, академик НАН Украины; *Н.И. Кравченко*, д-р техн. наук, проф.; *В.М. Кузмичев*, д-р физ.-мат. наук, проф.; *Л.Н. Литвиненко*, академик НАН Украины; *И.М. Неклюдов*, академик НАН Украины; *А.Г. Пащенко*, канд. физ.-мат. наук, доцент; *В.В. Поповский*, д-р техн. наук, проф.; *Э.Д. Прохоров*, д-р физ.-мат., проф.; *Е.Г. Прошкин*, д-р техн. наук, проф.; *А.И. Стрелков*, д-р техн. наук, проф.; *К.С. Сундучков*, д-р техн. наук, проф.; *Я.С. Шифрин*, д-р техн. наук, проф.; *С.Н. Шостка*, д-р техн. наук, проф.

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

Рекомендовано Ученым советом Харьковского национального университета радиоэлектроники, протокол № 9 от 04.02.2005.

Адрес редакционной коллегии: Харьковский национальный университет радиоэлектроники (ХНУРЕ), просп. Ленина, 14, Харьков, 61166, тел. (0572) 7021-397.

*Сборник «Радиотехника» включен в Каталог подписных изданий Украины.
подписной индекс 08391*

Выпуск подготовлен при поддержке и участии сотрудников АО «Институт информационных технологий»

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	5
ПЕРСПЕКТИВНЫЕ СИММЕТРИЧНЫЕ ШИФРЫ И ИХ СВОЙСТВА	
Горбенко І.Д., Гулак Г.М., Олійников Р.В., Руженцев В.І., Михайленко М.С. Аналіз властивостей алгоритмів блокового симетричного шифрування (за результатами міжнародного проекту Nessie)	7
Горбенко І.Д., Пушкарьов А.І., Олійников Р.В., Руженцев В.І., Горбенко Ю.І., Михайленко М.С. Порівняльний аналіз алгоритмів блокового симетричного шифрування (за результатами виконання проекту Nessie)	25
Алексейчук А.Н., Скрипник Л.В. Критерий примитивности группы подстановок, порожденной раундовыми шифрующими преобразованиями шифра Фейстеля	31
Горбенко Ю.І. Математична модель перекриття шифру для поточкових криптосистем	40
Бондаренко М.Ф., Лепеха А.Н. Исследование статистических свойств схемы разворачивания ключей FIPS-197	54
Лепеха А.Н. Сравнительный анализ схем разворачивания ключей в блочных симметричных шифрах ..	64
Орлова С.Ю. Современные подходы к оценке линейной сложности псевдослучайных последовательностей. Усовершенствованная методика тестирования линейной сложности на основе NIST STS.	70
КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ И ГИПЕРЭЛЛИПТИЧЕСКИХ КРИВЫХ	
Тевляшев А.Д., Збитнев С.И., Ковтун В.Ю. Алгоритмы скалярного умножения в группе точек эллиптической кривой и некоторые их модификации	82
Збитнев С.И., Ковтун В.Ю., Ильясова О.С. Арифметические операции на эллиптической кривой над двоичным полем в проективных координатах.	97
Балагура Д.С. Преобразования в оптимальных расширенных полях и их применение в криптографии.	108
Збитнев С.И., Ковтун В.Ю., Шевченко Д.М. Алгоритмы решения задачи дискретного логарифмирования в якобиане гиперэллиптической кривой.	116
Долгов В.И. Вибір еліптичних кривих до криптографічних застосувань.	133
МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ	
Потій О.В., Леншин А.В. Основні положення математичного апарату суб'єктивної логіки та його застосування для оцінки рівня зрілості систем забезпечення безпеки інформації.	144
Горбенко І.Д., Халимов О.Г. TWO-TRACK-MAC алгоритм высокой секретности.	161
Бондаренко М.Ф., Горбенко І.Д., Мелецкий А.П. Функциональность и характеристики криптобиблиотеки на языке JAVA.	168
Горбачев В.А., Степаненко В.В. Проектирование электронных систем со встроенным механизмом обеспечения безопасности.	174
КРИТЕРИИ, ПОКАЗАТЕЛИ И МЕТОДИКИ ОЦЕНКИ В КРИПТОГРАФИИ	
Ковальчук Л.В., Мельник С.В., Бездетный В.Т. Вероятностные характеристики схем генерации ключей с неравновероятным распределением.	181
Попович Е.В. Метод формирования профилей статистического тестирования для оценки свойств случайных последовательностей.	188
Попович Е.В., Горбенко Ю.І. Критерии оценки требований к генераторам случайных последовательностей в криптографических системах.	196
Куц Ю.В., Щербак Л.М. Фазові системи передачі інформації на основі чисельної системи залишкових класів.	205
Краснобаев В.А., Илюшко Я.В., Замула А.А. Универсальные алгоритмы сжатия табличных цифровых данных результатов выполнения арифметических операций в системе остаточных классов. ..	217
Горбенко І.Д., Мельникова О.А., Шумов О.І. Досвід підготовки та результати проведення VI Всеукраїнської олімпіади в напрямку «Інформаційна безпека».	226
РЕФЕРАТЫ	231

CONTENTS

PREFACE.....	5
PERSPECTIVE SYMMETRIC CIPHERS AND THEIR PROPERTIES	
<i>Gorbenko I.D., Gulak G.M., Oliynikov R.V., Ruzencev V.I., Mykhaylenko M.S.</i> The analysis of properties of block symmetric enciphering algorithms (by results of international project NESSIE)	7
<i>Gorbenko I.D., Pushkaryov A.I., Oliynikov R.V., Ruzhentsev V.I., Gorbenko U.I., Mykhaylenko M.S.</i> The comparative analysis of algorithms of block symmetric enciphering (by results of performance of project NESSIE)	25
<i>Aleksejchuk A.N., Skripnik L.V.</i> The primitivity criterion of the permutation group generated by the encryption round functions of Feistel cipher	31
<i>Gorbenko Y.I.</i> The mathematical model of cipher overlapping for stream-oriented cryptosystems	40
<i>Bondarenko M.F., Lepekha A.N.</i> Research on FIPS-197 key schedule statistical properties	54
<i>Lepekha A.N.</i> Comparative analysis of block symmetric ciphers key schedules	64
<i>Orlova S.Y.</i> Modern approaches to evaluation of the pseudo-random sequences linear complexity. Improved methods of linear complexity testing based on NIST STS	70
CRYPTOGRAPHIC TRANSFORMATIONS IN GROUPS OF POINTS OF THE ELLIPTIC AND HYPERELLIPTIC CURVES	
<i>Tevashev A.D., Zbitnev S.I., Kovtun V.Jur.</i> Scalar multiplication algorithms in group of elliptic curve points and some their modifications	82
<i>Zbitnev S.I., Kovtun V.Jur., Ilysova O.E.</i> Arithmetic operations on elliptic curve over binary field in projective coordinates	97
<i>Balagura D.S.</i> Transformation in optimal extended fields and their use in cryptography	108
<i>Zbitnev S.I., Kovtun V.Jur., Shevchenko D.V.</i> Arithmetic operations on elliptic curve over binary field in projective coordinates	116
<i>V.I. Dolgov.</i> Choice of elliptic curves for cryptographic applications	133
METHODS AND MEANS OF INFORMATION SECURITY	
<i>Potiy A.V., Lenshin A.V.</i> General mathematical mechanism foundations of Subjective Logic and its application in information security system maturity level evaluating	144
<i>Gorbenko I.D., Halimov O.G.</i> TWO-TRACK-MAC Algorithm of high privacy	161
<i>Bondarenko M.F., Gorbenko I.D., A.P. Melezky A.P.</i> Functionality and behavior of cryptography library based on Java language	168
<i>Gorbachev V.A., Stepanenko V.V.</i> Designing electronic systems embedded mechanism of security provision	174
CRITERIA, PARAMETERS AND TECHNIQUES OF THE ESTIMATION IN CRYPTOGRAPHY	
<i>Kovalchuk L.V., Melnik S.V., Bezdetny V.T.</i> Probability characteristics of key generation circuits with non equiprobable distribution	181
<i>Popovitch E.V.</i> Profiles forming method of statistical testing for random sequences properties evaluation ..	188
<i>Popovitch E.V., Gorbenko Y.I.</i> Evaluation criterion of requirements to pseudorandom sequences generators in cryptographic systems	196
<i>Kuts U. V., Scherbak L.N.</i> Phase systems of the information transfer based on numerical system of residual classes	205
<i>Krasnobaev V.A., Ilyshko Y.V., Zamula A.A.</i> Universal algorithms of compression of tabulated figures of results of performance of arithmetic operations in system of residual classes	217
<i>Gorbenko I.D., Melnikova O.A., Shumov A.I.</i> The experience in preparation and results of carrying out the VI all-Ukrainian competition in the field of information security	226
ABSTRACTS	231

Уважаемые читатели!

Интеграция Украины в мировое информационное пространство, ориентация государства на вступление в Европейское сообщество обусловили важность обеспечения взаимодействия органов государственной власти, местного самоуправления, юридических и физических лиц на мировом уровне с использованием иностранных и международных информационных и информационно-телекоммуникационных систем, различных информационных технологий, открытых систем (Интернет и т.д.). Согласно установленным международным нормам и стандартам всем пользователям должны предоставляться услуги по обеспечению подлинности, целостности, наблюдаемости, а в некоторых случаях и конфиденциальности информации на всех этапах её жизненного цикла. Одним из основных комплексных средств обеспечения указанных услуг является применение электронной цифровой подписи (ЭЦП).

На мировом уровне применение цифровой подписи во время обработки информации практически обязательно. В технологически развитых странах - США, Канада, Российская Федерация и страны Европейского содружества - созданы инфраструктуры открытых ключей (ИОК). В рамках ИОК обеспечиваются подтверждение целостности информации, подписываемой личным ключом пользователя, и идентификация отправителя (подтверждение подлинности подписанной им информации).

Украина сделала в этом направлении ряд очень важных шагов. Приняты законы "Про електронні документи та електронний документообіг", "Про електронний цифровий підпис", ряд Постановлений Кабинета министров и приказов ДСТСЗІ СБУ. Кроме того, введен в действие национальный стандарт на электронную цифровую подпись – ДСТУ 4145-2002. Эти документы определяют организационно-правовые основы использования электронных документов и электронных подписей, в том числе цифровых, в Украине. В них также определены условия и требования, при соблюдении и выполнении которых электронная цифровая подпись с использованием личного ключа приравнивается к собственноручной подписи, печати, штампу на бумажном документе. Таким образом, в масштабах государства начались работы по созданию систем ЭЦП, которые могут рассматриваться как первый этап создания Национальной ИОК.

Интеграция инфраструктуры открытых ключей, в том числе системы ЭЦП, с информационно-телекоммуникационными системами позволит решить такие задачи:

- внедрить новейшие технологии электронного документооборота в практику повседневной деятельности органов государственной власти, государственных ведомств и учреждений, предприятий, всех форм собственности;
- расширить возможности украинской экономики на международной арене, обеспечить равноправное вхождение Украины в мировое информационное пространство;
- повысить конкурентоспособность украинской экономики в целом, усилить процессы открытости и гласности в деятельности органов исполнительной власти и местного самоуправления;
- обеспечить целостность и подлинность (аутентификацию) информации, представленной в электронном виде, а также причастность субъектов информационных отношений и др.

Однако практическое применение ЭЦП в составе ИОК затруднено по следующим причинам:

1. Отсутствие Центрального Подтверждающего Органа (ЦПО) в системе ЭЦП влияет на определение требований, оценку возможностей, обоснование целесообразности и т.п.
2. В условиях неопределенности могут разрабатываться и внедряться ненадежные элементы системы ЭЦП и ИОК.

3. Однобокость при создании ИОК: она ориентирована только на предоставление услуг ЭЦП. Такие важные составляющие, как неотказуемость в целом и неотказуемость получателя вовсе не затрагиваются. Поэтому, как следствие, не создается подсистема обеспечения асимметричного (направленного) шифрования, а без неё предоставить услугу неотказуемости получателя невозможно.

4. Не все проблемы решены в управлении ключевыми данными, реализации безопасных и стойких криптографических протоколов, генерации и тестировании всех видов ключей, их хранении, обосновании использования общих параметров ДСТУ 4145-2002, доказательстве безопасности и стойкости используемых криптографических преобразований.

Для создания ИОК государственного масштаба существует множество путей. Основные из них:

- разработка, утверждение и осуществление целевых проектов и программ по созданию Национальной ИОК, внедрению ЭЦП, электронного документооборота и т.п.;
- непосредственное управление и контроль процессов создания организационно-технической составляющей Национальной ИОК со стороны ЦПО и контролирующего органа;
- координация деятельности органов государственной власти и местного самоуправления, министерств и ведомств, предприятий всех форм собственности по внедрению технологий ИОК, электронного документооборота с использованием цифровой подписи;
- создание и совершенствование системы обучения и подготовки специалистов соответствующей квалификации;
- сотрудничество с уполномоченными органами инфраструктуры открытых ключей других стран;
- материальное и финансовое обеспечение создания, внедрения, усовершенствования и применения Национальной ИОК.

Тематический выпуск посвящен решению перечисленных выше актуальных задач в области теории и практики информационной безопасности. Публикуемые статьи, по мнению специалистов Харьковского национального университета радиоэлектроники и ЗАО «Институт информационных технологий» (г. Харьков), отражают ряд острых противоречий и направлены на поиск путей разработки методов, методик, алгоритмов и средств их разрешения. Это издание позволит, на наш взгляд, довести до специалистов и интересующихся проблемами криптографической защиты информации, интересные результаты, обсудить состояние вопросов, продолжить ведущиеся в Украине дискуссии.

Большинство статей прошли обсуждение на кафедре «Безопасности информационных технологий» (БИТ) Харьковского национального университета радиоэлектроники 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 коллегам, специалистам и читателям, которые остаются неравнодушными к проблемам информационной безопасности

М.Ф. Бондаренко,
ректор ХНУРЭ,
доктор технических наук, профессор

И.Д. Горбенко
зав. кафедрой БИТ,
доктор технических наук, профессор

АНАЛІЗ ВЛАСТИВОСТЕЙ АЛГОРИТМІВ БЛОКОВОГО СИМЕТРИЧНОГО ШИФРУВАННЯ (ЗА РЕЗУЛЬТАТАМИ МІЖНАРОДНОГО ПРОЕКТУ NESSIE)

На сучасному етапі розвитку інформаційних технологій однією з обов'язкових вимог до систем, що обробляють інформацію з обмеженим доступом, є можливість забезпечення конфіденційності. Основним механізмом, що дозволяє реалізувати конфіденційність інформації, є шифрування. У числі найбільш широко використовуваних шифрів – блокові симетричні шифри (БСШ), що відрізняються від інших насамперед високою продуктивністю і простотою програмної та апаратної реалізацій. Рекомендований сьогодні до застосування на території України і країн СНД шифр ГОСТ 28147-89 був розроблений більш ніж 15 років тому, і, на наш погляд, не повною мірою відповідає сучасним вимогам. Тому виникає необхідність розгляду блокових симетричних шифрів, рекомендованих до застосування за результатами проекту NESSIE. Метою цієї статті є добір кращих алгоритмів в БСШ та аналіз їх властивостей. Порівняння їх властивостей та вибір претендента до застосування в Україні здійснюється в [1].

1 Методика ухвалення рішення на множині альтернативних варіантів

Відповідно до положень загальної теорії прийняття рішень, під ефективністю „функціонування” алгоритму шифрування будемо розуміти ступінь відповідності отриманих результатів функціонування схеми R_p необхідним R_n :

$$P = \langle R_p, R_n \rangle.$$

Необхідним результатом функціонування R_n є рівень (ступінь) забезпечення конфіденційності шляхом реалізації механізму шифрування. Досягнутим результатом функціонування R_p є реальний рівень забезпечення конфіденційності. Оцінка того, наскільки повно реалізована функція конфіденційності, або, іншими словами, наскільки R_p відповідає R_n , здійснюється за допомогою великого набору показників і критеріїв, що відбивають рівень виконання алгоритмом окремих складових загальної функціональної задачі.

На сьогоднішній день не існує систематизованого підходу для перетворення множини значень показників у єдиний показник, яким можна було б охарактеризувати ефективність функціонування схеми шифрування. Розроблювачі шифрів пропонують різні набори показників, що характеризують окремі сторони шифрів, але це не дозволяє представити картину в цілому. Для рішення таких задач на практиці часто застосовуються елементи теорії нечітких множин [2]. Розглянемо деякі елементи даної теорії.

Нехай U – повна множина, що охоплює всі об'єкти деякого класу. Нечітка підмножина F множини U , яку будемо називати нечіткою множиною, визначається через функцію приналежності $\mu_F(u)$, $u \in U$. Ця функція відображає елементи u множини U на множину ірраціональних чисел відрізка $[0,1]$, що вказують ступінь приналежності кожного елемента нечіткій множині F .

Якщо повна множина U складається з кінцевого числа елементів u_i , $i = 1, 2, \dots, n$, то нечітку множину F можна представити в такому вигляді:

$$F = \mu_F(u_1)/u_1 + \mu_F(u_2)/u_2 + \dots + \mu_F(u_n)/u_n,$$

Таким чином, для вирішення задачі вибору алгоритму шифрування можна скористатися елементами теорії нечітких множин. Експертні оцінки альтернативних варіантів за визначеними умовними критеріями можуть бути представлені як нечіткі множини або числа, виражені за допомогою функцій приналежності. Розглянемо одну з математичних постановок задачі прийняття рішень на основі теорії нечітких множин.

У даному випадку критерії визначають деякі поняття, а оцінки альтернатив являють собою ступені відповідності цим поняттям. Нехай є множина альтернатив $A = \{ a_1, a_2, \dots, a_m \}$ і множина критеріїв $C = \{ C_1, C_2, \dots, C_n \}$, при цьому оцінки альтернатив по кожному i -му критерію представлені нечіткими множинами:

$$C_i = \{ \mu_{C_i}(a_1)/a_1 + \mu_{C_i}(a_2)/a_2 + \dots + \mu_{C_i}(a_m)/a_m \}. \quad (1)$$

Правило вибору кращої альтернативи можна представити як перетинання нечітких множин:

$$D = C_1 \cap C_2 \cap \dots \cap C_n.$$

Операція перетинання нечітких множин може бути реалізована різними способами. Іноді перетинання виконується як множення, але частіше ця операція виконується як взяття мінімуму [2] для різних альтернатив:

$$\mu_D(a_j) = \min_{i=1, \dots, n} \mu_{C_i}(a_j), j = 1, \dots, m.$$

Кращою вважається альтернатива a^* , що має найбільше значення функції приналежності:

$$\mu_D(a^*) = \max_{j=1, \dots, m} \mu_D(a_j).$$

У випадку порівняння алгоритмів шифрування всі критерії мають різну важливість, тому їх внесок у загальне рішення можна представити як зважене перетинання [2]:

$$D = C_1^{\beta_1} \cap C_2^{\beta_2} \cap \dots \cap C_n^{\beta_n}, \quad (2)$$

де β_i – вагові коефіцієнти відповідних критеріїв, що повинні задовольняти умовам:

$$\beta_i \geq 0; i = 1, \dots, n; (1/n) \sum_{i=1}^n \beta_i = 1.$$

Коефіцієнти відносної важливості визначаються з використанням процедури попарного порівняння критеріїв.

Розглянемо цю процедуру, використовуючи метод попарного порівняння.

Метод попарного порівняння елементів [2] можна описати в такий спосіб. Будується матриця парних порівнянь. Парні порівняння проводяться в термінах домінування одного елемента над іншим. Отримані судження виражаються в цілих числах за дев'ятибальною шкалою (див. табл. 1).

Заповнення квадратних матриць парних порівнянь здійснюється за таким правилом. Якщо елемент E_i домінує над елементом E_j , то клітка матриці, що відповідає рядку E_i і стовпцю E_j , – цілим числом, а клітка, що відповідає рядку E_j і стовпцю E_i , заповнюється зворотним йому числом.

Для одержання кожної матриці експерт або особа, яка приймає рішення, виносить $n(n-1)/2$ суджень (тут n – порядок матриці парних порівнянь).

Розглянемо в загальному вигляді приклад формування матриці парних порівнянь.

Нехай $E_1, E_2, \dots, E_n$ – множина з n елементів (альтернатив) і $v_1, v_2, \dots, v_n$ – відповідно до їх ваги, або інтенсивності. Порівняємо попарно вагу, або інтенсивність, кожного елемента з вагою, або інтенсивністю, будь-якого іншого елемента множини стосовно загальної для них властивості або мети. У цьому випадку матриця парних порівнянь $[E]$ має такий вигляд:

$$[E] = \begin{bmatrix} & E_1 & E_2 & \dots & E_n \\ E_1 & v_1/v_1 & v_1/v_2 & \dots & v_1/v_n \\ E_2 & v_2/v_1 & v_2/v_2 & \dots & v_2/v_n \\ \dots & \dots & \dots & \dots & \dots \\ E_n & v_n/v_1 & v_n/v_2 & \dots & v_n/v_n \end{bmatrix}$$

Таблиця 1

Ступінь значимості	Визначення	Пояснення
1	Однакова значимість	Дві дії вносять однаковий вклад у досягнення мети
3	Деяка перевага значимості однієї дії над іншою (слабка значимість)	Існують поняття на користь переваги однієї з дій, однак ці поняття недостатньо переконливі
5	Істотна або сильна значимість	Є надійні дані або логічні судження для того, щоб показати перевагу однієї з дій
7	Очевидна або дуже сильна значимість	Переконливе свідчення на користь однієї дії перед іншою
9	Абсолютна значимість	Свідчення на користь переваги однієї дії над іншою найвищою мірою переконливі
2,4,6,8	Проміжні значення між двома сусідніми судженнями	Ситуація, коли необхідне компромісне рішення
Зворотні величини приведені вище ненульових величин	Якщо дії i при порівнянні з дією j приписується одне з визначених вище ненульових чисел, то дії j при порівнянні з дією i - зворотнє значення	Якщо погодженість була постульована при одержанні N числових значень для утворення матриці

При проведенні парних порівнянь варто відповідати на такі питання: який із двох порівнюваних елементів важливіший або має більший вплив, який більш ймовірний і який кращий. При порівнянні критеріїв запитують, який із критеріїв більш важливий, при порівнянні альтернатив стосовно критерію – яка з альтернатив більш краща або більш ймовірна [2].

Ранжирування елементів, що аналізуються з використанням матриці парних порівнянь $[E]$, здійснюється на підставі головних власних векторів, отриманих у результаті обробки матриць [2].

2 Вибір критеріїв оцінки алгоритмів шифрування

Наведені вище результати дозволяють виконати порівняння шифрів та вибір одного з них з використанням розвинутої методики.

Під критерієм будемо розуміти ознаку, на основі якої здійснюється оцінка, визначення чи класифікація чого-небудь. У такому розумінні узагальнений критерій порівняння алгоритмів шифрування можна представити у вигляді двох складових: безумовного та умовного критеріїв. Кожен з цих критеріїв є залежним від відповідного набору тих чи інших ознак.

Оцінку претендентів будемо виконувати в два етапи. На першому етапі алгоритми-кандидати перевірятимуться на відповідність безумовним критеріям. Оскільки безумовні критерії

допускають вибір тільки стійких алгоритмів, то кожний алгоритм, що відповідає безумовним критеріям, потенційно може бути використаний.

До безумовних критеріїв віднесено ті критерії (показники), виконання яких є обов'язковим для шифру.

Блоковий симетричний шифр повинен безумовно володіти наступними властивостями (задовольняти безумовному критерію):

1. Захищеність від усіх відомих та потенційно можливих криптоаналітичних атак, де під захищеністю розуміється той факт, що усі криптоаналітичні атаки мають складність більшу, ніж атака типу „груба сила”, що будемо позначати K_{61} .

2. Статистична безпечність алгоритму шифрування, під якою розуміється статистична незалежність зашифрованих блоків повідомлень та гам шифруючих від блоків відкритої інформації та ключів, що використовуються. Цю складову критерію будемо позначати K_{62} .

3. Надійність математичної бази з точки зору відсутності можливостей здійснювати атаки „універсальне” розкриття за рахунок недосконалості або закладеної навмисної специфічної математичної бази. При цьому вважається, що атака „універсальне розкриття” має складність, набагато меншу, ніж складність атаки „груба сила”. Будемо позначати цю складову критерію як K_{63} .

4. Практична захищеність алгоритму шифрування від силових атак, яка може досягатись на основі використання симетричних блокових криптоперетворень з довжиною блоку не менш, ніж 128 біт, та довжиною ключа не менше 128 біт (в деяких випадках 192 або 256 біт). Будемо позначати цю складову критерію як K_{64} .

5. Відсутність слабких початкових ключів та підозр на існування ключів, при яких складність криптоаналітичної атаки є меншою, ніж складність атаки „груба сила”. Цю складову критерію позначимо через K_{65} .

6. Складність прямого та зворотного перетворень не перевищує допустимої величини, крім того, складність розгортання ключів не перевищує заданої. Цю складову позначимо через K_{66} .

У використаних позначеннях часткових критеріїв нижній індекс означає номер часткового критерію.

Оскільки наведені часткові критерії є безумовними, то критерієм добору є логічна змінна Так/Ні (1/0).

Використавши позначення булевої алгебри, можемо записати:

$$(K_{61}, K_{62}, K_{63}, K_{64}, K_{65}, K_{66}) \in \{1, 0\}.$$

З урахуванням наведених вище критеріїв функція відповідності БСШ безумовному критерію (тобто усім $K_{61} - K_{66}$ частковим критеріям) може бути записана як

$$f_{6k}(\text{шифр}) = K_{61} \wedge K_{62} \wedge K_{63} \wedge K_{64} \wedge K_{65} \wedge K_{66}, \quad (3)$$

де символ „ $\wedge$ ” позначає операцію кон'юнкції булевих змінних.

Функція $f_{6k}(\text{шифр}) \in \{0, 1\}$ і при $f_{6k}(\text{шифр}) = 1$ оцінюваний алгоритм шифрування відповідає безумовному критерію.

Другою складовою загального критерію пропонується узагальнений критерій переваги K_7 . Цей критерій є умовним, в якості його складових пропонується використовувати наступні часткові умовні критерії (табл. 2).

В подальшому будемо вважати, що оцінка по кожному із часткових критеріїв здійснюється завдяки використанню показників $K_{71}, K_{72}, K_{73}, K_{74}, K_{75}, K_{76}, K_{77}$, де індекс $i = \overline{1, 7}$ означає номер часткового критерію.

Оцінку та порівняльний аналіз алгоритмів шифрування будемо здійснювати за допомогою наведених вище безумовного та умовного критеріїв. Кожен з цих критеріїв в свою чергу визначається через сукупність часткових критеріїв. Кожен з цих часткових критеріїв будемо застосовувати для прийняття рішення, використовуючи наступні показники або сукупність показників:

Критерій	Позначення
Можливість і умови вільного поширення алгоритму в Україні з урахуванням національного та міжнародного законодавства	K_{y1}
Реальна захищеність від відомих криптоатак при зменшеному числі циклів шифрування	K_{y2}
Рівень довіри до шифру	K_{y3}
Перспективність застосування алгоритму шифрування	K_{y4}
Складність програмної, апаратної, програмно-апаратної реалізації	K_{y5}
Можливість і умови використання в різних режимах роботи	K_{y6}
Гнучкість алгоритму шифрування	K_{y7}

1. Захищеність від усіх відомих атак. Для визначення цього критерію будемо спиратися на відомі джерела, присвячені пошуку криптоаналітичних атак на той чи інший шифр. За результатами експертних оцінок K_{61} приймає значення 1 чи 0.

2. Оцінку статистичної безпечності пропонується здійснювати, використовуючи ряд підходів, закладених для статистичного тестування у пакетах NIST STS та Стурт-Х.

На основі аналізу результатів тестування згідно цих методик експертами приймається рішення $K_{62} = 1$ чи 0 і, відповідно, алгоритм залишається для подальшого аналізу чи відкидається.

3. Оцінку надійності математичної бази можна здійснювати на основі експертних оцінок спеціалістів-криптологів. При цьому повинна враховуватись ступінь відкритості проектування та дослідження алгоритму шифрування. Основним же фактором є можливість еквівалентного (альтернативного) представлення криптографічних перетворень та виконання криптоаналізу з меншою або суттєво зменшеною складністю вказаних перетворень. В результаті приймається рішення $K_{63} = 1$ чи 0.

4. Оцінку практичної захищеності алгоритму шифрування від силових атак можна здійснювати, використовуючи розмір простору дозволених ключів та можливих криптограм. Виходячи з цього критерію, будемо розглядати шифри, для яких розмір блоку та ключа не нижче ніж, 128 біт. В інших випадках $K_{64} = 0$.

5. Оцінка відсутності чи наявності слабких початкових ключів є однією з найбільш складних задач. Її розв'язок повинен здійснюватись на основі експертної оцінки як прямого, так і зворотного алгоритму криптографічних перетворень і, перш за все, алгоритму розгортання ключів. Перша задача експертної оцінки повинна здійснюватись на основі визначення, чи не є деякі ключі слабкими при застосуванні того чи іншого методу криптоаналізу. Основними при цьому є такі методи: диференційний криптоаналіз, лінійний криптоаналіз, інтерполяційне вторгнення, інтегральна атака, вторгнення з частковим вгадуванням ключа, вторгнення з використанням зв'язаних ключів, пошук лазівок, вторгнення на основі обробки збоїв. У результаті виконання аналізу може бути прийняте рішення про наявність чи відсутність слабких початкових ключів і відповідно $K_{y5} = 1$ чи $K_{y5} = 0$.

6. Оцінка складності прямого та зворотного криптографічних перетворень, а також оцінка складності процедури розгортання ключа можуть бути здійснені на основі даних проекту NESSIE [5]. Тут наведено дані щодо складності перетворень різних шифрів, що реалізовані під найбільш поширені сьогодні процесори та функціонують під керуванням різних операційних систем. Оскільки в Україні більшість криптографічних алгоритмів працюють під керуванням операційної системи Windows та реалізовані мовою C++, то для порівняння алгоритмів будемо використовувати відповідні дані. Серед шифрів, що будуть аналізуватися, відкинемо найбільш повільні шифри за кожним з трьох показників. Для цих шифрів $K_{66} = 0$, для інших – $K_{66} = 1$.

3 Послідовність аналізу та вибору алгоритму шифрування

Виконується оцінка алгоритму шифрування з використанням умовних критеріїв на основі теорії нечітких множин. Будемо використовувати показник (2), що може бути представлений як

$$D = C_1^{\beta_1} \cap C_2^{\beta_2} \cap \dots \cap C_n^{\beta_n},$$

де C_i – числові значення приватних критеріїв узагальненого умовного критерію, отримані методом експертної оцінки і представлені нечіткими множинами (2);

β_i – ваговий коефіцієнт i -го критерію, отриманий за допомогою матриці попарних порівнянь відповідно до описаного в підрозділі 2.

Таким чином, вибір алгоритму шифрування пропонується здійснювати в такій послідовності:

1. Визначаються числові значення показників $K_{61}, K_{62}, K_{63}, K_{64}, K_{65}, K_{66}$, що входять до безумовної функції (3) $f_{6k}(A_i)$.

2. Визначаються значення функцій $f_{6k}(A_i)$. Якщо $f_{6k}(A_i)$ i -го алгоритму приймає значення “1”, то відповідний алгоритм задовольняє безумовному узагальненому критерію. Про це також свідчить значення функції $f_{6k}(A_i) = “1”$ (істина).

3. Для алгоритмів, що задовольняють узагальненому безумовному критерію, тобто $f_{6k}(A_i) = 1$, згідно до часткових умовних критеріїв обчислюються значення часткових умовних показників $K_{y1}, K_{y2}, K_{y3}, K_{y4}, K_{y5}, K_{y6}$ та K_{y7} .

4. Обчислюють значення узагальненого умовного показника K_y для кожного із альтернативних кандидатів.

5. На основі порівняння значень K_y для різних шифрів робиться висновок про більш кращі властивості того чи іншого алгоритму.

4 Огляд шифрів, що брали участь у конкурсі NESSIE

Наведемо стислий опис шифрів-учасників конкурсу NESSIE з тим, щоб обґрунтувати відповідність цих шифрів умовним та безумовним критеріям, викладеним у попередніх пунктах.

4.1 AES

AES [6] є блоковим симетричним шифром з довжиною блоку 128 біт і довжинами ключа 128, 192 і 256 біт. Алгоритм AES побудований на основі модифікованої SPN-структури (Substitution-Permutation Network) і є шифром, що використовує підстановку і лінійне перетворення (substitution-linear transformation network) з 10, 12 або 14 циклами шифрування, у залежності від довжини ключа. Алгоритм є байт-орієнтованим, і він побудований на основі попередньої розробки авторів – шифрі Square [7]. Основне нелінійне перетворення – підстановка (S-box) має математичну структуру і може бути представлена як обчислення зворотного елемента в полі $GF(2^8)$ і використання афінного перетворення. Підключі вволяться з використанням операції додавання по модулю 2 (XOR). Схема зашифрування криптоалгоритму представлена на рис. 1.

Процедура вироблення підключів має просту структуру. В разі використання в шифрі 128-бітного ключа, підключ, що використовується на 1-му циклі, збігається з майстер-ключем користувача, а всі інші підключі утворюються завдяки деякому перетворенню над попереднім підключем.

При проектуванні AES була врахована можливість застосування диференціального і лінійного криптоаналізу [6]. Максимальна ймовірність 4-циклової диференційної характеристики дорівнює 2^{-150} . З огляду на те, що розмір блоку AES складає 128 біт, а мінімальна кількість циклів шифрування дорівнює 10, можна стверджувати, що алгоритм є стійким до диференціального криптоаналізу і в шифрі закладений запас стійкості проти цієї атаки. Використана стратегія wide-trail гарантує також відсутність лінійних апроксимацій на 4 цикли з ймовірністю, що

перевищує 2^{-75} [6]. Цього досить, щоб запобігти атаці на весь шифр, як і у випадку з диференціальним криптоаналізом. Найбільш ефективним криптографічним нападом на скорочений

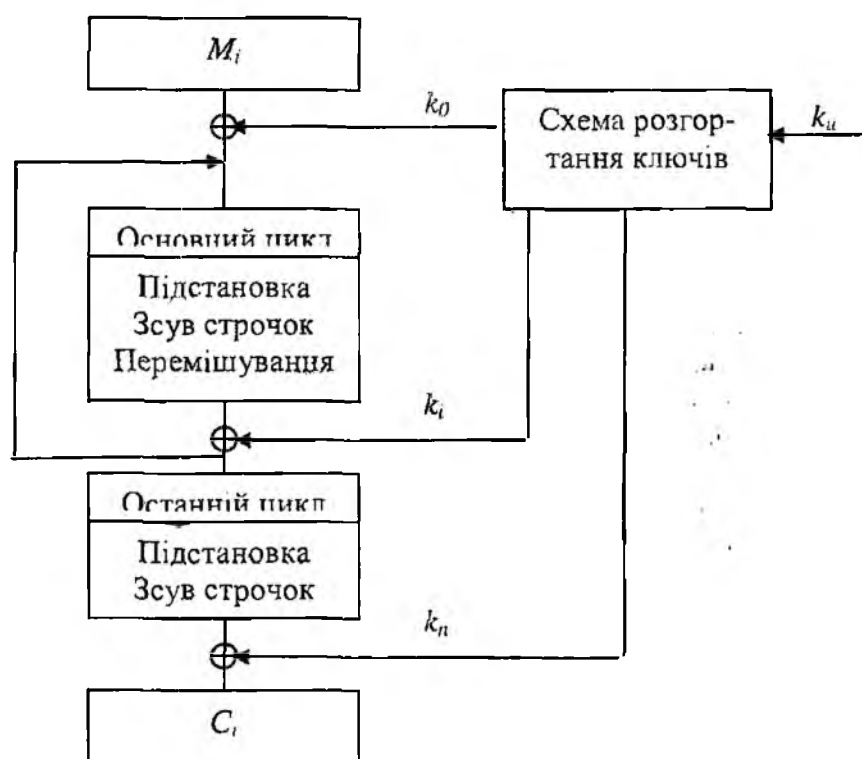


Рис. 1

шифр є інтегральна атака, але, як показано в [6], шифр з 7 та більшим числом циклів захищений від цієї атаки. В [8-11] запропоновані атаки колізії, атаки нездійснених диференціалів, усічених диференціалів, інтерполяційна атака, але загальним підсумком цих робіт є те, що шифр з повним числом циклів є захищеним від відомих криптоаналітичних атак.

В нашій роботі викладені основні результати реалізації найбільш ефективних криптоаналітичних нападів на цей шифр зі зменшеним числом циклів, які підтвердили справедливості розрахунків щодо захищеності шифру з повним числом циклів від цих атак.

4.2 Grand Cru

Grand Cru [12] є блоковим симетричним шифром з довжиною блока 128 біт і довжиною ключа 128 біт. Багато в чому структура криптоалгоритму Grand Cru схожа на структуру криптоалгоритму Rijndael. Криптостійкість алгоритму заснована на стратегії багаторівневої захищеності. Ідея якої полягає в послідовному використанні декількох криптостійких перетворень. Найбільш важливою перевагою використання такої стратегії є збільшена криптографічна стійкість проти диференціального і лінійного криптоаналізу, а також проведення атаки, спрямованої на реалізацію шифру [12]. Оскільки криптоалгоритм Grand Cru заснований на Rijndael, наведемо розходження між ними.

Схема зашифрування Rijndael для 128-бітного ключа K^0 може бути представлена в такий спосіб:

$$\sigma K_1^0 = \prod_{i=8}^n (\gamma \circ \pi \circ \theta \circ \sigma K_i^0) \circ \sigma K_{10}^0 \circ \gamma \circ \pi).$$

де γ – нелінійна підстановка (підстановка s-блоків);

π – байтове перетворення (циклічний зсув рядків);

θ – лінійне перетворення підмножини байт (перемішування в стовпцях);

σ – додавання з цикловим ключем.

Grand Cru додає три ключові операції, після чого підсумкова схема алгоритму шифрування має такий вигляд:

$$\psi K_1^3 \circ \nu^{-1} \circ \sigma K_{10}^0 \circ \beta K_9^2 \circ \pi K_8^1 \circ \gamma \circ \sigma K_7^0 \circ \prod_{i=8}^0 (\beta K_i^2 \circ \theta \circ \pi K_i^1 \circ \gamma \circ \sigma K_i^0) \circ \nu \circ \psi K_0^3,$$

де πK^1 – ключове перетворення, причому цикловий підключ K_i^1 може приймати $(4!)^5$ значень,

βK^2 – ключезалежний побайтовий циклічний зсув, де K_i^2 може приймати 2^{48} значень;

ψK^3 – додавання по модулю 2^8 ;

ν – додатковий рівень, що перемішує.

Схема розшифрування подібна до схеми зашифрування, тільки перетворення виконуються в зворотному порядку і циклічні ключі використовуються також в зворотному порядку. Загальна схема розгортання ключа така ж, як і в криптоалгоритмі Rijndael.

Що стосується s-блока, то криптоалгоритм використовує таблицю довжиною 256 байт, цілком ідентичну s-блоку Rijndael. До лінійних методів перемішування можна віднести побайтовий циклічний зсув і перемішування в стовпцях (множення на константну матрицю). З арифметичних операцій використовуються додавання по модулю 2^8 . В криптоалгоритмі присутнє забілювання у вигляді додавання відкритого тексту з підключем перед першим циклом зашифрування.

Розробники криптоалгоритму заявляють про стійкість Grand Cru до лінійного і диференціального криптоаналізу [13], стверджують, що, оскільки Grand Cru заснований на криптоалгоритмі Rijndael, то стійкість Rijndael до різних криптоатак поширюється і на Grand Cru. Відповідно до звіту NESSIE [13], на криптоалгоритм Grand Cru не було знайдено жодної криптоаналітичної атаки.

Зрозуміло також, що використання додаткових засобів захисту в криптоалгоритмі Grand Cru призводить до збільшення обчислювальних засобів, необхідних для його роботи.

4.3 NUSH

NUSH є блоковим симетричним шифром з довжиною блока 64, 128 або 256 біт і довжиною ключа 64, 128 або 256 біт. Для реалізації криптографічних перетворень автори алгоритму вибрали наступні операції: І, АБО, ВИКЛЮЧНЕ АБО, циклічний зсув і додавання по модулю 2 [14]. Вхідний блок даних розбивається на чотири блоки по 32 біти, кожен блок записується в окремий регістр. Для довжини блока 64 біти використовується 36 циклів шифрування, для довжини блока 128 біт – 68 циклів і для довжини блока 256 біт – 132 цикли. Один цикл зашифрування містить у собі чотири перетворення. У свою чергу, одне криптоперетворення складається з двох частин: лінійного перетворення в регістрах і додавання з результатом булевої функції над іншими двома регістрами. Таким чином, криптоперетворення можуть виконуватися паралельно на різних процесорах.

Схема одного циклу зашифрування подана на рис. 2

F-функція і G-функція представляють собою послідовне виконання над вхідними даними наступних операцій: додавання по модулю 2, циклічний зсув, логічне побітове додавання (OR).

В алгоритмі NUSH не застосовуються операції цілочисельного модульного множення та ділення, тому що криптоалгоритм розрахований на використання його в малопотужних електронних пристроях, таких як смарт-карти. Криптоалгоритм NUSH не використовує s-блоки, а нелінійність криптоалгоритму забезпечується за допомогою логічних операцій. У якості перед- та післязабілювання використовуються початкове і кінцеве криптоперетворення, в ході яких виконується додавання підключа. Схема розгортання підключів влаштована таким чином, що циклові ключі, які використовуються при шифруванні, виробляються простою перестановкою слів майстер-ключа. Таким чином, у схемі вироблення підключів не використовуються нелінійні функції.

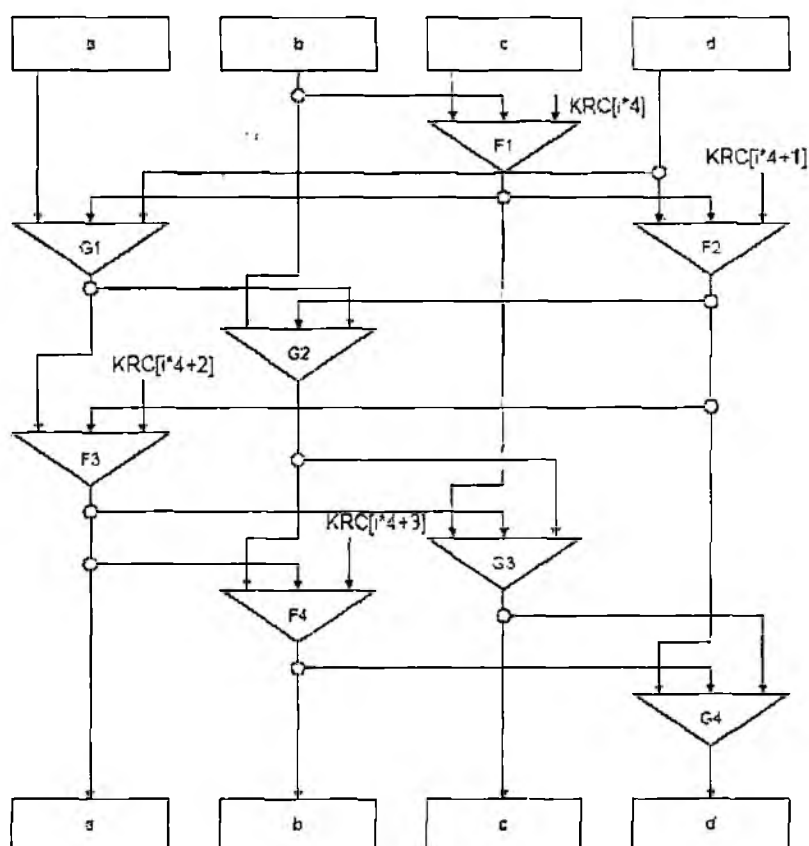


Рис. 2

Розробники стверджують, що цей криптоалгоритм стійкий до лінійних і диференціальних атак (атаки на основі слабких і зв'язаних ключів ними проаналізовані не були). Однак у ході досліджень NESSIE [15] було оголошено про існування лінійної атаки на криптоалгоритм із довжиною блока 64 і 128 біт при будь-якій довжині ключа, складність якої нижче складності повного перебору. Крім того, якщо не здійснювати два перших цикли зашифрування, то всі дев'ять варіантів криптоалгоритму стають уразливими до лінійного криптоаналізу, що говорить про дуже низький запас стійкості.

4.4 SC2000

SC2000 [16] – це симетричний блоковий криптоалгоритм з довжиною блока 128 біт, що працює з довжинами ключів 128, 192 і 256 біт. Алгоритм складається з 6,5 або 7,5 циклів шифрування. Алгоритм побудований з використанням ланцюга Фейстеля і SPN структур. SPN структура, використовувана в шифрі, є комбінацією 4 бітних s-блоків і лінійних криптоперетворень. F-функція в ланцюзі Фейстеля містить рівень з чотирьох 6-бітних s-блоків і восьми 5-бітних s-блоків.

При генерації s-блока використовувалися зведення в ступінь над полем Галуа та афінне перетворення. Для s-блока, побудованого на основі такої методики, доведено [17] досягнення мінімального значення максимальної імовірності перетворень на S-блочі. Схема вироблення циклових ключів генерує із майстер-ключа 56 32-бітних ключів (для довжини ключа 128 біт) і 64 32-бітних ключів (для довжин ключа 192 і 256 біт). Функція генерації ключів складається з допоміжної і розширеної функції генерації ключів.

Розробниками криптоалгоритму була описана атака на основі диференційного криптоаналізу на 4,5 цикла, у результаті було знайдено 28 біт у першому та останньому цикловому ключі [13]. Особливістю схеми розгортання ключів є те, що знання одного циклового ключа не розкриває значення інших циклових ключів і майстер-ключа.

4.5 Camellia

Довжина блока алгоритму Camellia [18] дорівнює 128 біт, дозволеними довжинами вихідного ключа є 128, 192 і 256 біт. Число циклів криптоалгоритму залежить від використовуваної довжини ключа, так, для довжини ключа 128 біт виконується 18 циклів зашифрування. При довжинах ключа 192 і 256 біт число циклів збільшується до 24. При довжині ключа 128 біт після 6 і 12 циклів знаходяться два FL/FL^{-1} функціональних рівні (при довжинах ключа 192, 256 біт функції FL/FL^{-1} розташовані після 6, 12 і 18 циклів). Крім того, перед першим і після останнього циклів виконуються забілювання – додавання по модулю 2 з цикловими ключами. Сам цикл має структуру Фейстеля. Процедура зашифрування наведена на рис. 3.

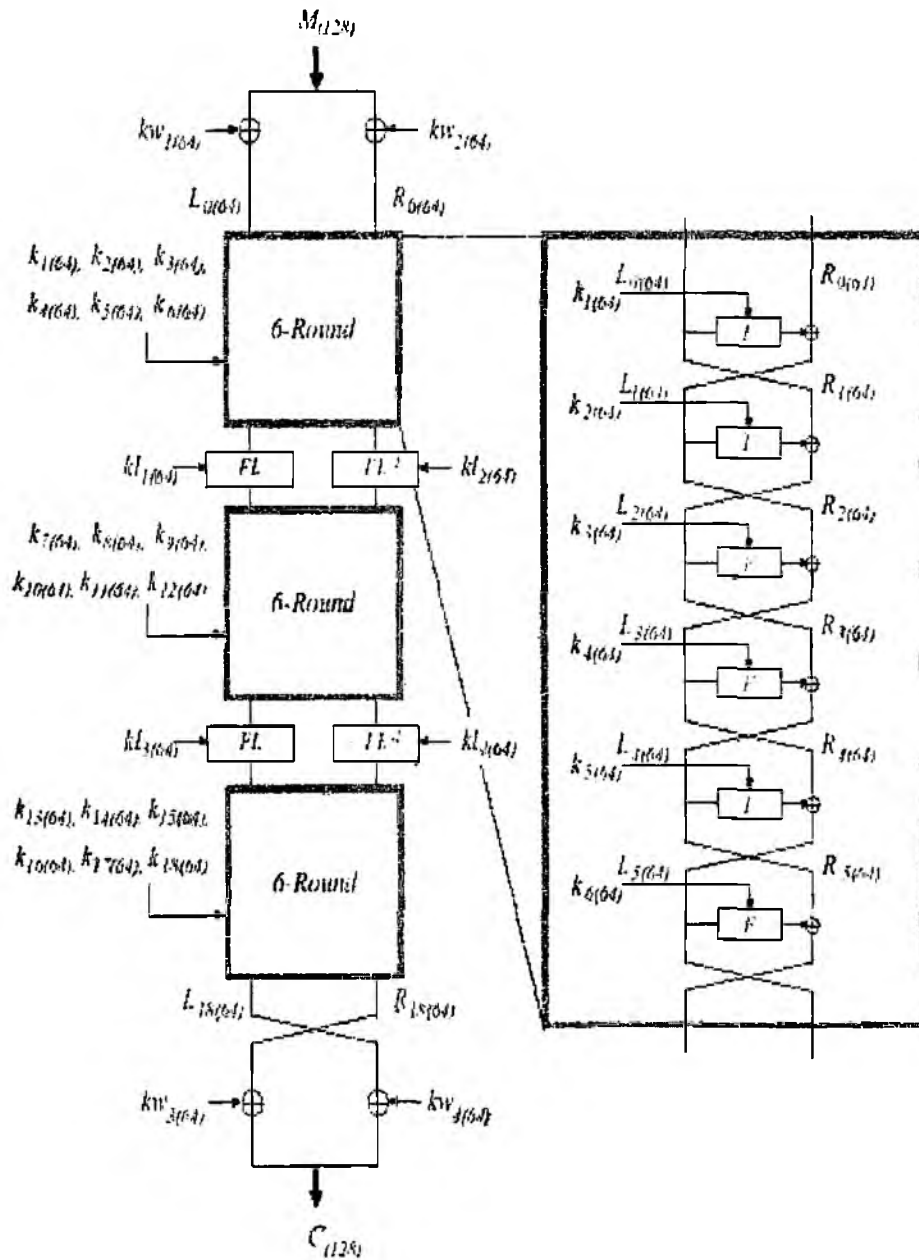


Рис.3

Схема розгортання ключів представлена на рис. 4. Як вихідні, так і циклові ключі, представлені у вигляді одновірних масивів. При генерації ключів шифру Camellia використовуються дві 128-бітові змінні $K_{L(128)}$, $K_{R(128)}$ і чотири 64-бітові змінні $K_{LL(64)}$, $K_{LR(64)}$, $K_{RL(64)}$ і $K_{RR(64)}$, що задані так, щоб задовольнялися наступні відносини:

$K_{(128)} = K_{L(128)}, K_{R(128)} = 0$ для 128-бітового ключа,

$K_{(192)} = K_{L(128)} \parallel K_{RL(64)}, K_{RR(64)} = \bar{K}_{RL(64)}$ для 192-бітового ключа,

$K_{(256)} = K_{L(128)} \parallel K_{R(128)}$ для 256-бітового ключа.

За допомогою цих змінних генеруються дві 128-бітових змінних $K_{A(128)}$ і $K_{B(128)}$, як показано на рис. 4, де $K_{B(128)}$ використовується тільки при довжині секретного ключа 192 або 256 біт. Спочатку виконується операція додавання по модулю 2 $K_{L(128)}$ з $K_{R(128)}$, і результат «шифрується» за два цикли з використанням постійних значень $\Sigma_{1(64)}$ і $\Sigma_{2(64)}$ як ключів; результат складається по модулю 2 з $K_{L(128)}$ і зашифровується ще на двох циклах з використанням $\Sigma_{3(64)}$ і $\Sigma_{4(64)}$. У підсумку результатом є $K_{A(128)}$. І, нарешті, виконується XOR-операція $K_{A(128)}$ з $K_{R(128)}$, результат шифрується за два цикли з використанням постійних значень $\Sigma_{5(64)}$ і $\Sigma_{6(64)}$. Результуючим значенням є $K_{B(128)}$.

Підключі $kw_{(64)}, k_{u(64)}$ і $kl_{v(64)}$ генеруються з (лівої або правої половини) циклічно до зсунутих значень $K_{L(128)}, K_{R(128)}, K_{A(128)}$ і $K_{B(128)}$.

Криптоалгоритм Camellia виявився стійким до усіх видів атак. відповідно до звіту NESSIE [13], у ході досліджень цього шифру не було знайдено жодної криптоаналітичної атаки, складність якої була б менше складності повного перебору. Наприклад, ймовірність знаходження лінійної і диференціальної характеристик дорівнює 2^{-28} для числа циклів шифрування, рівного 12, і ймовірність перебування цієї характеристики дорівнює 2^{-132} для 15 циклів шифрування.

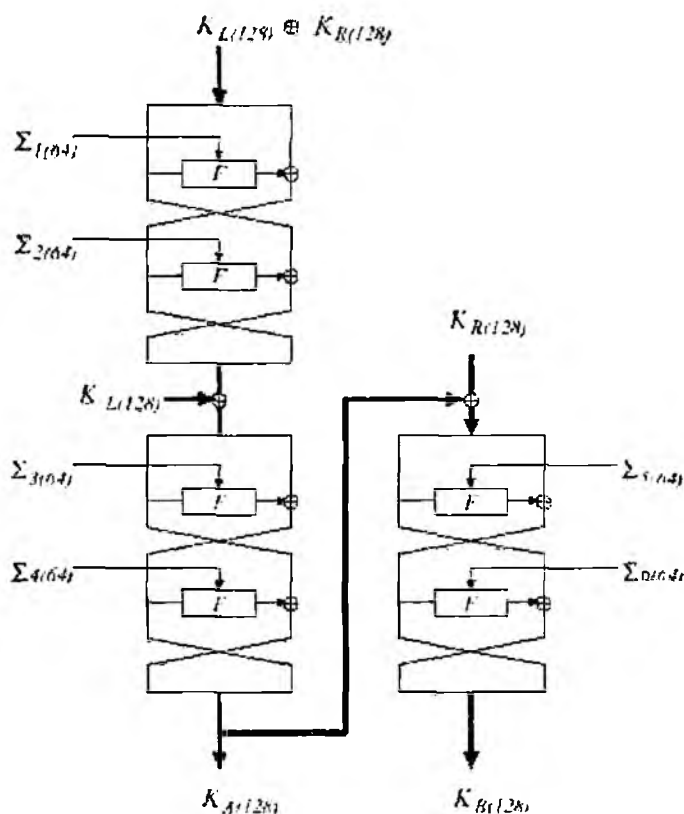


Рис. 4

4.6 SHACAL-2

SHACAL-2 [19] - це 256-бітовий блоковий шифр із 512-бітовим ключем, хоча він може бути змінений для використання 512-бітових блоків. SHACAL-2 зроблений на основі хеш-стандарту SHA-2, що був уведений NIST у 2000 році. SHA-2 - новий проект, що має деяку подібність до SHA-1, але є важливі розходження в структурі. Ми можемо відзначити, що функція стиску в

SHA-2 є 64-кроковою одноцикловою, на відміну від чотирициклової 80-крокової (4 цикли по 20 кроків) SHA-1. Таким чином, число кроків, що входять у функцію стиску, менше, ніж у SHA-1 (64 кроки в порівнянні з 80). З іншого боку, в SHA-2 дві змінні обновляються на кожному кроці функції стиску, а в SHA-1 тільки одна змінна.

Функція стиску SHA-2 може бути використана в режимі шифрування, з використанням ключа як повідомлення і відкритого тексту як початкової величини.

Коротко викладемо суть перетворень шифру Shacal. Поміщаємо 256-бітовий відкритий текст P у вісім 32-бітових змінних $AA, BB, CC, DD, EE, FF, GG, HH$, кожна 32-бітової довжини.

Псевдокод процедури зашифрування. На кожному з 64 кроків виконуються наступні перетворення:

$$\begin{aligned} H &+= \Sigma_0(A) + Maj(A, B, C) \\ D &+= H, \\ H &+= \Sigma_1(E) + Ch(E, F, G) + K_i + W_i, \\ A_i &= T_1 + T_2, \\ B_{i+1} &= A_i, \\ C_{i+1} &= B_i, \\ D_{i+1} &= C_i, \\ E_{i+1} &= D_i + T_1, \\ F_{i+1} &= E_i, \\ G_{i+1} &= F_i, \\ H_{i+1} &= G_i, \\ T_1 &= H + \Sigma_1(E) + Ch(E, F, G) + K_i + W_i, \\ T_2 &= \Sigma_0(A) + Maj(A, B, C), \end{aligned}$$

де W^i – 32-бітові крокові ключі; K^i – константи, різні на кожному кроці, $i = (\overline{0..n-1})$, і

$$\begin{aligned} Ch(XYZ) &= (X \& Y) \oplus (\bar{X} \& Z), \\ Maj(XYZ) &= (X \& Y) \oplus (X \& Z) \oplus (Y \& Z), \\ \Sigma_0(X) &= S_2(X) \oplus S_{13}(X) \oplus S_{22}(X), \\ \Sigma_1(X) &= S_6(X) \oplus S_{11}(X) \oplus S_{25}(X), \end{aligned}$$

де S_i – циклічний зсув вправо на i біт.

Результат виконання 64 кроків – значення восьми 32-бітових слів $A^{64}, B^{64}, C^{64}, D^{64}, E^{64}, F^{64}, G^{64}, H^{64}$. Криптограма C – є конкатенація значень попереднього пункту, тобто

$$C = [B^{64} || C^{64} || D^{64} || E^{64} || F^{64} || G^{64} || H^{64} || A^{64}]$$

У ході досліджень NESSIE не було знайдено вразливостей алгоритму SHACAL-2 [13]. SHA-2 – недавно розроблений примітив, тому необхідний час для здійснення обережної і ретельної оцінки безпеки SHA-2 і, відповідно, SHACAL-2.

4.7 Q

Q – блоковий симетричний шифр. Алгоритм шифрування перетворює блоки розміром 128 біт під керуванням ключа довжиною 128, 192 або 256 біт. Структура криптоалгоритму – це чергування процедур підстановки і перестановки. Підстановки для алгоритму взяті із шифру Rijndael, а процедури розсіювання – з алгоритму Serpent. За словами авторів, алгоритм є більш швидким і стійким, ніж Serpent. Алгоритм складається з 8 циклів перетворень, розглянемо послідовність перетворень окремого циклу.

Блок являє собою 16 байт. У кожному циклі спочатку виконується додавання (по XOR) з підключем (Key A Mixing), для кожного байта виконується підстановка байт-в-байт (підстановка взята із шифру Rijndael) (Byte Substitution), і ще раз виконується додавання з підключем (Key B Mixing). Потім виконується 32 рази підстановка 4 у 4 біти (для всіх 4 бітових кортежів) (Bit-slice substitution A), ще раз робиться додавання з підключем (Key n Mixing), далі слідує байтова перестановка (Permutation) і ще один шар 4-бітових підстановок (Bit-slice substitution B). Таким чином, послідовність перетворень кожного циклу така:

- 1) Key A Mixing
- 2) Byte Substitution
- 3) Key B Mixing
- 4) Bit-slice substitution A
- 5) Key n Mixing
- 6) Permutation
- 7) Bit-slice substitution B

При виробленні циклових ключів використовуються ті ж перетворення, що й в алгоритмі шифрування.

При аналізі стійкості цього шифру було розглянуто можливості організації атак диференціального, лінійного криптоаналізів, атак зі зв'язаними ключами, слайд-атаки, бумеранг-атаки, атаки нездійснених диференціалів. Було виявлено, що алгоритм із повним числом циклів вразливий до диференціальних і лінійних атак. Краща з атак на повний алгоритм вимагає наявності 2^{105} підібраних відкритих текстів і виконання 2^{77} операцій зашифрування. Для шифру з 192-бітовим ключем ці показники 2^{125} і 2^{96} , для 256-бітового ключа – 2^{125} і 2^{128} [13].

Можна зробити висновок, що алгоритм Q не відповідає вимогам, тому що для нього знайдено менш складні криптоаналітичні атаки у порівнянні з атакою типу "груба сила".

4.8 RC6

RC6 – алгоритм шифрування, що перетворює 128-бітні блоки під керуванням ключа з розміром до 256 біт. Структура RC6 – 20 циклів узагальненого ланцюга Фейстеля. Кожен цикл – послідовність операцій над чотирма 32-бітними словами. Основні операції: додавання по модулю 2^{32} , віднімання по модулю 2^{32} , побітне додавання по модулю 2 32-бітних слів, множення по модулю 2^{32} , параметричний зсув 32-бітних слів ($\lll$).

Схема одного циклу шифрування представлена на рис. 5.

На рис. 5 $f(x) = x \cdot (2x + 1)$, $S[i]$ – 32-бітний підключ.

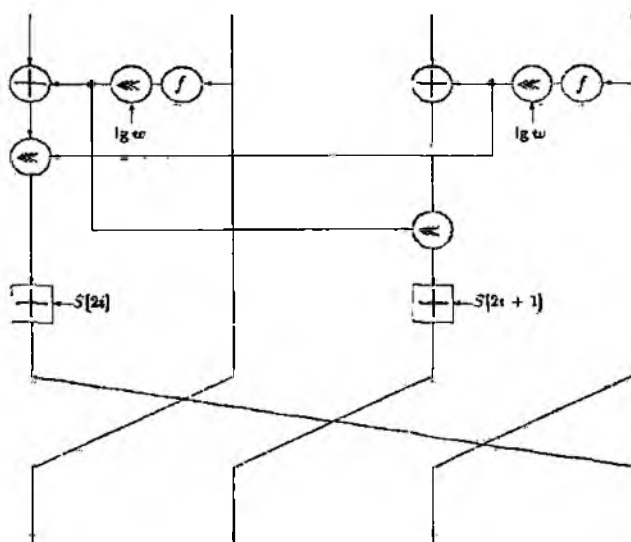


Рис. 5

Перед першим циклом і після останнього циклу виконується додаткове додавання двох з чотирьох 32-бітних слів з підключем.

При аналізі шифру показана його стійкість до диференціального і лінійного криптоаналізу. В роботі [20] запропоновано новий вид криптоаналізу, який названо χ^2 -атакою і який дозволяє здійснити ефективну атаку на 15-цикловий шифр. Крім того, показано, що серед кожних 2^{80} ключів існує один ключ, при використанні якого ця атака буде ефективною для 17-циклового шифру.

На наш погляд, в шифрі використовуються перетворення розсіювання та перемішування, які є далеко не найкращими серед відомих на сьогодні лінійних та нелінійних перетворень, і їх використання недостатньо обґрунтовано. Отже, був сумнів щодо надійності математичної бази цього шифру, тому було прийнято як $K_{\beta\beta} = 0$.

4.9 SAFER++

SAFER++ це представник сімейства блокових шифрів SAFER. Алгоритм працює з 128-бітним блоком під керуванням 128- або 256-бітного ключа. SAFER++ - це байт-орієнтований ітеративний шифр, що, у залежності від обраного розміру ключа, являє собою 7- або 10-циклову SPN-структуру. Два 16-байтових підключа використовуються в кожному циклі.

Цикл починається з першого подавання з підключем, причому частина байтів ключа (1,4,5,8,9,12,13,16) додається побітно за модулем 2 (XOR), а інші – побайтно за модулем 256.

Наступний крок складається з нелінійних S-блоків. Використовуються два види S-блоків: експонентний $X(a)=(45^a \bmod 257) \bmod 256$ і логарифмічний $L(a)=\log_{45} a \bmod 257$ з умовою, що $L(0)=128$ (7,128,46). Перший використовується для байтів 1,4,5,8,9,12,13,16, а другий – для інших.

На третьому кроці знову виконується змішування з підключем. Крок подібний до першого, але тепер до байтів 1,4,5,8,9,12,13,16 відповідні байти ключа додаються за модулем 256, а інші – за допомогою операції XOR.

Останній крок у циклі – це лінійне перетворення, що здійснюється в два етапи. Кожен етап складається з байтової перестановки і застосування 4-РНТ перетворень для кожного з 32-бітних фрагментів блоку. 4-РНТ перетворення може бути представлене як множення на матрицю розміром 4×4 байти.

Після останнього циклу виконується додаткове додавання з підключем, адалогічне тому, що виконується на початку кожного циклу. Загальна схема одного циклу перетворень представлена на рис. 6.

Автори розглядали стійкість шифру до диференціального і лінійного криптоаналізів. Вони стверджують, що шифр із цим і більше циклами стійкий до диференціального криптоаналізу, а 2,5 цикла забезпечують стійкість до лінійного криптоаналізу. Однак у роботі [21] була представлена лінійна атака на 4-цикловий шифр.

В результаті аналізу шифру виникли підозри щодо надійності математичної бази, застосованої в шифрі. Недостатньо обґрунтоване використання експонентного ($X(a)=(45^a \bmod 257) \bmod 256$) та логарифмічного ($L(a)=\log_{45} a \bmod 257$) S-блоків, які за такими криптографічними показниками, як максимальна ймовірність проходження нульової різниці або максимальна ймовірність лінійної апроксимації значень, що проходять через S-блок, є набагато гіршими, ніж відомі [17]. Тому було прийнято відповідне значення третього безумовного критерію: $K_{\beta\beta} = 0$.

4.10 ANUBIS

Структура шифру Anubis – чергування операцій перемішування (нелінійні підстановки) і розсіювання (лінійні перетворення). Алгоритм шифрування виконує перетворення 128-бітових блоків під керуванням ключа, розмір якого кращий 32 бітам (мінімальний розмір ключа 128 біт). Кількість циклів перетворення залежить від розміру ключа (мінімальна кількість циклів 12). При описі перетворень шифру зручно представити блок даних у вигляді матриці розміром 4 на 4 байти. Кожен цикл містить у собі побітове додавання по модулю 2 з підключем, нелінійні підстановки для кожного байта, лінійні перетворення, у ході яких виконується множення 32-бітних слів (колонки) на MDS-матрицю, і лінійне перетворення, що

полягає у виконанні транспозиції матриці. У цілому, як структура шифру, так і кожне окреме його перетворення, дуже схожі з шифром Rijndael. Основна відмінність від Rijndael полягає в тому, що Aneubis є інваріантним шифром, тобто при розшифруванні використовуються ті ж самі перетворення, що і при зашифруванні, оскільки кожне перетворення є інваріантним, тобто оберненим для самого себе.

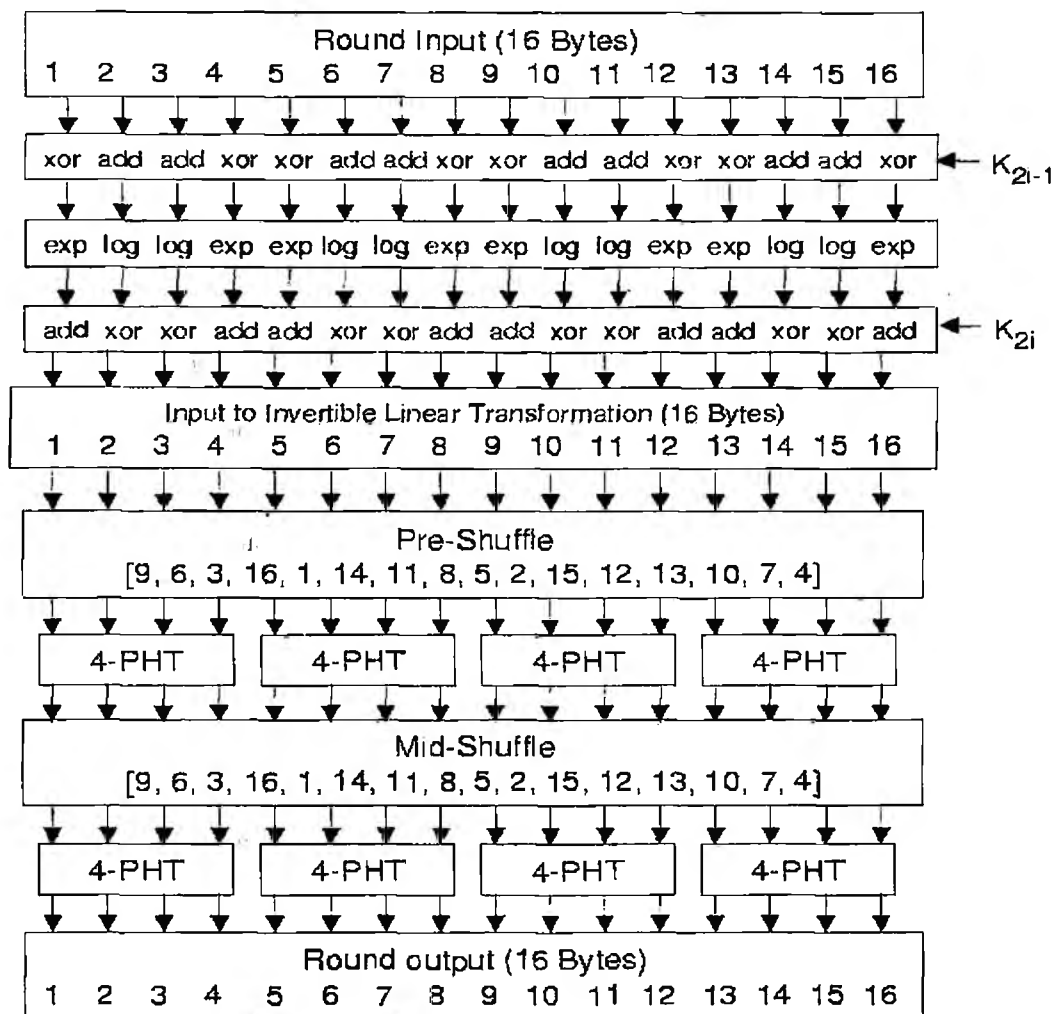


Рис. 6

Ще одна відмінність Aneubis від Rijndael полягає в схемі розгортання ключа. У цій схемі використовуються ті ж лінійні і нелінійні перетворення, що і в шифруючій функції, а також додавання з константами. На думку експертів [13], схема є досить складною, але при цьому і стійкою.

При аналізі криптостійкості шифру були розглянуті можливості здійснення диференціального, лінійного криптоаналізу, бумеранг-атаки, інтерполяційної атаки, атаки зв'язаних ключів, інтегральної атаки та атаки нездійснених диференціалів. Зроблено висновок [13] про стійкість шифру до цих видів атак. При цьому основним недоліком шифру є його практично повна ідентичність з алгоритмом Rijndael, через що шифр не пройшов навіть у другий раунд проекту NESSIE.

Таким чином, алгоритм шифрування Aneubis задовольняє сучасним вимогам до блокових симетричних шифрів, але при цьому є майже копією алгоритму AES.

4.11 HEROCRYPT-3

HEROCRYPT-3 – алгоритм шифрування, що перетворює 128-бітні блоки під керуванням ключа з розміром 128, 192 або 256 біт. Алгоритм шифрування містить у залежності від розміру ключа 6, 7 або 8 циклів. Схема перетворень одного циклу представлена на рис. 7

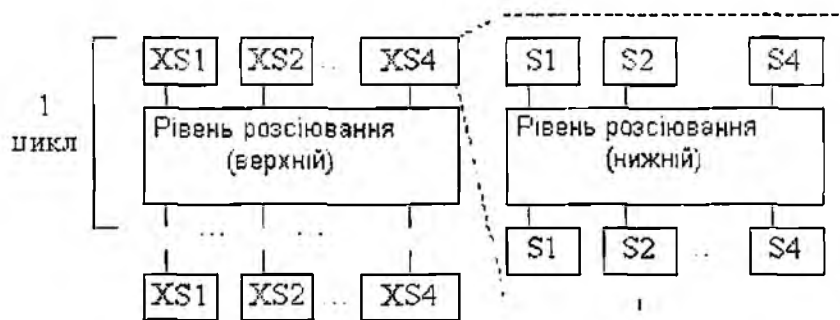


Рис. 7

S_i ($i = 1, \dots, 4$) являє собою послідовне виконання операції побітного додавання 128-бітного підключа по модулю 2 і виконання побайтних підстановок. Верхній і нижній рівні розсіювання представлені MDS-перетвореннями. Після останнього циклу виконується додатковий рівень XS перетворень.

Істотним недоліком алгоритму є повільна і складна процедура розгортання ключа. У цій процедурі використовується велике число додаткових перетворень.

Автори шифру розглянули його стійкість до диференціального і лінійного криптоаналізу, до інтегральної атаки, до атаки усічених диференціалів. У ході проекту NESSIE вдалося поліпшити практично усі з цих атак. Зазначимо, що для цього шифру вдалося оцінити верхню границю ймовірностей не тільки для окремих диференціальних і лінійних характеристик, але і для диференціалів і шаблонів лінійних апроксимацій, хоча довести теоретичну стійкість до диференціального і лінійного криптоаналізу не вдалося, оскільки ця границя виявилась недостатньо низькою (2^{-96}).

Таким чином, шифр Ніегосурт задовольняє основним сучасним вимогам, що висуваються до блокових симетричних шифрів.

4.12 NOEKEON

Цей шифр використовує 128-бітний блок і ключ. Шифр побудований за принципом SPN. Перетворення, що шифрують, складаються з 16 циклів. На початку кожного циклу до 128-бітного блока додається константа. Після цього блок розбивається на чотири підблоки, і виконується перетворення Theta, що забезпечує розсіювання (див. рис. 8).

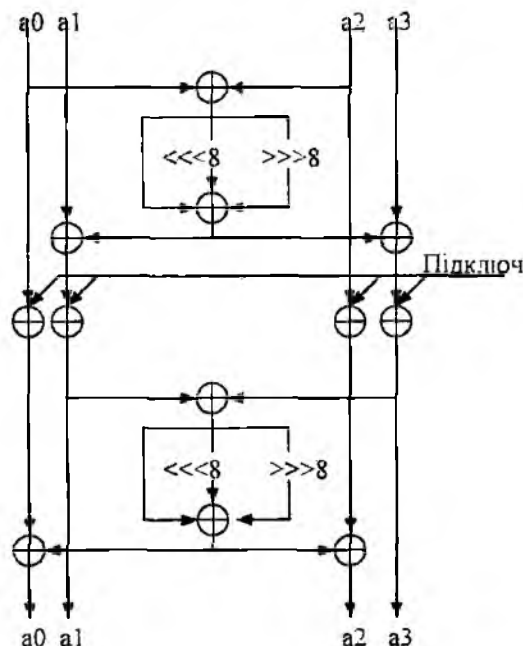


Рис. 8

У ході цього перетворення виконується, в тому числі, і додавання з підключем. Після перетворення Theta виконується циклічний зсув кожного з 32-бітних слів на різну кількість біт. Потім виконується рівень підстановок (32 підстановки 4 у 4 біти). Далі знову виконуються циклічні зсуви 32-бітних слів на те ж саме число біт, що і до цього, але в протилежну сторону.

Після останнього циклу виконується додаткове Theta-перетворення.

Шифр є інваріантним – та ж послідовність перетворень зі зворотним порядком використання підключів призводить до расшифрування вхідного повідомлення.

Авторами запропоновано дві схеми розгортання ключа. Перша пропонується до використання в тих випадках, коли немає погрози реалізації атаки зі зв'язаними ключами. Друга більш складна і, на думку авторів, досить стійка до цього виду атак. Але в ході проекту NESSIE показано, що і ця схема також вразлива до атак зі зв'язаними ключами [22].

Авторами шифру наведені досить обґрунтовані докази стійкості шифру до диференціального і лінійного криптоаналізів, а також до атаки усечених диференціалів і інтерполяційної атаки. Основним недоліком шифру можна вважати знайдені різниці для ключів, при яких, знаючи вхідну різницю відкритих текстів, з високою ймовірністю може бути передбачена різниця криптограм [22]. Викликає недовіру до шифру і той факт, що використання в шифрі інших нелінійних підстановок (S-блоків), що відповідають вимогам розробників шифру, призводить до того, що шифр стає вразливим до диференціального або лінійного криптоаналізів [22].

5 Порівняльний аналіз алгоритмів шифрування за безумовними критеріями

В табл. 3 представлені показники швидкості процедур за шифрування / розшифрування (такти процесора/байт) та процедури розгортання ключа (такти процесора / розгортання ключа), які були взяті із [5].

Таблиця 3

Назва шифру	Зашифрування	Місце	Розшифрування	Місце	Розгортання ключа	Місце	$K_{\text{сб}}$
AES(Rijndael)	23	1	23	1	494	4	1
Grand Cru	1856	12	2472	12	131000	11	0
NUSH	28	2	25	2	2579	8	1
SC2000	133	10	134	9	6988	10	0
Camellia	65	9	65	8	411	3	1
Shacal-2	60	7	59	6	1187	6	1
Q	61	8	146	11	1109	5	0
RC6	37	3	33	3	2073	7	1
Safer++	47	5	60	7	228	2	1
Anubis	37	4	37	4	4325	9	1
Hierocrypt-3	50	6	56	5	173000	12	0
Noekeon	167	11	137	10	58	1	0

Як вже зазначалося, шифри, що є трьома найгіршими по швидкості зашифрування, розшифрування або розгортання ключа, отримують $K_{\text{сб}} = 0$. Для інших - $K_{\text{сб}} = 1$.

Одержані значення показників $K_{\text{сб1}}$ - $K_{\text{сб6}}$ наведені у табл. 4.

З таблиці видно, що розглядати умовні критерії варто лише для шифрів AES, Camellia, Anubis та Shacal-2.

Що стосується шифру Anubis, то варто зазначити, що його структура майже цілком ідентична до структури шифру Rijndael, при цьому шифр набагато менше досліджений і вивчений, до того ж відомі роботи, в яких вказуються потенційні слабкості інваріантних шифрів ([23]), тому, на наш погляд, не слід розглядати його як претендента на перспективний шифр, який можна рекомендувати до використання на території України.

Таблица 4

Назва шифру	K_{61}	K_{62}	K_{63}	K_{64}	K_{65}	K_{66}
AES(Rijndael)	1	1	1	1	1	1
Grand Cru	1	1	1	1	1	0
NUSH	0	1	1	1	1	1
SC2000	1	1	1	1	1	0
Camellia	1	1	1	1	1	1
Shacal-2	1	1	1	1	1	1
Q	0	1	1	1	1	0
RC6	1	1	0	1	1	1
Safer++	1	1	0	1	1	1
Anubis	1	1	1	1	1	1
Hierocrypt-3	1	1	1	1	1	0
Noekeon	1	1	1	1	1	0

Список літератури: 1. Горбенко І.Д., Олійников Р.В., Руженцев В.І., Горбенко Ю.І., Михайленко М.С. Порівняльний аналіз алгоритмів блокового симетричного шифрування (за результатами виконання проекту NESSIE) // Радіотехніка, 2004 №138. 2. Андрейчиков А.В., Андрейчикова О.Н. Анализ, синтез, планирование решений в экономике. М.: Финансы и статистика, 2002. 386 с., ил. 3. Daemen J., Rijmen V. Security of a Wide Trail Design. <http://cryptonessie.org>. 4. Biham E. New Types of Cryptanalytic Attacks Using Related Keys. LNCS, Advances in Cryptology, proceeding of EUROCRYPT'93, 1993. 5. NESSIE public report D21. Performance of Optimized Implementations of the NESSIE Primitives. <http://cryptonessie.org>. 6. Daemen J. and Rijmen V., AES proposal: Rijndael. 1998. <http://www.nist.gov/aes>. 7. Daemen J., Knudsen L. R., Rijmen V. The block cipher Square. In Proceedings of Fast Software Encryption '97, no. 1267 in LNCS, Springer-Verlag, 1997. 8. Gilbert H., Minier M. A collision attack on seven rounds of Rijndael. In Proceedings of the Third Advanced Encryption Standard Conference. NIST. 2000. 9. Biham E., Keller N. Cryptanalysis of Reduced Variant of Rijndael. <http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html>. 10. Sugita M., Kobara K. Relationships among differential, truncated differential, impossible differential cryptanalyses against word-oriented block cipher like Rijndael, E2 // National Institute of Standards and Technology, <http://www.nist.gov/aes>. 11. Aoki K. Practical Evaluation of Security against Generalized Interpolation Attack. IEICE Transactions Fundamentals of Electronics, Communications and Computer Sciences (Japan), V. E83-A, No. 1, p. 33–38, 2000. (A preliminary version was presented at SAC'99). 12. Borst J., The Block Cipher: Grand Cru. Dept. Elektrotechnik-EAST/COSIC Kardinaal Mercierlaan 99. 13. NESSIE public report D20. NESSIE Security Report. <http://cryptonessie.org>. 14. Lebedev A. N., Volchikov A. A. Nush. Primitive submitted to NESSIE by LAN Crypto, Int., Sept. 2000. 15. Wenling W., Dengguo F. Linear cryptanalysis of NUSH block cipher. Science in China, V 45, Nr1, 2002. 16. Shimoyama T., Yanami H., Yokoyama K. Specification and Supporting Document of the Block Cipher SC2000. 17. Nyberg K. Differentially uniform mappings for cryptography. Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765, Helleseht T., Ed., Springer-Verlag, 1994, p. 55-64. 18. Aoki K., Ichikawa T., Kanda M., Matsui M., Moria S i, Nakajima J., and Tokita T. Camellia: A 128-bit block cipher suitable for multiple platforms. Primitive submitted to NESSIE by NTT Corp., Sept. 2000. See also <http://info.isl.ntt.co.jp/camellia/>. 19. Handschuh H., Naccache D. SHACAL. Submission Statement. 20. H. Gilbert, H. Handschuh et al. A Statistical Attack on RC6. Fast Software Encryption, vol. 1978 of Lecture Notes in Computer Science, Springer-Verlag. 2001. 21. Oswald E., Daemen J., Rijmen V. AES - The State of the Art of Rijndael's Security. 22. L. R. Knudsen, H. Raddum. On Noekeon. in Proceedings of the Second NESSIE Workshop, 2001 NES/DOC/UIB/WP3/009. 23. Biryukov A. Analysis of involutational ciphers: Khazad and Anubis. FSE'03, Lecture Notes in Computer Science, Springer-Verlag, 2003.

Харьковский национальный
университет радиозлектроники

Поступила в редколлегию 05.10.2004

І. Д. ГОРБЕНКО, д-р техн. наук, А. І. ПУШКАРЬОВ, Р. В. ОЛІЙНИКОВ, канд. техн. наук,
В. І. РУЖЕНЦЕВ, канд. техн. наук, Ю. І. ГОРБЕНКО, М. С. МИХАЙЛЕНКО

ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ БЛОКОВОГО СИМЕТРИЧНОГО ШИФРУВАННЯ (ЗА РЕЗУЛЬТАТАМИ ВИКОНАННЯ ПРОЕКТУ NESSIE)

В [1] обґрунтований підхід порівняння блокових симетричних шифрів на основі нечітких множин, обґрунтовано та вибрано критерії, а також проведено аналіз властивостей блокових симетричних шифрів, що розглядалися в проекті NESSIE [1].

В цій статті наведено результати порівняння алгоритмів блокового симетричного шифрування, що розглядалися у проекті NESSIE, та висновки щодо їх перспективності.

1 Порівняльний аналіз алгоритмів шифрування за умовними критеріями

Першим умовним критерієм є частковий критерій оцінки можливості та умови поширення алгоритму шифрування. За цим критерієм пропонується оцінювати:

- умови розробки та дослідження (наявність) обмежень уже на цьому етапі;
- відсутність обмежень на експорт;
- відсутність обмежень на імпорт;
- необхідність одержання ліцензії на використання алгоритму;
- контроль за застосуванням та обов'язковість звітування;
- наявність обмежень на експорт засобів, що реалізують алгоритм;
- наявність патентів на алгоритм та його складові.

Ці показники визначені на основі вивчення нормативно-правових актів, що діють на міжнародному і національному рівнях [3,4].

На основі описаних вище показників методом експертних оцінок устанавлюються переваги того або іншого претендента у вигляді вагових коефіцієнтів приватного критерію:

$$K_{y1} = \sum W_i$$

де W_i – числові значення усереднених експертних оцінок (вагового коефіцієнта) для кожного з варіантів.

Варто зазначити, що алгоритми FIPS-197 (AES) і Camellia використовуються як національні стандарти і дані про патентні обмеження на них добре описані у відкритих джерелах. а щодо алгоритму SHACAL-2 доводиться керуватися тільки даними проекту NESSIE.

У табл. 1 наведені результати оцінки за приватним умовним критерієм поширення алгоритму. Усього за цим критерієм будемо використовувати сім показників W_i , що наведені вище. Кожний з показників може приймати три значення:

- 0 – є обмеження;
- 1 – не визначений;
- 2 – дозволений або не обмежений.

Таблиця 1

Показники	W_1	W_2	W_3	W_4	W_5	W_6	W_7	K_{y1}
Алгоритми								
FIPS-197	2	2	2	2	2	0	2	12
Camellia	0	0	1	0	1	0	0	2
SHACAL-2	1	1	1	1	2	0	1	7

Оцінку реальної захищеності від відомих криптоаналітичних атак при зменшеному числі циклів проведено з використанням

$$K_{y2} = \sum W_i, \quad (1)$$

де n – число методів криптоаналізу, застосованих до розглянутого алгоритму;

W_i – числове значення (усереднені ваги) експертної оцінки стійкості відносно i -го методу криптоаналізу.

В якості основних атак будемо використовувати:

- диференціальний аналіз;
- атаку нездійснених диференціалів;
- колізії;
- Square атаку;
- зв'язані ключі;
- усічені диференціали;
- диференціали високого порядку;
- лінійний криптоаналіз;
- інтегральний криптоаналіз;
- атаку на реалізацію.

Оскільки відповідно до попередніх досліджень розроблювачів алгоритмів FIPS 197, Camellia і SHACAL-2, на повному числі циклів вони є стійкими проти відомих криптоаналітичних атак, то одним з показників оцінки стійкості можна вважати число циклів, при якому та або інша криптоаналітична атака може бути здійснена успішно.

Ваги коефіцієнтів W_i в (1) обрані в такий спосіб:

0 – якщо число циклів, на яких здійснена атака, більше $n_c/2$, де n_c повне число циклів;

1 – якщо число циклів менше $n_c/2$.

2 – якщо атака не може бути реалізована.

В табл. 2 наведено результати порівняння алгоритмів шифрування за частковим умовним критерієм K_{y2} .

Таблиця 2

Показники Алгоритми	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	K_{y2}
FIPS 197	2	1	1	1	1	2	2	2	1	1	14
Camellia	1	1	1	1	2	1	1	1	2	1	12
SHACAL-2	1	1	2	1	1	1	1	1	1	1	11

Критерій рівня довіри алгоритмові шифрування будемо визначати за складеними W_{31} , W_{32} , W_{33} (усереднені експертні оцінки) третього приватного умовного критерію K_{y3} :

$$K_{y3} = W_{31} + W_{32} + W_{33},$$

де W_{31} – оцінка ступеня відкритості правил проектування і прозорості використовуваних методів;

W_{32} – достатність кількості і авторитет числа незалежних дослідників, що аналізують стійкість і властивості шифру;

W_{33} – оцінка відсутності підозр на наявність вразливостей.

У табл. 3 наведені значення показника K_{y3} і його складових W_{31} , W_{32} і W_{33} . Оцінка зроблена, як і вище за трибальною системою $W_{3i} \in \{0, 1, 2\}$.

Таблиця 3

Показники Алгоритми	W_{31}	W_{32}	W_{33}	K_{y3}
FIPS 197	2	2	2	6
Camellia	2	1	1	4
SHACAL-2	1	1	1	3

Оцінка перспективності застосування алгоритму шифрування зроблена з використанням приватного умовного критерію K_{y4} і відповідних показників W_{4i} :

$$K_{y4} = W_{41} + W_{42} + W_{43} + W_{44},$$

де W_{41} – числове значення (вага) експертної оцінки про перевагу шифру на основі зведень про його використання в якості міжнародного стандарту (стандартів) і/або національного стандарту (стандартів);

W_{42} – числове значення експертної оцінки показника ступеня поширеності алгоритму;

W_{43} – числове значення експертної оцінки існування (відсутності) підозр на теоретичну можливість здійснення криптоаналітичної атаки і її погрози;

W_{44} – числове значення експертної оцінки можливості застосування в перспективних інформаційних технологіях.

Оцінка зроблена за трибальною системою показників.

У табл. 4 наведені значення показників і приватного критерію K_{y4} .

Таблиця 4

Показники	W_{41}	W_{42}	W_{43}	W_{44}	K_{y4}
FIPS 197	2	2	1	2	7
Camellia	1	1	1	2	5
SHACAL-2	0	0	1	2	3

Оцінка тимчасової і просторової складності, програмної, апаратної й апаратно-програмної реалізацій прямого і зворотного перетворень виконана з використанням приватного критерію K_{y5} :

$$K_{y5} = W_{51} + W_{52} + W_{53} + W_{54},$$

де W_{51} – числове значення експертної оцінки показника складності прямого криптографічного перетворення;

W_{52} – числове значення експертної оцінки показника складності зворотного криптографічного перетворення;

W_{53} – числове значення експертної оцінки просторової складності;

W_{54} – числове значення експертної оцінки розміру вихідного коду реалізації (реалізацій).

При оцінці використовувалися результати досліджень, наведені в першому і другому розділах. Результати оцінки наведені в табл. 5. Оцінка виконана за трибальною системою.

Таблиця 5

Показники Алгоритми	W_{51}	W_{52}	W_{53}	W_{54}	W_{55}	K_{y5}
FIPS 197	2	2	2	1	1	8
Camellia	1	1	2	1	2	7
SHACAL-2	1	1	2	1	2	7

Аналіз показника K_{y5} свідчить, що алгоритми шифрування за цим показником розміщені в такій послідовності FIPS 197, Camellia і SHACAL-2.

Оцінки режимів застосування розглянутих алгоритмів шифрування виконані згідно до часткового критерію K_{y6} :

$$K_{y6} = W_{61} + W_{62} + W_{63} + W_{64} + W_{65},$$

де $W_{61}, W_{62}, W_{63}, W_{64}, W_{65}$ – числові значення експертних оцінок використання алгоритму відповідно в режимах блокового, потокового шифрування з лічильником, потокового шифрування зі зворотним зв'язком, вироблення імітовставки і формування псевдовипадкових послідовностей.

При оцінці враховувався як сам факт наявності режиму роботи, так і його характеристики. У табл. 6 наведені результати експертних оцінок за трибальною системою.

Таблиця 6

Показники Алгоритми	W_{61}	W_{62}	W_{63}	W_{64}	W_{65}	K_{y6}
FIPS 197	2	2	2	2	2	10
Camellia	2	2	2	2	1	9
SHACAL-2	2	2	2	2	1	9

З табл. 6 випливає, що аналізовані алгоритми шифрування можуть застосовуватися у всіх режимах приблизно з однаковою ефективністю.

Оцінка ступеня гнучкості здійснювалася згідно критерію K_{y7} :

$$K_{y7} = W_{71} + W_{72} + W_{73} + W_{74} + W_{75},$$

де W_{71} – числове значення експертної оцінки можливості зміни довжини блока;

W_{72} – числове значення експертної оцінки можливості зміни довжини ключа;

W_{73} – числове значення експертної оцінки можливості реалізації на різних програмних платформах;

W_{74} – числове значення експертної оцінки можливості апаратної реалізації;

W_{75} – можливість використання для реалізації криптографічних протоколів. Результати оцінки зведені в табл. 7.

Таблиця 7

Показники Алгоритми	W_{71}	W_{72}	W_{73}	W_{74}	W_{75}	K_{y7}
FIPS 197	1	2	1	2	2	8
Camellia	1	2	1	2	2	8
SHACAL-2	1	1	2	1	2	7

З табл. 7 випливає, що алгоритми FIPS 197 і Camellia мають приблизно однакові показники за гнучкістю. Їм поступається алгоритм SHACAL-2, тому що в ньому фіксовані як довжина блоку, так і довжина ключа, а також немає даних про ефективну апаратну реалізацію алгоритму.

Тепер методом експертної оцінки для кожного з розглянутих алгоритмів шифрування для кожного критерію визначаємо нечіткі множини:

$$\begin{aligned} \mu_{K_{y1}} &= \{1/12 + 0,2/2 + 0,6/7\}; \\ \mu_{K_{y2}} &= \{1/14 + 0,9/12 + 0,85/11\}; \\ \mu_{K_{y3}} &= \{0,8/6 + 0,6/4 + 0,5/3\}; \\ \mu_{K_{y4}} &= \{0,7/7 + 0,5/5 + 0,3/3\}; \\ \mu_{K_{y5}} &= \{0,9/8 + 0,8/7 + 0,8/7\}; \\ \mu_{K_{y6}} &= \{0,6/10 + 0,55/9 + 0,55/9\}; \\ \mu_{K_{y7}} &= \{0,5/8 + 0,5/8 + 0,4/7\}; \end{aligned}$$

Цей запис розуміється так: для кожного часткового умовного критерію отримані значення $K_{y1} - K_{y7}$ – (табл. 1 – 6) перетворюються в нечіткі множини за допомогою згортки (2).

У нашому випадку всі часткові умовні критерії $K_{y1} - K_{y7}$ мають різну значимість при виборі найбільш кращого варіанта. У зв'язку з цим необхідно визначити вагові коефіцієнти β_i цих критеріїв. Один з можливих способів одержання значень вагових коефіцієнтів полягає в побудові матриці попарних порівнянь критеріїв [2]. Для приватних умовних критеріїв маємо відповідну табл. 8.

Таблиця 8

Умовні критерії	K_{y1}	K_{y2}	K_{y3}	K_{y4}	K_{y5}	K_{y6}	K_{y7}
K_{y1}	1	1/3	1	3	1/3	3	3
K_{y2}	3	1	5	5	2	3	5
K_{y3}	1	1/5	1	3	1/5	3	3
K_{y4}	1/3	1/5	1/3	1	1/5	3	3
K_{y5}	3	1/2	5	5	1	1	3
K_{y6}	1/3	1/3	3	1/3	1	1	1
K_{y7}	1/3	1/5	3	1/3	1/3	1	1

Ваговий коефіцієнт критеріїв β_i визначається на підставі обчислених значень правого часткового вектора матриці попарних порівнянь [2] α_i з наступним множенням на число критеріїв n :

$$\beta_i = \alpha_i n$$

Значення α_i і β_i , тобто власний вектор матриці попарних порівнянь критеріїв і їх вагові коефіцієнти, наведені в табл. 9.

Таблиця 9

	K_{y1}	K_{y2}	K_{y3}	K_{y4}	K_{y5}	K_{y6}	K_{y7}
Значення α_i	0,125	0,322	0,116	0,076	0,242	0,071	0,046
Значення β_i	0,875	2,254	0,812	0,532	1,694	0,497	0,323

Множина оптимальних альтернатив D з урахуванням різної важливості критеріїв ефективності визначається шляхом перетинання нечітких множин:

$$D = C_1^{\beta_1} \cap C_2^{\beta_2} \cap \dots \cap C_n^{\beta_n} \quad (2)$$

Множина оптимальних альтернатив з обчисленням отриманих вагових критеріїв за формулою (2):

$$D = \{ \min \{ 1, 0^{0,875}; 1, 0^{2,254}; 0,8^{0,812}; 0,7^{0,532}; 0,9^{1,694}; 0,6^{0,497}; 0,5^{0,323} \}, \\ \min \{ 0,2^{0,875}; 0,9^{2,254}; 0,6^{0,812}; 0,5^{0,532}; 0,8^{1,694}; 0,55^{0,497}; 0,5^{0,323} \}, \\ \min \{ 0,6^{0,875}; 0,85^{2,254}; 0,5^{0,812}; 0,3^{0,532}; 0,8^{1,694}; 0,55^{0,497}; 0,4^{0,323} \} \}$$

Згортка множини оптимальних варіантів для алгоритмів AES, Camellia і SHACAL-2 має вигляд:

1. За методом мінімуму: $\max_j \mu_D(a_j) = \max \{0,775; 0,244; 0,527\}$;

2. З урахуванням зваженої оцінки всіх критеріїв: $\max_j \mu_D(a_j) = \max \{0,358; 0,035; 0,066\}$

Запропонована методика дозволяє здійснити порівняльний аналіз перспективних алгоритмів шифрування. У табл. 10 наведені значення безумовного й умовного критеріїв оцінки алгоритмів шифрування за результатами порівняльного аналізу перспективних БСШ.

Таблиця 10

Алгоритми		Rijndael (FIPS-197)	SHACAL-2	Camellia
Критерії				
Безумовний K_6		1	1	1
Умовний K_y	Метод мінімуму	0,775	0,527	0,244
	Облік усіх критеріїв	0,358	0,066	0,035

Аналіз даних табл. 10 дозволяє констатувати, що найбільш кращим є алгоритм Rijndael (FIPS-197). На другому місці знаходиться алгоритм БСШ SHACAL-2, на третьому – Camellia. Тому, на наш погляд, можна зробити висновок про те, що в Україні для вирішення задач забезпечення конфіденційності інформації сьогодні доцільно рекомендувати до використання алгоритм блокового симетричного шифрування Rijndael (AES).

Список літератури: 1. Горбенко І.Д., Гулак Г.М., Олійников Р.В., Руженцев В.І., Михайленко М.С. Аналіз властивостей алгоритмів блокового симетричного шифрування (за результатами виконання проекту NESSIE). // Радіотехніка, 2004 №138. 2. Андрейчиков А.В., Андрейчикова О.Н. Анализ, синтез, планирование решений в экономике. М.: Финансы и статистика, 2002. 386 с., ил. 3. Daemen J., Rijmen V. The Wide Trail Design Strategy. <http://www.nist.gov/aes> 4. Harpes C., Kramer G.G., Massey J.L. A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma. LNCS, Advances in Cryptology, proceedings of Eurocrypt '95, 1995.

Харківський національний
університет радіоелектроніки

Надійшла до редколегії 04.10.2004

КРИТЕРИЙ ПРИМИТИВНОСТИ ГРУППЫ ПОДСТАНОВОК, ПОРОЖДЕННОЙ РАУНДОВЫМИ ШИФРУЮЩИМИ ПРЕОБРАЗОВАНИЯМИ ШИФРА ФЕЙСТЕЛЯ

Одним из общих требований к современным блочным шифрам является их обоснованная стойкость относительно алгебраических методов криптоанализа, основанных на группировании открытых, шифрованных сообщений или ключей шифра в классы эквивалентных (или близких, в том или ином смысле) объектов, позволяющем понизить трудоемкость алгоритмов решения соответствующих криптоаналитических задач. Стойкость блочных шифров к подобным методам криптоанализа, получившим в ряде публикаций название методов гомоморфных образов или гомоморфизмов [1 – 3], как правило, определяется алгебраическими свойствами различных групп подстановок, связанных с системой раундовых шифрующих преобразований данного блочного шифра $\mathcal{S}$ (точные определения основных понятий, используемых в статье, приведены ниже).

Изучению взаимосвязи между криптографической стойкостью шифра $\mathcal{S}$ и свойствами группы $G(\mathcal{S})$, порожденной его раундовыми шифрующими преобразованиями, посвящены работы [4 – 9]. В [5, 6] описаны атаки (на основе известных или выбираемых открытых сообщений) на блочные шифры, раундовые преобразования которых порождают группы подстановок малого порядка (например, такого, при котором возможно практическое перечисление всех шифрующих преобразований, действующих в определенных последовательных раундах шифрования).

В [4] показано, что необходимым условием стойкости блочного шифра $\mathcal{S}$ к алгебраическим атакам, основанным на группировании открытых и шифрованных сообщений, является примитивность группы $G(\mathcal{S})$. Показано также, что в случае, когда $G(\mathcal{S})$ является импримитивной группой подстановок, шифр $\mathcal{S}$ может содержать так называемые скрытые “лазейки”, существенно снижающие его практическую стойкость. В доказательство этого в [4] приведен пример 32-раундового шифра с длинами блока и ключа шифрования, равными соответственно 64 и 80 битам. Этот шифр по своей структуре аналогичен алгоритму DES, за исключением конструкции s-блоков, и является практически стойким к линейному и разностному криптоанализу. Вместе с тем, импримитивность группы, порожденной раундовыми преобразованиями этого шифра, позволяет осуществить на него алгебраическую атаку с использованием 2^{32} подобранных открытых текстов. Отметим, что в примере из [4] “лазейка” построена на основе системы импримитивности группы $G(\mathcal{S})$, блоками которой являются смежные классы аддитивной группы векторного пространства V_{64} (шифруемых сообщений) по некоторой подгруппе этой группы.

В [4] поставлен вопрос о том, существуют ли “неискусственные” примеры блочных шифров (аналогичных используемым в настоящее время шифрам Фейстеля и др.), группы, порожденные раундовыми преобразованиями, которых имеют системы блоков импримитивности, отличные от “линейных” систем, состоящих из смежных классов абелевой группы $(V_n, \oplus)$ по ее подгруппам.

Как следует из основного результата настоящей статьи и теоремы, доказанной в [10], ответ на этот вопрос по существу является отрицательным.

Цель данной статьи состоит в нахождении необходимых и достаточных условий, при которых раундовые шифрующие преобразования шифра Фейстеля $\mathcal{S}$ с обратной по каждому аргументу раундовой функцией ϕ порождают примитивную группу подстановок. В качестве примера применения полученного критерия примитивности рассматривается группа, порожденная раундовыми шифрующими преобразованиями блочного шифра GOST [11], и показывается, что эта группа является примитивной независимо от конкретных значений s-блоков указанного шифра.

1 Определения основных понятий и вспомогательный результат

Пусть V_n – пространство двоичных векторов длины $n = 2m$, где $m \geq 2$, $K = V_m$, $(f_k: V_n \rightarrow V_n, k \in K)$ – семейство подстановок на множестве V_n . Рассмотрим r -раундовый ($r \geq 2$) блочно-итерационный шифр $\mathfrak{S} = (V_n, \Lambda, F)$ с множеством открытых (шифрованных) сообщений V_n , множеством ключей $\Lambda = K^r$ и функцией шифрования $F: V_n \times \Lambda \rightarrow V_n$ [11]. Преобразование F_λ открытого сообщения $x \in V_n$ в шифрованный текст $y \in V_n$ на ключе $\lambda = (k(1), \dots, k(r)) \in \Lambda$ является композицией r раундовых шифрующих преобразований $f_{k(1)}, \dots, f_{k(r)}$, определяемых ключом λ :

$$y = F_\lambda(x) = f_{k(r)} \dots f_{k(1)}(x), x \in V_n.$$

Далее предполагается, что $\mathfrak{S}$ является шифром Фейстеля, а именно, существует отображение $\varphi: V_m \times V_m \rightarrow V_m$ такое, что для любого $k \in K$ выполняется равенство

$$f_k(x) = f_k(x_1, x_2) = (x_2, x_1 \oplus \varphi(x_2, k)), x = (x_1, x_2) \in V_n \quad (1)$$

(здесь и ниже символы x_1, x_2 обозначают соответственно левую и правую части длины m произвольного вектора $x \in V_n$). Кроме того, предполагается, что алгебраическая система (V_m, φ) является квазигруппой. (Напомним [3], что непустое множество A с заданной на нем бинарной алгебраической операцией $h: A \times A \rightarrow A$ называется *квазигруппой*, если для любых $a, b \in A$ каждое из уравнений $h(x, a) = b$, $h(a, y) = b$ имеет единственное решение $x \in A$ и $y \in A$, соответственно).

Пусть (A, h) – произвольная конечная квазигруппа, и $\varepsilon_1, \varepsilon_2, \varepsilon_3$ – отношения эквивалентности на множестве A . Следуя [2], назовем систему $\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ *конгруэнцией* квазигруппы (A, h) , если для любых $x, x', y, y' \in A$ соотношения

$$x \equiv x' \pmod{\varepsilon_1}, y \equiv y' \pmod{\varepsilon_2}$$

влекут соотношение

$$h(x, y) \equiv h(x', y') \pmod{\varepsilon_3}.$$

Конгруэнция $\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ называется *следствием конгруэнции* $\delta = (\delta_1, \delta_2, \delta_3)$, если выполняются условия $\varepsilon_1 \subseteq \delta_1, \varepsilon_2 \subseteq \delta_2, \varepsilon_3 \supseteq \delta_3$.

Обозначим через 0_A и 1_A наименьшее и наибольшее отношения эквивалентности на множестве A , соответственно. Конгруэнция $\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ квазигруппы (A, h) называется *нетривиальной*, если $\varepsilon_3 \neq 1_A$ и выполняется хотя бы одно из неравенств $\varepsilon_1 \neq 0_A, \varepsilon_2 \neq 0_A$. *Максимальной конгруэнцией* квазигруппы (A, h) называется ее произвольная конгруэнция ε , не имеющая нетривиальных следствий, отличных от ε [2]. Очевидно, что каждая нетривиальная конгруэнция квазигруппы (A, h) является следствием ее некоторой максимальной конгруэнции.

Пусть квазигруппа (A, h) является абелевой группой, и L – произвольная подгруппа группы A . Тогда *отношением смежности A по L* называется отношение эквивалентности на группе A , задаваемое посредством разложения A в смежные классы по подгруппе L .

В [2] описаны максимальные конгруэнции конечных квазигрупп, скрещенных по изотопным группам. Поскольку это описание существенно используется при обосновании основного результата, изложенного далее в статье, остановимся на нем подробнее, ограничив рассмотрение конкретным классом квазигрупп, связанных с шифрами Фейстеля.

Для любого $k \in K$ положим $\varphi_k(u) = \varphi(u, k)$, $u \in V_m$. Рассмотрим на множестве V_n бинарную операцию

$$h((x_1, x_2), (k_1, k_2)) = (x_1 \oplus \varphi_{k_1}(x_2), x_2 \oplus \varphi_{k_2}(x_1 \oplus \varphi_{k_1}(x_2))), \quad (2)$$

где $(x_1, x_2), (k_1, k_2) \in V_n$. Нетрудно видеть, что значение отображения h является результатом преобразования блока (x_1, x_2) на тактовых ключах k_1, k_2 шифра $\mathfrak{Z}$ в двух последовательных раундах шифрования. Так как, согласно сделанному выше предположению, алгебраическая система (V_m, ϕ) является квазигруппой, то для любых $x = (x_1, x_2), k = (k_1, k_2), y = (y_1, y_2) \in V_n$ уравнение

$$h((x_1, x_2), (k_1, k_2)) = (y_1, y_2)$$

однозначно разрешимо относительно x и k при фиксированных k, y и x, y соответственно. Таким образом, отображение h является квазигрупповой операцией на множестве V_n . При этом равенство (2) можно записать в виде

$$h((x_1, x_2), (k_1, k_2)) = (x_1, x_2) \oplus B((x_1, x_2), (k_1, k_2)), \quad (3)$$

где

$$B((x_1, x_2), (k_1, k_2)) \stackrel{\text{def}}{=} (\phi_{k_1}(x_2), \phi_{k_2}(x_1 \oplus \phi_{k_1}(x_2))) \quad (4)$$

есть обратимая справа бинарная операция на множестве V_n .

Соотношения (3), (4) означают, что квазигруппа (V_n, h) R -изотопна абелевой группе $(V_n, \oplus)$ при R -изотопии (id_{V_n}, B, id_{V_n}) , где id_{V_n} – тождественное отображение множества V_n [2]. Отсюда на основании теоремы 2 из [2] вытекает следующий результат.

Теорема 1 [2]. Пусть $\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ – нетривиальная максимальная конгруэнция квазигруппы (V_n, h) . Тогда существует собственная подгруппа L группы V_n такая, что отношение ε_1 совпадает с отношением смежности V_n по L , а отношение ε_3 содержит указанное отношение смежности.

Ниже символом S^X обозначается симметрическая группа подстановок произвольного конечного множества X . Напомним [12], что группа подстановок $G \subseteq S^X$ называется *транзитивной*, если для любых $a, b \in X$ существует элемент $g \in G$ такой, что $g(a) = b$. Множество $\Delta \subseteq X$ называется *блоком* группы $G \subseteq S^X$, если для любого $g \in G$ выполняется одно из условий $g(\Delta) = \Delta, g(\Delta) \cap \Delta = \emptyset$. В случае, когда $1 < |\Delta| < |X|$, блок Δ называется *нетривиальным*. Группа подстановок, не имеющая нетривиальных блоков, называется *примитивной*, а транзитивная группа, не являющаяся примитивной, – *импримитивной* группой подстановок.

Пусть $G \subseteq S^X$ – транзитивная группа, Δ – ее блок. *Полной системой блоков импримитивности* группы G (сопряженных с блоком Δ) называется совокупность всех попарно различных множеств вида $g(\Delta), g \in G$. Эта совокупность множеств называется *нетривиальной*, если Δ – нетривиальный блок группы G .

2 Основная теорема

Обозначим через $G(\mathfrak{Z}) = \langle f_k : k \in K \rangle$ подгруппу симметрической группы множества V_n , порожденную раундовыми шифрующими преобразованиями (1) шифра Фейстеля $\mathfrak{Z}$. Очевидно, что $G(\mathfrak{Z})$ является транзитивной на множестве V_n группой подстановок.

Основной результат настоящей статьи содержит следующая теорема, устанавливающая необходимые и достаточные условия примитивности указанной группы в терминах раундовой функции ϕ шифра Фейстеля $\mathfrak{Z}$.

Теорема 2. Пусть $\mathfrak{Z} = (V_n, \Lambda, F)$ – блочный шифр, раундовые шифрующие преобразования которого определяются по формуле (1), где отображение $\phi: V_m \times V_m \rightarrow V_m$ является квазигрупповой операцией на множестве V_m . Положим $G(\mathfrak{Z}) = \langle f_k : k \in K \rangle, G_\phi = \langle \phi_k : k \in K \rangle$.

Тогда, если $G(\mathfrak{Z})$ – импримитивная группа подстановок, то выполняется одно из следующих условий:

- (а) группа G_φ импримитивна и обладает полной системой блоков импримитивности, состоящей из смежных классов группы V_m по некоторой ее собственной ненулевой подгруппе;
- (б) существует невырожденное линейное преобразование g векторного пространства V_m такое, что для любого $k \in K$ выполняется равенство

$$\varphi_k(y) = g(y) \oplus g^{-1}(y) \oplus \varphi_k(0), \quad y \in V_m. \quad (5)$$

Обратно, при выполнении хотя бы одного из условий (а), (б) группа $G(\mathfrak{Z})$ является импримитивной на множестве V_n .

Для доказательства теоремы установим ряд вспомогательных утверждений.

Лемма 1. Пусть $G(\mathfrak{Z})$ является импримитивной группой подстановок, $\varepsilon = \{X_1, \dots, X_n\}$ – нетривиальная полная система блоков импримитивности этой группы. Тогда существует подгруппа L группы V_n такая, что ε является системой различных смежных классов V_n по L .

Доказательство. Справедливость леммы вытекает непосредственно из теоремы 1. Действительно, на основании определения понятия полной системы блоков импримитивности группы подстановок, тройка отношений эквивалентности $(\varepsilon, 0_{V_n}, \varepsilon)$ является конгруэнцией квазигруппы (V_n, h) . При этом, в силу условий $\varepsilon \neq 0_{V_n}$, $\varepsilon \neq 1_{V_n}$ указанная конгруэнция нетривиальна и, значит, является следствием некоторой максимальной конгруэнции $\varepsilon_{\max} = (\varepsilon_{\max}^{(1)}, \varepsilon_{\max}^{(2)}, \varepsilon_{\max}^{(3)})$ квазигруппы (V_n, h) . Согласно теореме 1 существует собственная подгруппа L группы V_n такая, что отношение $\varepsilon_{\max}^{(1)}$ совпадает с отношением V_n/L смежности группы V_n по подгруппе L , а отношение $\varepsilon_{\max}^{(3)}$ содержит это отношение смежности. Таким образом, справедливы включения

$$\varepsilon \subseteq \varepsilon_{\max}^{(1)} = V_n / L, \quad \varepsilon \supseteq \varepsilon_{\max}^{(3)} \supseteq V_n / L,$$

из которых следует равенство $\varepsilon = V_n / L$. Лемма доказана.

Итак, на основании леммы 1 группа подстановок $G(\mathfrak{Z})$ является импримитивной в том и только том случае, когда существует полная система блоков импримитивности этой группы, состоящая из попарно различных смежных классов группы V_n по ее собственной ненулевой подгруппе L .

Получим ряд следствий, вытекающих из этого утверждения.

1°. Для любых $x_2 \in V_m$, $k \in K$ справедлива импликация

$$(a_1, a_2) \in L \Rightarrow (a_2, a_1 \oplus \varphi_k(x_2 \oplus a_2) \oplus \varphi_k(x_2)) \in L. \quad (6)$$

Действительно, из условия $(a_1, a_2) \in L$ следует, что для любого $(x_1, x_2) \in V_n$ имеет место соотношение $(x_1, x_2) \equiv (x_1 \oplus a_1, x_2 \oplus a_2) \pmod{L}$, которое, в свою очередь, влечет соотношение $f_k(x_1, x_2) \equiv f_k(x_1 \oplus a_1, x_2 \oplus a_2) \pmod{L}$. Отсюда на основании равенства (1) непосредственно следует справедливость включения в правой части формулы (6).

2°. Для любого $a \in V_m$ условие $(a, 0) \in L$ влечет условие $(0, a) \in L$. Для доказательства достаточно положить в формуле (6) $a_1 = a$, $a_2 = 0$.

3°. Для любых $y_1 \in V_m$, $k \in K$ справедлива импликация

$$(a_1, a_2) \in L \Rightarrow (a_2 \oplus \varphi_k(y_1 \oplus a_1) \oplus \varphi_k(y_1), a_1) \in L. \quad (7)$$

Действительно, поскольку для любого $(y_1, y_2) \in V_n$

$$(y_1, y_2) \equiv (y_1 \oplus a_1, y_2 \oplus a_2) \pmod{L},$$

то

$$f_k^{-1}(y_1, y_2) \equiv f_k^{-1}(y_1 \oplus a_1, y_2 \oplus a_2) \pmod{L}.$$

Отсюда, принимая во внимание равенство

$$f_k^{-1}(y_1, y_2) = (y_2 \oplus \varphi_k(y_1), y_1), (y_1, y_2) \in V_n,$$

вытекающее из формулы (1), получим соотношение (7).

4⁰. Для любого $a \in V_m$ справедлива импликация $(0, a) \in L \Rightarrow (a, 0) \in L$ (достаточно положить в формуле (7) $a_1 = 0, a_2 = a$)

5⁰. Пусть H_1, H_2 – подгруппы группы V_m , определяются равенствами:

$$H_1 = \{a_1 \in V_m : (a_1, 0) \in L\}, H_2 = \{a_2 \in V_m : (0, a_2) \in L\},$$

соответственно. Тогда справедливы следующие утверждения:

(а) $H_1 = H_2$;

(б) для любых $h_1, h_2 \in H \stackrel{\text{def}}{=} H_1$ выполняется условие $(h_1, h_2) \in L$; в частности, $H \neq V_m$;

(в) для любых $x \in V_m, k \in K, h \in H$ выполняется условие

$$\varphi_k(x \oplus h) \oplus \varphi_k(x) \in H. \quad (8)$$

Действительно, равенство (а) следует непосредственно из утверждений 2⁰, 4⁰. Для доказательства (б) достаточно заметить, что по доказанному $(h_1, 0) \in L, (0, h_2) \in L$ и, следовательно, $(h_1, h_2) \in L$, так как L является подгруппой группы V_n .

Докажем равенство (8). Из условия $(0, h) \in L$ на основании утверждения 1⁰ вытекает соотношение $(h, \varphi_k(x \oplus h) \oplus \varphi_k(x)) \in L$. С другой стороны, справедливо соотношение $(h, 0) \in L$. Поскольку L – подгруппа группы V_n , то из полученных соотношений следует, что $(0, \varphi_k(x \oplus h) \oplus \varphi_k(x)) \in L$, то есть (в силу определения группы H) $\varphi_k(x \oplus h) \oplus \varphi_k(x) \in H$. что и требовалось доказать.

Используя утверждение 5⁰, нетрудно убедиться в справедливости следующей леммы.

Лемма 2. Пусть $G(\mathfrak{S})$ – импримитивная группа подстановок, L – собственная ненулевая подгруппа группы V_n такая, что смежные классы V_n по L образуют полную систему блоков импримитивности группы $G(\mathfrak{S})$. Положим $H = \{a \in V_m : (0, a) \in L\}$. Тогда при выполнении условия $H \neq 0$ смежные классы группы V_m по подгруппе H образуют нетривиальную полную систему блоков импримитивности группы подстановок $G_\varphi = \langle \varphi_k : k \in K \rangle$.

Обратно, если существует нетривиальная полная система блоков импримитивности группы G_φ , состоящая из смежных классов группы V_m по некоторой ее подгруппе H , то различные смежные классы группы V_n по подгруппе $L = H \oplus H \stackrel{\text{def}}{=}$ образуют нетривиальную полную систему блоков импримитивности группы $G(\mathfrak{S})$. В частности, $G(\mathfrak{S})$ является импримитивной группой подстановок.

Доказательство. Первое утверждение леммы следует непосредственно из 5⁰. Для доказательства второго утверждения заметим, что, поскольку $H \neq 0, H \neq V_m$, то $L \neq 0, L \neq V_n$. Далее, по условию леммы для любых $(a_1, a_2) \in L, (x_1, x_2) \in V_n, k \in K$ справедливы соотношения:

$$a_1, a_2 \in H, (x_1, x_2) \equiv (x_1 \oplus a_1, x_2 \oplus a_2) \pmod{L}, \varphi_k(x_2 \oplus a_2) \oplus \varphi_k(x_2) \in H,$$

из которых следует, что

$$f_k(x_1 \oplus a_1, x_2 \oplus a_2) \oplus f_k(x_1, x_2) = (a_2, a_1 \oplus \varphi_k(x_2 \oplus a_2) \oplus \varphi_k(x_2)) \in L.$$

Итак, элементы фактор-группы V_n/L образуют нетривиальную полную систему блоков импримитивности группы $G(\mathfrak{Z})$, что и требовалось доказать.

Доказательство теоремы 2. Пусть $G(\mathfrak{Z})$ – импримитивная группа подстановок. Покажем, что в этом случае выполняется одно из условий (а), (б), указанных в формулировке теоремы.

На основании леммы 1 существует подгруппа L группы V_n такая, что смежные классы V_n по L образуют нетривиальную полную систему блоков импримитивности группы $G(\mathfrak{Z})$. Пусть H – подгруппа группы V_m , заданная в условии первого утверждения леммы 2. Тогда $H \neq V_m$ (утверждение 5⁰), и, если $H \neq 0$, то, согласно лемме 2, выполняется условие (а) утверждения теоремы.

Предположим, что $H = 0$. Тогда на основании определения группы H для любых $a_1, a_2, b_1, b_1 \in V_m$ справедливы следующие импликации:

$$((a_1, a_2), (b_1, b_2)) \in L, a_1 = b_1 \Rightarrow (a_2 = b_2), \quad (9)$$

$$((a_1, a_2), (b_1, b_2)) \in L, a_2 = b_2 \Rightarrow (a_1 = b_1). \quad (10)$$

Рассмотрим множества:

$$M_1 = \{a_1 \in V_m \mid \exists a_2 \in V_m : (a_1, a_2) \in L\}, \quad M_2 = \{a_2 \in V_m \mid \exists a_1 \in V_m : (a_1, a_2) \in L\}.$$

В силу утверждений 1⁰, 3⁰ имеет место равенство $M_1 = M_2$. Положим $M = M_1$. Очевидно, что M – подгруппа группы V_m . Кроме того, из соотношений (9), (10) следует, что существует биективное отображение $g : M \rightarrow M$, удовлетворяющее условию

$$L = \{(a, g(a)) : a \in M\}. \quad (11)$$

Так как L – группа, то в силу равенства (11) g – эндоморфизм группы M .

Таким образом, окончательно получаем, что в случае $H = 0$ подгруппа L группы V_n имеет вид (11), где $g : M \rightarrow M$ является автоморфизмом группы M . Далее, применяя импликацию (7) к произвольному вектору $(a, g(a)) \in L$, получим, что для любых $y_1 \in V_m, k \in K$

$$(g(a) \oplus \varphi_k(y_1 \oplus a) \oplus \varphi_k(y_1), a) \in L$$

и, следовательно, согласно (11)

$$\varphi_k(y_1 \oplus a) \oplus \varphi_k(y_1) = g(a) \oplus g^{-1}(a). \quad (12)$$

Итак, для любых $a \in M, y_1 \in V_m, k \in K$ имеет место равенство (12).

Заметим теперь, в силу условия $L \neq 0, M$ является ненулевой подгруппой группы V_m . Если при этом $M \neq V_m$, то смежные классы группы V_m по подгруппе M образуют нетривиальную полную систему блоков импримитивности группы G_φ . Действительно, согласно равенству (12), для любых $a \in M, y_1 \in V_m, k \in K$ выполняется условие

$$\varphi_k(y_1 \oplus a) \oplus \varphi_k(y_1) \in M,$$

означающее, что образы векторов $y_1 \oplus a$ и y_1 при отображении φ_k принадлежат одному смежному классу группы V_m по подгруппе M . Если же $M = V_m$, то, полагая в формуле (12) $a = y_1$, получим, что для любого $k \in K$ отображение φ_k имеет вид (5).

Итак, если $G(\mathfrak{Z})$ – импримитивная группа и $H = 0$, то при $M \neq V_m$ выполняется условие (а) утверждения теоремы 2; в случае $M = V_m$ справедливо условие (б). Таким образом, первая часть теоремы полностью доказана.

Докажем вторую часть теоремы. На основании леммы 2 достаточно показать, что при выполнении условия (б) группа $G(\mathfrak{Z})$ является импримитивной на множестве V_n .

Предположим, что для любого $k \in K$ отображение φ_k имеет вид (5), где g – невырожденное линейное преобразование векторного пространства V_m .

Положим $L = \{(a, g(a)) : a \in V_m\}$ и заметим, что смежные классы группы V_n по ее (собственной ненулевой) подгруппе L образуют полную систему блоков импримитивности группы $G(\mathfrak{Z})$. Действительно, пусть $(x_1, x_2) \in V_n$; тогда на основании равенств (1), (5) для любых $a \in V_m, k \in K$ выполняются соотношения:

$$\begin{aligned} f_k(x_1 \oplus a, x_2 \oplus g(a)) \oplus f_k(x_1, x_2) &= (g(a), a \oplus \varphi_k(x_2 \oplus g(a)) \oplus \varphi_k(x_2)) = \\ &= (g(a), a \oplus g(x_2 \oplus g(a)) \oplus g^{-1}(x_2 \oplus g(a)) \oplus g(x_2) \oplus g^{-1}(x_2)) = \\ &= (g(a), a \oplus g(x_2) \oplus g^2(a) \oplus g^{-1}(x_2) \oplus a \oplus g(x_2) \oplus g^{-1}(x_2)) = \\ &= (g(a), g^2(a)) \in L. \end{aligned}$$

Итак, $G(\mathfrak{Z})$ является импримитивной группой подстановок.

Теорема полностью доказана.

3 Примитивность группы подстановок, порожденной раундовыми преобразованиями блочного шифра GOST

Покажем, используя теорему 2, что группа, порожденная раундовыми шифрующими преобразованиями шифра GOST, является примитивной, какими бы ни были s -блоки этого шифра.

Напомним [11], что GOST является шифром Фейстеля $\mathfrak{Z} = (V_{64}, \Lambda, F)$, раундовые преобразования $f_k: V_{64} \rightarrow V_{64}$ ($k \in K = V_{32}$) которого определяются по формуле (1), где отображение $\varphi: V_{32} \times V_{32} \rightarrow V_{32}$ имеет вид

$$\varphi(x, k) = \psi(x + k), \quad x, k \in V_{32}, \quad (13)$$

символ «+» обозначает операцию сложения по модулю 2^{32} двоичных целых чисел, соответствующих 32-мерным булевым векторам, а подстановка ψ определяется по формуле

$$\psi(z) = \rho_{32}^{(11)}(S^{(7)}(z^{(7)}), \dots, S^{(0)}(z^{(0)})), \quad (14)$$

где $\rho_{32}^{(11)}$ есть операция циклического сдвига “влево на 11 разрядов” на множестве V_{32} , $S^{(7)}, \dots, S^{(0)}$ – подстановки на множестве V_4 (s -блоки шифра GOST), и для любого $z = (z_{31}, \dots, z_1, z_0) \in V_{32}$

$$z^{(i)} \stackrel{\text{def}}{=} (z_{4i+3}, z_{4i+2}, z_{4i+1}, z_{4i}) \in V_4, \quad i \in \overline{0, 7}. \quad (15)$$

Из равенства (13) следует, что φ является квазигрупповой операцией на множестве V_{32} , и группа $G_\varphi = \langle \varphi_k : k \in K \rangle$, порожденная всевозможными подстановками $\varphi_k(u) = \psi(u + k)$, $u \in V_{32}$, совпадает с группой $\langle c, \psi \rangle$, порожденной подстановкой $c(x) = (x + 1) \bmod 2^{32}$, $x \in V_{32}$ и подстановкой ψ .

Лемма 3. Пусть $S^{(7)}, \dots, S^{(0)}$ – произвольные подстановки на множестве V_4 , и подстановка ψ определяется по формуле (14). Тогда выполняется равенство

$$G_\varphi = \langle c, \psi \rangle = S^{V_{32}}. \quad (16)$$

Доказательство. Покажем, что группа $\langle c, \psi \rangle$ является примитивной. В этом случае справедливость равенства (16) следует непосредственно из теоремы 1, полученной в [13].

Отождествим произвольное двоичное целое число $x = 2^{31}x_{31} + \dots + 2^0x_0$ с булевым вектором $(x_{31}, \dots, x_0) \in V_{32}$. Обозначим через c_i и ψ_i i -е ($i \in \overline{0, 31}$) координатные функции

подстановок c и ψ , соответственно, определяемые с помощью равенств $c(x) = (c_{31}(x), \dots, c_0(x))$, $\psi(x) = (\psi_{31}(x), \dots, \psi_0(x))$, $x \in V_{32}$.

Рассмотрим оргграф $S(c, \psi)$ с множеством вершин $\{0, 1, \dots, 31\}$, в котором из вершины i в вершину j направлена дуга в том и только том случае, когда хотя бы одна из булевых функций c_j, ψ_j существенно зависит от переменной x_i , $i, j \in \overline{0, 31}$. Известно [14], что группа $\langle c, \psi \rangle$ является примитивной тогда и только тогда, когда оргграф $S(c, \psi)$ – сильно связный. Очевидно также, что в $S(c, \psi)$ существует ориентированный путь, последовательно соединяющий вершины $0, 1, \dots, 31$. Отсюда вытекает, что для доказательства примитивности группы $\langle c, \psi \rangle$ достаточно установить существование целых чисел $j(1), j(2)$, удовлетворяющих следующим условиям:

- (1) $0 < j(1) \leq j(2) < 31$;
- (2) функция $\psi_{j(1)}$ существенно зависит от переменной x_{31} ;
- (3) функция ψ_0 существенно зависит от переменной $x_{j(2)}$.

Укажем такую пару чисел $j(1), j(2)$. Обозначим через $S_{31}, \dots, S_0$ координатные функции подстановки $S(z) = (S^{(7)}(z^{(7)}), \dots, S^{(0)}(z^{(0)}))$, $z \in V_{32}$, где булевы векторы $z^{(i)}$ ($i \in \overline{0, 7}$) определяются по формуле (15).

Заметим, что, поскольку отображения $S^{(7)}$ и $S^{(5)}$ являются подстановками на множестве V_4 , то существуют числа $j(1) \in \{7, 8, 9, 10\}$ и $j(2) \in \{20, 21, 22, 23\}$ такие, что булева функция $S_{j(1)-11}$ существенно зависит от переменной x_{31} , а булева функция S_{21} – от переменной $x_{j(2)}$ (арифметические операции над нижними индексами символа S выполняются по модулю 32). Отсюда на основании равенства (14) вытекает, что функции $\psi_0 = S_{21}$ и $\psi_{j(1)} = S_{j(1)-11}$ зависят существенно от переменных $x_{j(2)}$ и x_{31} , соответственно.

Итак, указанные числа $j(1)$ и $j(2)$ удовлетворяют сформулированным выше условиям (1) – (3), и, следовательно, $\langle c, \psi \rangle$ является примитивной группой подстановок. Лемма доказана.

Непосредственно из утверждений теоремы 2 и леммы 3 вытекает следующая теорема.

Теорема 3. Для любых s -блоков $S^{(7)}, \dots, S^{(0)}$ шифра GOST группа, порожденная раундовыми шифрующими преобразованиями этого шифра, является примитивной на множестве V_{64} .

Доказательство. Достаточно заметить, что, поскольку $G_{\varphi} = S^{132}$ является 32-транзитивной (и заведомо примитивной) группой подстановок, то для нее не выполняются оба условия (а), (б) первого утверждения теоремы 2. (При выполнении условия (б) подстановки φ_k являются аффинными преобразованиями векторного пространства V_{32} и порождают, самое большее, полную аффинную группу $AGL(32, 2)$, которая не является 32-транзитивной [12]).

Теорема доказана.

Полученная теорема свидетельствует о стойкости шифра GOST к алгебраическим атакам, описанным в [4–6], а также об отсутствии в конструкции этого шифра скрытых “лазеек”, основанных на импримитивности группы, порожденной его раундовыми шифрующими преобразованиями [4].

Предполагается, что указанная группа совпадает со знакопеременной группой подстановок на множестве V_{64} , однако в настоящее время авторы статьи не располагают доказательством этого утверждения или контрпримером к нему.

Список литературы: 1. Горчинский Ю.Н. О гомоморфизмах многоосновных универсальных алгебр в связи с криптографическими приложениями // Труды по дискретной математике. М.: ТВП, 1997. Т.1. С.67–84. 2. Шапошников И.Г. О конгруэнциях конечных многоосновных универсальных алгебр // Дискретная математика. 1999. Т.11. В.3. С.48–62. 3. Бабаи А.В., Шанкин Г.П. Криптография. М.: Солон-Р, 2002. 511с. 4. Paterson K G. Imprimitve permutation groups and trapdoors in iterated block ciphers // FSE'99, Proceedings. Springer Verlag, 1999. P. 201–214. 5. Campbell K W., Wiener M. DES is not a group // Advances in Cryptology – CRYPTO'92, Proceedings. Springer Verlag, 1993. P. 512–520. 6. Kaliski B.S., Rivest R.L., Sherman A.T. Is the Data Encryption Standard a group? (Results of cycling experiments on DES) // Journal of Cryptology. 1988. № 1. P. 3–36. 7. Even S., Goldreich O. DES-like func-

tions can generate the alternating group // *IEEE Trans. on Information Theory*. 1983. V. 29. P. 863-865.

8. *Pieprzyk J., Zhang X.-M.* Permutation generators of alternating groups // *AUSCRYPT'90, Proceedings*. Springer Verlag, 1990. P. 237-244.

9. *Wernsdorf R.* The one-round functions of the DES generate the alternating group // *Advances in Cryptology – EUROCRYPT'92, Proceedings*. Springer Verlag, 1993. P. 99-112.

10. *Алексейчук А.Н.* Критерий примитивности группы подстановок, порожденной раундовыми преобразованиями Rijndael-подобного блочного шифра // *Ресстрація, зберігання і обробка даних*. 2004 (в печати).

11. *Харин Ю.С., Берник В.И., Матвеев Г.В.* Математические основы криптологии. Минск: Изд-во БГУ. 1999. 319 с.

12. *Глухов М.М., Елизаров В.П., Нечаев А.А.* Алгебра: Учебник. В 2-х т., М.: Гелиос АРВ. 2003. Т. II. 416 с.

13. *Погорелов Б.А.* Примитивные группы подстановок, содержащие 2^m -цикл // *Алгебра и логика*. 1980. Вып. 19. № 2. С. 236-247.

14. *Глухов М.М., Зубов А.Ю.* О длинах симметрических и знакопеременных групп подстановок в различных системах образующих (обзор) // *Математические вопросы кибернетики*. 1999. Вып. 8. № 3. С. 5-32.

*Киевский
политехнический университет*

Поступила в редколлегию 12.11.2004

МАТЕМАТИЧНА МОДЕЛЬ ПЕРЕКРИТТЯ ШИФРУ ДЛЯ ПОТОКОВИХ КРИПТОСИСТЕМ

Одним з основних показників ефективності поточкового шифру є так звана ймовірність перекриття шифру [1–3]. Її значення суттєво впливає на якість надання такої важливої послуги, як конфіденційність. Як правило, допустима величина ймовірності перекриття шифру задається в тактико-технічному завданні або технічному завданні на розробку симетричного шифру. В [1] показано, що при перекритті шифру складність розв'язку задачі успішного криптоаналізу носить поліноміальний характер і не представляє для криптоаналітика жодних труднощів [1–3]. Аналіз підсумків виконання Європейського проекту *Nessie* зі створення стандартних криптографічних примітивів, в тому числі поточкових шифрів, підтверджує відсутність адекватних математичних моделей оцінки перекриття шифру.

Аналіз показує [1–2], що в поточкових режимах шифрування гама Γ_i є функцією

$$\Gamma_i = \phi(K_j, СП_i, P_r) \quad (1)$$

ключа K_j , синхропослідовності $СП_i$ та, можливо, параметрів P_r .

При цьому співвідношення (1) справедливе тільки на строк дії K_j ключа, оскільки різні $СП_i$ формують один і той же, по суті, ізоморфізм гами (різні відрізки послідовності періоду L) [2]. Крім того, ідентичні гами можуть виникати за рахунок використання в мережі однакових $СП_i$, або випадково. В першому випадку, тобто коли $СП_i = СП_j$, і використовується один і той же ключ, будуть сформовані ідентичні гами, причому незалежно від довжини гами. В другому випадку ймовірність перекриття, по суті, є ймовірністю відбуття колізії. Докладно це питання розглянуте в [3–5].

Метою цієї статті є розроблення та аналіз математичної моделі перекриття поточкового шифру, яка, по-перше, враховувала б взаємозв'язок таких показників та параметрів, як ймовірність перекриття шифру $P_{\text{пш}}$, період повторення гами шифруючої L , число одночасно функціонуючих станцій n та число перезапусків r , коли зашифровані повідомлення M мають довільну довжину l_M .

1 Методи оцінки ймовірності перекриття гам шифруючих

Нехай гами Γ_i формуються згідно виразу (1) та мають довжини l_i . Приймемо, що ключ K_j не змінюється в інтервалі Δt , але для кожної о повідомлення або різних сеансів використовуються різні синхропослідовності з однаковою довжиною l_c . Покладемо, що в режимі лічильника період гами буде L . Оскільки згідно (1) при формуванні гами змінюються тільки синхропослідовності, то Γ_i буде періодичною із періодом L . Представимо таку гаму у замкнутому вигляді (1) для загального випадку з періодом L [2].

Будемо вважати, що дві гами Γ_i та Γ_j перекриваються, якщо $\Gamma_i = \Gamma_j$ не менше, ніж в Σ символах. Зашифровані з їх допомогою повідомлення M_i та M_j будуть перекриватися у вигляді криптограм C_i та C_j . Нехай передаються r криптограм $C_1, C_2, \dots, C_r$ з довжинами $l_1, l_2, \dots, l_r$. Будемо вважати, що перекриття відсутнє, якщо усі пари криптограм не мають перекриття, тобто при їх створенні не використовуються гами, що співпадають не менше, ніж в Σ бітах. Перекриттям в цьому випадку є використання для шифрування не менше Σ одних і тих же символів періодичної послідовності (рис. 1).

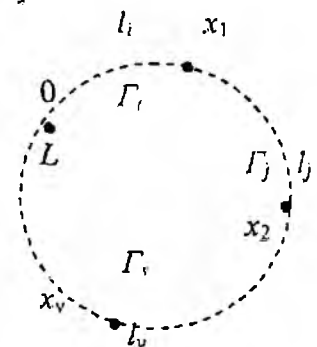


Рис. 1

Сформуємо в дещо іншому вигляді, ніж в [2], теорему про перекриття гам шифруючих.

Теорема 1 [2]. Нехай криптограми $C_1, C_2, \dots, C_r$ є результат потокового зашифрування повідомлень $M_1, M_2, \dots, M_r$ з використанням гам шифруючих $\Gamma_1, \Gamma_2, \dots, \Gamma_r$ з довжинами $l_1, l_2, \dots, l_r$, причому різні гами отримуються за рахунок використання різних синхропослідовностей:

$$CP_1, CP_2, \dots, CP_r, \text{ і } \sum_{i=1}^r l_i = Z, \quad (2)$$

а період гами в режимі лічильника дорівнює L , тоді ймовірність перекриття шифру

$$P_{\text{пш}} = 1 - \frac{(r-1)! \cdot C_{L-Z+r-1}^{r-1}}{L^{r-1}}. \quad (3)$$

Доведення проведемо, враховуючи, що ймовірність перекриття шифру $P_{\text{пш}}$ та ймовірність відсутності перекриття $P_{\text{вп}}$ складають повну групу подій:

$$P_{\text{пш}} + P_{\text{вп}} = 1. \quad (4)$$

Для доведення теореми визначимо ймовірність відсутності перекриття шифру $P_{\text{вп}}$, спираючись на рис. 1.

Ймовірність відсутності перекриття $P_{\text{вп}}$ визначимо як

$$P_{\text{вп}} = \frac{n_{\text{вп}}}{n_{\Sigma}}, \quad (5)$$

де $n_{\text{вп}}$ – число розміщень r повідомлень (гам) на періоді L без перекриття;

n_{Σ} – загальне число гам, що можуть бути сформовані (вибрані) для r спроб (повідомлень).

При кожному звертанні до алгоритму формування гами існує (рис. 1) L варіантів вибору. Тому що ці звертання є випадкові (повинні бути), то на r звертаннях

$$n_{\Sigma} = L^r. \quad (6)$$

Визначимо $n_{\text{вп}}$ за r звертань. При першому звертанні існує $n_1 = L$ варіантів вибору гами. При другому і наступних вибирається $r-1, r-2, \dots$ і т.д. варіантів вибору відрізків. Всього можна вибрати

$$n_2 = (r-1)! \quad (7)$$

гам в різній послідовності (варіантів).

Далі, оскільки відрізки не перекриваються, то відстані між ними $x_1, x_2, \dots, x_r \geq 0$. Всього згідно рис. 1 таких відстаней буде r .

Справедливим є таке співвідношення:

$$\sum_{i=1}^r x_i + \sum_{i=1}^r l_i = L. \quad (8)$$

Із виразу (8) виходить, що число вільних позицій на періоді визначається як

$$\sum_{i=1}^r x_i = L - \sum_{i=1}^r l_i = L - Z. \quad (9)$$

Визначимо число різних розстановок $r-1$ перегородок між $L-Z$ предметами (число способів, за якими можна розмістити $L-Z$ предметів по r ящиках):

$$n_3 = C_{L-Z+r-1}^{L-Z} = C_{L-Z+r-1}^{r-1}, \quad (10)$$

тобто воно дорівнює числу сполучень із r об'єктів по $L-Z$ з повторенням.

Дійсно, це виходить із теореми 8.69 [6], яка стверджує, що число різних сполучень із k об'єктів по n дорівнює:

$$C_{n+k-1}^k = C_{n+k-1}^{k-1} = \frac{(n+k-1)!}{n!(k-1)!}.$$

У випадку, що розглядається:

$$n = L - Z, \quad k = r, \text{ тому } C_{L-Z+r-1}^{r-1} = \frac{(L - Z + r - 1)!}{(L - Z)!(r - 1)!}.$$

Підставивши (6), (7) та (10) в (5), маємо

$$\begin{aligned} P_{\text{вп}} &= \frac{n_1 \cdot n_2 \cdot n_3}{n_z} = \frac{L \cdot (r - 1)! C_{L-Z+r-1}^{r-1}}{L^r} = \frac{(r - 1)! C_{L-Z+r-1}^{r-1}}{L^{r-1}} = \\ &= \frac{(r - 1)!(L - Z + r - 1)!}{L^{r-1}(L - Z)!(r - 1)!} = \frac{(L - Z + r - 1)!}{L^{r-1}(L - Z)!} \end{aligned} \quad (11)$$

Враховуючи (4), маємо

$$P_{\text{пш}} = 1 - P_{\text{вп}} = 1 - \frac{(r - 1)! C_{L-Z+r-1}^{r-1}}{L^{r-1}} = 1 - \frac{(L - Z + r - 1)!}{L^{r-1}(L - Z)!}. \quad (12)$$

Справедливість (10) і відповідно (12) виходить із (9). В [7] показано, що число розв'язків рівняння (9) визначається як число різних сполучень із $r = k$ об'єктів по $L - Z = n$ об'єктів.

Таким чином, (12) може бути використане для обчислення ймовірності перекриття шифру. Вхідними даними при обчисленні є:

- r – число гам шифрування, сформованих з різними синхропослідовностями;
- L – період повторення гами при зафіксованому ключі;
- Z – число біт гами, що містяться в r різних гамах $\Gamma_1 \div \Gamma_r$.

Важливими для перекриття є випадки, коли повідомлення (гами) мають однакові довжини, а також випадок, коли $r = 2$, тобто визначається ймовірність перекриття для двох повідомлень.

Із співвідношення (3) при $r = 2$ маємо

$$P_{\text{пш}} = 1 - \frac{L - Z + 1}{L} = \frac{Z - 1}{L}.$$

Для повідомлень (гам) однакової довжини l із виразу (11) маємо

$$P_{\text{пш}} = 1 - \frac{(r - 1)! C_{L-r(l-1)-1}^{r-1}}{L^{r-1}}. \quad (13)$$

При урахуванні (11) для випадку (13) маємо

$$P_{\text{пш}} = 1 - \frac{(r - 1)!(L - Z + r - 1)!}{L^r(L - Z)!(r - 1)!} = 1 - \frac{(L - r(l - 1) - 1)!}{L^r(L - rl)!}. \quad (14)$$

Формули (12) та (14) дозволяють точно обчислити значення ймовірностей перекриття шифру.

Проведений аналіз показав, що прямі обчислення ймовірностей перекриття шифру за формулами (12) та (14) вже при $L \geq 2^{64}$ є дуже складними, оскільки необхідно обчислювати біноміальні коефіцієнти, розраховуючи факторіали при великих значеннях аргументів.

Значення факторіалів можна обчислювати, використовуючи асимптотичну формулу Стерлінга [8]:

$$n! = \sqrt{2\pi n} n^n e^{-n}. \quad (15)$$

Формула (15) дає досить малі відносні похибки, наприклад, при $n = 100$ похибка складас не більше 0,08 %. При довільному значенні n точне значення $n!$ відрізняється від асимптотичного, обчисленого за допомогою (15), тільки множителем $e^{\theta(n)}$, причому $0 < \epsilon_n < \frac{1}{12n}$.

З урахуванням цієї складової замість виразу (15) можна використовувати формулу

$$n! = \sqrt{2\pi n} n^n e^{-n} e^{\theta(n)}. \quad (16)$$

Якщо застосувати формулу (15), то одержимо:

$$(L-Z+r-1)! = \sqrt{2\pi} (L-Z+r-1)^{(L-Z+r-1/2)} \cdot \exp(-(L-Z+r-1)), \quad (17)$$

$$(L-Z)! = \sqrt{2\pi} (L-Z)^{(L-Z+1/2)} \cdot \exp(-(L-Z)). \quad (18)$$

Далі, підставивши вирази (17) та (18) в (12), одержимо, що

$$\begin{aligned} P_{пш} &= 1 - \frac{(L-Z+r-1)^{(L-Z+r-1/2)} \cdot \exp(-(L-Z+r-1))}{L^{r-1} (L-Z)^{(L-Z+1/2)} \cdot \exp(-(L-Z))} = \\ &= 1 - \frac{(L-Z+r-1)^{(L-Z+r-1/2)}}{(L-Z)^{(L-Z+1/2)} L^{r-1}} \cdot \exp(-r+1) \end{aligned} \quad (19)$$

Для спрощення (19) зробимо такі математичні перетворення. Представимо чисельник у такому вигляді:

$$\begin{aligned} &(L-Z+r-1)^{(L-Z+r-1/2)} \cdot \frac{(L-Z)^{(L-Z+r-1/2)}}{(L-Z)^{(L-Z+r-1/2)}} = \\ &= \left(1 + \frac{r-1}{L-Z}\right)^{(L-Z+r-1/2)} \cdot (L-Z)^{(L-Z+r-1/2)}. \end{aligned} \quad (20)$$

Підставивши (20) в (19), одержимо

$$\begin{aligned} P_{пш} &= 1 - \frac{(L-Z)^{(L-Z+r-1/2)}}{(L-Z)^{(L-Z+1/2)} L^{r-1}} \left(1 + \frac{r-1}{L-Z}\right)^{(L-Z+r-1/2)} \exp(-r+1) = \\ &= 1 - \left(\frac{L-Z}{L}\right)^{(r-1)} \left(1 + \frac{r-1}{L-Z}\right)^{(L-Z+r-1/2)} \cdot \exp(-r+1). \end{aligned} \quad (21)$$

Очевидно, формула (21) є найменш обчислювально складною, адже вона не містить обчислення факторіалів. Але треба зважати на те, що вона дозволяє одержати тільки наближені результати, оскільки ми використовуємо формулу Стерлінга. Похибка в цьому випадку розраховується з використанням (16), причому її значення залежить тільки від величини n .

Іншою умовою, справедливою в практичних ситуаціях, є те, що $L \gg Z$, в найгіршому випадку $L \geq 10 \cdot Z$, тому $r \ll L-Z$. В пілому для співвідношення (21) можна одержати таке наближення:

$$\begin{aligned} P_{пш} &= 1 - \exp(-r+1) \left(\frac{L-Z}{L}\right)^{(r-1)} \left(1 + \frac{r-1}{L-Z}\right)^{(L-Z+r-1/2)} = \\ &= 1 - \exp(-r+1) \left(1 - \frac{Z}{L}\right)^{(r-1)} \left(1 + \frac{r-1}{L-Z}\right)^{L-Z} \left(1 + \frac{r-1}{L-Z}\right)^{r-1/2}. \end{aligned} \quad (22)$$

При умові, що $Z \leq 0,1L$, можна скористатись апроксимацією $(1-x) \approx e^{-x}$. Тому при $x = Z/L$

$$\left(1 - \frac{Z}{L}\right)^{r-1} = \left(\exp\left(-\frac{Z}{L}\right)\right)^{r-1} = \exp\left\{-\frac{(r-1)Z}{L}\right\}. \quad (23)$$

Зведемо до експоненти також і наступні співмножники (22). Так,

$$\left(1 + \frac{r-1}{L-Z}\right)^{L-Z} = \left(1 - \frac{1-r}{L-Z}\right)^{L-Z} = \exp\{-(1-r)\} = \exp(r-1), \quad (24)$$

$$\left(1 + \frac{r-1}{L-Z}\right)^{r-1/2} = \exp\left\{\frac{r-1}{L-Z}(r-0,5)\right\}. \quad (25)$$

Підставляючи (23), (24) та (25) у (22), одержимо

$$\begin{aligned} P_{\text{пш}} &= 1 - \exp(-r+1) \exp\left\{-\frac{(r-1)Z}{L}\right\} \exp(r-1) \exp\left\{\frac{r-1}{L-Z}(r-0,5)\right\} = \\ &= 1 - \exp\left\{(r-1)\left(\frac{r-0,5}{L-Z} - \frac{Z}{L}\right)\right\} \end{aligned} \quad (26)$$

Таким чином, в подальшому основними робочими формулами для обчислення ймовірностей перекриття будуть (12), (13) та (26). При цьому (12) є точною в межах прийнятої моделі. Вираз (21) дає наближений результат, а похибка визначається з використанням співвідношення (16). Вираз (26) є найменш складним, але у ньому є похибки, пов'язані з апроксимацією за формулою Стерлінга (15), а також похибки, пов'язані з апроксимаціями (23), (24) та (25).

2 Параметризація задачі оцінки ймовірності перекриття шифру

При розрахунках необхідно задати число перезапусків r , що можуть бути виконані, обсяг виробленої гами Z (зашифрованої інформації) та значення мінімально допустимої довжини періоду L . Задаючи ці параметри, можна обчислити ймовірності перекриття, що досягаються при відповідних значеннях параметрів r , Z та L . По суті, це параметрична задача, розв'язок якої наводиться нижче.

Формула (26) для випадку, коли повідомлення (пакети) мають однакову довжину, має вигляд

$$P_{\text{пш}} = 1 - \exp\left\{(r-1)\left(\frac{r-0,5}{L-r \cdot l_n} - \frac{r \cdot l_n}{L}\right)\right\}, \quad (27)$$

де l_n – довжина пакета.

Для визначення r необхідно розв'язати рівняння (27) відносно r . Запишемо (27) у вигляді

$$1 - P_{\text{пш}} = \exp\left\{(r-1)\left(\frac{r-0,5}{L-r \cdot l_n} - \frac{r \cdot l_n}{L}\right)\right\}$$

та прологарифмуємо одержане рівняння. Тоді:

$$\ln(1 - P_{\text{пш}}) = (r-1)\left(\frac{r-0,5}{L-r \cdot l_n} - \frac{r \cdot l_n}{L}\right). \quad (28)$$

Розв'язавши рівняння (28) відносно r , можна обчислити допустиме число перезапусків при шифруванні пакетів. Дійсно,

$$\begin{aligned} (r-1)\left(\frac{r-0,5}{L-r l} - \frac{r l}{L}\right) &= (r-1)\left(\frac{L(r-0,5) - r l L + r^2 l^2}{L(L-r l)}\right) = \\ &= \frac{l^2 r^3 + (l^2 + L - L l) r^2 + (L l - L - 0,5 L) r + 0,5 L}{L^2 - L r l} = \ln(1 - P_{\text{пш}}) \end{aligned} \quad (29)$$

Далі, зробивши ряд перетворень, одержуємо:

$$l^2 r^3 + (l^2 + L - L l) r^2 + (L l - 1,5 L) r + 0,5 L = -\ln(1 - P_{\text{пш}}) L l r + l^2 \ln(1 - P_{\text{пш}})$$

чи

$$l^2 r^3 + (l^2 + L(1-l)) r^2 + (L l - 1,5 L + L l \ln(1 - P_{\text{пш}})) r - L^2 \ln(1 - P_{\text{пш}}) + 0,5 L = 0$$

Введемо позначення:

$$a = l^2; \quad (30)$$

$$b = l^2 + L(1 - l); \quad (31)$$

$$c = Ll - 1,5L + Ll \ln(1 - P_{пш}) = L(l - 1,5 + l \ln(1 - P_{пш})) = L(l(1 + \ln(1 - P_{пш})) - 1,5); \quad (32)$$

$$d = L(L \ln(1 - P_{пш}) + 0,5). \quad (33)$$

Параметри перекриття можна визначити на основі розв'язку кубічного рівняння:

$$ar^3 + br^2 + cr + d = 0. \quad (34)$$

Далі, за аналогією з (28), можна одержати рівняння і для загального випадку (27):

$$\ln(1 - P_{пш}) = \left\{ (r-1) \left(\frac{r-0,5}{L-Z} - \frac{Z}{L} \right) \right\}.$$

Після розкриття правої частини маємо

$$(r-1) \left(\frac{r-0,5}{L-Z} - \frac{Z}{L} \right) = (r-1) \left(\frac{(r-1)L + 0,5L - LZ - Z^2}{(L-Z)L} \right).$$

Якщо позначити

$$y_1 = 0,5L - LZ - Z^2, \quad (35)$$

$$y_2 = (L-Z)L, \quad (36)$$

$$x = (r-1), \quad (37)$$

то одержимо:

$$\ln(1 - P_{пш}) = x \left(\frac{xL + y_1}{y_2} \right)$$

$$\text{або} \quad (38)$$

$$Lx^2 + y_1x - y_2 \ln(1 - P_{пш}) = 0.$$

Необхідно підкреслити, що рівняння (38) по суті містить дві невідомі величини $x = r - 1$ та Z , яке також залежить від r . Тому його можна розглядати як параметричне рівняння. Основним же для визначення числа допустимих перезапусків є рівняння вигляду (34).

3 Оцінка ймовірності перекриття на основі узагальненого „парадоксу” про день народження

Розглянемо ще один випадок, коли при відомих значеннях $P_{пш}$, Z та L необхідно визначити допустиму величину перезапусків. Перш за все, скоротимо у формулі (12) на $(L-Z)!$. Тоді маємо

$$\begin{aligned} P_{пш} &= 1 - \frac{(L-Z+r-1)(L-Z+r-2)\dots(L-Z+1)}{L^{r-1}} = \\ &= 1 - \left(\frac{L-Z+r-1}{L} \right) \left(\frac{L-Z+r-2}{L} \right) \dots \left(\frac{L-Z+1}{L} \right) = \\ &= 1 - \left(1 - \frac{Z-r+1}{L} \right) \left(1 - \frac{Z-r+2}{L} \right) \dots \left(1 - \frac{Z+1}{L} \right). \end{aligned} \quad (39)$$

Розглянемо випадок, коли $Z-r$ та $r < 0,1L$. Ця умова на практиці завжди виконується. Тоді можна скористатись формулою $(1-x) \approx e^{-x}$ і вираз (39) представити у вигляді

$$P_{пш} = 1 - e^{-\frac{Z-r+1}{L}} \cdot e^{-\frac{Z-r+2}{L}} \dots e^{-\frac{Z+1}{L}} = 1 - e^{-\frac{Z-r+1}{L} - \frac{Z-r+2}{L} - \dots - \frac{Z+1}{L}} \quad (40)$$

Оскільки члени суми ступеня виразу (40) являють собою арифметичну прогресію зі знаменником прогресії $q = \frac{1}{L}$ та числом членів $n = Z + 1 - (Z - r + 1) = r$, то

$$\frac{Z-r+1}{L} + \frac{Z-r+2}{L} + \dots + \frac{Z+1}{L} = \frac{(2Z-r+2)r}{2L} = \frac{(Z+1+0,5r)r}{L} \quad (41)$$

Підставивши (41) в (42), одержимо:

$$\begin{aligned} P_{пш} &= 1 - e^{-\frac{(Z+1+0,5r)r}{L}} = 1 - e^{-\frac{0,5r^2 - (Z+1)r}{L}} = \\ &= 1 - \exp\left\{\frac{(0,5r^2 - (Z+1)r)}{L}\right\}. \end{aligned} \quad (42)$$

Для випадку, коли $Z = lr$, наприклад при шифруванні пакетів, формула (42) приймає вигляд

$$P_{пш} = 1 - \exp\left\{\frac{(0,5r^2 - (lr+1)r)}{L}\right\}. \quad (43)$$

Якщо скористатись виразом (43), то можна оцінити також і допустиме число перезапусків r . За аналогією з раніше розглянутим для (43) одержимо

$$\frac{(0,5r^2 - lr^2 - r)}{L} = \ln(1 - P_{пш})$$

або

$$(0,5 - l)r^2 - r - L \ln(1 - P_{пш}) = 0. \quad (44)$$

Розв'язавши рівняння (44), одержимо оцінки для числа перезапусків r , при якому ймовірність перекриття не перевищує допустимої величини.

Для загального випадку (44) за аналогією маємо

$$0,5r^2 - (Z+1)r - L \ln(1 - P_{пш}) = 0. \quad (45)$$

Аналіз показує, що в формулі (45) змінна Z пов'язана з r , оскільки Z – це загальне число біт гами шифруючої, що сформоване при r запусках.

Таким чином, рівняння (44) та (45), а також (34) та (38) дозволяють визначити допустиме число перезапусків при заданих обмеженнях на допустимі величини $P_{пш}$, L та Z .

Крім того, як показано в [9], існує ще одна апроксимація формули (12) визначення ймовірності $P_{пш}$:

$$P_{пш} = 1 - \left[1 - \frac{Z}{L}\right]^{r-1} \left[1 + \frac{r-1}{L-Z}\right]^{-1/2} \quad (46)$$

Допустиме число перезапусків для цього випадку можна визначити як

$$r_{\text{доп}} \geq \sqrt{\frac{P_{пш} L}{l}} \quad (47)$$

де l – довжина пакета (криптограми).

4 Порівняльний аналіз складності та точності оцінки параметрів перекриття шифру

Крім основних співвідношень оцінки ймовірності перекриття, узагальнена математична модель повинна містити математичні співвідношення оцінки складності та точності оцінки. Вище одержано ряд аналітичних співвідношень, що дозволяють опініти ймовірність перекриття шифру як функцію

$$P_{пш} = f(L, Z, r). \quad (48)$$

Так, для загального випадку одержано різні формули, основні з них є наступні:

$$P_{пш} = 1 - \frac{(L - Z + r - 1)!}{L^{r-1} (L - Z)!}. \quad (12)$$

$$P_{пш} = 1 - \exp \left\{ (r - 1) \left(\frac{r - 0.5}{L - Z} - \frac{Z}{L} \right) \right\}. \quad (26)$$

$$P_{пш} = 1 - \exp \left\{ \frac{(0.5r^2 - (Z + 1)r)}{L} \right\}. \quad (42)$$

Для випадку перезапуску через кожні l біт гами шифруючої також одержано часткові випадки, що відповідають (12), (26) та (43):

$$P_{пш} = 1 - \exp \left\{ (r - 1) \left(\frac{r - 0.5}{L - r \cdot l_n} - \frac{r \cdot l}{L} \right) \right\} \quad (27)$$

та

$$P_{пш} = 1 - \exp \left\{ \frac{(0.5r^2 - (lr + 1)r)}{L} \right\}. \quad (43)$$

Важливим при оцінці (48) є знання похибок, з якими здійснюються обчислення, а також врахування складності обчислень ймовірності $P_{пш}$. Ці задачі і розглядаються та розв'язуються в параграфі.

Аналогічно (48) допустиме число перезапусків можна розглядати у вигляді функції

$$r = \psi(L, P_{пш}, Z) \quad (49)$$

Для випадку передачі пакетів однакової довжини рівняння мають вигляд

$$ar^3 + br^2 + cr + d = 0,$$

де

$$\begin{aligned} a &= l^2; \\ b &= l^2 + L(1 - l); \\ c &= Ll - 1.5L + Ll \ln(1 - P_{пш}) = L(l - 1.5 + l \ln(1 - P_{пш})) = \\ &= L(l(1 + \ln(1 - P_{пш})) - 1.5); \\ d &= L(L \ln(1 - P_{пш}) + 0.5). \end{aligned}$$

Для випадку (39) рівняння має вигляд

$$(0.5 - l)r^2 - r - L \ln(1 - P_{пш}) = 0. \quad (44)$$

В цілому, для загального випадку (26) рівняння має вигляд

$$Lx^2 + y_1x - y_2 \ln(1 - P_{пш}) = 0. \quad (38)$$

де

$$y_1 = 0.5L - LZ - Z^2; \quad y_2 = (L - Z)L; \quad x = (r - 1).$$

Для випадку (39) рівняння має вигляд

$$0,5r^2 - (Z+1) - L \ln(1 - P_{\text{пш}}) = 0. \quad (45)$$

Аналіз показує, що особливістю рівнянь (38) та (45) є те, що вони є параметричними в тому змісті, що змінна Z пов'язана з r .

З урахуванням вказаного оцінку складності обчислень знайдемо для (12), (26), (42) та (46).

Для формули (12) маємо

$$I_{12} = I_{\phi}(L - Z + r - 1) + I_c(L^{r-1}) + I_{\phi}(L - Z) + I_r(\cdot) + I_n(\cdot). \quad (50)$$

де $I_{\phi}(\cdot)$ – складність обчислення значення факторіалу від аргументу;

I_c – складність піднесення числа L до $r-1$ ступеня;

$I_g(\cdot)$ – складність ділення;

$I_n(\cdot)$ – складність віднімання.

Аналіз показує, що $I_{\phi} \gg I_c(\cdot) + I_g(\cdot) + I_n(\cdot)$. Тому для обчислення $I_{\phi}(\cdot)$ необхідно виконати $I_M = (L - Z + r - 1)$ множень. При цьому один із співмножників має розмір $\log_2(L - Z + r - 1)$, а другий – змінне значення факторіала, тобто змінну, що збільшується. Подальшу оцінку зробимо через часову t та просторову складності.

Часову складність t_c обчислимо у вигляді математичного сподівання часу обчислення значення факторіала:

$$t_c = \frac{I_M}{\gamma K}, \quad (51)$$

де γ – потужність комп'ютера, що здійснює обчислення;

$K = 3,15 \cdot 10^7$ с/рік – число секунд у році.

Необхідно, щоб обчислення здійснювались над числами змінного розміру, максимальна довжина одного з них $V_1 = \log_2(L - Z + r - 1)$, а другого $V_2 = \log_2(L - Z + r - 1)!$.

Для запам'ятовування цих чисел необхідно відповідно V_1 та V_2 біт.

В табл. 1 наведено значення t_c , V_1 для різних значень числа $V = \log_2(L - Z + r - 1)$ та для значення $\gamma = 10^{14}$ опер/с.

Таблиця 1

V	2^{63}	2^{127}	2^{191}	2^{255}	2^{512}
t_c (біт)	0,0029	$5,3 \cdot 10^{16}$	$1 \cdot 10^{56}$	$1,8 \cdot 10^{24}$	$4,1 \cdot 10^{32}$
V_1 (біт)	63	127	191	255	512

Таким чином, обчислення $P_{\text{пш}}$ за формулою (12) є фізично неможливими, оскільки часова складність обчислення для реальних значень V перевищує фізичні та енергетичні можливості.

За аналогією з (12) та (13) для (26) та (42) маємо

$$I_{27} = 5I_n + 2I_o + I_M + I_e, \quad (52)$$

де I_n – складність операції віднімання;

I_o – складність операції ділення;

I_M – складність операції множення;

I_e – складність обчислення експоненти;

I_c – складність додавання.

У виразі (26) покладено, що числа мають V довжину. Для рівняння (42) одержимо

$$I_{42} = 2I_n + I_c + I_o + 2I_M + I_e. \quad (53)$$

Аналіз показує, що (53) має дещо меншу складність ніж, (52), але порядок складності однаковий, тому при обчисленні $P_{\text{пш}}$ можна використовувати як формулу (26), так і вираз (42).

Аналогічно до (52) та (53) для оцінки складності перекриття шифру згідно (27) та (43) маємо

$$I_{27} = 5I_n + 2I_o + 3I_M + I_e. \quad (54)$$

У випадку (43) одержимо

$$I_{43} = 2I_a + I_c + 3I_m + I_d + I_e + I_{ke}. \quad (55)$$

Із порівняння (54) та (55) виходить, що складність (55) є дещо меншою, оскільки вимагає на дві операції віднімання менше, а операція піднесення до квадрата є простішою в порівнянні з операцією ділення.

Для випадку (39) із прямого підрахунку одержимо

$$I_{39} = 2I_a + I_c + (r-1)I_g + (r-2)I_m + I_n = (r+1)I_a + I_c + (r-2)I_m + I_n. \quad (56)$$

Аналіз (56) показує, що точне обчислення значення $P_{\text{ПШ}}$ значно складніше, ніж за допомогою наближень (27) та (43). Але при реальних значеннях r складність I_{40} є кінцевою та значно меншою за складність обчислення (12). Тому для точного визначення $P_{\text{ПШ}}$ можна використовувати формулу (39).

Як показує аналіз, оцінку точності можна здійснювати, використовуючи абсолютну чи відносну похибки. Абсолютною похибкою числа a [10] називається різниця $x - a$, де a – дане число, яке розглядається як наближене значення деякої величини, точне значення якої x . Відносною похибкою числа a називається відношення

$$\frac{x-a}{a} = \delta(a).$$

Число $\delta(a)$ таке, що [10]

$$\left| \frac{x-a}{a} \right| \leq \delta(a),$$

називається границею відносної похибки.

Використовуючи (15) та (16), спочатку оцінимо похибку, що вноситься за рахунок наближення (15). В результаті для абсолютної похибки маємо

$$|\Delta_a| = \left| \sqrt{2\pi n} n^r e^{-n} e^{\theta(n)} - \sqrt{2\pi n} n^r e^{-n} \right| = \sqrt{2\pi n} n^r e^{-n} |e^{\theta(n)} - 1|. \quad (57)$$

Відносна похибка

$$\delta(a) = \frac{|\Delta_a|}{\sqrt{2\pi n} n^r e^{-n}} = \frac{\sqrt{2\pi n} n^r e^{-n} |e^{\theta(n)} - 1|}{\sqrt{2\pi n} n^r e^{-n}} = |e^{\theta(n)} - 1|. \quad (58)$$

Відзначимо, що в (58) та (59) $|\theta(n)| \leq \frac{1}{12n}$, де n – значення, для якого обчислюється факторіал.

Виконаємо аналіз похибки, що пов'язана з використанням формули (26). При її виведенні було використано наближення (15), тобто не було враховано значення $e^{\theta(n)} \leq e^{1/(12n)}$. Якщо переходити від наближення (15) до (16), то в формулах (17) та (18) додатково з'являється відповідно два множники:

$$\psi_1 = e^{1/(12(L-Z+r-1))} \quad (59)$$

та

$$\psi_2 = e^{1/(12(L-Z))}. \quad (60)$$

Далі з урахуванням (59) та (60) формула (26) матиме вигляд

$$P'_{\text{ПШ}} = 1 - \exp \left\{ (r-1) \left(\frac{r-0,5}{L-Z} - \frac{Z}{L} \right) \frac{\psi_1}{\psi_2} \right\} = 1 - \exp \left\{ (r-1) \left(\frac{r-0,5}{L-Z} - \frac{Z}{L} \right) \right\} \exp \left\{ \frac{1}{12(r-1)} \right\}. \quad (61)$$

Аналізуючи (61), можна зробити висновок, що величина похибки не залежить від періоду повторення L та об'єму зашифрованих даних на одному і тому ж ключі Z , а залежить тільки від числа перезапусків, тобто кількості використаних синхропослідовностей r .

Необхідно зазначити, що похибку можна розрахувати і для $P_{\text{ПШ}}$ та $P'_{\text{ПШ}}$ аналогічно (58). В результаті

$$\delta(a) = \left| \frac{P' - P}{P} \right| = \left| 1 - \exp \left\{ (r-1) \left(\frac{r-0,5}{L-Z} - \frac{Z}{L} \right) \right\} \exp \left\{ \frac{1}{12(r-1)} \right\} \right| = \left| 1 - \exp \left\{ \frac{1}{12(r-1)} \right\} \right|. \quad (62)$$

Порівнюючи (58) та (62), можна зробити висновок про співпадання оцінок похибки.

Зробимо оцінку погрешностей, що дає формула (42). По-перше, природа похибки витікає із апроксимації $1-x \approx e^{-x}$, використаної при переході від виразу (39) до виразу (42). По-друге, оскільки в виразі (40) присутні різні значення x , то похибка, що виникає при переході від $1-x$ до e^{-x} , очевидно, не може бути використана для оцінки похибки, що дають формули (42) та (43).

У зв'язку із вказаним похибку, що дають формули (42) та (43), будемо оцінювати відносно значення, що дає формула (39). Позначимо значення, що дають формули (42) та (43) через $P'_{пш}$. Тоді, використовуючи формулу (57), маємо

$$\begin{aligned} \delta(P'_{пш}) &= \frac{P_{пш} - P'_{пш}}{P'_{пш}} = \\ &= \frac{1 - \left(1 - \frac{Z-r+1}{L}\right) \left(1 - \frac{Z-r+2}{L}\right) \cdot \left(1 - \frac{Z+1}{L}\right) - 1 + \exp\{[0,5r^2 - (Z+1)r]/L\}}{1 - \exp\{[0,5r^2 - (Z+1)r]/L\}} = \\ &= \frac{\left(1 - \frac{Z-r+1}{L}\right) \left(1 - \frac{Z-r+2}{L}\right) \cdot \left(1 - \frac{Z+1}{L}\right) - \exp\{[0,5r^2 - (Z+1)r]/L\}}{1 - \exp\{[0,5r^2 - (Z+1)r]/L\}}. \end{aligned} \quad (63)$$

Далі, аналіз формули (19) показує, що значення похибки можуть бути одержані, тільки експериментальним шляхом, тобто розрахунками згідно до формули (63).

Оцінимо похибки відносно розв'язку рівнянь (34), (38), (44) та (45).

Для цього розглянемо рівняння (38). Поділивши на L , одержимо рівняння (38) в нормованій формі:

$$x^2 + \frac{y_1}{L} - \frac{y_2}{L} \ln(1 - P_{пш}) = 0. \quad (64)$$

Підставивши в (64) значення y_1 (35) та y_2 (36), одержимо

$$x^2 + \frac{0,5L - LZ - Z^2}{L} x - \frac{(L-Z)L}{L} \ln(1 - P_{пш}) = 0$$

або

$$x^2 + \frac{0,5L - LZ - Z^2}{L} x - (L-Z) \ln(1 - P_{пш}) = 0. \quad (65)$$

Далі, враховуючи, що $x = r-1$, розв'язуємо (65) і одержуємо значення r .

Оцінку допустимого значення r числа перезапущів можна провести, використовуючи формулу (45). В цьому випадку рівняння має вигляд

$$r^2 - 2(r+1)r - 2L \ln(1 - P_{пш}) = 0. \quad (66)$$

Розв'язавши це рівняння, також одержимо оцінку допустимого значення числа перезапущів. Але при цьому необхідно враховувати, що рівняння (65) та (66) є параметричними, оскільки Z залежить від r .

Для випадку, коли пакети або повідомлення мають однакову довжину, можна використати рівняння (44). Воно може бути зведене до вигляду

$$r^2 - \frac{1}{0,5-l} r' - \frac{L \ln(1 - P_{пш})}{0,5-l} = 0.$$

Враховуючи, що $x^2 > r'$, наближеною оцінкою є:

$$r = \sqrt{\frac{L \ln(1 - P_{\text{пш}})}{0,5 - l}} \quad (67)$$

Таким чином, оцінку значення r можна виконати, використовуючи вираз (67), а більш точне рішення дає рівняння (66).

5 Практичне дослідження ймовірностей перекриття шифру

На практиці, при розрахунках строку дії ключа виходять із допустимої величини ймовірності перекриття шифру $P_{\text{доп}}/\Delta t$. Прийmemo в якості $P_{\text{доп}}/\Delta t = 10^{-9}, 10^{-12}, 10^{-16}, 10^{-20}, 10^{-24}$ та 10^{-36} . Виберемо такі значення періоду повторення гами $L_1 = 2^{64}, 2^{128}, 2^{192}, 2^{256}$ та 2^{512} .

Параметр r при дослідженнях виберемо $r = 2, 2^8, 2^{12}, 2^{16}, 2^{32}$.

Будемо вважати, що шифрування здійснюється зі швидкостями 1, 10, 100, 1000 та 4000 Мбіт/с.

В табл. 2 наведені значення Z в залежності від швидкості шифрування за умови, що ключові дані змінюються один раз за добу.

Таблиця 2

Z	V	10^6 Мг біт/с	10^7	10^8	10^9	$4 \cdot 10^9$
Z		$8,64 \cdot 10^{10}$	$8,64 \cdot 10^{11}$	$8,64 \cdot 10^{12}$	$8,64 \cdot 10^{13}$	$3,46 \cdot 10^{14}$

В табл. 3 наведені значення Z в залежності від швидкості шифрування за умови, що ключ змінюється один раз за тиждень.

Таблиця 3

Z	V	10^6 Мг біт/с	10^7	10^8	10^9	$4 \cdot 10^9$
Z		$6,05 \cdot 10^{11}$	$6,05 \cdot 10^{12}$	$6,05 \cdot 10^{13}$	$6,05 \cdot 10^{14}$	$2,42 \cdot 10^{15}$

В табл. 4 наведено значення Z в залежності від швидкості шифрування за умови, що ключ змінюється один раз на добу і в мережі на цьому ключі одночасно працює q станцій.

Таблиця 4

Z	V	q	10^6 Мг біт/с	10^7	10^8	10^9	$4 \cdot 10^9$
Z		2	$1,72 \cdot 10^{11}$	$1,72 \cdot 10^{12}$	$1,72 \cdot 10^{13}$	$1,72 \cdot 10^{14}$	$6,91 \cdot 10^{14}$
Z		16	$1,35 \cdot 10^{12}$	$1,35 \cdot 10^{13}$	$1,35 \cdot 10^{14}$	$1,35 \cdot 10^{15}$	$5,4 \cdot 10^{15}$
Z		64	$5,53 \cdot 10^{12}$	$5,53 \cdot 10^{13}$	$5,53 \cdot 10^{14}$	$5,53 \cdot 10^{15}$	$2,16 \cdot 10^{16}$
Z		256	$2,16 \cdot 10^{13}$	$2,16 \cdot 10^{14}$	$2,16 \cdot 10^{15}$	$2,16 \cdot 10^{16}$	$8,64 \cdot 10^{16}$
Z		1024	$8,84 \cdot 10^{13}$	$8,84 \cdot 10^{14}$	$8,84 \cdot 10^{15}$	$8,84 \cdot 10^{16}$	$3,46 \cdot 10^{17}$

Обчислення ймовірностей перекриття шифру в поточному режимі здійснювалось з використанням відповідно до формул: (12) – еталон або (39) – еталон, (26) та (42) та (46). Формули (12) та (39) є точними відповідно до прийнятої моделі. Формули (26) та (39) є наближеними. Ще одна апроксимація формули (12) розглянута в [9], вона має вигляд (46).

Спочатку дослідження ймовірності перекриття шифру провадилися з метою експериментального порівняння точності. В табл. 5 наведені значення ймовірностей перекриття шифру в залежності від періоду повторення гами L (лічильника) при фіксованих значеннях довжина повідомлення $l_m = 10^8$ біт, число перезапусків $r = 2^{16} \approx 6,6 \cdot 10^4$. Це еквівалентно об'єму передаваних даних $Z = 6,6 \cdot 10^{12}$, що передаються протягом однієї доби.

Таблиця 5

Період L Формула	2^{64}	2^{128}	2^{192}	2^{256}	2^{512}
3.13	2.301480	1.262215	6.842486	3.709319	3.203430
3.40	3380E-02	9606E-21	4337E-41	3281E-60	690E-137
3.27	2.301375 794E-02	1.262158 1760E-21	6.842173 1850E-41	3.709149 5160E-60	3.203284 041E-137
3.43	2.301410 5150E-02	1.262177 4420E-21	6.842277 6240E-41	3.709206 1320E-60	3.203332 936E-137
3.47	2.328270 8980E-02	1.262158 1830E-21	6.842173 2190E-41	3.709149 5340E-60	3.203284 057E-137
3.49	2.301375 7940E-02	1.262158 1760E-21	6.842173 1850E-41	3.709149 5160E-60	3.203284 041E-137

В табл. 6 наведені значення ймовірності перекриття шифру в залежності від періоду повторення гами при довжині повідомлення $l_m = 10^8$ біт та числі перезанусків $r = 2^{32} \approx 4,3 \cdot 10^9$. Це відповідає величині $Z = 4,3 \cdot 10^{17}$, що еквівалентно роботі в мережі 1024 станцій пакетів протягом однієї доби зі швидкістю $V = 10^8$ біт/с. Величина $Z = 4,3 \cdot 10^{17}$ еквівалентна роботі 1024 станцій протягом 40 днів (тобто більше одного місяця). При цьому за вказаний час може бути здійснено $2^{32} \approx 4,9 \cdot 10^9$ перезанусків, тобто сформовано та передано таку кількість синхропослідовностей.

Таблиця 6

Період L Формула	2^{64}	2^{128}	2^{192}	2^{256}	2^{512}
	1.0E+00	5.421052 0850E-12	2.938758 2240E-31	1.593104 0250E-50	1.375831 489E-127
	1.0E+00	5.421052 1140E-12	2.938758 2390E-31	1.593104 0340E-50	1.375831 496E-127
	1.00000760090 E ≠ 00	5.421052 1120E-31	2.938758 2390E-31	1.593104 0330E-50	1.375831 496E-127
	1.0E+00	5.421052 0850E-12	2.93875 82240E-31	1.593104 0250E-50	1.375831 489E-127

В табл. 7 наведені результати обчислення ймовірностей перекриття шифру у залежності від тих же значень r та L при довжинах повідомлень (пакетів) $l_m = 10^5$ та 10^6 біт, а також у залежності від періоду повторення гами L .

Таблиця 7

r L	2^8		2^{16}		2^{32}		2^{64}	
	1000	10000	1000	10000	1000	10000	1000	10000
2^{32}	0,782	1	1	1	1	1	1	1
2^{40}	5,92E-03	5,76E-02	1	1	1	1	1	1
2^{42}	1,48E-03	1,47E-02	1	1	1	1	1	1
2^{46}	9,28E-05	9,27E-04	0,998	1	1	1	1	1
2^{48}	2,32E-05	2,32E-04	0,783	1	1	1	1	1
2^{50}	5,80E-06	5,80E-05	0,317	0,978	1	1	1	1
2^{64}	4,10E-10	3,5E-09	2,33E-05	2,33E-04	1	1	1	1
2^{128}	1,92E-29	1,92E-28	1,26E-24	1,26E-23	5,42E-15	5,42E-14	1	1
2^{192}	1,04E-48	1,04E-47	6,84E-44	6,84E-43	2,93E-34	2,93E-33	5,42E-15	5,42E-14
2^{256}	5,63E-68	5,64E-67	3,71E-62	3,71E-62	1,59E-53	1,59E-52	2,94E-34	2,94E-33

Одержані результати обчислень ймовірностей перекриття шифру, наведені в табл. (5 – 7) дозволяють зробити наступні висновки.

1. При використанні, наприклад алгоритму FIPS 197 в режимі лічильника може бути забезпечена необхідна ймовірність перекриття шифру. За даними таблиць можна вибрати параметри r , Z , l_m , L та визначити $P_{\text{пш}}$.

2. Із даних, наведених в табл. 6 також виходить, що при довжині блоку $l_b = 128$ біт (FIPS 197) забезпечуються достатні рівні захищеності від перекриття шифру.

Так, при $r = 2^{32}$ та $l_m = 10^4$ забезпечується передача $Z = r \cdot l_m \approx 4,3 \cdot 10^{13}$ біт інформації з максимальною ймовірністю перекриття $5,4 \cdot 10^{-16}$.

Це еквівалентно роботі станцій з обміном вказаним числом біт між собою зі швидкістю 10^8 біт/с протягом приблизно шести днів, причому ймовірність перекриття не перевищує 10^{-15} .

При $r = 2^{32}$ та $l_m = 10^6$ забезпечується передача $Z = 4,3 \cdot 10^{15}$ біт інформації з максимальною ймовірністю перекриття $5,4 \cdot 10^{-14}$. Це еквівалентно роботі 1024 станцій зі швидкістю 10^8 біт/с протягом однієї доби.

Список літератури: 1. Харин Ю.С., Берник В.И., Матвеев Г.В. Математические основы криптологии: Минск: Изд-во БГУ, 1999. 319 с. 2. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002. 816 с. 3. Бабац А.В., Шанкин Г.П. Криптография. М.: Солон-Р, 2002. 511 с. 4. Колчин В.Ф., Севастьянов Б.А., Чистяков В.П. Случайные размещения. М.: Наука, 1976. 223 с. 5. Михайлов В.Г. О повторяемости состояний датчика псевдослучайных чисел при его многократном использовании // Теория вероятн. и ее примен. 1995. Т. 40. Вып. 4. С. 786 – 797. 6. Михайлов В.Г. О перекрытиях путей на графе генератора псевдослучайных чисел // Обзорение прикл. и промышл. матем. 1995. Т.2. Вып. 6. С. 888 – 905. 7. Савчук М.Н. О предельных распределениях некоторых случайных величин, связанных с перекрытиями сообщений в дискретных системах передачи информации. Некоторые вопросы применения марковских процессов для моделирования систем. К.: АН УССР, Ин-т кибернетики, 1983. С. 29 – 36. 8. Савчук М.Н. О размещении комплектов частиц на окружности. // Вероятностные методы в дискретной математике: Тезисы докладов Всесоюзной конференции. Петрозаводск: Карельский ф-л АН СССР. 1983. С. 77 – 78. 9. Коваленко И.Н., Левитская А.А., Савчук М.Н. Избранные задачи вероятностной комбинаторики. К.: Наук. думка. 1986. 224 с. 10. Ивченко Г.И. Об одной модели случайных размещений. // Вероятностные процессы и их приложения: Межвузовский сборник. М.: МИЭМ. 1987. С. 3 – 12. 11. Ивченко Г.И. Предельные теоремы в дискретной задаче о покрытии окружности // Вероятностные процессы и их приложения: Межвузовский сборник. М.: МИЭМ, 1990. С. 27 – 34. 12. Janson S. Random covering of the circle by arcs with restricted endpoints // J. Appl. Probab. 1988. V. 25. № 1. P. 215 – 219. 13. Михайлов В.Г. Предельная теорема Пуассона в схеме размещения частиц комплектами // Теория вероятн. и ее примен. 1977. Т. 22. Вып. 1. С. 156 – 160.

Харківський національний
університет радіоелектроніки

Надійшла до редколегії 10.10.2004

ИССЛЕДОВАНИЕ СТАТИСТИЧЕСКИХ СВОЙСТВ СХЕМЫ РАЗВОРАЧИВАНИЯ КЛЮЧЕЙ FIPS-197

Введение

При разработке блочного симметричного шифра (БСШ) важным этапом является оценка стойкости шифра к различным атакам, что подтверждают требования предъявляемые к конкурсантам участвовавшим в конкурсах AES и NESSIE [1]. Так современный БСШ должен предоставлять гарантированную безопасность против атак грубой силы и против известных аналитических методов криптоанализа. Условно аналитические атаки можно разбить на два важных класса: на цикловую функцию и на схему разворачивания ключей. Наибольшую угрозу сегодня представляют дифференциальный и линейный криптоанализ, а также большое разнообразие атак являющихся их разновидностями. В связи с этим, большое внимание при разработке БСШ уделяется конструированию цикловой функции, итеративное повторение которой, должно обеспечивать гарантированную безопасность к известным видам криптоанализа. Такой подход к созданию шифров привел к тому, что разработчиками уделяется мало внимания к проектированию схем разворачивания ключей. В большинстве современных БСШ используются алгоритмы, которые формируют цикловые подключи по некоторому простому линейному закону. Однако в конце 90-х появился ряд мощных методов криптоанализа, использующих линейность формирования цикловых подключей, к которым оказались уязвимы БСШ стойкие к атакам на цикловую функцию. Наиболее известными из них являются "связанные ключи" [1], "скольжения" [2], "встреча по середине" [3], "простые зависимости и эквивалентные ключи" [4]. Особенностью таких атак, по сравнению с атаками на цикловую функцию БСШ, является большая эффективность, в случае ее проведения, и слабая зависимость от количества итераций использующихся в шифре [2]. Сформулируем условия, при которых имеется потенциальная возможность осуществления какой-либо из указанных атак:

1. Возможность подобрать связь между цикловыми ключами (наличие линейных связей между битами цикловых ключей K_n^i , где i – номер итерации) [1].
2. Возможность установить зависимость между битами секретного ключа K и цикловых ключей K_n^i [15].
3. Если некоторые биты криптограммы не зависят от всех бит секретного ключа K [9].
4. Схема разворачивания ключей формирует цикловые ключи K_n^i с некоторой периодичностью [2].
5. Использование цикловой функции с плохими свойствами рассеивания.
6. Биты цикловых ключей K_n^i не зависят от всех бит секретного ключа K .

Для устранения потенциальной возможности реализации атак на схемы разворачивания ключей, некоторые авторы [5, 9] предлагают использовать цикловую функцию с высокими криптографическими свойствами и избегать применения в качестве схемы разворачивания ключей алгоритмов, формирующих цикловые подключи линейным способом. В некоторых современных БСШ, представленных на конкурсах AES и NESSIE, наблюдается тенденция использования нелинейных преобразований для формирования цикловых подключей. Однако обзор конкурсантов [11], показывает, что такой подход к решению проблемы не всегда является достаточным и БСШ может быть уязвим к атакам на схему разворачивания ключей. Другими словами криптоаналитик имеет возможность определить закономерности формирования цикловых подключей. Таким образом, стоит задача достижения статистической независимости между битами цикловых подключей K_n^i , и между битами личного

секретного ключа K и цикловых подключей K'_i . Во-первых, для этого каждый бит любого циклового ключа K'_i должен являться результатом некоторой нелинейной функции от всех бит секретного ключа K . Во-вторых, между битами ключа K и цикловых подключей K'_i , а также между битами самих цикловых подключей K'_i , должны отсутствовать линейные связи. В-третьих, вычисление каких-либо бит ключа K или бит цикловых подключей K'_i , из некоторых известных бит любого циклового подключа, должно быть вычислительно трудным (невозможным) [5, 6]. Однако выявление таких закономерностей не всегда является тривиальной задачей, особенно, если для формирования цикловых подключей используются нелинейные преобразования.

В статье приводится методика оценки статистических свойств нелинейных схем разворачивания ключей, позволяющая определить наличие линейных связей, которые могут стать потенциальной возможностью построения какой-либо атаки. При проведении исследований использовались методы корреляционного анализа, суть которых и необходимость их использования рассматривается далее.

В качестве объекта исследований выбрана схема разворачивания ключей FIPS-197 (длина блока $l_b = 128$ бит, длина ключа $l_k = 128, 192, 256$ бит). Исследования шифра, проводившиеся в рамках проекта NESSIE, показали, что FIPS-197 устойчив ко всем известным методам криптоанализа. Отметим, что наиболее эффективными из них являются атаки на схему выработки цикловых подключей [12, 13]. Как показывают авторы атак, возможность их проведения связана с тем, что схема разворачивания ключей FIPS-197 формирует цикловые подключи, обладающие линейными связями, которые используются при проведении этих атак.

Целью исследований схемы разворачивания ключей FIPS-197 является углубленный анализ конструкции для выявления скрытых "аномалий" при формировании цикловых подключей, которые нельзя выявить аналитическим путем. Косвенно исследования позволяют понять особенности функционирования схемы разворачивания ключей FIPS-197.

В связи с этим ставились задачи исследовать зависимость каждого бита всех цикловых подключей от каждого бита исходного ключа. Проанализировать, каким образом изменение любого (одного) бита в исходном ключе K оказывает влияние на изменение бит в цикловых подключях. Выявить наличие линейных связей между битами цикловых подключей.

Методы исследования схемы разворачивания ключей FIPS-197

Для решения сформулированных задач использовались методы корреляционного анализа, адаптированные для исследования схемы разворачивания ключа. За основу взяты три показателя стойкости (статистической безопасности [7]), которые принято использовать при проверке многоцикловых процедур современных блочных систем шифрования:

1. Проверка наличия "лавинного" эффекта в развернутом ключе.
2. Оценка "полноты", характеризующая степень зависимости любого бита в развернутом ключе от всех бит исходного ключа.
3. Оценка "строгости лавинного критерия", который показывает равновероятность изменения любого бита в развернутом ключе при изменении одного (любого) бита в исходном ключе.

Дополнительно оценивалось количество бит, которое изменяется в развернутом ключе K_{ext} при изменении любого одного бита в исходном ключе K . Обозначим количество бит, которые изменились как M .

Далее будем использовать обозначения принятые в работе [7]: для показателей "лавинного" эффекта – d_a , "полноты" – d_c и "строгости лавинного критерия" – d_{3a} . Описание данных методов и порядок проведения исследования приводятся в [8].

Схему разворачивания ключа FIPS-197 будем рассматривать как отображение “исходный ключ – развернутый ключ”, задаваемое некоторой функцией $f : (GF(2))^n \rightarrow (GF(2))^m$ или $f : (K)^n \rightarrow (K_{ext})^m$, где n – количество бит исходного ключа; m – количество бит развернутого ключа K_{ext} . Развернутый ключ K_{ext} является битовым массивом из которого формируются цикловые подключи K_y^i . Для алгоритма с длиной личного ключа $n = 128, 192, 256$ бит, в соответствии со спецификацией [10], длина развернутого ключа составляет $m = 1408, 1664, 1920$ бит соответственно.

Обоснуем необходимость использования указанных методов исследования для проверки выдвинутых во введении предположений об необходимости зависимости каждого бита циклового подключа от всех бит секретного ключа, а также наличия нелинейной связи между битами самих цикловых подключей, битами секретного ключа и цикловых подключей.

Если каждый бит развернутого ключа K_{ext} зависит от каждого бита входного ключа K , то ожидаемое значение показателя “полноты” $d_c = 1$, как следует из работы [7]. Показатели “лавинного” эффекта и “строгого лавинного критерия” равные единице ($d_a \approx 1 \pm 0,1$ и $d_{sa} \approx 1 \pm 0,1$, соответственно), говорит о том, что любой (один) бит исходного ключа K влияет равновероятно на каждый выходной бит развернутого ключа K_{ext} .

Наличие “лавинного” эффекта в развернутом ключе K_{ext} дает гарантию, что в среднем при изменении любого (одного) бита в исходном ключе K изменится $m/2$ бит. Если изменение одного бита исходного ключа будет приводить к незначительным изменениям в развернутом ключе, то это может привести к возможности определения групп ключей, с близким весом Хемминга, и, соответственно, криптограмм, с очень близкими значениями веса Хемминга [9]. Это может позволить криптоаналитику сократить подбор ключа, методом перебора, используя поиск по сформированным группам ключей. В этом случае, алгоритм шифрования должен обладать цикловой функцией с высокими показателями рассеивания и перемешивания, а также иметь достаточное [9] количество итераций, чтобы компенсировать появление указанных свойств.

Изменение какого-либо (одного) бита исходного ключа K должно приводить к равновероятному изменению любого бита в развернутом ключе, что проверяется оценкой “строгого лавинного критерия”. Если биты развернутого ключа будут изменяться неравновероятно, то это говорит о том, что между битами исходного и развернутого ключей можно установить закон формирования бит, а значит, и определить (подобрать) связи между битами самих цикловых ключей. Это может служить основой для реализации атаки типа “связанные ключи” и ее разновидностей.

Оценка “полноты” должна показать, что каждый бит развернутого ключа K_{ext} зависит от каждого бита исходного ключа K . Если биты исходного ключа воздействуют не на все биты развернутого ключа, что имеется потенциальная возможность осуществления ряда аналитических атак. Например, “встреча по середине”. Оценка “полноты” служит дополнением к оценке “лавинного” эффекта и строгого “лавинного” критерия.

Обозначения и параметры тестирования:

K – исходный ключ ($K = GF(2)^n = \{0,1\}^n$; $n = l_k = 128, 192$ и 256 бит);

W – векторное пространство 32-битных элементов (“слов”) ($W = GF(2)^{32} = \{0,1\}^{32}$);

r – количество всех итераций для исследуемой версии алгоритма ($r = 10$ для $l_k = 128$, $r = 12$ для $l_k = 192$, $r = 14$ для $l_k = 256$);

K_{ext} – развернутый ключ, который состоит из элементов массива, образованных “словами” W ($K_{ext} = W \dots W = GF(2) \dots GF(2) = \{0,1\} \dots \{0,1\}$; r – количество итераций);

K_y^r – подключ для итерации r ($K_y^r = W^r = GF(2)^4 = \{0,1\}^4$);

N_B – количество ключей K , задействованных в тестировании ($N_B = 10, 100, 5000, 10000$).

Исследование статистических свойств схемы разворачивания ключа FIPS-197

Исследования проводились в два этапа. На первом этапе сделана общая оценка статистических свойств развернутого ключа с использованием методов корреляционного анализа. Производился расчет показателей лавинного эффекта, полноты и строгого лавинного критерия.

Выполнить статистические исследования для всех возможных ключей K – вычислительно невыполнимая задача, поэтому использовался ограниченный набор ключей, сгенерированных случайно и равновероятно из полного множества $\{0,1\}^{128}$, $\{0,1\}^{192}$, $\{0,1\}^{256}$ и образующих выборку объемом N_B . Объем выборки N_B изменялся переменным до тех пор, пока значения рассчитываемых параметров d_a , d_{sa} , d_c , для двух соседних выборок с различным количеством исследуемых ключей $K \in N_B$, отличались на величину не более чем принятая погрешность $\alpha = \pm 0,1$ [7].

На втором этапе анализировалось влияние каждого бита исходного ключа $K \in N_B$ на биты цикловых ключей. Подсчитывалось количество бит M , которое изменялось в каждом из элементов массива W , из которых состоит развернутый ключ K_{ext} , при изменении любого (одного) бита в ключе K .

Этап 1

Исследования проводились для выборки объемом $N_B = 10$, $N_B = 100$, $N_B = 5000$, $N_B = 10000$. Результаты исследований приведены в табл. 1 для алгоритма с длиной ключа $l_k = 128$.

Таблица 1

Количество испытаний, N_B	10	100	5000	10000
Тип теста				
d_c	0,574391	0,574929	0,574929	0,574929
d_a	0,583950	0,582674	0,582683	0,582752
d_{sa}	0,428958	0,523148	0,560929	0,562586
M	410,100781	409,189284	409,20897	409,15610
$\max N$	608	608	628	626
$\min N$	208	184	133	138

Полученные результаты (табл. 1) для $N_B = 5000$, $N_B = 10000$ при расчете для d_a , d_c и d_{sa} не уточняются больше чем на 0,01, что меньше, чем принятое значение погрешности $\alpha = \pm 0,1$ [7]. Далее будем использовать рассчитанные величины, полученные для выборки объема $N_B = 10000$.

Анализируя полученные в ходе эксперимента показатели d_a , d_c и d_{sa} (табл. 1) можно отметить, что показатель “лавинного” эффекта в развернутом ключе отличается от единицы на величину, большую, чем принятое значение погрешности $\alpha = \pm 0,1$: $d_a = 0,582752$. Учитывая принятую трактовку значений рассчитываемых величин [7], полученный результат свидетельствует о том, что “лавинный” эффект отсутствует. Следовательно, биты исходного ключа K оказывают незначительное воздействие на биты развернутого ключа K_{ext} , что подтверждается показаниями низкого значения степени “полноты” – $d_c = 0,574929$. Среднее количество изменяющихся бит в развернутом ключе, при изменении одного входного бита исходного ключа составило $M \approx 409$ (при общей длине развернутого ключа – 1408 бит), что меньше чем ожидаемое $M \approx 708$. Показания строгого “лавинного” критерия $d_{sa} = 0,562586$, из чего следует, что при изменении любого (одного) бита исходного ключа K биты в развернутом ключе K_{ext} изменяются неравновероятно. Из этого следует, что между битами исходного и развернутого ключей существует определенная зависимость, а значит, имеется возможность определить (подобрать) связи между битами самих цикловых ключей.

Аналогичные расчеты проведены для алгоритма с длиной ключа $l_k = 192$ (табл. 2).

Таблица 2

Количество испытаний, N_B	10	100	5000	10000
Тип теста				
d_c	0,548910	0,549279	0,549279	0,549279
d_a	0,557743	0,557121	0,557501	0,557332
d_{sa}	0,410788	0,499322	0,534434	0,535977
M	463,04218	462,52489	462,84044	462,69999
maxN	680	685	698	700
minN	292	247	208	205

Полученные результаты (табл. 2) для $N_B = 5000$, $N_B = 10000$ при расчете для d_a , d_c и d_{sa} не уточняются больше чем на 0,01, что меньше, чем принятое значение погрешности $\alpha = \pm 0,1$ [7]. Далее будем использовать результаты для выборки объема $N_B = 10000$.

Проведем аналогичный статистический анализ для схемы разворачивания ключа с длиной исходного ключа $l_k = 256$ бит. Результаты исследований представлены в табл. 3.

Сравнивая полученные результаты при исследовании схемы разворачивания ключей с длиной ключа $l_k = 192$ бита (табл. 2) и $l_k = 256$ бит (табл. 3) с результатами, полученными для секретного ключа K длиной $l_k = 128$ бит (табл. 1), можно отметить, что показатели для d_a , d_c и d_{sa} отличаются не более, чем на 0,1, что не выше принятой погрешности $\alpha = \pm 0,1$. Учитывая, что для разворачивания ключа длиной $l_k = 256$ бит используется другой алгоритм [8,10], можно говорить, что оба алгоритма обладают одинаковыми статистическими показателями. Следовательно, выводы, сделанные по результатам для схемы разворачивания ключей с ключом $l_k = 128$ бит ($l_k = 256$ бита) (табл. 1, 2), справедливы и для алгоритма с длиной ключа $l_k = 192$ бит.

Сравнивая полученные результаты при исследовании схемы разворачивания ключей с длиной ключа $l_k = 192$ бита (табл. 2) и $l_k = 256$ бит (табл. 3) с результатами, полученными для секретного ключа K длиной $l_k = 128$ бит (табл. 1), можно отметить, что показатели для d_a , d_c и d_{sa} отличаются не более, чем на 0,1, что не выше принятой погрешности $\alpha = \pm 0,1$. Учитывая, что для разворачивания ключа длиной $l_k = 256$ бит используется другой алгоритм [8,10], можно говорить, что оба алгоритма обладают одинаковыми статистическими показателями. Следовательно, выводы, сделанные по результатам для схемы разворачивания ключей с ключом $l_k = 128$ бит ($l_k = 256$ бита) (табл. 1, 2), справедливы и для алгоритма с длиной ключа $l_k = 192$ бит.

Таблица 3

Количество испытаний, N_B	10	100	5000	10000
Тип теста				
d_c	0,482788	0,483122	0,483154	0,483146
d_a	0,491599	0,490116	0,490406	0,490444
d_{sa}	0,360397	0,439228	0,470780	0,472179
M	470,934766	469,511445	469,790155	469,82640
maxN	713	716	733	747
minN	247	192	187	180

Однако этого, как показывают результаты (табл. 3), недостаточно для того, чтобы все биты исходного ключа K в полной мере воздействовали на биты развернутого ключа K_{ext} , изменяющихся бит в цикловых ключах.

Таким образом, цикловые ключи, формируемые обоими алгоритмами разворачивания ключей для FIPS-197, обладают сформулированными во введении свойствами, которые могут стать причиной для проведения одной из атак на схемы разворачивания ключей.

Этап 2

Проведен углубленный статистический анализ схемы разворачивания ключа FIPS-197. Цель этого исследования – расширить предыдущие исследования и выявить, какие биты исходного ключа K и в какой степени, воздействуют на конкретные биты в развернутом ключе.

Первоначально исследовалась схема разворачивания ключей FIPS-197 для длин $l_k = 128$ бит. “Лавинный” эффект наблюдался в каждом “слове” $W_0 \dots W_{44}$ развернутого ключа K_{ext} , при изменении любого (одного) бита во входном ключе K . Испытания проводились для $N_B = 10000$ случайных входных векторов (исходные ключи). Результаты исследования представлены графиком (рис. 1).

Как известно [7], лавинный эффект присутствует в том случае, если при изменении любого входного бита в среднем изменяется половина всех бит в выходном векторе. Очевидно, что в каждом выходном “слове” $W_0 \dots W_{44}$, из которых состоит развернутый ключ K_{ext} , должно изменяться около половины бит (16 бит). Если “лавинный” эффект присутствует во всех “словах” $W_0 \dots W_{44}$ развернутого ключа, то и для всей последовательности развернутого ключа будет удовлетворяться условие наступления “лавинного” эффекта.

Из полученных результатов можно увидеть, что количество изменяющихся бит, равное или близкое к половине, наблюдается в словах $W_{17} \dots W_{44}$. Диапазон изменившихся бит для этих “слов” изменяется с периодичностью от 10 до 16 бит в каждом “слове”. Изменение бит исходного ключа K приводит к недостаточному изменению бит для “слов” $W_0 \dots W_{16}$ (1,26-8, табл. 4), что говорит об отсутствии лавинного эффекта в этих “словах”. Низкие значения лавинного эффекта говорят о недостаточной зависимости между битами исходного ключа K и битами в “словах” развернутого ключа K_{ext} . Соответственно, чем меньше бит изменилось в “слове”, тем меньше для этого слова будет зависимость от бит входного ключа. Очевидно, что на биты в словах $W_0 \dots W_{16}$ биты исходного ключа K оказывают незначительное влияние. Максимальные показатели лавинного эффекта и связь между битами секретного ключа K и развернутого K_{ext} наблюдается в “словах”: $W_{21}, W_{25}, W_{29}, W_{33}, W_{37}, W_{41}$.

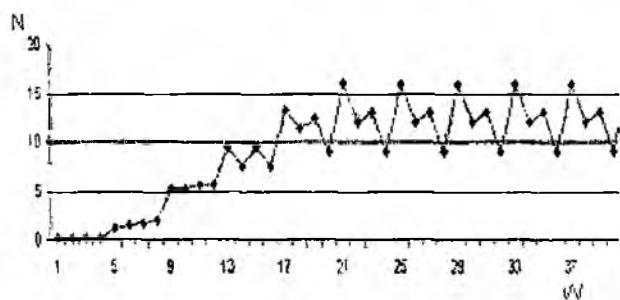


Рис. 1

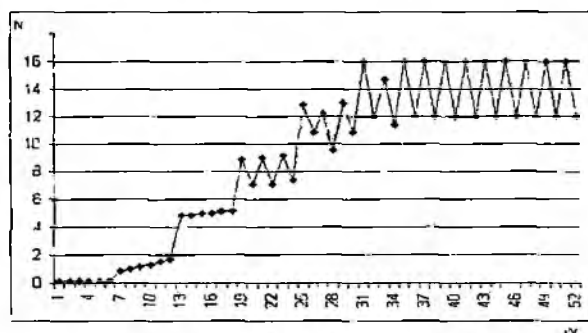


Рис. 2

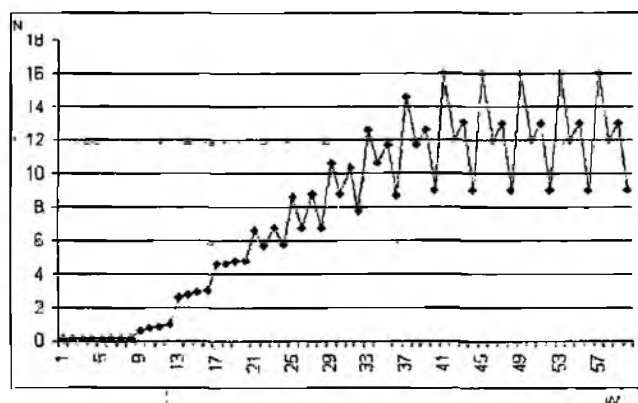


Рис.3

Из полученных результатов можно сделать выводы, что биты исходного ключа K влияют на определенные (рис. 1, рис. 2 и рис. 3) биты развернутого ключа K_{ext} , что подтверждает проведенные на этапе 1 расчеты “лавинного” эффекта.

В связи с этим возникает вопрос, каким образом влияет каждый отдельный бит исходного ключа на биты в развернутом ключе. Это позволит однозначно выявить зависимости, наличие которых уже было подтверждено в предыдущих исследованиях.

Проанализируем влияние любого (одного) i -го бита исходного ключа K на биты в каждом “слове” $W_0...W_{44}$ для версии схемы разворачивания ключей с длиной исходного ключа $l_k = 192$ бит. Определим количество бит, которые изменяются в “словах” $W_0...W_{44}$, развернутого ключа K_{ext} , при изменении каждого бита в исходном ключе K .

На рис. 4-7 изображены графики, отображающие влияние каждого бита ключа K на количество изменяющихся бит (ось ординат) в каждом “слове” $W_0...W_{44}$ развернутого ключа K_{ext} (ось абсцисс). При проведении испытаний было установлено, что изменение некоторых бит исходного ключа приводит к одинаковым изменениям бит в одних и тех же “словах” развернутого ключа. По этому признаку все графики были отсортированы по группам. На рис. 4 изображены графики, показывающие изменение количества бит в W_i “словах” при изменении 1-32 бита в исходном ключе K . На рис. 5 собраны графические отображения изменения бит в ключевой последовательности K_{ext} при изменении 33-64 бит исходного ключа K . На рис. 6 изображена зависимость изменения бит в “словах” развернутого ключа для 65-96 бит ключа K . Степень лавинного эффекта в ключевой последовательности K_{ext} при изменении 97-128 бит в пользовательском ключе K отображена на рис. 7. Таким образом биты пользовательского ключа K могут быть разбиты на группы по 32 бита, в которых биты объединяются по степени своего воздействия на биты развернутого ключа K_{ext} . Очевидно, что биты с 1 по 96 вызывают лавинный эффект (в “слове” должно изменяться в среднем около 16 бит только в части “слов”, из которых состоит развернутый ключ. Причем в “словах” W_1-W_{16} лавинный эффект проявляется слабо, так как максимально зафиксированное количество изменившихся бит – 13,27. Из рис. 7 видно, что биты с 97 по 128 исходного ключа K вызывают максимальное воздействие на биты развернутого ключа K_{ext} . В “словах” $W_{17}-W_{44}$ наблюдается равномерный лавинный эффект. А для младших “слов” W_1-W_{16} фиксируется значительно большее количество изменившихся бит в развернутом ключе, чем при изменении младших бит 1-96. Следовательно, старшие биты 97-128 пользовательского ключа K , оказывают наибольшее влияние на статистические характеристики развернутого ключа K_{ext} .

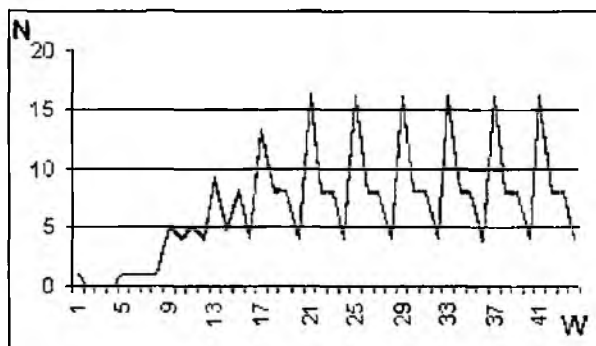


Рис. 4

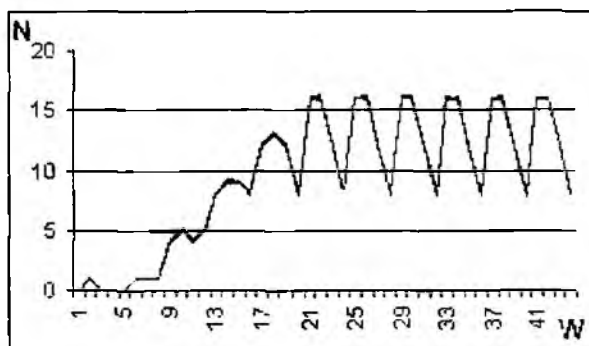


Рис. 5

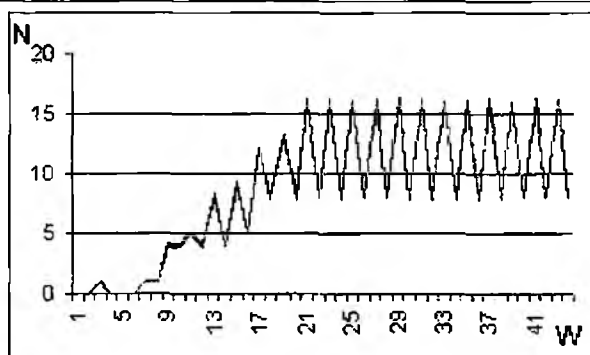


Рис. 6

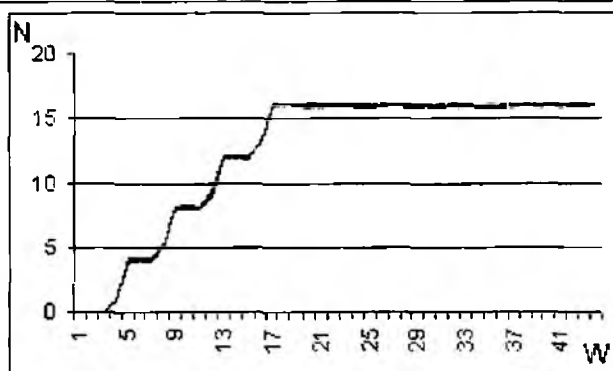


Рис. 7

Аналогичный развернутый анализ влияния каждого бита исходного ключа K на наличие “лавиного” эффекта в каждом “слове” развернутого ключа K_{ex} проведен для схемы разворачивания ключа с длинами $l_k = 192$ и $l_k = 256$ бит. Результаты исследований здесь не приводятся в связи с ограниченным объемом статьи. Отметим, что наблюдается схожее разделение бит исходного ключа K на группы, объединенные по признаку одинакового воздействия на биты развернутого ключа (по 32 бита). Однако, если для версии схемы разворачивания ключа $l_k = 128$ бит максимальное воздействие на биты развернутого ключа оказывали только старшие биты (97-128), то для ключа длиной $l_k = 192$ бита выделяются группы бит: 33-64, 97-128 и 161-192. Для версии схемы разворачивания ключа $l_k = 256$ бит максимальное воздействие на достижение лавинного эффекта в “словах” развернутого ключа оказывают группы бит: 97-128 и 225-256. В целом эпюры для схемы разворачивания ключей с длинами ключей $l_k = 192$ и $l_k = 256$ бит идентичны эпюрам для $l_k = 128$ бит. Отличие заключается только в крутизне нарастания “лавиного” эффекта, что не играет существенной роли.

Особенности выработки цикловых ключей FIPS-197

В предыдущем разделе было показано, что битовая последовательность развернутого ключа отличается от случайной и характеризуется влиянием определенных бит исходного ключа на группы бит развернутого ключа. Объясним полученные результаты статистического анализа на основе принципов работы схемы разворачивания ключей FIPS-197.

Напомним, что для выработки цикловых ключей используются два алгоритма разворачивания ключей [10]. Первый предназначен для версии алгоритма с длиной ключа $l_k = 128$ бит ($l_k = 192$ бита), второй – для разворачивания ключей длиной $l_k = 256$ бит. Выбор необходимого алгоритма производится в соответствии с параметром N_k , который представляет собой количество столбцов матричного представления исходного ключа K и, соответственно,

равен $N_k = l_k / 32 = 128 / 32 = 4$, для $l_k = 192$ бита $N_k = l_k / 32 = 192 / 32 = 6$ и для $l_k = 256$ бит $N_k = l_k / 32 = 256 / 32 = 8$.

В качестве первого циклового ключа используется исходный ключ K ($K_{11}^0 = K$), биты которого являются первыми элементами массива W расширенного ключа K_{ext} . Остальные K_{11}^i цикловые ключи (i – это номер итерации) формируются из “слов” развернутого ключа K_{ext} , по следующему правилу:

$$W_j = \begin{cases} W_{j-1} \oplus W_{j-N_k}, \\ W_{j-N_k} \oplus SubByte(ROL(W_{j-1})) \oplus RCon_i, \quad j \equiv 0 \pmod{N_k} \end{cases} \quad (1)$$

где: $SubByte$ – байтовая подстановка;

ROL – циклический сдвиг;

$RCon$ – константы.

При использовании ключа длиной $l_k = 256$ бит, “слова” цикловых подключей формируются с использованием дополнительного нелинейного преобразования для элементов расширенного ключа K_{ext} :

$$W_{j-N_k} \oplus SubByte(W_{j-1}) \oplus RCon_i, \quad j \equiv 4 \pmod{N_k} \quad (2)$$

Анализируя алгоритм выработки байт цикловых подключей, представленный формулами (1) и (2), видим, что схема разворачивания ключей FIPS-197 не относится к линейным [5], однако существует сильная зависимость между некоторыми байтами цикловых ключей, которые определяются выражением: $W_j = W_{j-1} \oplus W_{j-N_k}$. Именно эти связи были выявлены при статистическом исследовании, результаты которого приведены в табл. 1 – 3.

В проведенных исследованиях было установлено, что некоторые биты исходного ключа K оказывают воздействие на группы бит развернутого ключа различной степени. Сделаем обоснование этого факта для схемы разворачивания ключей с длиной ключа $l_k = 128$ бит. Периодичность изменения бит в различных “словах” $W_1 - W_{44}$ (рис. 4–7) объясняется особенностями выработки цикловых ключей, что следует из формул (1) и (2). Скачкообразность появления “лавинного” эффекта “словах” наблюдается там, где суммируются “слова” в которых количество бит максимально изменилось. Изменяя биты в “слове” W_1 (1–32 бит) получим максимальный лавинный эффект в “словах” с индексами W_{1+N_k} , то есть W_5, W_9, W_{13}, W_{17} и т.д. При статистическом анализе было установлено, что старшие биты исходного ключа (97–128 бит) оказывают большее влияние на биты во всех “словах” по сравнению с другими битами исходного ключа (рис. 7). Этот факт объясняется тем, что данные биты расположены в “слове” W_4 . Данный элемент массива кратен длине исходного ключа ($N_k = 4$), в связи с чем биты этого “слова” проходят через ряд преобразований $SubByte$, $RotByte$ и сложение с константами. Комбинирование нелинейных ($SubByte$) и линейных ($RotByte$) операций приводит к значительным изменениям в выходных битах “слова” при изменении уже одного бита. А так как каждое “слово” зависит от предыдущего, то размножение ошибок в этом случае будет иметь максимальный эффект в остальных “словах” W_i развернутого ключа K_{ext} .

Для разворачивания ключа длиной $l_k = 192$ бита биты исходного ключа размещаются в $W_1 - W_6$ словах развернутого ключа K_{ext} , а для $l_k = 256$ бит – в “словах” $W_1 - W_8$. Разделение бит исходного ключа K на группы по 32 бита, объединенные по признаку одинакового воздействия на биты развернутого ключа K_{ext} , объясняется тем, что выработка “слов” W_i развернутого ключа K_{ext} аналогична, как и для алгоритма с длиной ключа $l_k = 128$ бит. Дополнительное нелинейное преобразование ($SubByte$), используемое для разворачивания ключей $l_k = 256$ бит, ускоряет процесс нарастания изменений в “словах” развернутого ключа K_{ext} .

Заключение

На примере исследования схемы разворачивания ключей FIPS-197 показано, каким образом влияет связь между выявленными статистическими связями между битами цикловых ключей, либо битами цикловых ключей и секретного ключа, на предрасположенность к проведению атак на схему разворачивания ключей. Действительно, линейность выработки цикловых ключей в FIPS-197 делает возможным осуществление таких атак. Обзор атак на FIPS-197 проведенный в [11] показывает, что атаки, использующие недостатки схемы разворачивания ключей, являются наиболее эффективными. Одним из недостатков, указанных для FIPS-197 при проведении конкурса NESSIE – это низкая граница безопасности (всего 2 цикла для ключа длиной $l_k = 128$ бит). В связи с чем рекомендовано использование стандарта шифрования с ключами длиной $l_k = 192$ и $l_k = 256$ бит, для которых используется большее количество циклов шифрования – 12 (для $l_k = 192$ бита) и 14 (для $l_k = 256$ бит). Очевидным решением этой проблемы может стать разработка в будущем улучшенной схемы разворачивания ключей, что может быть актуальным, если FIPS-197 будет принят в качестве национального стандарта Украины.

Используемая в работе методика проведения исследований может служить в качестве шаблона для испытаний современных схем разворачивания ключей.

Также, из работы вытекает важность разработки схем разворачивания ключей, которые формируют цикловые подключи нелинейным способом. Такой подход, при разработке алгоритма разворачивания ключей, может быть универсальным способом защиты от атак на схему разворачивания ключей.

Список литературы: 1. *Biham E.* New Types of Cryptanalytic Attack Using Related Keys // *J. of Cryptology*, Vol. 7, 1994, pp. 229-246. 2. *Biryukov A., Wagner D.* Slide Attack // *FSE'99*, LNCS 1636, Springer-Verlag, 1999, pp. 245-259. 3. *van Oorschot P.C., Wiener M.* Improving Implementable Meet-In-The-Middle Attack by Order of Magnitude // *Bell-Northern Research*, P.O. Ontario, Canada, 1996. 4. *Kelsey J., Schneier B., Wagner D.* Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER and 3-DES // *CRYPTO'96*, Springer-Verlag, 1996, pp. 237-251. 5. *Knudsen L. R.* Practically secure Feistel ciphers. In R. Anderson, editor, *Fast Software Encryption 1993*, Cambridge Security Workshop (FSE1), Volume 809 of Lecture Notes in Computer Science, 1994. Springer-Verlag. 6. *Aoki K. et al.* Supporting Document on E2 // *Nippon Telegraph and Telephone Corporation*, June 14, 1998. 7. *Pascale Serf* The degrees of completeness, of avalanche effect, and of strict avalanche criterion for MARS, RC6, Rijndael, Serpent and Twofish with reduced number of rounds // *Siemens AG. ZT IK 3*, April 3, 2000. 8. *Горбенко И.Д., Скрыпник Л.В., Головашич С.А., Гриненко Т.А.* Стандарт симметричного шифрования 21 века: свойства, режимы работы, реализация // *Радиотехника. Всеукр. межвед. науч.-техн. сб.* 2001. Вып. 119. С. 22–35. 9. *Heys H. M. and Tavares S. E.* Key Clustering in Substitution-Permutation Network Cryptosystems // *Department of Electrical Engineering Queen's University Kingston, Ontario, Canada*, Dec. 1993. 10. *National Institute of Standards and Technology.* Advanced encryption algorithm (AES) development. // *FIPS 197*, U.S. Department of Commerce, Nov. 2001. 11. *Горбенко И.Д., Лепеха А.Н.* Сравнительный анализ блочных симметричных шифров представленных в проекте NESSIE // *Радиотехника. Всеукр. межвед. науч.-техн. сб.* 2001. Вып. 136. С. 38–53. 12. *Lucks S.* Attacking seven rounds of Rijndael under 192-bit and 256-bit key // *In Proceedings of the Third Advanced Encryption Standard Conference*, NIST, Apr. 2000. [p. 64, 65, 96]. 13. *Ferguson N., Kelsey J., Lucks S., Schneier B., Stay M., Wagner D., and Whiting D.* Improved cryptanalysis of Rijndael // *Fast Software Encryption'2000*, P. 213, 2000. 14. *NESSIE Call for Cryptographic Primitives*, Version 2.2, 8 March 2000: <http://cryptonessie.org>. 15. *Carlisle M. Adams* DES-80 // *Communication Security Establishment Of Canadian Government*, May 1997. 16. *Жильников В.* От папируса до компьютера. М.: АБФ. 1996. 336 с.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 05.12.2004

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СХЕМ РАЗВОРАЧИВАНИЯ КЛЮЧЕЙ БЛОЧНЫХ СИММЕТРИЧНЫХ ШИФРОВ

Условно блочный симметричный шифр можно представить в виде взаимодействия двух алгоритмов: схемы выработки подключей (схема разворачивания ключей) и шифрующего блока. Блок шифрования представляет собой комбинацию криптографических преобразований, объединенных в некоторую функцию F . Процедура зашифрования открытого блока данных X в блок криптограммы Y может быть выражена как отображение $X \rightarrow Y$, которое задается цикловой функцией $F : (GF(2))^n \rightarrow (GF(2))^n$, повторяемой итеративно некоторое количество раз, что определяется требуемыми показателями защищенности от атак на цикловую функцию. Уникальность определяемых функцией F отображений $X \rightarrow Y$ достигается за счёт применения различных цикловых ключей K_j^i , получаемых из исходного секретного ключа K , в результате работы алгоритма разворачивания ключа (процедура выработки подключей). Поэтому свойства получаемой последовательности множества криптограмм $\{Y\}$ могут быть проанализированы через связи (зависимости) между битами цикловых подключей $\{K_j\}$, которые сохраняются в выходной криптограмме Y . Если выявленные зависимости (связи) можно проанализировать и описать математически, то это может служить основой для проведения атаки. Таким образом, способы формирования и использования цикловых подключей при помощи схем разворачивания ключа играют важную роль в криптографической безопасности всего шифра.

Сегодня активно развиваются методы криптоанализа, которые используют особенности формирования цикловых ключей и их использования: «связанные ключи» [1], «скольжения» [2], «встреча посередине» [3], «простые зависимости и эквивалентные ключи» [4]. Исследования современных блочных шифров, проводившиеся в рамках проекта NESSIE [5] показали необходимость совершенствования методов формирования цикловых подключей в связи с прогрессом в развитии методов криптоанализа и вычислительных систем. Причиной сложившейся ситуации является недостаточное внимание со стороны исследователей к проблемам построения схем разворачивания ключей. Как итог – отсутствие четких требований, методов оценки эффективности и критериев проектирования процедур формирования цикловых подключей. Такие исследования особенно актуальны для Украины в связи с отсутствием национального блочного симметричного стандарта шифрования, отвечающего требованиям высокой криптографической стойкости.

В статье приводится способ классификации схем разворачивания ключей блочных симметричных шифров, который может быть использован для анализа существующих и разработки перспективных методов выработки цикловых подключей. Сделан сравнительный анализ схем разворачивания ключей ряда современных блочных симметричных шифров: DES, IDEA, Misty1, RC6, Rijndael, SAFER (SAFER+), Camellia, с целью показать достоинства и недостатки различных методов выработки цикловых подключей. Для оценки эффективности схем разворачивания ключей использовались критерии предоставляемой стойкости к известным криптоаналитическим атакам и тактико-технических показателей.

Для дальнейшего изложения воспользуемся рядом обозначений и определений. Пусть $K = (x_1, x_2, \dots, x_n)$ будет исходный (секретный) ключ (n – разрядность (бит) ключа K , причем $(x_1, x_2, \dots, x_n) \in F_2^n$ биты этого ключа). Тогда $K_{ext} = (x_1, x_2, \dots, x_m)$ будет развернутый ключ, разрядности m бит и полученный как результат работы некоторой схемы разворачивания ключа. Развернутый ключ K_{ext} служит материалом для создания цикловых ключей K_{ij}^i (j – номер итерации). Будем далее рассматривать, что развернутый ключ представляет собой конкатенацию цик-

ловых ключей, и, следовательно: $K_{ext} = K_{it}^1, K_{it}^2, \dots, K_{it}^r$ (r – количество итераций, необходимых для зашифрования / расшифрования одного блока информации). Длина каждого циклового ключа $K_{it}^j = (x_1, x_2, \dots, x_{m/r})$. Схему разворачивания ключей далее будем рассматривать как некоторую функцию φ , которая выполняет отображение множества исходных ключей $\{K^n\}$ во множество развернутых ключей $\{K_{ext}^m\}$: $\varphi: (K)^n \rightarrow (K_{ext})^m$. Множество всех отображений $\{\varphi: K^n \rightarrow K_{ext}^m\}$ биективно соответствует множеству всех систем функций

$$\{f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_m(x_1, x_2, \dots, x_n)\},$$

где $f_i(x_1, x_2, \dots, x_n): F_2^n \rightarrow F_2$, $i = 1, \dots, m$. Далее функции $f_i(x_1, x_2, \dots, x_n)$ будем называть координатными функциями, причем $f_j(x_1, x_2, \dots, x_n): (K^n) \rightarrow K_{ext}^j$ ($j = 1, \dots, m$ – бит развернутого ключа). Тогда отображения, задаваемые процедурой разворачивания ключей, представляют совокупность булевых функций, каждая из которых вычисляет значение одного выходного бита развернутого ключа как функцию от некоторой совокупности входных аргументов (битов) личного ключа $K = (x_1, x_2, \dots, x_n)$.

Анализ схем разворачивания ключей современных блочных симметричных шифров, представленных на конкурсе AES и в проекте NESSIE, позволил сделать классификацию схем разворачивания ключей (рис.1).

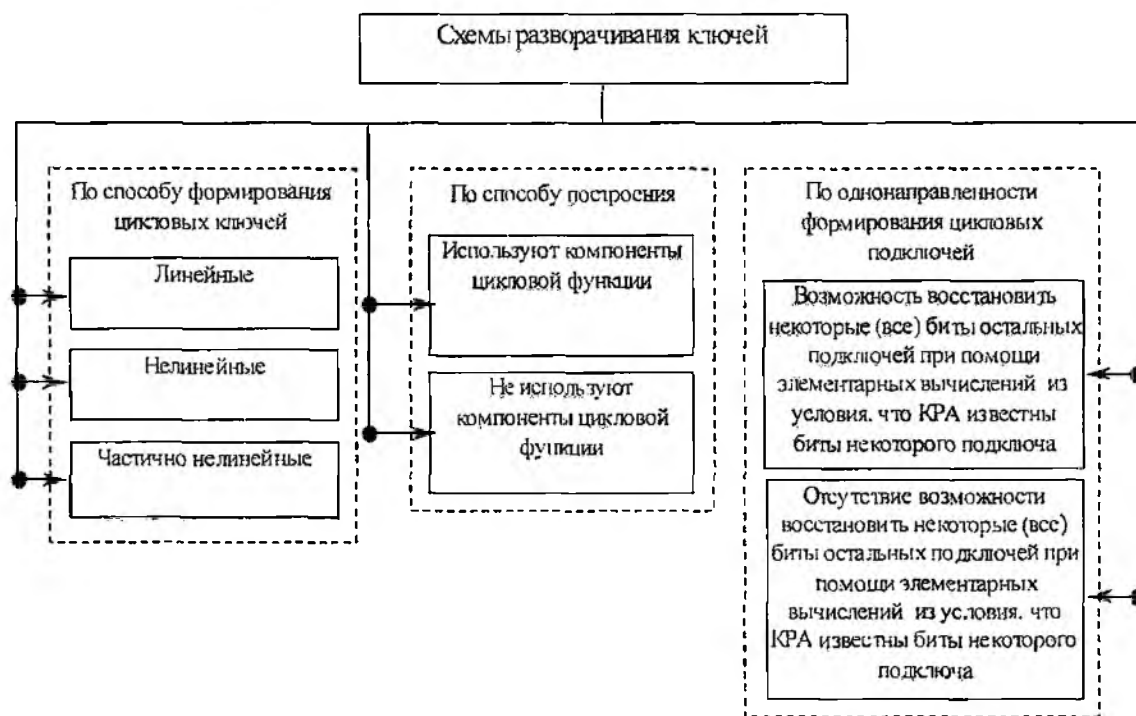


Рис. 1

Схемы разворачивания ключей далее будем делить на три больших класса, по способу выработки цикловых ключей:

Тип 1: Линейные схемы разворачивания ключей. К таким методам формирования цикловых подключей будем относить алгоритмы, которые задают отображение $\varphi: (K)^n \rightarrow (K_{ext})^m$, где биты развернутого ключа K_{ext} , есть координатные функции $f_i(x_1, x_2, \dots, x_n)$, являющиеся

линейными многочленами, то есть $f_i(x_1, x_2, \dots, x_n) = \bigoplus_{i=1}^m a_i x_i$ (a – ненулевой элемент поля F_2^n , m – разрядность развернутого ключа).

Тип 2: Нелинейные схемы разворачивания ключей. Цикловые ключи формируются как результат некоторой функции φ , выполняющей нелинейное отображение всех битов исходного ключа в биты цикловых ключей: $\varphi: (K)^n \rightarrow (K_{ext})^m$. Такие схемы будем называть «нелинейными». Причем отображение φ считается «нелинейным», если все его координатные функции $f_i(x_1, x_2, \dots, x_n)$ являются нелинейными.

Тип 3: Частично нелинейные схемы разворачивания ключей. Такие алгоритмы формируют развернутый ключ, биты которого есть координатные функции, некоторые биты которых имеют линейную зависимость от битов исходного ключа, а другие биты – нелинейную, то есть некоторые координатные функции $f_i(x_1, x_2, \dots, x_n)$ зависят от некоторой переменной (некоторых переменных) x_i линейно: $f_i(x_1, x_2, \dots, x_n) = a \cdot x_i \oplus \psi(y)$ (где y – это исходный ключ K без x_i переменной, вектор $a \in F_2^n$).

С учетом конструктивных особенностей можно провести классификацию по способу построения. К схемам разворачивания ключей, которые не используют базовых компонентов схемы шифратора, относятся все схемы разворачивания ключей типа 1 (линейные). Это объясняется тем, что цикловые ключи формируются по некоторому линейному закону (перестановка, сложение с константами). Схемы разворачивания ключей типа 2 [6, 7], как правило, используют различные нелинейные преобразования, в качестве которых применяются различные элементы цикловой функции. В некоторых случаях (например, RC6) могут использоваться комбинации математических линейных и нелинейных операций. Конструктивные особенности схемы разворачивания ключей оказывают влияние на затраты при реализации, что важно при аппаратном исполнении.

По характеру элементарных преобразований и способу их объединения в составе схемы разворачивания ключей алгоритмы формирования цикловых подключей условно можно разделить на две группы: на базе «легко» обратимых преобразований; на базе однонаправленных (либо «трудно» обратимых) преобразований.

«Легко» обратимыми будем называть преобразования φ_k , для которых обратное $\gamma_k = \varphi_k^{-1}$ имеет вычислительную сложность того же порядка, что и φ_k . Однонаправленными, в данном контексте, будем называть преобразования, не удовлетворяющие определению легко обратимых. Очевидно, что если криптоаналитик не имеет вычислительно «простого» способа определить биты цикловых ключей или исходного ключа по некоторым известным (полученным в ходе какой-либо успешной атаки), то это является положительным качеством схемы разворачивания ключей. Однонаправленность формирования битов развернутого ключа будем рассматривать как косвенный показатель безопасности.

Выясним преимущества и недостатки различных типов схем разворачивания ключей. Для этого проанализируем ряд блочных симметричных шифров, представленных в проекте NESSIE: DES, IDEA, Misty1, RC6, Rijndael, SAFER (SAFER+), Camellia, которые имеют различные типы схем разворачивания ключей, в соответствии с приведенной классификацией. К первому типу (линейные) относятся схемы разворачивания ключей блочных симметричных шифров DES, IDEA, SAFER (SAFER+). Ко второму (нелинейные) типу можно отнести RC6, Misty1. К третьему типу (частично нелинейные) относятся схемы разворачивания ключей криптоалгоритмов Camellia и Rijndael.

Сравнительный анализ схем разворачивания ключей указанных алгоритмов будем проводить по криптографической стойкости к известным методам криптоанализа, по сложности восстановления цикловых подключей либо исходного ключа при условии, что криптоаналитику известны биты других цикловых подключей, по скорости формирования ключей.

Сравнение указанных шифров по возможности реализации атак, связанных с особенностями выработки цикловых ключей, приведено в табл. 1.

Таблица 1

Криптоалгоритм	Тип схемы разворачивания ключей	Циклы	Техника	Время	Данные
3-DES ₆₄	линейная	все	2 ²⁸ связанных ключей «Связанные ключи»	2 ⁸⁴	2 ²⁸
DES ₆₄	линейная	все	«Встреча по середине» «Скольжения» «простые связи» (свойство дополнения)	1.3*2 ¹⁰⁴ 2 ⁵⁵	2 ³² 128
IDEA ₆₄	линейная	4,5 2,5 4 8,5	«Встреча посередине» «Связанные ключи» / SQUIRE 2 ¹⁷ связанных ключей «Связанные ключи» / дифференциально-линейный (2 связанных ключа) «Слабые ключи» 2 ⁵⁴ -2 ⁶⁴ «Бумеранг» используя 2 ⁶⁴ слабых ключа	2 ¹¹² 2 ³² 38,8 2 ¹⁶	2 ⁶⁴ 2 38 2 ¹⁶
SAFER _{64, 128} SAFER+ _{64, 128} SAFER++ _{64, 128}	Линейная		«Связанные ключи» (2 ⁸ связанных ключа) «Связанные шифры» «Связанные ключи» (2 связанных ключа) —	2 ⁴⁴ 2 ²⁰⁰	2 ⁴⁴ 2 ²⁹ 3 2 ³²
Mistyl ₆₄	нелинейная		—		
RC6 ₁₂₈	нелинейная		—		
Rijndael _{128, 192, 256}	Частично нелинейная	9\7	«Связанные ключи» / Square 256 связанных ключей. ks/Square	2 ²²⁴ —	2 ⁷⁷ —
Camellia _{128, 192, 256}	Частично нелинейная		—	—	—

Сравним схемы разворачивания ключей указанных выше шифров по возможности восстановить часть битов исходного ключа или других цикловых ключей при условии, что криптоаналитик определил биты какого-либо циклового ключа. Определив биты цикловых ключей для блочных симметричных шифров DES, IDEA, SAFER (линейные схемы разворачивания ключей), криптоаналитик тем самым определяет биты исходного ключа и части битов других цикловых ключей. Очевидно, что таким образом сокращается пространство поиска остальных бит личного ключа путем перебора. Для алгоритма Rijndael по битам одного циклового ключа криптоаналитик может восстановить все остальные цикловые подключи путем простых вычислений. В блочном симметричном шифре Camellia криптоаналитик может восстановить либо часть бит других цикловых ключей, либо часть бит секретного ключа, в зависимости от того, биты какого циклового ключа удалось определить. В шифрах с нелинейными схемами разворачивания ключей Mistyl и RC6 криптоаналитик не имеет простого способа восстановить биты других цикловых ключей или секретного ключа.

Сравним рассматриваемые шифры по скоростным показателям. Для расчетов будем использовать количество тактов процессора для PIII/Windows 2000 затрачивается на разворачивание ключа (табл. 2).

Таблица 2

Криптоалгоритм	Длина развернутого ключа (исходный ключ), бит	Время, тактов процессора на ключ	Время, тактов процессора на байт
DES ₆₄	768 (56)	929	132,7
IDEA ₆₄	832 (128)	395	24,6
Misty1 ₆₄	512 (128)	125	7,8
SAFER ₆₄	1152 (128)	1692	105,7
SAFER ₁₂₈	1920, 2688 (128, 256)	1464; 1865	91,5; 116,5
RC6 ₁₂₈	1408 (128, 192, 256)	1040, 2856, 1814	65; 119; 56,6
Rijndael ₁₂₈	1408 (128, 192, 256)	497, 552, 775	31; 23; 24,2
Camellia ₁₂₈	1664, 2176 (128, 192, 256)	597, 1797, 1805	37,3; 74,8; 56,4

Комментируя полученные результаты для блочных симметричных шифров с линейными схемами разворачивания ключей, можно сформулировать следующие достоинства и недостатки.

Достоинства:

1. Простота реализации. При формировании цикловых подключей используются элементарные математические операции над битами исходного ключа (циклические сдвиги сложения с константами), которые легко и с одинаковой сложностью реализуются как при программной, так и при аппаратной реализации с приблизительно одинаковой сложностью.

2. Время разворачивания ключа ниже, чем у нелинейных схем разворачивания ключей. Однако это справедливо для БСШ, в которых цикловые подключи формируются путем простого выбора битов исходного ключа (ГОСТ 28147-89 или IDEA).

Недостатки:

1. Предрасположенность к атакам на схемы разворачивания ключей. В связи с этим, безопасность таких шифров зависит от цикловой функции, а именно от количества используемых итераций, порядка ввода цикловых подключей в цикловую функцию на итерациях, цикловая функция должна обладать высокими криптографическими свойствами.

2. Возможность легко восстанавливать биты других цикловых подключей или исходного секретного ключа по известному (-ным) цикловым подключам.

Нелинейные схемы разворачивания ключей характеризуются следующими достоинствами и недостатками:

Достоинства:

1. Сложность построения атак на схему разворачивания ключей.

2. Сложность восстановления битов цикловых подключей (секретного ключа) по известным цикловым ключам.

3. Возможность использования в качестве нелинейных элементов компоненты цикловой функции, что позволяет сократить затраты на реализацию. Другой положительной стороной является уменьшение времени разворачивания ключа за счет сокращения количества операций.

Недостатки:

1. Сложность реализации.

2. Малые показатели скорости по сравнению с линейными схемами разворачивания ключей.

Таким образом, учитывая, что обеспечение криптографической стойкости является всегда самым важным требованием, необходимо говорить о развитии нелинейных конструкций для построения схем разворачивания ключей. Причем, как показывают результаты в табл. 1, использование конструкций линейно-нелинейного типа позволяет достичь компромисса между безопасностью, скоростными показателями и сложностью реализации таких схем на процессорах различной разрядности, однако существование атак на такие конструкции не исключается (Rijndael). Очевидно, что целесообразным является развитие нелинейных конструкций. Такие конструкции имеют значительное превосходство в обеспечении криптографической стойкости.

учетом того, что такие конструкции не имеют значительного проигрыша в скорости разворачивания ключей. В качестве направления исследований и развития нелинейных схем разворачивания ключей можно определить использование в качестве процедуры разворачивания ключа одностороннего генератора псевдослучайных последовательностей. При этом желательно, чтобы каждый бит «развёрнутой» выходной последовательности зависел от всех битов исходного пользовательского ключа. Под псевдослучайностью здесь понимается непредсказуемость последующих и предыдущих элементов последовательности рабочих подключей по текущему элементу. Под односторонностью понимается высокая вычислительная сложность восстановления состояния генератора (и соответственно неизвестных элементов последовательности) по некоторому множеству известных элементов выходной последовательности. Такое требование, в большинстве случаев реализации блочных симметричных шифров, является избыточным, однако является универсальным способом защиты от атак на схему разворачивания ключа и, кроме того, позволяет построить безопасную (устойчивую к коллизиям) хеш-функцию.

Список литературы: 1. *Bihani E.* New Types of Cryptanalytic Attack Using Related Keys // J. of Cryptology, Vol. 7, 1994, p. 229-246. 2. *Biryukov A., Wagner D.* Slide Attack // FSE'99, LNCS 1636, Springer-Verlag, 1999, p. 245-259. 3. *van Oorschot P.C., Wiener M.* Improving Implementable Meet-In-The-Middle Attack by Order of Magnitude // Bell-Northern Research, P.O., Ontario, Canada, 1996. 4. *Kelsey J., Schneier B., Wagner D.* Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER and 3-DES // CRYPTO'96, Springer-Verlag, 1996, p. 237-251. 5. *NESSIE Call for Cryptographic Primitives*, Version 2.2, 2000: <http://cryptonessie.org>. 6. *Knudsen L. R.* Practically secure Feistel ciphers. In R. Anderson, editor, Fast Software Encryption 1993, Cambridge Security Workshop (FSE1), Volume 809 of Lecture Notes in Computer Science, p. 211. 1994. Springer-Verlag. 7. *Aoki K et.al.* Supporting Document on E2 // Nippon Telegraph and Telephone Corporation, June 14, 1998.

*Харьковский национальный
университет радиозлектроники*

Поступила в редколлегию 30.11.2004

СОВРЕМЕННЫЕ ПОДХОДЫ К ОЦЕНКЕ ЛИНЕЙНОЙ СЛОЖНОСТИ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ. УСОВЕРШЕНСТВОВАННАЯ МЕТОДИКА ТЕСТИРОВАНИЯ ЛИНЕЙНОЙ СЛОЖНОСТИ НА ОСНОВЕ NIST STS

Линейная сложность является одним из основных и наиболее важных показателей при изучении и оценке криптографических свойств поточных шифров и шифрообразующих устройств в целом. Этот показатель должен в первую очередь учитываться при вынесении окончательного решения относительно использования псевдослучайной последовательности (генератора) в криптографических целях.

Как известно, линейной сложностью конечной двоичной последовательности называется длина L кратчайшего линейного рекуррентного регистра (ЛРР), который генерирует символы данной последовательности в качестве первых элементов выхода. Это определение исходит из того факта, что существует эффективная процедура – алгоритм Берлекампа-Мэсси [2], для нахождения самого короткого ЛРР, генерирующего данную последовательность, после наблюдения потока длиной $2L$ бит.

Использование алгоритма Берлекампа-Мэсси дает возможность реализовать универсальную криптоаналитическую атаку на генераторы гамм, поскольку позволяет заменить любой шифргенератор его кратчайшим линейным эквивалентом. Большое значение линейной сложности последовательности шифрующей гаммы – это необходимое (но не достаточное) условие для практической стойкости всякого аддитивного поточного шифра и любого шифрообразующего устройства, а оценка линейной сложности является актуальной задачей как в практическом, так и в теоретическом плане.

В настоящее время при разработке поточных шифров на основе линейных рекуррентных регистров широкое распространение получило использование регистров в двоичных расширенных полях $GF(2^m)$. В качестве примеров можно назвать такие современные шифры, как Snow [3, 4], Turing [5], последние модификации шифров семейства SOBER [6, 4].

Известно [1, 7], что, если ЛРР, функционирующий в простом поле, имеет в качестве функции обратной связи примитивный полином степени L , то при любом ненулевом начальном состоянии этот ЛРР будет формировать выходную последовательность, линейная сложность которой будет равна L .

Исследования показали, что значение линейной сложности выходных последовательностей ЛРР, функционирующих в расширенных полях, превышает показатель степени образующего полинома. В связи с этим возникает необходимость адекватно оценить сложность выходных последовательностей таких регистров, поскольку от результатов анализа свойств базовых ЛРР зависит оценка стойкости шифрообразующего устройства в целом.

Известные подходы к оценке линейной сложности линейных рекуррентных регистров посвящены регистрам над простыми полями. В то же время какие-либо сведения, касающиеся исследований сложности ЛРР в расширенных полях, в открытой литературе отсутствуют.

Учитывая возрастающий интерес к свойствам регистров с линейной обратной связью, функционирующих в расширенных полях, в настоящей работе ставится цель разработать подходы к оценке сложности ЛРР такого типа. Выводится расчетное соотношение для определения конкретного значения показателя линейной сложности и обосновываются предложения по совершенствованию существующей методики тестирования регистров над расширенными полями.

1 Основные определения и свойства линейной сложности

Кратко напомним сначала некоторые понятия и определения теории линейной сложности, которые потребуются нам в дальнейшем.

Объектом рассмотрения будут последовательности символов на выходе некоторого генератора (автомата). Пусть s обозначает бесконечную последовательность, символы

которой $s_0, s_1, s_2, \dots$. Конечную последовательность, состоящую из первых n элементов, будем обозначать через $s^n = s_0, s_1, \dots, s_{n-1}$.

Считается, что ЛПР *генерирует* последовательность s , если существует некоторое начальное состояние, при котором на выходе регистра появится s [1]. В частном случае утверждается, что ЛПР *генерирует* конечную последовательность s^n , если существует некоторое начальное состояние, при котором на выходе данного регистра появится данная последовательность.

В дальнейшем будем рассматривать последовательности, элементами которых являются 0 и 1.

Согласно работы [1], линейная сложность бесконечных двоичных последовательностей s обозначается $\Lambda(s)$ и имеет следующие свойства:

- 1) если s – нулевая последовательность $s = 0, 0, 0, \dots$, то $\Lambda(s) = 0$;
- 2) если никакой ЛПР не может генерировать s , то $\Lambda(s) = \infty$;
- 3) во всех остальных случаях, $\Lambda(s)$ – длина кратчайшего ЛПР, который генерирует s .

Линейной сложностью конечной двоичной последовательности s^n называется длина L самого короткого ЛПР, который генерирует последовательность s^n в качестве n первых элементов выхода [1]. В соответствии с принятыми выше обозначениями будем записывать ее как $\Lambda(s^n)$.

В работе [2] предложено эквивалентное определение: *линейная сложность* $\Lambda(s^n)$ – это наименьшее неотрицательное целое число L такое, что существует линейная рекуррента с фиксированными константами $c_0, c_1, \dots, c_L$, удовлетворяющая равенству

$$s_j + c_1 s_{j-1} + \dots + c_L s_{j-L} = 0, (1)$$

где $L \leq j \leq 1$, выходные символы и коэффициенты обратной связи $c_0, c_1, \dots, c_L$ принадлежат некоторому конечному либо бесконечному полю.

Известно [1], что линейная сложность двоичных последовательностей обладает следующими свойствами. Если s и t – двоичные последовательности, то:

- 1) для любого $n \geq 1$ линейная сложность подпоследовательности s^n удовлетворяет $0 \leq \Lambda(s^n) \leq n$;
- 2) $\Lambda(s^n) = 0$ тогда и только тогда, когда s^n является нулевой последовательностью длиной n ;
- 3) $\Lambda(s^n) = n$ тогда и только тогда, когда $s^n = 0, 0, 0, \dots, 0, 1$;
- 4) если s – периодическая последовательность с периодом N , тогда $\Lambda(s) < N$;
- 5) $\Lambda(s \oplus t) \leq \Lambda(s) + \Lambda(t)$, где $s \oplus t$ обозначает побитный XOR s и t .

Как известно, высокая линейная сложность является необходимым, но не достаточным, условием для применения последовательности в криптографических целях. Так, например, свойство 3 показывает, что линейная сложность периодической последовательности с периодом n , состоящей из одной 1, остальные биты которой равны 0, будет равна n , поскольку никакое линейное соотношение более короткое, чем $s_{i+n} = s_i$, для любого $i \geq 0$ не будет удовлетворять каждому биту последовательности. Однако, очевидно, что такая вырожденная последовательность не может использоваться в качестве ключевого потока, так как $n-1$ бит равны нулю.

В середине 1980-х годов Райнером Рюппелем было введено понятие профиля линейной сложности последовательности. Если $s' = s_0, s_1, \dots, s_{i-1}$ и $\Lambda(s^n)$ – линейная сложность подпоследовательности $s^n = s_0, s_1, \dots, s_{n-1}$, $0 \leq n \leq i$, тогда последовательность $\Lambda_1(s'), \Lambda_2(s'), \dots, \Lambda_n(s')$ называется *профилем линейной сложности* последовательности s' [10]. Заметим, что использование этого понятия позволяет описать и выявить последовательности вырожденного типа. Установлено [11], что график зависимости профиля линейной сложности от длины последовательности является неубывающим и близким к значению $n/2$. Свойства профиля линейной сложности и примеры его построения можно найти в [1].

В соответствии с результатами этой же работы [11] математическое ожидание линейной сложности последовательности $s^n = s_0, s_1, \dots, s_{n-1}$, состоящей из n независимых и равномерно распределенных двоичных случайных величин, определяется соотношением

$$E[\Lambda(s^n)] = \frac{n}{2} + \frac{4 + R_2(n)}{18} - \frac{1}{2^n} \left(\frac{n}{3} + \frac{2}{9} \right), \quad (2)$$

где $R_2(n)$ обозначает остаток от деления n на 2.

Это соотношение позволяет установить, что значение линейной сложности случайно выбранной последовательности s^n ожидается близким к $n/2$. Для больших значений n

$$E[\Lambda(s^n)] \cong \frac{n}{2} + \frac{4 + R_2(n)}{18}, \quad n \gg 1. \quad (3)$$

Здесь $E[\Lambda(s^n)]$ отличается от $n/2$ только сдвигом на $2/9$ в случае, если n четное, и на $5/18$ в случае, если n нечетное.

Дисперсия линейной сложности последовательности $s^n = s_0, s_1, \dots, s_{n-1}$, состоящей из n независимых и равномерно распределенных двоичных случайных величин, определяется соотношением

$$\text{Var}[\Lambda(s^n)] = \frac{86}{81} - \frac{1}{2^n} \left(\frac{14 - R_2(n)}{27} n + \frac{82 - 2R_2(n)}{81} \right) - \frac{1}{2^{2n}} \left(\frac{1}{9} n^2 + \frac{4}{27} n + \frac{4}{81} \right), \quad (4)$$

где $R_2(n)$, как и ранее, обозначает остаток от деления n на 2. Для больших значений n

$$\lim_{n \rightarrow \infty} \text{Var}[\Lambda(s^n)] = \frac{86}{81}. \quad (5)$$

то есть, является константой, близкой к 1.

В [11] Рюппель также показывает, что значение математического ожидания $E[\Lambda(\tilde{z})]$ линейной сложности $\Lambda(\tilde{z})$ периодически повторяющейся случайной двоичной последовательности $\tilde{z}$, для двух исключительных случаев, когда период $T = 2^n$, где $n \geq 0$, и когда $T = 2^n - 1$, где n – простое число, почти равно периоду T . В то же время он предположил, что значение $E[\Lambda(\tilde{z})]$ будет близким к T в общем случае. В [12] Даи и Янг вычислили пределы для $E[\Lambda(\tilde{z})]$ и дисперсии $\text{Var}[\Lambda(\tilde{z})]$ для всех возможных значений T . Они показали, что значение математического ожидания линейной сложности случайной непериодической последовательности близко к периоду, что, в частности, подтвердило предположение Рюппеля.

Приведенные соотношения будут использованы в дальнейшем при исследовании показателей сложности ЛРР над расширенными полями.

2 Оценка линейной сложности по методике NIST

Согласно методике, предложенной в статистическом наборе NIST STS, основу теста составляет вычисление длины линейного рекуррентного регистра, способного сгенерировать заданную последовательность. Главная задача – определить, является ли последовательность достаточно сложной, чтобы принять ее в качестве случайной. Очевидно, что случайные последовательности характеризуются более длинными эквивалентными ЛРР, слишком короткие ЛРР предполагают неслучайность.

В соответствии с [8], в процессе тестирования последовательность $s^n = s_0, s_1, \dots, s_{n-1}$ длиной n разбивается на N независимых блоков по M бит, где $n = MN$. При помощи алгоритма Берлекампа-Мэсси, вычисляется линейная сложность $\Lambda_i(s^M) = L_i$ каждого из N блоков ($i = 1, \dots, N$). L_i – i -м длина самого короткого линейного рекуррентного регистра последовательности, который генерирует все биты в i -м блоке. Внутри любой L_i -битовой последовательности некоторая комбинация бит после сложения по модулю 2, порождает следующий бит в последовательности ($L_i + 1$ бит) (см. уравнение (1)).

Пример. Пусть $M = 13$, а тестируемый блок имеет значения: 1101011110001. Тогда при помощи алгоритма Берлекампа-Мэсси можно установить, что $L_i = 4$, а последовательность формируется путем сложения 1 и 2-го бит внутри текущей 4-битовой подпоследовательности, в результате чего формируется следующий (5-й) бит. Процесс формирования такой последовательности происходит следующим образом:

	1 бит	2 бит	3 бит	4 бит	5 бит
Первых 4 бита и результирующий 5-й бит:	1	1	0	1	0
2-5 бит и результирующий 6-й бит:	1	0	1	0	1
3-6 бит и результирующий 7-й бит:	0	1	0	1	1
⋮	1	0	1	1	1
⋮	0	1	1	1	1
⋮	1	1	1	1	0
⋮	1	1	1	0	0
⋮	1	1	0	0	0
9-12 бит и результирующий 13-й бит:	1	0	0	0	1

Для случайной последовательности из M бит, в соответствии с (2) может быть вычислено теоретическое значение математического ожидания $E[\Lambda(s^M)] = \mu$.

Так, для случая, рассмотренного в примере, $\mu = \frac{13}{2} + \frac{(9 + (-1)^{13+1})}{36} - \frac{(13/3 + 2/9)}{2^{13}} = 6.777222$.

Далее для каждой подстроки вычисляется промежуточное значение статистической характеристики критерия проверки T_i , где $T_i = (-1)^M (L_i - \mu) + 2/9$.

Например, $T_i = (-1)^{13} (4 - 6.777222) + 2/9 = 2.999444$.

В методике NIST, статистическая характеристика критерия проверки нулевой гипотезы (в данном случае, предполагается, что последовательность случайна) рассматривается как функция случайных величин, которая подчиняется известному закону распределения. Ввиду дискретного характера этого распределения и ограниченности дисперсии (см. выражение (5)), случаи, при которых M четное и M нечетное (одно из них принимает только отрицательные значения), рассматриваются отдельно для двух различных предельных распределений.

В соответствии с методикой определяются частоты, $v_0, v_1, \dots, v_K$ попаданий значений T_M для каждой i -й подстроки длиной M в один из выбранных $K+1$ интервалов. Для достаточно больших значений M и N определены следующие интервалы ($K=6$): $\{T \leq -2.5\}$, $\{-2.5 < T \leq -1.5\}$, $\{-1.5 < T \leq -0.5\}$, $\{-0.5 < T \leq 0.5\}$, $\{0.5 < T \leq 1.5\}$, $\{1.5 < T \leq 2.5\}$ и $\{T > 2.5\}$.

Затем значения T_i отображаются в $v_0, \dots, v_6$ следующим образом:

Если: $T_i \leq -2.5$ Увеличение v_0 на 1
 $-2.5 < T_i \leq -1.5$ Увеличение v_1 на 1
 $-1.5 < T_i \leq -0.5$ Увеличение v_2 на 1
 $-0.5 < T_i \leq 0.5$ Увеличение v_3 на 1
 $0.5 < T_i \leq 1.5$ Увеличение v_4 на 1
 $1.5 < T_i \leq 2.5$ Увеличение v_5 на 1
 $T_i > 2.5$ Увеличение v_6 на 1

В результате формируется массив v_i .

На следующем этапе, в соответствии со значениями, полученными для $v_0, \dots, v_6$, вычисляется статистическая характеристика критерия проверки гипотезы случайности $\chi^2(obs)$ – мера соответствия наблюдаемого количества появлений ЛРР фиксированной длины ожидаемому количеству появлений при условии случайности. Опорное распределение статистической характеристики критерия – распределение χ^2 с K степенями свободы. Значение $\chi^2(obs)$ вычисляется по формуле:

$$\chi^2(obs) = \sum_{i=0}^K \frac{(v_i - N\pi_i)^2}{N\pi_i},$$

где K – количество степеней свободы, $K=6$;

π_i – теоретические вероятности промежуточных характеристик, соответствующие выбранным интервалам, $\pi_0 = 0.01047$, $\pi_1 = 0.03125$, $\pi_2 = 0.125$, $\pi_3 = 0.5$, $\pi_4 = 0.25$, $\pi_5 = 0.0625$, $\pi_6 = 0.02078$.

По статистической характеристике критерия проверки вычисляется соответствующее значение вероятности P :

$$P = \frac{\int_{\chi^2(obs)}^{\infty} e^{-u/2} u^{K/2-1} du}{\Gamma(K/2) 2^{K/2}} = \text{igamc}\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right),$$

где igamc – неполная гамма-функция $Q(a, x) \equiv 1 - P(a, x) \equiv \frac{\Gamma(a, x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^{\infty} e^{-t} t^{a-1} dt$, причем

$Q(a, 0) = 1$ и $Q(a, \infty) = 0$.

Значение вероятности P есть вероятность того, что статистическая характеристика критерия примет значение большее, чем наблюдаемое в опыте со случайными последовательностями. Характеристика критерия проверки гипотезы случайности строится таким образом, чтобы ее большие значения указывали на какой-либо дефект случайности последовательности. Следовательно, малые значения P (например, $P < 0.01$) интерпретируются как доказательство того, что последовательность неслучайна. Окончательное решение относительно последовательности принимается в соответствии с выбранным уровнем значимости α . Ожидается, что для фиксированного α определенная часть значений P укажет на неудачный исход теста. Например, если выбрать уровень значимости $\alpha = 0.01$, тогда ожидается, что примерно 1% последовательностей тест не пройдет. Считается, что последовательность прошла статистический тест, когда $P \geq \alpha$.

Для успешного проведения проверки согласно данной методике разработчики пакета рекомендуют выбрать длину тестируемой последовательности $n \geq 10^6$. Значение M должно попадать в интервал $500 \leq M \leq 5000$, а $N \geq 200$ для того, чтобы результирующее значение χ^2 было действительным.

3 Экспериментальные результаты тестирования последовательностей, формируемых ЛРР современных шифров

В этом разделе приводятся экспериментальные результаты тестирования выходных последовательностей ЛРР, взятых из современных поточных шифров, которые прошли во второй тур конкурса NESSIE, а именно, Snow, Sober-t16 и Sober-t32. Все указанные схемы являются синхронными поточными шифрами, в основе которых используются регистры, функционирующие в расширенном поле $GF(2^w)$.

В структурной схеме шифрообразующего устройства Snow используется ЛРР длиной 16 над $GF(2^{32})$ с примитивным полиномом обратной связи:

$$p(x) = x^{16} + x^{13} + x^7 + \alpha.$$

Элементы поля $GF(2^{32})$ формируются с использованием неприводимого полинома вида $q(x) = x^{32} + x^{29} + x^{20} + x^{15} + x^{10} + x + 1$ над полем $GF(2)$, причем $q(\alpha) = 0$. Период такой линейной рекурсии равен $2^{16 \cdot 32} - 1 = 2^{512} - 1 \approx 1.3 \cdot 10^{154}$.

Основу схем Sober-t16 и Sober-t32 составляют ЛРР с примитивными полиномами обратной связи над полями $GF(2^{16})$ и $GF(2^{32})$, соответственно. Используемые полиномы:

$$p(x) = x^{17} + \varphi \cdot x^{15} + x^4 + \xi \text{ для Sober-t16,}$$

$$p(x) = x^{17} + x^{15} + x^4 + \xi \text{ для Sober-t32,}$$

где $\varphi = 0xE382$, $\xi = 0x67C3$, $\psi = 0xC2DB2AA3$. Все арифметические операции в поле $GF(2^w)$ выполняются по модулю соответствующих полиномов вида

$$q(x) = x^{16} + x^{14} + x^{12} + x^7 + x^6 + x^4 + x^2 + x + 1, \text{ для Sober-t16,}$$

$$q(x) = x^{32} + (x^{24} + x^{16} + x^8 + 1)(x^6 + x^5 + x^2 + 1), \text{ для Sober-t32,}$$

над $GF(2)$. При этом обеспечивается формирование последовательности с периодом повторения $2^{17 \cdot 16} - 1 = 2^{272} - 1 \approx 7.6 \cdot 10^{81}$ для Sober-t16 и $2^{17 \cdot 32} - 1 = 2^{544} - 1 \approx 5.6 \cdot 10^{163}$ для Sober-t32. Подробное описание схем Snow, Sober-t16 и Sober-t32 можно найти в [4].

Для осуществления проверки описанные выше регистры были реализованы программно. К последовательностям, полученным с выходов ЛРР, применен статистический тест линейной сложности из пакета NIST STS.

Для проведения тестирования выбраны следующие параметры:

1. Длина тестируемой последовательности $n = 10^6$ бит.
2. Количество тестируемых последовательностей $m = 100$, и, таким образом, объем тестируемой выборки составил $10^6 \times 100 = 10^8$ бит.
3. Уровень значимости $\alpha = 0.01$.

В соответствии с рекомендациями разработчиков, оценка линейной сложности осуществлялась на основе определения доверительного интервала, в который должно попасть значение коэффициента r – доли последовательностей, прошедших статистический тест [9]. Коэффициент определялся путем вычисления отношения суммы значений вероятностей P , превышающих уровень значимости, к общему количеству последовательностей m . Границы доверительного интервала определялись согласно выражению

$$r_{\max(\min)} = \bar{p} \pm 3 \sqrt{\frac{\bar{p}(1-\bar{p})}{m}},$$

где $\bar{p} = 1 - \alpha$. Считалось, что последовательность прошла статистический тест, если значение коэффициента r находится внутри доверительного интервала $[r_{\min}, r_{\max}]$. Нижняя граница в этом случае составила $r_{\min} = 0.96015$.

Окончательное решение о прохождении последовательностью статистического теста принималось после анализа распределения значений P , которое должно соответствовать равномерному закону распределения. Равномерность определялась путем вычисления статистической характеристики критерия χ^2 и вычисления значения вероятности P_0 , которое соответствует распределению характеристики критерия согласия, по значениям P (для разных блоков последовательности), полученным в результате статистического теста (то есть, вычислялось “общее” значение P_0 по всем значениям P , полученным в результате тестирования всех блоков последовательности). Если $P_0 \geq 0.0001$, то считалось, что последовательность равномерно распределена.

В табл. 1 представлены сводные результаты по прохождению регистрами теста линейной сложности по правилу доверительного интервала и приведены “общие” значения вероятности P_0 .

Таблица 1

Регистр	Количество последовательностей, не прошедших тестирование	Доля последовательностей, прошедших статистический тест	Значение вероятности P_0
ЛРР(Snow)	1	0.99	0.779188
ЛРР(Sober-t16)	3	0.97	0.350485
ЛРР(Sober-t32)	2	0.98	0.867692

Из данных табл. 1 видно, что доля последовательностей, прошедших статистический тест, для всех регистров попадает в доверительный интервал, а значения вероятности $P_0 \geq 0.0001$. В результате можно сказать, что все регистры прошли тест для 1% уровня значимости.

На рис. 1-3 графически изображены результаты тестирования каждого ЛРР.

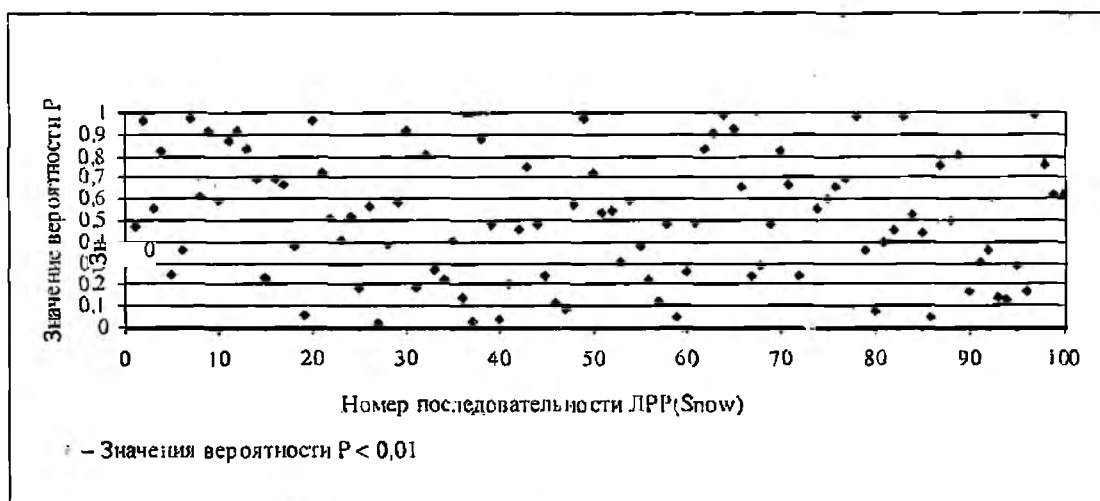


Рис. 1



Рис. 2

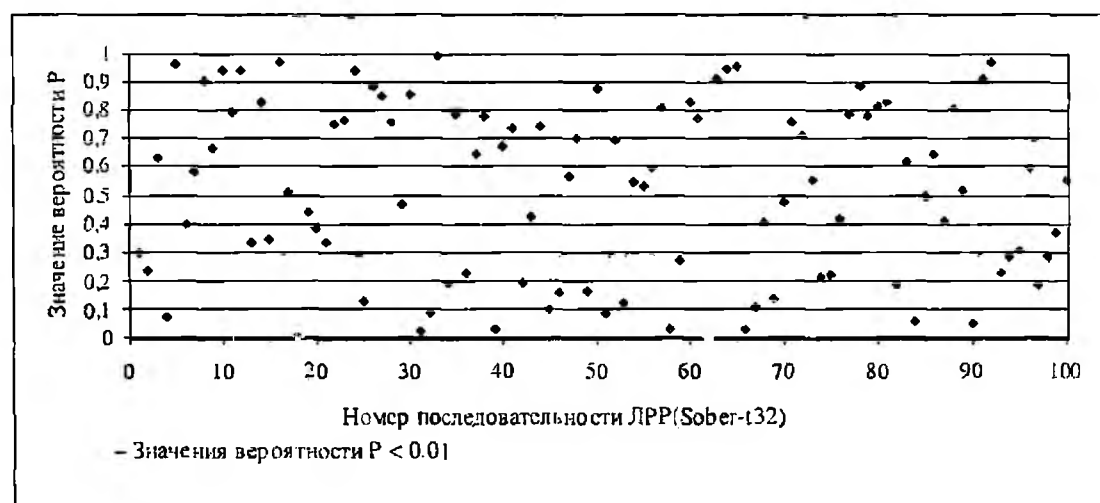


Рис. 3

Приведенные результаты позволяют сделать вывод о том, что линейная сложность ЛРР над расширенными полями существенно превосходит значение линейной сложности регистров, функционирующих в простых полях. Нашей следующей задачей стоит вывод расчетного соотношения для оценки линейной сложности ЛРР над расширенными полями.

4 Линейная сложность ЛРР над расширенными полями

Рассмотрим особенности функционирования линейных рекуррентных регистров в расширенных полях. В отличие от двоичных регистров, оперирующих отдельными битами, регистры, основанные на рекуррентном соотношении над полем $GF(2^w)$, работают с объектами установленного объема w , которые представляют собой элементы базового поля.

Стандартным представлением элемента A в поле $GF(2^w)$ является w -битное слово, биты которого $(a_{w-1}, a_{w-2}, \dots, a_1, a_0)$ представляют собой полином $a_{w-1}z^{w-1} + \dots + a_1z + a_0$. Элементы можно складывать и перемножать: сложение элементов в поле эквивалентно операции XOR. Для того, чтобы перемножить два элемента поля, перемножаются соответствующие полиномы по модулю 2, а затем результирующий полином сокращается по модулю выбранного неприводимого полинома степени w [5].

Известно [13], что закон формирования выходной последовательности $s_0, s_1, \dots$ над произвольным конечным полем $GF(q)$ m -каскадного регистра сдвига задается линейным рекуррентным соотношением:

$$s_{t+m} = c_{m-1}s_{t+m-1} + c_{m-2}s_{t+m-2} + \dots + c_0s_t + c, \quad t = 0, 1, \dots, \quad (6)$$

где c и все c_i также принадлежат полю $GF(q)$, $0 \leq i \leq m-1$.

Вектор состояния $S_0 = (s_0, s_1, \dots, s_{m-1})$, где каждое значение s_i , $0 \leq i \leq m-1$, является w -битным словом, называется вектором начального состояния регистра. Таким образом, вектор $S_t = (s_t, s_{t+1}, \dots, s_{t+m-1})$ естественно называть внутренним состоянием регистра сдвига с обратной связью на t -м такте работы или вектором t -го состояния линейной рекуррентной последовательности. Соотношение вида (6) называется однородным, если $c = 0$, в противном случае такое соотношение называется неоднородным. Соответственно, формируемая последовательность называется однородной линейной рекуррентной последовательностью, иначе неоднородной. Так как любое неоднородное линейное рекуррентное соотношение (последовательность) можно свести к однородному [13], то в дальнейшем мы будем рассматривать только однородные рекуррентные соотношения (последовательности).

Докажем следующее утверждение.

Утверждение. Пусть линейный рекуррентный регистр, формирующий на выходе последовательность максимальной длины, функционирует в расширенном поле вида $GF(2^w)$. Если m – степень характеристического полинома над $GF(2^w)$, описывающего закон функционирования заданного регистра, то линейная сложность выходной последовательности

$$L = m \cdot w^2. \quad (7)$$

Доказательство. Если $s_0, s_1, \dots$ – однородная линейная рекуррентная последовательность над полем $GF(q)$, то ей соответствует характеристический многочлен, принадлежащий этому же полю, вида:

$$p(x) = x^m + c_{m-1}x^{m-1} + \dots + c_0. \quad (8)$$

Согласно [13], с однородной линейной рекуррентной последовательностью над произвольным конечным полем $GF(q)$ можно связать так называемую сопровождающую матрицу C , принадлежащую полю $GF(q)$, размера $m \times m$ следующего вида:

$$C = \begin{pmatrix} 0 & 0 & \dots & 0 & c_0 \\ 1 & 0 & \dots & 0 & c_1 \\ 0 & 1 & \dots & 0 & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & c_{m-1} \end{pmatrix}. \quad (9)$$

При этом для векторов состояний регистра справедливо равенство

$$S_t = S_0 \cdot C^t, \quad t = 0, 1, \dots$$

Так как мы имеем дело с расширенными полями $GF(2^w)$, то есть $q = 2^w$ и в процессе формирования последовательности все операции выполняются над элементами поля $GF(2^w)$, то, вектор состояния и сопровождающая матрица также будут содержать элементы, принадлежащие этому полю.

Известно [14], что умножение элементов в поле $GF(2^w)$ можно описать через двоичную квадратную матрицу порядка w :

$$M = \begin{pmatrix} \mu_{0,0} & \mu_{0,1} & \dots & \mu_{0,w-1} \\ \mu_{1,0} & \mu_{1,1} & \dots & \mu_{1,w-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{w-1,0} & \mu_{w-1,1} & \dots & \mu_{w-1,w-1} \end{pmatrix},$$

где $\mu_i \in GF(2)$.

Произведение элементов $a = (a_0, a_1, \dots, a_{w-1})$ и $b = (b_0, b_1, \dots, b_{w-1})$, $a \cdot b = d$, при помощи матрицы M выполняется по правилам:

$$\begin{aligned} d_0 &= (a_0 a_1 \dots a_{w-1}) M \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{w-1} \end{pmatrix}, \\ d_1 &= (a_1 a_2 \dots a_0) M \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_0 \end{pmatrix}, \\ &\dots \\ d_{w-1} &= (a_{w-1} a_0 \dots a_{w-2}) M \begin{pmatrix} b_{w-1} \\ b_0 \\ \vdots \\ b_{w-2} \end{pmatrix}, \end{aligned} \quad (10)$$

то есть, при вычислении первого бита d_0 вектора-произведения d векторы a и b остаются без изменений, для получения оставшихся $(w-1)$ бит вектора d к множителям a и b последовательно применяется операция циклического сдвига влево на один бит.

Поскольку в регистре при умножении один из множителей является константой, матрица умножения также известна, то каждый элемент матрицы C можно заменить двоичной матрицей, описывающей умножение соответствующих векторов состояния регистра на этот элемент:

$$C' = \begin{pmatrix} 0 & 0 & \dots & 0 & M \cdot c_0 \\ M \cdot 1 & 0 & \dots & 0 & M \cdot c_1 \\ 0 & M \cdot 1 & \dots & 0 & M \cdot c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & M \cdot 1 & M \cdot c_{m-1} \end{pmatrix}, \quad (11)$$

где $M \cdot c_0$ и $M \cdot 1$ – матрицы умножения порядка $w \times w$ над $GF(2)$, на элементы c_i и 1 , соответственно.

В результате получим квадратную матрицу C' порядка $m \times m$ над $GF(2)$, которая будет выдавать прореженную последовательность с шагом w , поскольку внутренние матрицы

$M \cdot c_0$ и $M \cdot 1$ учитывают циклические сдвиги только второго, известного множителя. Согласно (10), при выполнении операции умножения элементов в двоичном расширенном поле необходимо также учесть циклические сдвиги первого множителя. Для корректного выполнения данной операции и, как следствие, получения каждого бита выходной последовательности, к каждому w -битному слову вектора входного состояния регистра должна применяться операция циклического сдвига влево на j бит, $j = \overline{0, w-1}$. Тогда сопровождающая матрица примет вид

$$C^* = \begin{pmatrix} 0 & 0 & \dots & 0 & C' \\ 1 & 0 & \dots & 0 & C' \\ 0 & 1 & \dots & 0 & C' \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & C' \end{pmatrix}, \tag{12}$$

где 1 – квадратная матрица порядка $m \times m \times w$ над $GF(2)$, описывающая циклический сдвиг влево на 1 бит каждого w -битного слова вектора входного состояния регистра длиной $m \cdot w$ бит.

Двоичная матрица C^* порядка $m \cdot w^2$, полученная по формуле (12), будет соответствовать двоичному характеристическому полиному степени $m \cdot w^2$. По условию ЛРР на выходе формирует последовательность максимальной длины, следовательно, линейная сложность выходной последовательности будет равна степени полученного двоичного характеристического полинома, то есть $L = m \cdot w^2$.

Для экспериментального подтверждения этого результата выходные последовательности, сформированные рассмотренными регистрами, подавались на вход алгоритма Берлекампа-Мэсси. Результаты работы алгоритма представлены в табл. 2.

Таблица 2

Регистр	Степень характеристического полинома, m	Степень порядка поля, w	Значение линейной сложности, L
ЛРР-Snow	16	32	$16384 = 16 \cdot 32^2$
ЛРР- Sober-t16	17	16	$4352 = 17 \cdot 16^2$
ЛРР- Sober-t32	17	32	$17408 = 17 \cdot 32^2$

5 Предложения по совершенствованию методики тестирования линейной сложности NIST

Как известно [1, 2, 11], для того чтобы получить значение линейной сложности при помощи алгоритма Берлекампа-Мэсси, необходимо проверить $2L$ бит исследуемой последовательности. Так, например, чтобы получить значение сложности выходной последовательности регистра, взятого из шифра Snow, необходимо было проверить 32768 бит. Предполагается, что после применения нелинейной функции сложность последовательности на выходе шифрообразующего устройства должна значительно увеличиться. Однако разработчики пакета NIST STS рекомендуют выбирать размер тестируемого блока M последовательности из диапазона [500, 5000]. Очевидно, что объем указанного интервала не соответствует современному уровню сложности шифрующих последовательностей. Предлагается увеличить диапазон в соответствии с требованиями криптостойкости шифрообразующего устройства.

Фактически, согласно [7], закон формирования выходной последовательности регистров, функционирующих в расширенном поле, можно представить как эквивалентную реализацию w параллельных двоичных ЛРР, каждый длиной $m \cdot w$, где m – степень характеристического полинома над расширенным полем $GF(2^w)$. Эти регистры будут иметь одинаковый закон формирования выходной последовательности, задаваемый полиномом над $GF(2)$.

Такое представление является изоморфным, но не идентичным [5]. Так, для регистра, взятого из Snow, линейная рекурсия над $GF(2^{32})$ будет эквивалентна реализации 32 параллельных двоичных ЛРР, каждый длиной $16 \cdot 32 = 512$. Для регистра из Sober-t16 она будет эквивалентна реализации 16 параллельных двоичных ЛРР, каждый длиной $17 \cdot 16 = 272$. И для Sober-t32 длина 32 таких регистров составит $17 \cdot 32 = 544$.

Учитывая данное свойство реализации, при исследовании выходных последовательностей данных регистров проверялся каждый w -й бит выходного блока. В результате на выходе алгоритма Берлекампа-Мэсси были получены значения линейной сложности равные $m \cdot w$. А именно, для исследования последовательности с выхода ЛРР шифра Snow проверялся каждый 32 бит, полученное при этом значение линейной сложности равно 512; в случае исследования регистра поточного шифра Sober-t16 проверялся каждый 16 бит выходной последовательности, полученное значение линейной сложности – 272; для случая Sober-t32 тестировался каждый 32 бит, значение сложности – 544. Так как линейная сложность во всех случаях сохраняет постоянное значение, то, согласно методике теста линейной сложности из пакета NIST STS, значения статистических критериев рассчитываться не могут. Следовательно, последовательности тест не пройдут.

Таким образом, можно сделать следующие выводы. Реализация ЛРР в расширенных полях позволяет не только улучшить скоростные характеристики [4, 5, 6], но и значительно увеличить линейную сложность выходной последовательности. Полученное выражение (7) позволяет численно оценить сложность таких последовательностей.

Безусловно, при генерации шифрующих последовательностей линейные рекуррентные регистры не используются в чистом виде. К выходу таких устройств применяются функции нелинейной фильтрации, которые повышают их криптографическую стойкость, в частности, увеличивая значение линейной сложности. Но, так как последовательности, полученные при помощи современных ЛРР, обладают достаточно высокой сложностью, то для того чтобы обнаружить эффект воздействия нелинейных функций при помощи предлагаемой методики статистического анализа, диапазон выбора тестируемого блока исследуемой последовательности должен быть увеличен, в зависимости от требуемого уровня обеспечения криптографической стойкости генератора. Так, если сложность современных ЛРР в среднем составляет 12715, то для ее расчета потребуется последовательность длиной 25430 бит. Поэтому целесообразно проверять последовательности, длина которых приближается к этому значению.

С другой стороны, согласно рассматриваемой методике, количество проверяемых блоков должно быть не менее 200 для того, чтобы значение статистического критерия было корректным [8], и таких последовательностей должно быть не менее 100 для обеспечения минимального уровня значимости, поэтому длина последовательности, подаваемой на вход теста, значительно возрастет, а, следовательно, для анализа потребуются значительно большие вычислительные затраты. В связи с этим исследование линейной сложности удобнее проводить отдельно от остальных тестов.

Хотя на практике при выполнении процедур зашифрования / расшифрования используется каждый бит шифрующей последовательности, тем не менее, если генератор такой последовательности построен на основе ЛРР, функционирующего в расширенном поле $GF(2^m)$, и в дальнейшем все нелинейные преобразования шифрообразующего устройства выполняются в том же расширенном поле, то для получения более точной оценки генератора тестирование выходной последовательности рекомендуется выполнять, проверяя каждый w -й бит выходного блока, например, проверять каждый старший бит.

Как отмечалось в разд. 1, высокие показатели линейной сложности являются недостаточным условием для принятия окончательного решения относительно применения последовательности в криптографических целях, а построение профиля линейной сложности – достаточно трудоемкий процесс, требующий больших временных затрат. Поэтому дополнительные проверки критериев случайности рекомендуется проводить при помощи остальных тестов, входящих в пакет NIST STS.

При разработке и исследовании поточных шифров и шифрогенераторов в целом необходимо учитывать, прежде всего, статистические свойства выходных последовательностей основных компонентов шифрообразующего устройства. Одной из главных составляющих анализа этих свойств является проверка линейной сложности, поскольку используемый при этом показатель позволяет определить, насколько сложно восстановить символы последовательности, зная все предшествующие. Так как в настоящее время наибольшую популярность получил метод построения поточных шифров на основе ЛПР, функционирующих в расширенных полях, то для вычисления линейной сложности таких регистров предлагается использовать выражение (7).

Для осуществления статистического анализа линейной сложности ключевых последовательностей в целом можно использовать методику, предложенную в NIST STS. Учитывая современные подходы при реализации шифрообразующих устройств, соответствующие возросшим требованиям криптографической стойкости, выдвигаются следующие предложения по совершенствованию существующей методики:

1. Увеличить диапазон тестируемых блоков.
2. При тестировании генераторов на основе ЛПР, функционирующих в расширенном поле $GF(2^m)$, кроме обычной проверки (исследования каждого бита выходной последовательности), выполнять проверку выходной последовательности, состоящей из бит, взятых с шагом m .
3. В связи с возросшими вычислительными затратами проверку линейной сложности рекомендуется выполнять отдельно от остальных тестов.

Список литературы: 1. *Menezes A., P. van Oorschot, and Vanstone S. Handbook of Applied Cryptography.* CRC Press, 1997. 2. *Massey J.L., Shift-Register Synthesis and BCH Decoding // IEEE Transactions on Information Theory, January 1969. V. IT-15, no. 1.* 3. *Ekdahl P., Johansson T., Snow – a new stream cipher. Proceedings of First Open NESSIE Workshop, KU-Leuven. 2000.* 4. *Горбенко И.Д., Потий А.В., Избенко Ю.И., Орлова С.Ю. Анализ схем поточного шифрования, представленных на европейский конкурс NESSIE // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2002. Вып. 5. С.92-110.* 5. *Hawkes P., Rose G., Turing, a Fast Stream Cipher // Pre-proceedings of Fast Software Encryption FSE2003. February 1999. P. 307-324.* 6. *Hawkes P., Rose G. Primitive Specification for Sober-128. Cryptology ePrint Archive Report 2003/081, April 2003.* 7. *Herlestam T. On functions of linear shift register sequences // Proceedings of EUROCRYPT'85, Lecture Notes in Computer Science. 1985. P. 199-129.* 8. *Soto J. Randomness Testing of the Advanced Encryption Candidate Algorithms // NIST, 1999.* 9. *Потий А., Орлова С., Гриненко Т. 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. Вып. 2. С.206-214.* 10. *Фомичев В.М. Дискретная математика и криптология: Курс лекций / Под ред. Н.Д. Подуфалова. М.: ДИАЛОГ-МИФИ. 2003. 400 с.* 11. *Rueppel R.A. Linear complexity and random sequences // Advances in Cryptology – Eurocrypt'85, Lecture Notes in Computer Science. 1986. V. 219. P. 167-188.* 12. *Dai Z.-D., Yang J.-H. Linear complexity of periodically repeated random sequences // Advances in Cryptology – EUROCRYPT '91 (LNCS 547). 1991. P. 168-175.* 13. *Лидл Р., Нидеррайтер Г. Конечные поля: В 2-х т. / Пер. с англ. М.: Мир, 1988. Т. 2. 822 с.* 14. *IEEE P1363 / D11 (Draft Version 11). Standard Specifications for Public Key Cryptography. Annex A (Informative). Number-Theoretic Background. 1999.*

Харьковский национальный
университет радиозлектроники

Поступила в редколлегию 12.12.2004

КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ И ГИПЕРЭЛЛИПТИЧЕСКИХ КРИВЫХ

УДК 681.3.06

А. Д. ТЕВЯШЕВ, д-р техн. наук, С. И. ЗБИТНЕВ, канд. техн. наук, В. Ю. КОВТУН

АЛГОРИТМЫ СКАЛЯРНОГО УМНОЖЕНИЯ В ГРУППЕ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ И НЕКОТОРЫЕ ИХ МОДИФИКАЦИИ

Введение

Теория алгебраических кривых и в частности преобразования в группе точек эллиптической кривой, сегодня стали основой построения ряда новых криптографических стандартов [1, 7, 8, 36, 37]. Интерес к этому математическому аппарату обусловлен открывшимися возможностями уменьшения, по сравнению с другими стандартами эквивалентной стойкости, длины ключа и блока, на котором выполняются криптопреобразования.

Одним из актуальных направлений развития современных криптографических преобразований в группе точек эллиптической кривой является уменьшение сложности выполнения базовых операций и как следствие скалярного умножения точек эллиптической кривой. Задачи уменьшения сложности привлекли внимание многих специалистов, в результате исследований и разработки которых созданы средства выполнения операций скалярного умножения с повышенной скоростью. В статье ставится цель изложить сущность и особенности различных алгоритмов уменьшения сложности скалярного умножения, которые удалось выделить из доступных публикаций, выполнить их систематизацию по вычислительной сложности и определить наиболее перспективные направления оптимизации. Представленные материалы вместе с обширной библиографией, на наш взгляд, могут стать полезными для специалистов-практиков, занимающихся разработкой средств криптографической защиты информации на основе преобразований в группе точек эллиптических кривых, якобиане дивизоров гиперэллиптических кривых, а также специалистов криптоаналитиков, успешное решение задач криптоанализа которыми в существенной мере зависит от сложности выполнения скалярного умножения.

В статью также включены предложения авторов по дополнительному улучшению некоторых алгоритмов скалярного умножения, появившиеся в процессе ее подготовки.

Алгоритмы скалярного умножения

Сразу отметим, что речь будет идти об операциях с точками эллиптической кривой (ЭК), рекомендованных в [37], которые рассматриваются над полями характеристики 2.

Операция скалярного умножения (СУ) состоит в вычислении точки $Q = kP$, где $P \in E(GF(2^n))$ – точка ЭК, имеющая известный порядок n , k – целое, $k \in [1, n-1]$ [1, 5, 7, 8].

Приведем сначала алгоритм умножения, следующий из общего определения СУ точек ЭК.

Алгоритм 1 Двоичное умножение слева направо

Вход: $k = (k_{t-1}, \dots, k_0)_2$, $k_{t-1} = 1$, $P \in E(GF(2^n))$

Выход: kP

-
1. $Q \leftarrow O$.
 2. For $i = t-1$ downto 0 do
 - 2.1. $Q \leftarrow 2 \cdot Q$.
 - 2.2. If $k_i = 1$ then $Q \leftarrow Q + P$.
 3. Return Q .
-

В этом алгоритме, количество единиц в двоичном представлении скалярного множителя k в среднем составляет $l/2 \approx m/2$, что соответствует количеству операций сложения точек, в то время как количество операций удвоения равно m (см. алгоритм 1). Поэтому сложность алгоритма 1 в групповых операциях составляет:

$$I(A_1^{aver}) = \frac{m}{2} I_{add} + m I_{dbl}. \quad (1)$$

Дальнейшая конкретизация этого выражения связана с выбором базиса представления точек (аффинного, проективного, смешанного). Например, в аффинных координатах точек сложность алгоритма 1 в полевых операциях составит [1]:

$$I(A_1^{affn}) = 3m I_{mul} + \frac{3m}{2} I_{inv}, \quad (2)$$

Максимальной производительности групповых операций позволяет добиться представление точек ЭК в смешанном базисе [6]. Если в алгоритме 1 представить точку Q в модифицированных проективных координатах Лопеса-Дахаба, а точку P в аффинных – удвоение на шаге 2.1 и сложение на шаге 2.2 могут быть выполнены с использованием выражений из [6]:

$$\begin{aligned} X_3 &= A^2 + K + L + a \cdot Z_3, \\ Y_3 &= X_3(K + Z_3) + L \cdot C \cdot (F \cdot I + E \cdot J), \\ Z_3 &= D^2, \end{aligned}$$

где $E = Y_1 \cdot Z_2^2$, $F = Y_2 \cdot Z_1^2$, $A = F + E$, $I = X_1 \cdot Z_2$, $J = X_2 \cdot Z_1$, $B = I + J$, $C = Z_1 \cdot Z_2$, $D = B \cdot C$, $K = A \cdot D$, $L = B^2 \cdot D$.

В этом случае для оценки сложности алгоритма 1 в полевых операциях можно получить выражение

$$I(A_1^{proj}) = (5m + 5m + 2) I_{mul} + I_{inv} = (10m + 2) I_{mul} + I_{inv}. \quad (3)$$

При получении этого выражения учтено, что сложность перехода из проективного представления Лопеса-Дахаба $(X : Y : Z)$ к аффинному $(X/Z, Y/Z)$ равна $I(T) = 2I_{mul} + I_{inv}$.

Очередной возможностью повышения эффективности выполнения операции СУ предлагается учет того факта, что вычитание точек ЭК над двоичным полем эффективнее сложения (точке $P(x, y) \in E(GF(2^m))$ соответствует точка $-P = (x, x + y)$). Использование этого свойства требует изменения представления скалярного множителя k с двоичного на знаковое в несмежной форме (NAF – non-adjacent form) $NAF(k) = \sum_{i=0}^{l-1} k_i 2^i$, где $k_i \in \{0, \pm 1\}$. Полезность NAF [3, 5, 20, 28] состоит в том, что в нем смежные коэффициенты не могут быть одновременно отличными от нуля. Кроме того, NAF имеет наименьшее количество коэффициентов $k_i \neq 0$ среди других знаковых представлений и $NAF(k)$ длиннее двоичного представления числа k не более чем на 1 [31, 32]. Каждое положительное целое k имеет единственное NAF(k) представление. Эффективное вычисление NAF(k) выполняется согласно алгоритму 2 [3, 5, 20, 28].

Алгоритм 2 Вычисление NAF целого числа

Вход: Целое число k

Выход: NAF(k)

1. $i \leftarrow 0$.

2. While $k \geq 1$ do

2.1. If $k \bmod 2 = 1$ then $k_i \leftarrow 2 - (k \bmod 4)$, $k \leftarrow k - k_i$; Else $k_i \leftarrow 0$.

2.2. $k \leftarrow k/2$, $i \leftarrow i + 1$.

3. Return $(k_{i-1}, \dots, k_0)$.

Модифицированный алгоритм 3 с учетом использования $\text{NAF}(k)$ имеет вид [3]:

Алгоритм 3 Двоичное NAF умножение

Вход: $\text{NAF}(k) = \sum_{i=0}^{l-1} k_i 2^i$, $P \in E(GF(2^m))$

Выход: kP

1. $Q \leftarrow O$.
2. For $i = l-1$ downto 0 do
 - 2.1. $Q \leftarrow 2 \cdot Q$.
 - 2.2. If $k_i = 1$ then $Q \leftarrow Q + P$.
 - 2.3. If $k_i = -1$ then $Q \leftarrow Q - P$.
3. Return Q .

Сложность алгоритма 3 с учетом того, что средняя плотность всех ненулевых коэффициентов в $\text{NAF}(k)$ длины m равна $\frac{m}{3}$ [3, 20], составит

$$I(A_3) = \frac{m}{3} I_{add} + m I_{dbl}. \quad (4)$$

В случае реализации алгоритма без ограничения памяти, время его работы можно сократить посредством использования, так называемого, «оконного» метода предложенного в [3], суть которого состоит в следующем. На каждом шаге итерации анализируется w бит числа k за итерацию. При этом множитель k в $\text{NAF}_w(k)$ представляется в виде $\text{NAF}_w(k) = \sum_{i=0}^{l-1} k_i 2^i$, где каждый ненулевой коэффициент k_i – нечетный, $k_i < 2^{w-1}$, в остальных свойства NAF и $\text{NAF}_w(k)$ аналогичны [3]. Заметим, что $\text{NAF}_2(k) = \text{NAF}(k)$. $\text{NAF}_w(k)$ вычисляется по алгоритму 4 [3, 5].

Алгоритм 4 Вычисление $\text{NAF}_w(k)$ целого числа

Вход: Целое число k

Выход: $\text{NAF}_w(k)$

1. $i \leftarrow 0$.
2. While $k \geq 1$ do
 - 2.1. If $k \bmod 2 = 1$ then $k_i \leftarrow (k \bmod 2^n)$, $k \leftarrow k - k_i$, Else $k_i \leftarrow 0$.
 - 2.2. $k \leftarrow k/2$, $i \leftarrow i+1$.
3. Return $(k_{i-1}, \dots, k_0)$.

Согласно рекомендациям [3], ширина «окна» предвычислений для СУ в проективных координатах $w = 4$ и $w = 5$ для $m = 163$ и $m = 233$ ($m = 283$) соответственно позволяет достичь минимальной сложности СУ. В дальнейшем принимается $w = 4$, т.к. в этом случае сложность незначительно выше, чем при $w = 5$, и значительно проще программная реализация.

В работах [3, 5, 20] предложен еще один алгоритм – 5, использующий NAF представление и «окно» предвычислений, реализация которого представлена ниже:

Алгоритм 5 «Оконное» NAF умножение

Вход: $\text{NAF}(k) = \sum_{i=0}^{l-1} k_i 2^i$, $P \in E(GF(2^m))$

Выход: kP

1. Вычислить $P_i = iP$, для $i \in \{1, 3, 5, \dots, 2^{w-1} - 1\}$.
2. $Q \leftarrow O$.
3. For $i = l-1$ downto 0 do
 - 3.1. $Q \leftarrow 2Q$.
 - 3.2. If $k_i \neq 1$ then

If $k_i > 0$ then $Q \leftarrow Q + P_{k_i}$.

Else $Q \leftarrow Q - P_{k_i}$.

3. Return (Q) .

Средняя плотность ненулевых коэффициентов в представлении $\text{NAF}_w(k)$ длиной m равна $m/(w+1)$ [3, 5, 20]. Для оценки сложности алгоритма 5 получено выражение

$$I(A_5) = (1I_{abl} + (2^{w-2} - 1)I_{add}) + (m/(w+1)I_{add} + mI_{dbl})$$

Частным случаем описанного выше алгоритма для $w = 2$ следует считать алгоритм СУ в проективных координатах Якоби, представленный в стандартах [2, 4, 5]:

Алгоритм 6. Умножение в проективных координатах Якоби со сложением и вычитанием

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_2$, $k_{t-1} = 1$, $3k = (h_{r-1}, \dots, h_0)_2$, $P = (x, y) \in E(GF(2^m))$

Выход: kP

1. If $(k = 0)$ or $(Z = 0)$ then Return $(1, 1, 0)$.

2. $e = k \bmod n$, где n – порядок точки P (если n неизвестно, тогда $e = k$).

3. $X' \leftarrow X$, $Z' \leftarrow Z$, $Z_1 \leftarrow 1$.

4. If $(k < 0)$ then $k \leftarrow (-k)$, $Y' \leftarrow XZ + Y$.

5. else $Y' \leftarrow Y$.

6. If $Z' = 1$ then $X_1 \leftarrow X'$, $Y_1 \leftarrow Y'$.

7. else $X_1 \leftarrow X'/(Z')^2$, $Y_1 \leftarrow Y'/(Z')^3$.

8. For $i = t - 1$ downto 1 do

8.1 $(X', Y', Z') \leftarrow 2(X', Y', Z')$.

8.2. If $(h_i = 1)$ and $(k_i = 0)$ then $(X', Y', Z') \leftarrow (X', Y', Z') + (X_1, Y_1, Z_1)$.

8.3. If $(h_i = 0)$ and $k_i = 1$ then $(X', Y', Z') \leftarrow (X', Y', Z') - (X_1, Y_1, Z_1)$.

9. Return (X', Y', Z') .

В основе следующего алгоритма лежит идея Монтгомери, предложенная в работе [13]. Ее сущность состоит в том, что x – координата точки $Q_1(x_1, y_1) + Q_2(x_2, y_2)$, $Q_1 \neq \pm Q_2$ может быть получена из x – координат точек $Q_1, Q_2, Q_1 - Q_2$ в соответствии с алгоритмом:

Алгоритм 7 Двоичное умножение Монтгомери

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_2$, $k_{t-1} = 1$, $P = (x, y) \in E(GF(2^m))$

Выход: kP

1. $P_1 \leftarrow P$, $P_2 \leftarrow 2P$.

2. For $i = t - 1$ downto 0 do

2.1. If $k_i = 1$ then $P_1 \leftarrow P_1 + P_2$, $P_2 \leftarrow 2P_2$.

2.2. else $P_2 \leftarrow P_2 + P_1$, $P_1 \leftarrow 2P_1$.

3. Return (P_1) .

Сложность алгоритма 7 составит

$$I(A_7) = (m - 1)I_{add} + (m - 1)I_{dbl}. \quad (6)$$

Модифицированный алгоритм СУ получается путем перехода в алгоритме 7 к операциям над точками $Q_1(x_1, y_1)$, $Q_2(x_2, y_2)$, $Q_1 \neq \pm Q_2$ в аффинных координатах. Если обозначить $Q_1 + Q_2 = (x_3, y_3)$ и $Q_1 - Q_2 = (x_4, y_4)$, то формула для вычисления суммы точек принимает вид:

$$x_3 = x_4 + \frac{x_1}{x_1 + x_2} + \left(\frac{x_1}{x_1 + x_2} \right)^2. \quad (7)$$

На j -й итерации алгоритма вычисляется $T_j = (k_j P, (k_j + 1)P)$. Затем, если старший коэффициент $k_{j+1} = 0$, то $T_{j+1} = (2k_{j+1}P, (2k_{j+1} + 1)P)$ иначе $T_{j+1} = ((2k_{j+1} + 1)P, (2k_{j+1} + 2)P)$. Координаты результирующей точки $kP = (x_1, y_1)$ определяются как:

$$y_1 = x^{-1}(x_1 + x)[(x_1 + x)(x_2 + x) + x^2 + y] + y.$$

Представленные преобразования, использующие аффинное представление точек ЭК, реализованы в алгоритме 8.

Алгоритм 8. Умножение Монтгомери в аффинных координатах

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_2$, $k_{t-1} = 1$, $P = (x, y) \in E(GF(2^m))$

Выход: kP

1. $x_1 \leftarrow x, x_2 \leftarrow x^2 + b/x^2$.
2. For $i = t - 2$ downto 0 do
 - 2.1. $t \leftarrow x_1/(x_1 + x_2)$.
 - 2.2. If $k_i = 1$ then
 - 2.2.1. $x_2 \leftarrow x + t^2 + t, x_2 \leftarrow x_2^2 + b/x_2^2$.
 - 2.3. else
 - 2.3.1. $x_2 \leftarrow x + t^2 + t, x_1 \leftarrow x_1^2 + b/x_1^2$.
3. $r_1 \leftarrow x_1 + x, r_2 \leftarrow x_2 + x, y_1 \leftarrow r_1(r_2 + x^2 + y)/(x + y)$.
4. Return (x_1, y_1) .

Сложность алгоритма 8, по нашим оценкам, составляет:

$$I(A_8) = (t - 2)(2I_{mul} + 2I_{inv} + 2I_{sqr}) + (1I_{inv} + 1I_{mul} + 1I_{sqr}). \quad (8)$$

В алгоритме 9, изложенном в работах [3], [11], авторы использовали идею Монтгомери но реализовали операции над точками ЭК в стандартных проективных координатах.

Алгоритм 9. Умножение Монтгомери в стандартных проективных координатах

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_2$, $k_{t-1} = 1$, $P = (x, y) \in E(GF(2^m))$

Выход: kP

1. Вычисление $(P, 2P)$: $X_1 \leftarrow x, Z_1 \leftarrow 1, X_2 \leftarrow x^4 + b, Z_2 \leftarrow x^2$.
2. For $i = t - 2$ downto 0 do
 - 2.1. If $k_i = 1$ then
 - 2.1.1. $T \leftarrow Z_1, Z_1 \leftarrow (X_1 Z_2 + X_2 Z_1)^2, X_1 \leftarrow x Z_1 + X_1 X_2 T Z_2$.
 - 2.1.2. $T \leftarrow X_2, X_2 \leftarrow X_2^4 + b Z_2^4, Z_2 \leftarrow T^2 Z_2^2$.
 - 2.2. else
 - 2.2.1. $T \leftarrow Z_2, Z_2 \leftarrow (X_1 Z_2 + X_2 Z_1)^2, X_2 \leftarrow x Z_2 + X_1 X_2 T Z_1$.
 - 2.2.2. $T \leftarrow X_1, X_1 \leftarrow X_1^4 + b Z_1^4, Z_1 \leftarrow T^2 Z_1^2$.
3. $x_3 \leftarrow X_1/Z_1$.
4. $y_3 \leftarrow (x + x_3)[(X_1 + x Z_1)(X_2 + x Z_2) + (x^2 + y)(Z_1 Z_2)](x Z_1 Z_2)^{-1} + y$.
5. Return (x_3, y_3) .

Для оценки сложности вычислений по алгоритму 9 можно получить выражение

$$I(A_9) = (t - 2)(6I_{mul} + 5I_{sqr}) + (1I_{inv} + 1I_{mul} + 3I_{sqr}). \quad (9)$$

Наш анализ показал, что существует дополнительная возможность улучшить производительность этого алгоритма. Для этого предлагается в п.3 алгоритма 8 не вычислять непосредственно инверсию Z_1 , т.к. она может быть получена как $Z_1^{-1} = x \cdot Z_2 \cdot (xZ_1Z_2)^{-1}$ в процессе вычислений п.4. Преимуществом алгоритма 8 перед алгоритмами типа 5 является отсутствие необходимости в хранении результатов предвычислений.

Алгоритм 10 представляет модификацию алгоритма 9, предложенную авторами, в которой используются проективные координаты Лопеса-Дахаба [6].

Алгоритм 10 Умножение Монтгомери в проективных координатах Лопеса-Дахаба

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_2$, $k_{t-1} = 1$, $P = (x, y) \in E(GF(2^m))$

Выход: kP

1. Вычисление $(P, 2P)$: $X_1 \leftarrow x, Z_1 \leftarrow 1, X_2 \leftarrow x^4 + b, Z_2 \leftarrow x^2$.

2. For $i = t - 2$ downto 0 do

2.1. If $k_i = 1$ then

2.1.1. $A \leftarrow X_1Z_2, B \leftarrow X_2Z_1, E \leftarrow A + B, T \leftarrow E^2, X_1 \leftarrow xT + AB$,

2.1.2. $Z_1 \leftarrow T, C \leftarrow X_2^2, D \leftarrow Z_2^2, Z_2 \leftarrow CD, X_2 \leftarrow C^2 + bD^2$.

2.2. else

2.2.1. $A \leftarrow X_2Z_1, B \leftarrow X_1Z_2, E \leftarrow A + B, T \leftarrow E^2, X_2 \leftarrow xT + AB$,

2.2.2. $Z_2 \leftarrow T, C \leftarrow X_1^2, D \leftarrow Z_1^2, Z_1 \leftarrow CD, X_1 \leftarrow C^2 + bD^2$.

3. $x_r \leftarrow X_1/Z_1$.

4. $C \leftarrow Z_1Z_2$.

5. $y_r \leftarrow (x + x_r)(X_1X_2 + xE + yC)(xC)^{-1} + y$.

6. Return (x_r, y_r) .

Сложность алгоритма 10 составляет:

$$I(A_{10}) = (t - 2)(6I_{mul} + 5I_{sqr}) + (I_{inv} + 10I_{mul}). \quad (10)$$

Аналогично алгоритму 9 в п. 3 можно не выполнять инверсию Z_1 , т.к. она может быть получена из $x \cdot Z_2 \cdot (xC)^{-1}$.

Рассмотрим теперь подходы к уменьшению сложности СУ, основанные на использовании предвычислений. Следуя [3], воспользуемся представлением числа k в системе исчисления по основанию 2^w в виде: $(k_{d-1}, \dots, k_0)_{2^w}$, где $d = \lceil l/w \rceil$. Пусть точка Q_i такая, что $Q_i = \sum_{r: k_r = j} 2^{wr} P$. Тогда произведение kP можно представить в виде [3]:

$$\begin{aligned} kP &= \sum_{i=0}^{d-1} k_i (2^{wi} P) = \sum_{j=1}^{2^w-1} \left(j \sum_{r: k_r = j} 2^{wr} P \right) = \sum_{j=1}^{2^w-1} j Q_j \\ &= Q_{2^{w-1}} + (Q_{2^{w-1}-1} + Q_{2^{w-1}-2}) + \dots + (Q_{2^{w-1}-1} + Q_{2^{w-1}-2} + \dots + Q_1), \end{aligned} \quad (11)$$

Результат (11) позволил авторам работы [3] предложить алгоритм, представленный ниже.

Алгоритм 11 «Оконное» умножение точки

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_{2^w}$, ширина окна w , $d = \lceil t/w \rceil$, $P = (x, y) \in E(GF(2^m))$

Выход: kP

1. Вычисление: $Q_i = 2^{wi} P$, $i = 0, d - 1$.

2. $A \leftarrow O$, $B \leftarrow O$.

3. For $j = 2^{w-1}$ downto 1 do

3.1. For each i for which $k_i = j$ do

3.1.1. $B \leftarrow B + Q$.

3.2. $A \leftarrow A + B$.

4. Return A .

В предположении, что вероятности $\Pr\{k_{(w-1)+1}, \dots, k_t = 0\} = 1/2^w$ и $\Pr\{k_{(w-1)+1}, \dots, k_t \neq 0\} = (2^w - 1)/2^w$, для оценки сложности алгоритма 11 в среднем и худшем случаях, соответственно, можно получить выражения:

$$I(A_{11}^{aver}) = ((d(2^w - 1)/2^w - 1) + (2^w - 2))I_{odd}, \quad (12)$$

$$I(A_{11}^{worst}) = (d + (2^w - 2))I_{odd}. \quad (13)$$

Этот алгоритм можно дополнительно улучшить, если на шаге предвычислений учесть, что количество операций удвоения соответствует количеству четных чисел, меньших 2^w , а количество операций сложения соответствует количеству нечетных чисел, меньших 2^w . Тогда сложность шага предвычислений для алгоритма 11:

$$I(A_{11}^{pre}) = (2^{w-1} - 1)I_{odd} + (2^{w-1} - 1)I_{dbl}. \quad (14)$$

Предвычисления используются и в работе [5] при построении оригинального алгоритма СУ, получившего название Lim-Lee [14], который является модификацией алгоритма Brickell-Gordon-McCurley-Wilson (BGMW) [30]. Кратко остановимся на основной идее этого алгоритма.

Скалярный множитель k длиной t представляется в виде h блоков длины $a = \lceil t/h \rceil$, каждый такой блок обозначается через K_i . Далее каждый блок K_i представляется в виде v блоков длины $b = \lceil a/v \rceil$. Таким образом:

$$k = \sum_{r=0}^{h-1} \sum_{s=0}^{v-1} \sum_{t=0}^{b-1} k_{vbr+bs+t} 2^{vbr+bs+t}. \quad (15)$$

Авторами [5] предлагается представить в соответствии с (15) произведение kP следующим образом

$$kP = \sum_{t=0}^{b-1} 2^t \sum_{s=0}^{v-1} P_{s,u}, \quad (16)$$

где $P_{s,u}$ – массив предвычисленных значений для $s \in [0, v[$, $u \in [1, 2^h[$, $u = (u_{h-1}, \dots, u_0)_2$, причем

$$u = I_{s,t} = \sum_{r=0}^{h-1} k_{vbr+bs+t} 2^r, \quad (17)$$

где $s \in [0, v-1[$, $t \in [0, b[$.

Элементы массива предвычислений определяются следующим образом:

$$P_{0,u} = \sum_{r=0}^{h-1} u_r 2^{rvb} P, \quad P_{s,u} = 2^{sb} P_{0,u}. \quad (18)$$

Выбор параметров h и v приводит к выбору между производительностью и объемом памяти, которую можно выделить для предвычислений.

Алгоритм 12 Умножение Lim-Lee фиксированной точки

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_2$, h , v , $s \in [0, v[$, $u \in [1, 2^h[$, $P = (x, y) \in E(GF(2^m))$

Выход: kP

1. For $u = 1$ to $2^h - 1$ do
 - 1.1. For $s = 0$ to $v - 1$ do
 - 1.1.1. $u \leftarrow (u_{h-1}, \dots, u_1, u_0)_2$.
 - 1.1.2. $P_{s,u} \leftarrow 2^{sb} \sum_{i=0}^{h-1} u_i 2^{bi} P$.
 2. $Q \leftarrow O$.
 3. For $t = b - 1$ downto 0 do

3.1. $Q \leftarrow 2Q$.

3.2. For $s = v-1$ downto 0 do

3.2.1. $I_{s,t} \leftarrow \sum_{i=0}^{h-1} 2^i K_{vb_i+bs+t}$.

3.2.2. If $(I_{s,t} \neq 0)$ then $u \leftarrow I_{s,t}$, $Q \leftarrow Q + P_{s,u}$.

4. Return Q .

Алгоритм 12 требует хранить $v(2^h - 1)$ точек. Средняя сложность и сложность в худшем случае, соответственно, составят:

$$I(A_{12}^{aver}) = (((2^h - 1)/2^h)vb - 1)I_{add} + (d - 1)I_{dbl}, \quad (19)$$

$$I(A_{12}^{worst}) = (vb - 1)I_{add} + (d - 1)I_{dbl}. \quad (20)$$

Упрощенная версия алгоритма 12 рассматривается в работах [3, 5, 12]. Она и получила название комбинированного умножения. Согласно алгоритму, множитель k записан в виде w строк и колонки полученной матрицы обрабатываются одна за другой:

$$[a_{v-1}, \dots, a_1, a_0]P = a_{v-1}2^{(w-1)d}P + \dots + a_12^dP + a_0P, \quad (21)$$

где $d = \lceil l/w \rceil$ и $a_i \in Z_2$.

Приведем алгоритм, реализующий описанный выше подход.

Алгоритм 13 Комбинированное умножение фиксированной точки

Вход: $k = (k_{l-1}, \dots, k_1, k_0)_2$, ширина окна w , $d = \lceil l/w \rceil$, $P = (x, y) \in E(GF(2^m))$

Выход: kP .

1. Вычисление: $[(a_{w-1}, \dots, a_0)]P$, $\forall (a_{w-1}, \dots, a_0) \in Z_{2^w}$, другими словами, вычислить

$$\sum_{i=0}^{2^w-1} a_i 2^{di} P, \quad \forall (a_{w-1}, \dots, a_0) \in Z_{2^w}.$$

2. Представить множитель в виде:

$$k = k_{d(w-1)+(d-1)}2^{d(w-1)+(d-1)} + \dots + k_{d(w-1)}2^{d(w-1)} + k_{d(w-2)+(d-1)}2^{d(w-2)+(d-1)} + \dots + k_{d(w-w)}2^{d(w-w)}.$$

3. $Q \leftarrow O$.

4. For $i = d-1$ downto 0 do

4.1. $Q \leftarrow 2Q$.

4.2. $Q \leftarrow Q + [(k_{d(w-1)+i}, \dots, k_{d(w-w)+i})]P$.

5. Return Q .

В п. 4.2. может возникнуть ситуация, когда $(k_{d(w-1)+i}, \dots, k_{d(w-w)+i}) = 0$, тогда операция сложения не выполняется. Следовательно, $P\{(k_{d(w-1)+i}, \dots, k_{d(w-w)+i}) = 0\} = 1/2^w$ и, наоборот, $P\{(k_{d(w-1)+i}, \dots, k_{d(w-w)+i}) \neq 0\} = (2^w - 1)/2^w$, исходя из этого сложность в среднем и сложность в худшем случае алгоритма 14, соответственно, составит:

$$I(A_{13}^{aver}) = ((d-1)(2^w - 1)/2^w)I_{add} + (d-1)I_{dbl}, \quad (22)$$

$$I(A_{13}^{worst}) = (d-1)I_{add} + (d-1)I_{dbl}. \quad (23)$$

Сложность шага 1 предвычислений алгоритма 13:

$$I(A_{13}^{pre}) = w2^{w-1}I_{add} + d(w-1)I_{dbl}. \quad (24)$$

Количество операций сложения соответствует количеству разрядов $a_i = 1$, $\forall (a_{w-1}, \dots, a_0) \in Z_{2^w}$ в [23], другими словами? $\sum_{i=1}^w iC_w^i = w2^{w-1}$. Количество операций удвоения соответствует степени множителя $2^{(w-1)d}$ в [23].

Как указано в [5], комбинированный метод с фиксированным основанием отличается от оконного метода с фиксированным основанием необходимостью в незначительно меньшем объеме памяти.

Как указано в [5], комбинированный метод с фиксированным основанием отличается от оконного метода с фиксированным основанием необходимостью в незначительно меньшем объеме памяти.

Другой разновидностью алгоритма 13 является алгоритм de Rooij [9], представленный алгоритмом 14 требует значительно меньших объемов предвычислений.

Алгоритм 14 Умножение de Rooij фиксированной точки

Вход: $k = (k_{d-1}, \dots, k_1, k_0)_{2^w}$, заранее предвычислены $P_i = 2^{(i-1)w} P, i = \overline{1, d}$,
 $P = (x, y) \in E(GF(2^n))$

Выход: kP

1. Определить $M \in [0, d-1]$ такой, что $z_M \geq z_i$ для $i = \overline{0, d-1}$.
2. Определить $N \in [0, d-1]$, $N \neq M$ такой, что $z_N \geq z_i$ для всех $i = \overline{0, d-1}, i \neq M$.
3. For $i = 0$ to $d-1$ do
- 3.1. $z_i \leftarrow k_i$.
4. Вычислить M и N для $(z_{d-1}, \dots, z_0)$.
5. While $z_N \geq 0$ do
- 5.1. $q \leftarrow \lfloor z_M / z_N \rfloor$, $P_N \leftarrow qP_M + P_N$, $z_M \leftarrow z_M \pmod{z_N}$.
- 5.2. Вычислить M и N для $(z_{d-1}, \dots, z_0)$.
6. $C \leftarrow z_M P_M$.
7. Return (C) .

Заметим, что п. 5.1 требует умножения P_M на q , где $0 \leq q \leq 2^w$, которое выполняется посредством одного из описанных выше алгоритмов.

Сложность алгоритма 14 составляет $\log_2 k + W_H(k)$ групповых операций, где $W_H(k)$ – вес по Хеммингу множителя k . В среднем сложность составляет $1.5 \log_2(k)$ групповых операций. Используя NAF представление множителя и «окно» предвычислений, сложность можно уменьшить до $1.2 \log_2(k)$ групповых операций.

Число предвычисленных точек равно $2^w - 2$, значение w определяется из условия $2^{(d+2)w} \geq \#E(GF(p^n))$, где $d = \lceil t/w \rceil$.

Рассмотрим случай, когда умножается базовая точка P , являющаяся фиксированной, тогда СУ точки может быть ускорено за счет предвычисления значений $2P, 2^2P, \dots, 2^{t-1}P$, откуда следует, что сложность алгоритма 1 двоичного умножения справа налево составит:

$$I(A_{15}) = \lceil m/2 \rceil I_{add}. \quad (25)$$

Алгоритм 15 Двоичное умножение слева направо фиксированной точки с предвычисленными удвоениями

Вход: $k = (k_{t-1}, \dots, k_0)_2$, $k_{t-1} = 1$, $P \in E(GF(2^n))$

Выход: kP

1. $Q \leftarrow O$.
2. For $i = t-1$ downto 0 do
- 2.1. If $k_i = 1$ then $Q \leftarrow Q + 2^i P$, где $2^i P$ являются заранее предвычисленными значениями.
3. Return Q .

В работе [10] предлагаются выражения для вычисления $2^i P, i = \overline{1, 4}$ на $E(GF(2^n))$ в аффинном представлении точек. Формулы с уменьшенной сложностью приведены в работе [11]. Их эффективность обусловлена граничным отношением $T_{inv}/T_{mul} \geq 2.3$ [11] в поле

$GF(2^m)$. Для проективных координат подобный подход не эффективен [11]. Аналогичный [10, 11], но менее эффективный подход описан в [12].

В том же ракурсе следует отметить материалы [33, 34], посвященные эффективному вычислению выражений $2P+Q$, $3P+Q$, $4P+Q$ и $2P\pm Q$, $3P\pm Q$, $4P\pm Q$, $P, Q \in E(GF(2^m))$.

Альтернативой к описанным выше подходам для повышения скорости вычисления $2^k P$ на $E(GF(2^m))$ являются результаты [3, 5], которые представлены как алгоритм 16. Условием применения алгоритма является $T_{inv}/T_{mul} < 1$ [5]. Основная идея алгоритма заключается в том, что точка $P(x, y) \in E(GF(2^m))$, $x \neq 0$ может быть представлена парой (x, λ) . Элемент λ вычисляется как $\lambda \leftarrow x + y/x$, но при этом пара (x, λ) не является точкой ЭК. Для выполнения операции удвоения $2P$ представление точки в виде $P(x, \lambda)$, по сравнению с аффинным представлением $P(x, y)$, позволяет сэкономить одну операцию умножения.

Алгоритм 16 Удвоение в аффинных координатах (эффективное инвертирование в поле)

Вход: k , $P = (x, y) \in E(GF(2^m))$

Выход: $2^k P$

1. $\lambda \leftarrow x + y/x$.
 2. For $j = 1$ to $k - 1$ do
 - 2.1. $x_2 \leftarrow \lambda^2 + \lambda + a$, $\lambda_2 \leftarrow \lambda^2 + a + b/(x^4 + b)$, $x \leftarrow \lambda^2$, $\lambda \leftarrow \lambda_2$.
 3. $x_2 \leftarrow \lambda^2 + \lambda + a$, $y_2 \leftarrow x^2 + (\lambda + 1)x_2$.
 4. Return (x_2, y_2) .
-

Сложность алгоритма 16 составит:

$$I(A_{16}) = (k-1)I_{inv} + kI_{mul} + (3(k-2)+2)I_{sqr}. \quad (26)$$

Предложенная авторами модификация предыдущего алгоритма представлена алгоритмом 17.

Алгоритм 17 Удвоение в стандартных проективных координатах

Вход: k , $P = (x, y) \in E(GF(2^m))$

Выход: $2^k P$

1. $X \leftarrow x, Y \leftarrow y, Z \leftarrow 1$.
 2. $\lambda \leftarrow X^2 + Y, Z_2 \leftarrow X$.
 2. For $j = 1$ to $k - 1$ do
 - 2.1. $L \leftarrow \lambda^2, Z_2 \leftarrow Z_2^2, A \leftarrow a \cdot Z_2, X_2 \leftarrow L + \lambda \cdot Z_2 + A$.
 - 2.2. $Z_{\lambda_2} \leftarrow Z_2 \cdot C, B \leftarrow b \cdot Z_2^4, C \leftarrow (X^4 + B), \lambda_2 \leftarrow (L + Z_2) \cdot C + B \cdot Z_2$.
 - 2.3. $X \leftarrow L, Z \leftarrow Z_2, \lambda \leftarrow \lambda_2, Z_2 \leftarrow Z_{\lambda_2}$.
 3. $\lambda \leftarrow \lambda/Z_2$, $x_2 \leftarrow \lambda^2 + \lambda + a$, $x_2 \leftarrow X_2/Z_2$, $y_2 \leftarrow x^2 + (\lambda + 1)x_2$.
 4. Return (x_2, y_2)
-

В приведенном алгоритме операции над точками выполняются в стандартных проективных координатах, что позволяет исключить инвертирование в поле. Этот алгоритм следует позиционировать как алгоритм с эффективным умножением в поле в отличие от алгоритма 16.

Сложность алгоритма 17 составляет:

$$I(A_{17}) = 2I_{inv} + (4(k-2)-3)I_{mul} + (6(k-2)+2)I_{sqr}. \quad (27)$$

Следующим способом, позволяющим снижать сложность СУ, является использование смешанных проективных координат [6], это позволит сэкономить одну операцию умножения на каждом сложении точек.

В различных криптографических схемах, таких как ECDSA, ECGDSA, ECKDSA, при проверке цифровой подписи возникает необходимость в вычислении выражения $kP + lQ$. В произведении kP , точка P – фиксирована, а в lQ , Q – произвольная. Авторы работ [5, 20] предлагают одновременно умножать и вычислять сумму. Алгоритм 18, реализует этот подход.

Алгоритм 18 Одновременное умножение

Вход: $k = (k_{d-1}, \dots, k_1, k_0)_{2^w}$, $l = (l_{d-1}, \dots, l_1, l_0)_{2^w}$, $d = \lceil t/w \rceil$, $P, Q \in E(GF(2^m))$

Выход: $kP + lQ$

1. Вычисление: $D_{i,j} \leftarrow iP + jQ$, $i = 0, 2^w - 1$, $j = 0, 2^w - 1$.
2. $R \leftarrow O$.
3. For $i = d - 1$ downto 0 do
 - 3.1. $R \leftarrow 2^w \cdot R$.
 - 3.2. $R \leftarrow R + D_{i,j}$.
4. Return R .

Алгоритм хранит предвычисленными 2^{2w} точек ЭК.

Сложность алгоритма 18 составляет:

$$I(A_{18}) = (2^{2w} - 3)I_{add} + (d - 1)(2^{2w} - 1)/2^{2w} I_{add} + (d - 1)wI_{dbl}. \quad (28)$$

Сложность этапа предвычислений алгоритма 18:

$$I(A_{18}^{pre}) = 2((2^{w-1} - 1)I_{add} + (2^{w-1} - 1)I_{dbl}) + (2^{2w} - 2^{w+1} + 1)I_{add}. \quad (29)$$

Сложность этапа предвычислений возможно уменьшить. Если все необходимые предвычисления для базовой точки сделать во время инициализации системы, то первое слагаемое уменьшится в два раза.

Авторами предложено развитие подхода, лежащего в основе алгоритма 18, посредством идеи алгоритмов 10 и 12. Получившийся алгоритм был назван как алгоритм 19 – одновременного умножения на основе умножения Lim-Lee фиксированной точки, алгоритм 12, и умножения Монтгомери в проективных координатах Лопеса-Дахаба, алгоритм 10. Каждый из них выполняется последовательно один за другим, в конце, результаты складываются.

Сложность алгоритма 19, соответственно, составит:

$$I(A_{19}^{over}) = (d(2^w - 1)/2^w)I_{add} + dI_{dbl} + \\ + (t - 2)(6I_{mul} + 5I_{sqr}) + (1I_{in.} + 11I_{mul} + 3I_{sqr}). \quad (30)$$

Схожие относительно идеи алгоритма, были рассмотрены в работе [3]. Отличие состоит в том, что в качестве алгоритма умножения базовой точки был использован алгоритм 13, который обладает большей вычислительной сложностью в сравнении с алгоритмом 12.

Ниже приведен алгоритм одновременного умножения, предложенный в [23].

Алгоритм 20 Одновременное умножение Shamir'a

Вход: $k = (k_{t-1}, \dots, k_1, k_0)_{2^w}$, $l = (l_{t-1}, \dots, l_1, l_0)_{2^w}$, $P, Q \in E(GF(2^m))$

Выход: $kP + lQ$

1. Вычисление: $G = P + Q$.
2. $A \leftarrow O$.
3. For $i = t - 1$ downto 0 do
 - 3.1. $A \leftarrow 2A$.

- 3.2. If $(k_i, l_i) = (1, 0)$ then $A \leftarrow A + P$
 3.3. else If $(k_i, l_i) = (0, 1)$ then $A \leftarrow A + Q$
 3.3.1. else If $(k_i, l_i) = (1, 1)$ then $A \leftarrow A + G$
 4. Return (A) .

Сложность алгоритма 20 и этапа предвычислений, соответственно, составит:

$$I(A_{20}^{over}) = (\frac{3}{2}t + 1)I_{add} + (t - 1)I_{dbl}, \quad (31)$$

$$I(A_{20}^{pre}) = 1I_{add}. \quad (32)$$

Известно, что число сложений точек зависит от общего веса по Хеммингу чисел k и l длиной t . Им ставится в соответствие матрица размером $2 \times t$. В ней строки соответствуют двоичным знаковым или беззнаковым представлением этих чисел. *Общий вес по Хеммингу* $JHW(k, l)$ (JHW – joint Hemming weight) чисел k и l – число отличных от нуля строк этой матрицы. Например, если k и l имеют вес по Хеммингу $t/2$, то $JHW(k, l) = (3/4)t$. Для NAF представления с $JHW(k, l) = \frac{5}{9}t$ сложность алгоритма 20 составит:

$$I(A_{20}^{over} NAF) = (\frac{5}{9}t + 2)I_{add} + 1I_{dbl}. \quad (33)$$

Отметим, что п.1 алгоритма 20 требует вычисления не только $P + Q$, но и $P - Q$.

Альтернативой к NAF представлению скалярного множителя считается смежное разреженное представление JSF (joint sparse form) [24]. JSF – знаковое представление, где любые три последовательные колонки матрицы включают нулевую колонку, и если две смежные строки отличны от нуля, тогда они или (11) или $(-1 -1)$ и соответствующие записи из другой строки будут (± 10) . Например, два числа $(4773)_{10}$ и $(4395)_{10}$ могут быть представлены в JSF как:

$$\begin{bmatrix} 4773 \\ 4395 \end{bmatrix} = \begin{bmatrix} 1010\tilde{1}\tilde{1}0100101 \\ 1000100110\tilde{1}0\tilde{1} \end{bmatrix}.$$

В отличие от других представлений, для пары чисел k и l $JHW(k, l) = \frac{1}{2}t$. Благодаря этому замечательному свойству, сложность алгоритма 20 со скалярными множителями, представленными в JSF, составит:

$$I(A_{20}^{over} JSF) = (\frac{1}{2}t + 2)I_{add} + 1I_{dbl}. \quad (34)$$

Кардинально отличающийся подход описан в работе [16]. Суть его состоит в переходе от операции удвоения к операции деления пополам, т.е. вычисляется такая точка Q , что $2Q = P$. Этот подход применим лишь для половины $E(GF(2^m))$ с $Tr(a) = 1$. Основными операциями в этом подходе являются: извлечение квадратного корня, вычисление следа элемента, решение в поле квадратных уравнений вида $x^2 + x = s$, $s \in GF(2^m)$. Описанный выше метод не эффективен при полиномиальном представлении элементов поля, т.к. требует для достижения приемлемого быстродействия использования значительных предвычислений и большого количества памяти. В то же время, он эффективно реализуется в оптимальном нормальном базисе [7, 17, 21, 22] при аппаратной реализации криптопреобразований.

Сравнение алгоритмов СУ по вычислительной сложности

С учетом сложности реализаций, представленные алгоритмы СУ можно разделить на следующие группы:

- умножение без предвычислений;
- умножение с предвычислениями;
- умножение фиксированной точки;
- одновременное умножение;
- умножение на скаляр вида 2^m , $m > 2$;
- умножение на скаляр вида 2^m , $2 < m \leq 5$.

В табл. 1 представлены результаты обобщения оценок вычислительной сложности алгоритмов скалярного умножения без предвычислений.

Таблица 1

Наименование алгоритма	Сложность
Алгоритм 1. Двоичное умножение слева направо	$I(A_1^{aver}) = \frac{1}{2} I_{add} + I_{dbl}$
Алгоритм 3. Двоичное NAF умножение	$I(A_3) = \frac{1}{3} I_{add} + I_{dbl}$
Алгоритм 6. Умножение в проективных координатах Якоби со сложением и вычитанием	$I(A_6) = (t-2) \left(\frac{1}{4} (I_{add} + I_{sub}) + I_{dbl} \right) + (2I_{inv} + 3I_{mul})$
Алгоритм 7. Двоичное умножение Монтгомери	$I(A_7) = (t-1) I_{add} + (t-1) I_{dbl}$
Алгоритм 8. Умножение точек Монтгомери в аффинных координатах	$I(A_8) = (t-2) (2I_{mul} + 2I_{inv} + 2I_{sqr}) + (I_{inv} + I_{mul} + I_{sqr})$
Алгоритм 9. Умножение Монтгомери в стандартных проективных координатах	$I(A_9) = (t-2) (6I_{mul} + 5I_{sqr}) + (I_{inv} + I_{mul} + 3I_{sqr})$
Алгоритм 10. Умножение Монтгомери в проективных координатах Лопеса-Дахаба	$I(A_{10}) = (t-2) (6I_{mul} + 5I_{sqr}) + (I_{inv} + 10I_{mul})$
Алгоритм 16. Удвоение в аффинных координатах (эффективное инвертирование)	$I(A_{16}) = (k-1) I_{inv} + k I_{mul} + (3(k-2)+2) I_{sqr}$
Алгоритм 17. Удвоение в стандартных проективных	$I(A_{17}) = 2I_{inv} + (4(k-2)+3) I_{mul} + (6(k-2)+2) I_{sqr}$

Из данных, приведенных в табл. 1, следует, что среди алгоритмов СУ без предвычислений минимальная сложность обеспечивается при использовании алгоритма 9 или 10 с авторскими улучшениями. Одним из наиболее предпочтительных, среди известных алгоритмов удвоения, без предвычислений, является алгоритм 17.

В табл. 2 представлены результаты обобщения оценки вычислительной сложности алгоритмов СУ с предвычислениями.

Таблица 2

Наименование алгоритма	Сложность
Алгоритм 5. «Оконное» умножение	$I(A_5) = (I_{dbl} + (2^{w-2} - 1) I_{add}) + (t/(w+1) I_{add} + m I_{dbl})$
Алгоритм 11. «Оконное» умножение фиксированной точки	$I(A_{11}^{aver}) = ((d(2^w - 1)/2^w - 1) + (2^w - 2)) I_{add}$
Алгоритм 12. Умножение Lim-Lee фиксированной точки	$I(A_{12}^{aver}) = (((2^h - 1)/2^h)nb - 1) I_{add} + (d-1) I_{dbl}$

Алгоритм 13. Комбинированное умножение фиксированной точки	$I(A_{14}^{over}) = ((d-1)(2^w-1)/2^w)I_{add} + (d-1)I_{dbl}$
Алгоритм 15. Двоичное умножение слева направо фиксированной точки, с предвычисленными удвоениями	$I(A_{15}) = \lceil \frac{L}{2} \rceil I_{add}$

Анализ соотношений, приведенных в табл. 2, показывает, что среди алгоритмов умножения с предвычислениями наименьшую сложность имеет алгоритм 13, который обладает предпочтительным отношением сложность – память. При отсутствии ограничения на количество точек, которые необходимо хранить в памяти, следует выделить алгоритм 12. При аппаратной реализации наиболее предпочтительным является алгоритм 15, но при этом он требует для хранения данных предвычислений наибольшего количества памяти.

В табл. 3 представлены результаты оценки сложности одновременного скалярного умножения.

Таблица 3

Наименование алгоритма	Сложность
Алгоритм 18. Одновременное умножение	$I(A_{18}) = (2^{2w} - 3)I_{add} + (d-1)(2^{2w} - 1)/2^{2w} I_{add} + (d-1)wI_{dbl}$
Алгоритм 19. Одновременное умножение на основе комбинированного умножения фиксированной точки	$I(A_{19}^{over}) = (d(2^w - 1)/2^w)I_{add} + dI_{dbl} + (t-2)(6I_{mul} + 5I_{sqr}) + (1I_{inv} + 11I_{mul} + 3I_{sqr})$
Алгоритм 20. Одновременное умножение Shamir'a (двоичное представление).	$I(A_{20}^{over}) = (\frac{3}{4}t + 1)I_{add} + (t-1)I_{dbl}$
Алгоритм 20. Одновременное умножение Shamir'a (NAF)	$I(A_{20}^{over} NAF) = (\frac{5}{9}t + 2)I_{add} + tI_{dbl}$
Алгоритм 20. Одновременное умножение Shamir'a (NAF)	$I(A_{20}^{over} JSF) = (\frac{1}{2}t + 2)I_{add} + tI_{dbl}$

Анализ соотношений, представленных в табл. 3 позволяет сделать вывод о том, что среди алгоритмов одновременного умножения, использующих предвычисления, занимает ведущее место авторский алгоритм 19. Среди алгоритмов, не использующих предвычисления, необходимо выделить алгоритм 20, использующий JSF представление скалярного множителя.

В целом необходимо также отметить [19, 25], что алгоритмы 7-10, более предпочтительны в случаях, когда необходимо противостоять атакам на реализацию. Детальное описание этих алгоритмов представлено в работах [18, 26]. Указанные подходы повышения устойчивости к атакам на реализацию излагаются в работах [19, 25, 27].

Таким образом, представленные в статье результаты позволяют оценить сложность выполнения скалярного умножения в группе точек эллиптической кривой, выбрать из них наиболее предпочтительные в зависимости от требований и ограничений, которые накладываются на средства их реализации.

Список литературы: 1. ДСТУ 4541-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка. 2. Kobliž N. CM-curves with good cryptographic properties. Advances in cryptology. CRYPTO'91, LNCS 576, 1992, 279-287. 3. Hankerson D., Lopez J., Menezes A. Software implementation of elliptic curve cryptography over binary fields. In Cetin K. Koc and C. Paar editors, Workshop and embedded systems (CHES'99) LNCS 1717 p. 1-24, Springer-Verlag, august 2000. 4. Ковтун В.Ю., Головашич С.А., Стасев Ю.В. Сравнительный анализ алгоритмов умножения и приведения по модулю в поле $GF(2^n)$ // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2003. Вып. 135. С. 129-141. 5. J. Lopez, Dahab R. An overview of elliptic curve cryptography. 2000. 6. Ковтун В.Ю., Стасев Ю.В., Смирнов О.А., Стасева Я.Ю. Анализ методов представления точек эллиптической кривой над двоичными полями // Сборник научных трудов ХВУ. Харьков: ХВУ.

Системы обработки информации. 2002. Вып. 5(21). 7. *IEEE P1363 / D9* (Draft Version 9). Standard Specifications for Public Key Cryptography. 8. *AMERICAN NATIONAL STANDARD X9.62-1998* (Draft version), Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). 9. *Woodbury A.D.*. Efficient algorithms for elliptic curve cryptosystems on embedded systems. Worcester Polytechnic Institute. -September, 2001. 10. *Guajardo J., Paar C.* Efficient algorithms for elliptic curve cryptosystems. *Advanced in cryptology, Proc. CRYPTO' 97*. LNCS 1294. P.342-356, Springer-Verlag. 1997. 11. *Ковтун В.Ю., Вишньенко В.В., Певнев В.Я., Смирнов А.А.*. Модифікований алгоритм скалярного добутку точок еліптичної кривої над двійковими полями. Вісник Житомирського державного технологічного університету вип. 2(26) Т.1, Житомир, 2003. С. 187-192. 12. *Lopez J., Dahab R.* Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation, *Proceedings of the first workshop on cryptographic hardware and embedded systems – CHES'99*, LNCS 1717 p.316-327, Springer-Verlag, 1999. 13. *R. Schroepfel.* Faster elliptic calculations in $GF(2^m)$, Preprint. 1998. 14. *Lim C.H., Lee P.J.* More flexible exponentiation with precomputation, In *Advances in Cryptology-CRYPTO'94* p95-107, Springer-Verlag. 1994. 15. *Itoh K., Takenaka M., Torii N., Temma S., Kurihara Y.* Fast implementation pf public-key cryptography on DSP TMS320C6201. In *proceedings of the first workshop on cryptography hardware and embedded systems (CHES'99)*, LNCS 1717. P.66-72, Springer-Verlag, 1999. 16. *Knudsen E. W.* Elliptic scalar multiplication using point halving. In *Asiacrypt'99*, LNCS 1716 P.135-149, Springer-Verlag, 1999. 17. *Solinas J.* An improved algorithm for arithmetic on family of elliptic curves. *Advanced in cryptology, CRYPTO' 97*, LNCS 1294, B. Kaliski. P.357-371, Springer-Verlag, 1997. 18. *Kocher P.C.* Timing attacks on implementations of Diffie-Helman, RSA, DSS, and other systems. *Advances in cryptology. CRYPTO'96*, LNCS 1109. P.104-113, Springer-Verlag, 1996. 19. *Izu T., Takagi T.* A fast parallel elliptic curve multiplication resistant against side channel attacks. Research Report CORR 02-03, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2002. 21 pages. 20. *Sakai Y., Sakurai K.* An efficient representation of scalars for simultaneous elliptic scalar multiplication. Special section on discrete mathematics and its applications. *IEICE TRANS. FUNDAMENTALS*, vol E86-A, No.5 may 2003. 21. *Bailey D.V., Paar C., Sarcozy G., Hofri M.* Computation on optimal extension fields. May 2000. 22. *Bailey D.V.* Optimal extension fields for fast arithmetic in public-key algorithms. April 1998. 23. *ELGamal T.* Public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE trans. Inf. Theory*, Vol.31, N.4, p.469-472, 1985. 24. *J. A. Solinas.* Some computational speedups and bandwidth improvements for curves over prime fields. 25. *B. Moller.* Securing elliptic curve point multiplication against side-channel attacks. ICS 2001. 26. *Okeya K., Sakurai K.* Power analysis breaks elliptic curve cryptosystems even secure against the timing attack. *Progress in cryptology – INDOCRYPT'2000* (2000), B.K. Roy, E. Okamoto, Eds., LNCS 19977 p.178-190. 27. *Fischer W., Giraud C., Knudsen E.W., Seifert J.-P.* Parallel scalar multiplication on general elliptic curves over F headed against non-differential side-channel attacks. 28. *Chi-Sung, Wen-Chung Kuo.* Speeding up the computations of elliptic curves crypto schemes. *Computers math. Applic.*, Vol. 33, N. 5. P. 29-36, 1997. 29. *Weimerskirch A., Paar C., Shantz S.C.* Elliptic curve cryptography on a Palm OS device. The 6th Australasian conference on information security and privacy, ACISP'01, 2001. 30. *Brickell E.F., Gordon D.M., McCurley K.S., Wilson D.B.* Fast exponentiation with precomputation (Extended abstract). 31. *Mur J.A., Stinson D.R.* Minimality and other properties of the width-w nonadjacent form. Research Report CORR 04-08, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada. 2004. 32. *Mur J.A., Stinson D.R.* Alternative digit sets for nonadjacent representation. Research Report CORR 03-04, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2003. 33. *Ciet M., Joye M., Lauter K.* Trading inversions for multiplications in elliptic curve cryptography. Available at: www.dicc.ucl.ac.be/cryptol/. 34. *Eisentrager K., Lauter K., P. Montgomery L.* An efficient procedure to double and add points on an elliptic curve. August 5, 2002. 35. *FIPS 186-2.* Recommended elliptic curves for federal government use. July 1999. 36. *ISO/IEC FCD 15946-2:* Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital signatures, Final Committee Draft, 1999. 37. *ISO/IEC FCD 15946-4:* Information technology – Security techniques – Cryptographic techniques based on elliptic curves. Part 4: Digital signatures giving message recovery, Final Committee Draft, 2001.

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ НА ЭЛЛИПТИЧЕСКОЙ КРИВОЙ НАД ДВОИЧНЫМ ПОЛЕМ В ПРОЕКТИВНЫХ КООРДИНАТАХ

Введение

Массовое создание, внедрение и эксплуатация информационных систем привели к возникновению спектра новых проблем в сфере безопасности личности, общества и государства. Внимание к этим проблемам закономерно. Если коммерческая организация допускает утечку более 20% важной внутренней информации, то она в 60 случаях из 100 банкротится [18]. Утверждают также [19], 93% компаний, лишившихся доступа к собственной информации на срок более 10 дней, покинули бизнес, причем половина из них заявила о своей несостоятельности немедленно.

Потребность в обеспечении безопасности связана с тем, что существует множество субъектов и структур, весьма заинтересованных в чужой информации и готовых заплатить за неё высокую цену. Так, стоимость устройств подслушивания, продаваемых только в США, составляет в среднем около 900 млн. дол. в год. Суммарный урон, нанесенный организациям, против которых осуществлялось прослушивание, составляет ежегодно в США около 8 млрд. дол. А ведь существуют и, соответственно, приобретаются устройства для несанкционированного доступа к информации и по другим каналам: проникновение в информационные системы, перехват и дешифровка сообщений и т.д. В результате, по данным SANS Institute, средний размер убытка от одной атаки в США на корпоративную систему для банковского и ИТ-секторов экономики составляет около полумиллиона долларов [21]. Примерная структура последствий неэффективного обеспечения информационной безопасности в американских организациях такова [20]: кража конфиденциальной информации – 20-25% от общего годового ущерба; фальсификация финансовой информации – 21-25%; заражение вредоносными программами – 11-12%; нарушение доступа к Web-сайтам – 1-11%; срыв работы информационной системы – 4-10%; незаконный доступ сотрудников к информации – 4-9%; другие виды ущерба – 14-33%.

Не вызывает сомнений, что мероприятия по защите критически важных информационных систем должны соответствовать многочисленным международным, национальным, корпоративным, нормативным и методическим документам [1-3]. К их числу следует отнести и украинский стандарт ДСТУ 4145-2002, в основу которого положены криптографические преобразования на эллиптических кривых (ЭК). Популярность этого математического аппарата обусловлена возможностью применения малой длины ключа и блока преобразований. Наряду с этим, остается актуальным вопрос минимизации вычислительной сложности.

Методы реализации и характеристики криптопреобразований на ЭК зависят от следующих параметров [4]:

1. Видов поля $GF(q)$, над которым задается ЭК: $GF(p)$ или $GF(2^m)$, где p – простое, m – целое.
2. Представления элементов поля в расширенном поле Галуа $GF(2^m)$ (полиномиальное или нормальное).
3. Видов ЭК $E(GF(q))$ (случайная кривая, кривая Коблица).
4. Представления точек ЭК (аффинного или проективного).

Последнее позволяет повысить производительность без ущерба безопасности криптосистем, т.е. не уменьшает сложность решения задачи дискретного логарифма в группе точек ЭК, и поэтому является одним из основных подходов к уменьшению сложности преобразований в группе точек ЭК. Предварительный анализ известных выражений для операций над точками ЭК в различных координатных системах [1-3, 5, 7-9, 12, 13] показывает, что большинство из них могут быть улучшены.

Целью настоящей работы является анализ существующих способов представления точек ЭК, изложение позиций по их усовершенствованию и применению.

1 Арифметика точек ЭК в известных представлениях. Арифметика в аффинных координатах [1-3, 5]

Рассмотрим ЭК $E(GF(2^m))$, которая описывается усеченной формой уравнения Вейерштрасса:

$$y^2 + xy = x^3 + ax^2 + b, \quad (1)$$

где $a, b \in GF(2^m)$ при $b \neq 0$.

Напомним, что вычисление точки пересечения выполняется методом секущих Диофанта [16]. При пересечении кривой порядка 3 кривой порядка 1 в общем случае будет три точки пересечения. На рис. 1 приведены операции сложения и инвертирования точек эллиптической кривой.

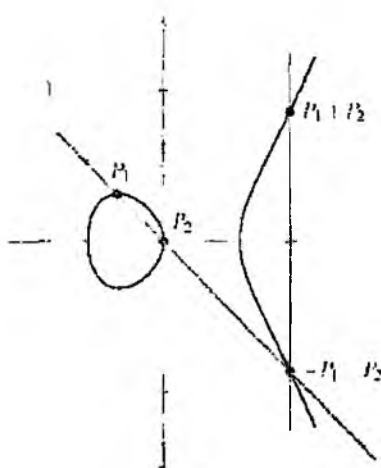


Рис. 1

Дана ЭК $E(GF(2^m))$, заданная уравнением (1) и $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ точки на ней, причем $P_1 \neq P_2$, тогда точка $P_3(x_3, y_3) = P_1 + P_2$ вычисляется следующим образом:

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a, \quad (2)$$

$$y_3 = (x_1 + x_3)\lambda + x_3 + y_1, \quad (3)$$

где

$$\lambda = \frac{y_1 + y_2}{x_1 + x_2}, \text{ если } P_1 \neq P_2, \quad 3.1. (4)$$

$$\lambda = \frac{y_1}{x_1} + x_1, \text{ если } P_1 = P_2 \quad (5)$$

Вычислительная сложность операции сложения и удвоения точек, соответственно, составит:

$$I_{add}(Af) = I_{inv} + 2I_{mul} + I_{sqr} + 9I_{add}, \quad (6)$$

$$I_{dbl}(Af) = I_{inv} + 2I_{mul} + I_{sqr} + 7I_{add}, \quad (7)$$

где I_{inv} – сложность операции инвертирования элемента в поле; I_{mul} – сложность операции умножения в поле; I_{sqr} – сложность операции возведения в квадрат в поле; I_{add} – сложность операции сложения в поле.

Известно, что наиболее вычислительно сложной операцией является инверсия элемента поля. Одним из возможных способов исключения операции инвертирования при сложении и удвоении точки является переход к проективным координатам [1-5].

Арифметика в проективных координатах [6]

На сегодняшний день в литературе по имплементации криптографических преобразований в группе точек эллиптической кривой описано несколько видов проективных координат.

Арифметика в стандартных проективных координатах [1-3, 5, 7, 8]

Проективной точке $(X:Y:Z)$, $Z \neq 0$ ставится в соответствие точка с аффинными координатами:

$$\left(\frac{X}{Z}, \frac{Y}{Z}\right). \quad (8)$$

Уравнение кривой в стандартных проективных координатах имеет вид:

$$Y^2Z + XYZ = X^3 + aX^2Z + bZ^3. \quad (9)$$

Для заданных точек $P_1(X_1:Y_1:Z_1)$, $P_2(X_2:Y_2:Z_2)$, причем точки P_1 и P_2 не принадлежат одному классу смежности, координаты результирующей точки $P_3(X_3:Y_3:Z_3) = P_1 + P_2$ определяются соотношениями:

$$X_3 = (X_1Z_2 + X_2Z_1)(Y_1Z_2 + Y_2Z_1)^2Z_1Z_2 + (Y_1Z_2 + Y_2Z_1) \times \\ \times (X_1Z_2 + X_2Z_1)^2Z_1Z_2 + (X_1Z_2 + X_2Z_1)^4 + aZ_1Z_2(X_1Z_2 + X_2Z_1)^3, \quad (10)$$

$$Y_3 = (X_1Z_2 + X_2Z_1)^2(X_1Z_2Y_2Z_1 + X_2Z_1Y_1Z_2) + \\ + X_3(X_1Z_2 + X_2Z_1 + Y_1Z_2 + Y_2Z_1)Z_1Z_2, \quad (11)$$

$$Z_3 = (X_1Z_2 + X_2Z_1)^3Z_1Z_2, \quad (12)$$

Обозначим $E = Y_1 \cdot Z_2$, $F = Y_2 \cdot Z_1$, $I = X_1 \cdot Z_2$, $J = X_2 \cdot Z_1$, $B = E + F$, $A = I + J$, $C = Z_1 \cdot Z_2$, $D = C \cdot B$. С учетом введенных обозначений выражения (10)-(11) примут вид:

$$X_3 = (A^2 + A \cdot B) \cdot D + a \cdot Z_3 + B^4, \quad (13)$$

$$Y_3 = B^2 \cdot (F \cdot I + E \cdot J) + X_3 \cdot (A + B), \quad (14)$$

$$Z_3 = B^2 \cdot D. \quad (15)$$

В случае, если точки P_1 и P_2 принадлежат одному классу смежности, выражения для вычисления координат результирующей точки имеют вид

$$X_3 = X_1Z_1(Y_1Z_1 + X_1^2)^2 + (X_1Z_1)^2(Y_1Z_1 + X_1^2) + a(X_1Z_1)^3, \quad (16)$$

$$Y_3 = X_1^3 + X_3(Y_1Z_1 + X_1^2 + X_1Z_1), \quad (17)$$

$$Z_3 = (X_1Z_1)^3. \quad (18)$$

Введем обозначения $A = Y_1 \cdot Z_1 - C$, $B = X_1 \cdot Z_1$, $C = X_1^2$. С учетом введенных обозначений выражения (16)-(18) примут вид

$$X_3 = A \cdot B \cdot (A + B) + a \cdot Z_3, \quad (19)$$

$$Y_3 = X_3 \cdot (A + B) + C \cdot X_1, \quad (20)$$

$$Z_3 = B^2 \cdot B. \quad (21)$$

3.2. Сложность операции сложения и удвоения, соответственно, составит

$$I_{add}(StdPrj) = 14I_{mul} + 3I_{sqr} + 8I_{add}, \quad (22)$$

$$I_{dbl}(StdPrj) = 8I_{mul} + 2I_{sqr} + 3I_{add}. \quad (23)$$

Арифметика в проективных координатах Якоби [1-3, 5, 7, 12, 13]

Проективной точке $(X:Y:Z)$, $Z \neq 0$, ставится в соответствие точка с аффинными координатами:

$$\left(\frac{X}{Z^2}, \frac{Y}{Z^3} \right). \quad (24)$$

Уравнение кривой в проективных координатах Якоби представляется в виде

$$Y^2 + XYZ = X^3 + aX^2Z^2 + bZ^6. \quad (25)$$

Для заданных точек $P_1(X_1:Y_1:Z_1)$, $P_2(X_2:Y_2:Z_2)$, причем точки P_1 и P_2 не принадлежат одному классу смежности, координаты результирующей точки $P_3(X_3:Y_3:Z_3) = P_1 + P_2$ определяются соотношениями

$$X_3 = (Y_1Z_2^3 + Y_2Z_1^3)^2 - Z_1Z_2(Y_1Z_2^3 + Y_2Z_1^3)(X_1Z_2^2 + X_2Z_1^2) + (X_1Z_2^2 + X_2Z_1^2 + aZ_1Z_2)(X_1Z_2^2 + X_2Z_1^2)^2, \quad (26)$$

$$Y_3 = X_1Z_2^3(Y_1Z_2^3 + Y_2Z_1^3)(X_1Z_2^2 + X_2Z_1^2)^2 + X_3(Y_1Z_2^3 + Y_2Z_1^3 - Z_3) + Y_1Z_1^3(X_1Z_2^2 + X_2Z_1^2)^3. \quad (27)$$

3.3. Преобразуем выражения (26) и (27) к виду, описанному в стандарте [3]:

$$X_3 = aZ_3^2 + (Y_1Z_2^3 + Y_2Z_1^3)(Y_1Z_2^3 + Y_2Z_1^3 + Z_3) + (X_1Z_2^2 + X_2Z_1^2)^3, \quad (28)$$

$$Y_3 = X_3(Y_1Z_2^3 + Y_2Z_1^3 + Z_3) + (X_2(Y_1Z_2^3 + Y_2Z_1^3) - Y_2Z_1(X_1Z_2^2 + X_2Z_1^2))Z_1^2(X_1Z_2^2 + X_2Z_1^2)^2, \quad (29)$$

$$Z_3 = Z_1Z_2(X_1Z_2^2 + X_2Z_1^2). \quad (30)$$

Обозначим $A = Y_1 \cdot Z_2^3 + Y_2 \cdot Z_1^3$, $B = X_1 \cdot Z_2^2 + X_2 \cdot Z_1^2$, $C = B \cdot Z_1$, $E = Y_2 \cdot Z_1$. С учетом введенных обозначений выражения (28)-(30) преобразуются к виду

$$X_3 = A \cdot (A + Z_3) + B^2 \cdot B + a \cdot Z_3^2, \quad (31)$$

$$Y_3 = X_3 \cdot (A + Z_3) + C^2 \cdot (A \cdot X_2 + B \cdot E), \quad (32)$$

$$Z_3 = C \cdot Z_2. \quad (33)$$

В случае, если точки P_1 и P_2 принадлежат одному классу смежности, выражения для вычисления координат результирующей точки имеют вид

$$X_3 = (X_1 + cZ_1^2)^4, \quad (34)$$

$$Y_3 = X_3(Y_1Z_1 + X_1^2 + Z_3) + X_1^4Z_3, \quad (35)$$

$$Z_3 = X_1 \cdot Z_1^3. \quad (36)$$

Введем обозначения $A = X_1^2$, $B = Z_1^2$, $c = b^{1/4}$. С учетом введенных обозначений выражения (34)-(36) примут вид

$$X_3 = (X_1 + c \cdot B)^4, \quad (37)$$

$$Y_3 = A^2 \cdot Z_3 + X_3 \cdot (Z_3 + Y_1 \cdot Z_1), \quad (38)$$

$$Z_3 = X_1 \cdot B. \quad (39)$$

Сложность операций сложения и удвоения, соответственно, составит

$$I_{add}(JacPrj) = 15I_{mul} + 5I_{sqr} + 8I_{add}, \quad (40)$$

$$I_{dbl}(JacPrj) = 5I_{mul} + 5I_{sqr} + 4I_{add}. \quad (41)$$

Арифметика в проективных координатах Лопеса-Дахаба [8]

Проективной точке $(X:Y:Z)$, $Z \neq 0$ ставится в соответствие точка с аффинными координатами:

$$\left(\frac{X}{Z}, \frac{Y}{Z^2} \right). \quad (42)$$

Уравнение кривой в проективных координатах Лопеса – Дахаба имеет вид

$$Y^2Z + XYZ = X^3Z + aX^2Z^2 + bZ^4. \quad (43)$$

Для заданных точек $P_1(X_1:Y_1:Z_1)$, $P_2(X_2:Y_2:Z_2)$, причем точки P_1 и P_2 не принадлежат одному классу смежности, координаты результирующей точки $P_3(X_3:Y_3:Z_3) = P_1 + P_2$ определяются соотношениями:

$$X_3 = A^2 + I \cdot (D + a \cdot C^2) + A \cdot D, \quad (44)$$

$$Y_3 = Z_3 \cdot (X_3 + I \cdot E) + A \cdot B \cdot (F \cdot Z_3 + X_3 \cdot I), \quad (45)$$

$$Z_3 = D^2, \quad (46)$$

где $E = Y_2 \cdot Z_1^2$, $A = Y_1 \cdot Z_2^2 + E$, $F = X_1 \cdot Z_2$, $B = F + X_2 \cdot Z_1$, $C = Z_1 \cdot Z_2$, $D = B \cdot C$, $I = B^2$.

В случае, если точки P_1 и P_2 принадлежат одному классу смежности, выражения для вычисления координат результирующей точки имеют вид

$$X_3 = B^2 + A, \quad (47)$$

$$Y_3 = A \cdot Z_3 + X_3 \cdot (a \cdot Z_3 + Y_1^2 + A), \quad (48)$$

$$Z_3 = B \cdot C, \quad (49)$$

где $A = b \cdot C^2$, $B = X_1^2$, $C = Z_1^2$.

3.4. Сложность операций сложения и удвоения, соответственно, составит

$$I_{add}(LDPPrj) = 15I_{mul} + 6I_{sqr} + 8I_{add}, \quad (50)$$

$$I_{dbl}(LDPPrj) = 5I_{mul} + 5I_{sqr} + 4I_{add}. \quad (51)$$

Арифметика в проективных координатах Чудновского [12, 13]

Отличительной особенностью представления точек в координатах Чудновского от Якобиановых является сопоставление точки в аффинном представлении и точки вида $(X:Y:Z:Z^2:Z^3)$, $Z \neq 0$. При сложении точек, такое представление позволяет сэкономить несколько операций возведения в квадрат.

Сложности операций сложения и удвоения, соответственно, составят:

$$I_{add}(ChudPrj) = 15I_{mul} + 3I_{sqr} + 8I_{add}, \quad (52)$$

$$I_{dbl}(ChudPrj) = 6I_{mul} + 5I_{sqr} + 4I_{add}. \quad (53)$$

Арифметика в модифицированных проективных координатах Якоби [9, 5, 12]

Отличительной особенностью представления точек в координатах Якоби является сопоставление точки в аффинном представлении точки $(X : Y : Z : Z^2)$, $Z \neq 0$. При сложении точек, такое представление позволяет сэкономить несколько операций возведения в квадрат.

Сложности операций сложения и удвоения, соответственно, составят:

$$I_{add}(JacPrjMdf) = 15I_{mul} + 3I_{sqr} + 8I_{add}, \quad (54)$$

$$I_{dbl}(JacPrjMdf) = 5I_{mul} + 5I_{sqr} + 4I_{add}. \quad (55)$$

Арифметика в модифицированных проективных координатах Лопеса-Дахаба [9, 12, 13]

Отличительной особенностью представления точек в координатах Лопеса-Дахаба является сопоставление точки в аффинном представлении точки $(X : Y : Z : Z^2)$, $Z \neq 0$, что позволяет сэкономить одну полевую операцию возведения в квадрат. Для заданных точек $P_1(X_1 : Y_1 : Z_1)$, $P_2(X_2 : Y_2 : Z_2)$, причем точки P_1 и P_2 не принадлежат одному классу смежности, координаты результирующей точки $P_3(X_3 : Y_3 : Z_3) = P_1 + P_2$, определяются соотношениями (45)-(47).

В случае, если точки P_1 и P_2 принадлежат одному классу смежности, выражения для вычисления координат результирующей точки определяются соотношениями (48)-(50).

3.5. Сложность операций сложения и удвоения, соответственно, составит:

$$I_{add}(LDPPrjMdf) = 15I_{mul} + 5I_{sqr} + 8I_{add}, \quad (56)$$

$$I_{dbl}(LDPPrjMdf) = 5I_{mul} + 5I_{sqr} + 4I_{add}. \quad (57)$$

Арифметика в смешанных координатах

В работах [8, 13] предлагается оригинальный способ сложения точек, когда одна из точек представлена в аффинных, а другая – в проективных координатах $P_1(X_1 : Y_1 : Z_1)$, $P_2(X_2 : Y_2 : 1)$.

Для сложения точки в стандартных проективных координатах $P_1(X_1 : Y_1 : Z_1)$ и точки $P_2(X_2 : Y_2 : 1)$ в аффинных координатах, причем они не принадлежат одному классу смежности, выражения (19)-(21) примут вид

$$X_3 = A \cdot C \cdot (A + B) + B^4 + a \cdot Z_3, \quad (58)$$

$$Y_3 = X_3 \cdot (A + B) \cdot Z_1 + B^2 \cdot (F \cdot Y_1 + E \cdot X_1), \quad (59)$$

$$Z_3 = C \cdot B^2, \quad (60)$$

где $E = Y_2 \cdot Z_1$, $F = X_2 \cdot Z_1$, $A = Y_1 + E$, $B = X_1 + F$, $C = Z_1 \cdot B$.

Сложность операции сложения в смешанных стандартных проективных координатах составляет

$$I_{add}^{mix}(StdPrj) = 12I_{mul} + 2I_{sqr} + 8I_{add}. \quad (61)$$

В случае, когда одна из точек представлена в проективных координатах Якоби $P_1(X_1 : Y_1 : Z_1)$, а другая в аффинных координатах, $P_2(X_2 : Y_2 : 1)$, причем они не принадлежат одному классу смежности, выражения (31) – (33) примут вид

$$X_3 = a \cdot E + A \cdot (A + Z_3) + B^2 \cdot B, \quad (62)$$

$$Y_3 = X_3 \cdot (A + Z_3) + E \cdot (X_2 \cdot Y_1 + D \cdot X_1), \quad (63)$$

$$Z_3 = B \cdot Z_1, \quad (64)$$

где $C = Z_1^2$, $D = Y_2 \cdot Z_1$, $A = Y_1 + C \cdot D$, $B = X_1 + X_2 \cdot C$, $E = Z_3^2$.

Сложность операции сложения в смешанных проективных координатах Якоби составляет

$$I_{add}^{mix}(JacPrj) = 11I_{mul} + 3I_{sqr} + 8I_{add}. \quad (65)$$

Далее рассмотрим сложение в проективных координатах Лопаса-Дахаба точки $P_1(X_1 : Y_1 : Z_1)$ и точки, представленной в аффинных координатах $P_2(X_2 : Y_2 : 1)$, причем они не принадлежат одному классу смежности, выражения (45) – (47) [8] примут вид

$$X_3 = A^2 + D + B^2 \cdot (C + a \cdot E), \quad (66)$$

$$Y_3 = D \cdot (X_2 \cdot Z_3 + X_3) + Z_3 \cdot (X_3 + Y_2 \cdot Z_3), \quad (67)$$

$$Z_3 = C^2, \quad (68)$$

где $A = Y_1 + Y_2 \cdot E$, $B = X_1 + X_2 \cdot Z_1$, $C = Z_1 \cdot B$, $D = A \cdot C$, $E = Z_1^2$.

Сложность операции сложения в смешанных проективных координатах Якоби составляет:

$$I_{add}^{mix}(LDPrj) = 10I_{mul} + 4I_{sqr} + 8I_{add}. \quad (69)$$

При сложении в смешанных координатах Чудновского точек $P_1(X_1 : Y_1 : Z_1 : Z_1^2 : Z_1^3)$ и $P_2(X_2 : Y_2 : 1 : 1 : 1)$ используются выражения (63)–(65). Отличие этих координат состоит в экономии одной операции возведения в квадрат за счет предвычислений.

3.6. Сложность операции сложения в смешанных проективных координатах Чудновского составляет

$$I_{add}^{mix}(ChudnPrj) = 11I_{mul} + 2I_{sqr} + 8I_{add}. \quad (70)$$

Для сложения точки в модифицированных проективных координатах Якоби $P_1(X_1 : Y_1 : Z_1 : Z_1^2)$ и точки в аффинных координатах $P_2(X_2 : Y_2 : 1 : 1)$ используются выражения (64)–(66).

Сложность операции сложения в смешанных модифицированных проективных координатах Якоби составляет:

$$I_{add}^{mix}(JacPrjMdf) = 11I_{mul} + 3I_{sqr} + 8I_{add}. \quad (71)$$

Сложение точек в проективных модифицированных координатах Лопаса-Дахаба $P_1(X_1 : Y_1 : Z_1 : Z_1^2)$ и в аффинных координатах $P_2(X_2 : Y_2 : 1 : 1)$ выполняется аналогично сложению точек в проективных координатах Лопаса-Дахаба согласно выражениям (67)–(69).

В этом случае сложность операции сложения составит

$$I_{add}^{mix}(LDPrjMdf) = 10I_{mul} + 4I_{sqr} + 8I_{add}. \quad (72)$$

2 Операции сложения точек в проективных координатах с пониженной вычислительной сложностью

Анализируя арифметические операции в существующих проективных координатах, авторы провели вывод выражений для вычисления координат результирующей точки. Полученные результаты выгодно отличаются от приведенных в открытой печати [1-3, 7-10, 12] и содержат на несколько полевых операций меньше. Рассмотрим операции сложения с уменьшенной сложностью.

Сложение точек в проективных координатах Якоби

Преобразуем выражение (27) Y -координаты к виду

$$Y_3 = X_3(Y_1Z_2^3 + Y_2Z_1^3 + Z_3) + (X_2Z_1^2(Y_1Z_2^3 + Y_2Z_1^3) + Y_2Z_1^3(X_1Z_2^2 + X_2Z_1^2))(X_1Z_2^2 + X_2Z_1^2)^2. \quad (73)$$

Введем обозначения $L = Z_2^2$, $E = Y_1 \cdot L \cdot Z_2$, $F = Y_2 \cdot K \cdot Z_1$, $I = X_1 \cdot L$, $J = X_2 \cdot K$, $A = E + F$, $B = I + J$, $C = Z_1 \cdot Z_2$, $K = Z_1^2$, $M = B^2$. В новых обозначениях выражения (73), (28) и (30) примут вид

$$X_3 = A \cdot (A + Z_3) + M \cdot B + a \cdot Z_3^2, \quad (74)$$

$$Y_3 = X_3 \cdot (A + Z_3) + M \cdot (A \cdot J + B \cdot F), \quad (75)$$

или, преобразовав $(A \cdot J + B \cdot F)$ из (75), имеем

$$Y_3 = X_3 \cdot (A + Z_3) + M \cdot (E \cdot J + F \cdot I), \quad (76)$$

$$Z_3 = C \cdot B. \quad (77)$$

3.7. Сложность операции сложения в проективных координатах, полученных авторами:

$$I_{add}(JacPrj*) = 15I_{mul} + 4I_{sqr} + 8I_{add}. \quad (78)$$

Сложение точек в проективных координатах Лопеса-Дахаба

Преобразуем выражения (2) и (3) к проективному виду

$$X_3 = (Y_1Z_2^2 + Y_2Z_1^2)^2 + Z_1Z_2(Y_1Z_2^2 + Y_2Z_1^2)(X_1Z_2 + X_2Z_1) + Z_1Z_2(X_1Z_2 + X_2Z_1 + aZ_1Z_2)(X_1Z_2 + X_2Z_1)^2, \quad (79)$$

$$Y_3 = (X_1Z_2 + X_2Z_1)^3(Z_1Z_2)^2(Y_1Z_2^2X_2Z_1 + Y_2Z_1^2X_1Z_2) + X_3(Z_1Z_2(Y_1Z_2^2 + Y_2Z_1^2)(X_1Z_2 + X_2Z_1) + Z_3), \quad (80)$$

$$Z_3 = (Z_1Z_2)^2(X_1Z_2 + X_2Z_1)^2. \quad (81)$$

3.8. Обозначим $E = Y_1 \cdot Z_2^2$, $F = Y_2 \cdot Z_1^2$, $A = F + E$, $I = X_1 \cdot Z_2$, $J = X_2 \cdot Z_1$, $B = I + J$, $C = Z_1 \cdot Z_2$, $D = B \cdot C$, $K = A \cdot D$, $L = B^2 \cdot D$. В рамках введенных обозначений конечные формулы представим в виде:

$$X_3 = A^2 + K + L + a \cdot Z_3, \quad (82)$$

$$Y_3 = X_3 \cdot (K + Z_3) + L \cdot C \cdot (F \cdot I + E \cdot J), \quad (83)$$

$$Z_3 = D^2. \quad (84)$$

Сложность операции сложения:

$$I_{add}(LDPrj*) = 14I_{mul} + 5I_{sqr} + 8I_{add}. \quad (85)$$

Сложение точек в смешанных координатах Лопеса-Дахаба и аффинных

В смешанных координатах Лопеса-Дахаба выражения для вычисления координат точек (82)-(85) упростятся:

$$X_3 = A^2 + D + C \cdot B^2 + a \cdot Z_3, \quad (86)$$

$$Y_3 = Z_3 \cdot (X_2 \cdot D + Y_2 \cdot Z_3) + X_3 \cdot (D + Z_3), \quad (87)$$

$$Z_3 = C^2, \quad (88)$$

где $A = Y_1 + Y_2 \cdot Z_1^2$, $B = X_1 + X_2 \cdot Z_1$, $C = Z_1 \cdot B$, $D = A \cdot C$.

Сложность операции сложения в смешанных координатах:

$$I_{add}^{mix}(LDPPrj^*) = 10I_{mul} + 4I_{sqr} + 8I_{add}. \quad (89)$$

Сложение точек в проективных координатах Чудновского

3.9. В качестве отправной точки примем выражения (31)–(33). Обозначим $E = Y_1 \cdot Z_2^3$, $F = Y_2 \cdot Z_1^3$, $I = X_1 \cdot L$, $J = X_2 \cdot K$, $A = E + F$, $B = I + J$, $C = Z_1 \cdot Z_2$, $K = Z_1^2$, $L = Z_2^2$, $M = B^2$. Согласно введенным обозначениям, формулы для вычисления результирующей точки примут вид

$$X_3 = A \cdot (A + Z_3) + M \cdot B + a \cdot Z_3^2, \quad (90)$$

$$Y_3 = X_3 \cdot (A + Z_3) + M \cdot (E \cdot J + F \cdot I), \quad (91)$$

$$Z_3 = C \cdot B. \quad (92)$$

3.10. Сложность операции сложения точек в координатах

$$I_{add}(ChudPrj^*) = 14I_{mul} + 2I_{sqr} + 8I_{add}. \quad (93)$$

Сложение точек в модифицированных проективных координатах Якоби

Введем обозначения $E = Y_1 \cdot L \cdot Z_2$, $F = Y_2 \cdot K \cdot Z_1$, $I = X_1 \cdot L$, $J = X_2 \cdot K$, $A = E + F$, $B = I + J$, $C = Z_1 \cdot Z_2$, $K = Z_1^2$, $L = Z_2^2$, $M = B^2$. Воспользуемся введенными обозначениями и выражениями (31)–(33). В результате выражения (31)–(33) преобразуются к виду

$$X_3 = A \cdot (A + Z_3) + M \cdot B + a \cdot Z_3^2, \quad (94)$$

$$Y_3 = X_3 \cdot (A + Z_3) + M \cdot (E \cdot J + F \cdot I), \quad (95)$$

$$Z_3 = C \cdot B. \quad (96)$$

Сложность операции сложения:

$$I_{add}(JacPrjMdf^*) = 15I_{mul} + 2I_{sqr} + 8I_{add}. \quad (97)$$

Сложение точек в модифицированных проективных координатах Лопеса-Дахаба

Воспользуемся ранее полученными выражениями (44)–(46), с учетом введенных обозначений $E = Y_1 \cdot Z_2^2$, $F = Y_2 \cdot Z_1^2$, $A = F + E$, $I = X_1 \cdot Z_2$, $J = X_2 \cdot Z_1$, $B = I + J$, $C = Z_1 \cdot Z_2$, $D = B \cdot C$, $K = A \cdot D$, $L = B^2 \cdot D$, формулы (45)–(47) примут вид

$$X_3 = A^2 + K + L + a \cdot Z_3, \quad (98)$$

$$Y_3 = X_3 \cdot (K + Z_3) + L \cdot C \cdot (F \cdot I + E \cdot J), \quad (99)$$

$$Z_3 = D^2. \quad (100)$$

Сложность операции сложения:

$$I_{add}(LDPPrjMdf^*) = 14I_{mul} + 3I_{sqr} + 8I_{add}. \quad (101)$$

3 Сравнение

В сводной табл. 1 приведены полученные сложности арифметических операций на ЭК. Строки, помеченные символом (*), содержат результаты, полученные авторами.

С целью сравнения сложности преобразований в различных координатах выразим вычислительную сложность операции инверсии и возведения в квадрат через сложность операции умножения в поле следующим образом: $I_{inv} \approx 10,5I_{mul}$, $I_{sq} \approx 0,11I_{mul}$ [8]. Операция сложения не учитывалась в силу своей малой вычислительной сложности, что не приводит к существенному влиянию на результирующую сложность.

Таблица 1

Система координат	Общее сложение				Общее сложение (смешанные координаты)				Удвоение			
	/	$\wedge^2$	*	Σ^*	/	$\wedge^2$	*	Σ^*	/	$\wedge^2$	*	Σ^*
Аффинная, (x, y)	1	1	2	12,6	-	-	-		1	1	2	12,6
Стандартная проективная, [8], $(x/Y, Y/Z)$	-	3	14	14,3	-	2	12	12,2	-	2	8	8,22
Проективная Якоби, [8], $(x/Z^2, Y/Z^3)$	-	5	15	15,5	-	3	11	11,3	-	5	5	5,55
Проективная Якоби*, $(x/Z^2, Y/Z^3)$	-	4	15	15,4	-	3	11	11,3	-	5	5	5,55
Модифицированная проективная Якоби, [8], $(x/Z^2, Y/Z^3)$	-	3	15	15,3	-	3	11	11,3	-	5	5	5,55
Модифицированная проективная Якоби*, $(x/Z^2, Y/Z^3)$	-	2	15	15,2	-	3	11	11,3	-	5	5	5,55
Проективная Чудновского, $(x/Z^2, Y/Z^3)$	-	3	15	15,3	-	2	11	11,2	-	5	6	6,55
Проективная Чудновского*, $(x/Z^2, Y/Z^3)$	-	2	14	14,2	-	2	11	11,2	-	5	6	6,55
Проективная Лопес-Дахаба, [8], $(x/Z, Y/Z^2)$	-	6	15	15,7	-	4	10	10,4	-	5	5	5,55
Проективная Лопес-Дахаба*, $(x/Z, Y/Z^2)$	-	5	14	14,5	-	4	10	10,4	-	5	5	5,55
Модиф. проективная Лопес-Дахаба, $(x/Z, Y/Z^2)$	-	5	15	15,5	-	4	10	10,4	-	5	5	5,55
Модифицированная проективная Лопес-Дахаба*, $(x/Z, Y/Z^2)$	-	3	14	14,3	-	4	10	10,4	-	4	5	5,44

* – авторская оптимизация.

Заключение

Из данных, приведенных в табл. 1 следует, что операция сложения в проективных координатах Чудновского, стандартных проективных координатах, а также полученные авторами выражения для сложения в модифицированных координатах Лопеса-Дахаба наиболее эффективны. Если рассматривать весь комплекс операций с точками на кривой, а именно, операции, необходимые для выполнения скалярного умножения точек, тогда лидирующие позиции занимают арифметические операции в координатах Лопес-Дахаба, согласно выражениям, полученным авторами.

Внешне полученный авторами выигрыш в несколько полевых операций кажется незначительным, однако он становится ощутимым при выполнении большого количества операций скалярного умножения. Примером может служить процессинговый центр банка по работе с электронными транзакциями. Известно, что каждая транзакция содержит авторскую электронную подпись, которую необходимо проверить. Так, при проверке электронной подписи на 1 млн. транзакций выигрыш от применения выражений предложенных авторами в проективных координатах Лопеса-Дахаба, составит 44,8 тыс. транзакций.

Использование проективных координат позволяет уйти от операции инвертирования, время на выполнение которой наиболее ощутимо и составляет $I_{inv} \approx 10,5I_{mul}$ [8], в то время

как граничное условие $I_{inv} \geq 3I_{mul}$ [14]. Следовательно, использование проективных координат будет актуальным и в ближайшем будущем.

В последующих работах будет рассмотрен подход [9, 13], основанный на комбинировании различных координат, с целью уменьшения количества операций умножения и инвертирования при сложении и удвоении точек ЭК.

Список литературы: 1. ДСТУ 4541-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка. 2. IEEE P1363 / D9 (Draft Version 9). Standard Specifications for Public Key Cryptography, 1999. 3. AMERICAN NATIONAL STANDARD X9.62-1998 (Draft version). Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). 4. D. Johnson, A. Menezes, S. Vanstone. The elliptic curve digital signature. Certicom research 2001. Canada. 5. Анализ методов представления точек эллиптической кривой над двоичными полями / Ю.В. Стасев, В.Ю. Ковтун, О.А. Смирнов, Я.Ю. Стасева // Системы обработки информации. 2002. Вып. 5(21). 6. Elliptic curves. J.S. MILNE. University of Michigan. 1996. 7. Agnew, G. B., Mullin, R. C. and Vanstone, S. A. On the development of a fast elliptic curve cryptosystem. Advances in Cryptology EuroCrypt'92. 8. Darell Hankerson, Julio Lopez Hernandez, Alfred Menezes. Software implementation of elliptic curve cryptography over binary fields. Advances in Cryptology Crypto '99. 9. Cohen H., Miyaji A., Ono T. Efficient elliptic curve exponentiation using mixed coordinates. 10. J. Lopez and R. Dahab, Improved algorithms for elliptic curve arithmetic's in $GF(2^n)$, Selected Areas in Cryptography –SAC'98, LNCS 1556, 1999. 201-212. 11. Point Multiplication on Ordinary Elliptic Curves over Fields of Characteristic Three. N.P. Smart, E.J. Westwood. Computer Science Department University of Bristol. United Kingdom. 12. Chudnovsky D.V., Chudnovsky G.V. "Sequence of number generated by addition in formal group and new primality and factorization test", Advanced in Applied Math., 8 (1986), 385-434. 13. Збитнев С.И. Проективная геометрия – не все так гладко // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2002. Вып. 126. 14. Вишнько В.В., Ковтун В.Ю., Певнев В.Я., Смирнов А.А. Модифікований алгоритм скалярного добутку точок еліптичної кривої над двійковими полями // Вісник ЖДТУ. 2003. №2(26). Т.1 / Технічні науки. С.187-192. 15. И.Р. Шафаревич. Основы алгебраической геометрии. М. 1972 С. 568. 16. Соловьев Ю.П. Рациональные точки на эллиптических кривых // Соросовский образовательный журнал. 1997. №10. С. 138-143. 17. Степанов С.А. Арифметика алгебраических кривых. М.: Наука, 1999. 18. Ездаков А., Макарова О. Как защитить информацию // Сети 1997 № 8. 19. Сабынин В. Специалисты, давайте говорить на одном языке и понимать друг друга. Информост-Средства связи. № 6. 20. Сэйер П. Lloyd страхует от хакеров. Computerworld // Россия. 2000. № 30. 21. Хметев Л. Оценка эффективности мер безопасности, закладываемых при проектировании электронно-информационных систем // Труды науч.-техн. конференции "Безопасность информационных технологий". Пенза, 2001.

Харьковский национальный
университет радиоэлектроники

Поступила в редакцию 15.11.2004

ПРЕОБРАЗОВАНИЯ В ОПТИМАЛЬНО РАСШИРЕННЫХ ПОЛЯХ И ИХ ПРИМЕНЕНИЕ В КРИПТОГРАФИИ

Введение

В наш век глобального использования электронных информационных систем большое внимание уделяется решению задач обеспечения безопасности информации в этих системах. Для решения вопросов безопасности используются криптографические алгоритмы двух типов – симметричные и несимметричные. Арифметика в полях является неотъемлемой частью многих несимметричных криптографических алгоритмов. В общем, поля $GF(p^m)$ можно разделить на несколько основных типов:

- поля, в которых p – большое простое число, $m = 1$;
- поля, в которых $p > 1$ и $m > 1$;
- поля, в которых $p = 2$ и $m > 1$.

Поля первого типа достаточно сложны для выполнения вычислений на современных компьютерах. Поля второго типа наиболее оптимальны для аппаратной реализации, но при этом так же сложны для программной реализации. В данной статье рассматривается один из видов полей третьего типа, а именно оптимально расширенные поля. Ранее рассмотрение таких полей проводилось в нескольких работах. В [1] проводился анализ использования поля вида $GF((2^n)^m)$ с $n = 16$, $m = 11$. Такое представление удобно в связи с использованием подполя $GF(2^{16})$. Небольшая размерность подполя позволяет при вычислении сформировать таблицу Кэли, которая занимает относительно небольшой объем памяти. Кроме того, в некоторых работах [2,3,4] достаточно полно рассмотрен оптимальный нормальный базис. Анализ оптимально расширенных полей проводился не настолько полно, наилучшим образом они рассмотрены в [5,6,7,8]. Информация об оптимально расширенных полях, возможности их использования в криптографии и реализациях разрознена и не систематизирована. Целью данной статьи является систематизация и обобщение информации об оптимально расширенных полях и алгоритмах выполнения операций над полями такого типа. Кроме того, немаловажной задачей является попытка реализации элементарных математических операций в оптимально расширенных полях и анализ возможности ускорения таких операций.

1 Оптимально расширенные поля. Формирование оптимально расширенных полей по заданным параметрам

В оптимально расширенных полях в общем случае p является простым числом достаточно большого размера, m – небольшим целым числом.

Для решения задачи увеличения скорости арифметических операций можно использовать несколько известных методов оптимизации, применимых для программной реализации. Для оптимизации арифметики в поле $GF(p^m)$ предлагаются такие ограничения [7,8]:

1. При выборе простого числа p необходимо руководствоваться соотношением $p < w$, где w – размер машинного слова.
2. Кроме того, p должно быть псевдо-Мерсеновым числом, то есть иметь вид $2^n \pm c$, где $\log_2 c \leq \frac{1}{2}n$. Такой выбор числа p позволяет эффективно выполнять приведение по модулю в подполе.
3. Проводить выбор m таким образом, чтобы была возможность получить неприводимый бином вида $x^m - \omega$, такой выбор числа m позволяет выполнять эффективное приведение по модулю в расширенном поле. Степень расширения

может быть достаточно мала, если размеры машинного слова позволяют выбирать простые числа p большого размера. Например, если число P имеет разрядность 32 бита, то для получения расширенного поля с размерностью 160 бит число $m = 5$.

Исходя из этих ограничений в [5] были сформулированы определения, касающиеся оптимально расширенных полей. Приведем некоторые из них.

Псевдо-Мерсенновы простые числа – это простые числа вида $2^n \pm c$, где $\log_2 c \leq \frac{1}{2}n$.

Как уже отмечалось, для построения оптимально расширенного поля (OEF – Optimal Extended Field) необходимо выполнение некоторых условий. Формально оптимально расширенное поле – это конечное поле $GF(p^m)$, для которого выполняются следующие условия:

- p является псевдо-Мерсенновым простым числом;
- неприводимый бином $P(x) = x^m - \omega$ существует над $GF(p)$.

Существует два специальных случая оптимально расширенных полей, которые обладают дополнительными преимуществами при выполнении арифметических операций. Такие поля называются OEF вида I и OEF вида II.

OEF вида I строится на основе p вида $2^n \pm 1$. Поля вида I позволяют значительно понизить сложность выполнения операции приведения по модулю в подполе.

OEF вида II имеет неприводимый бином $x^m - \omega$. Поля вида II позволяют значительно понизить сложность выполнения операции приведения по модулю в расширенном поле, так как умножение на ω , в случае, когда $\omega = 2$, может быть выполнено как простой сдвиг.

Выбор степени расширения m зависит от разложения (факторизации) числа $p - 1$.

В общем, процесс построения OEF с учетом ограничений, указанных выше, мы можем разделить на следующие шаги.

1) Выбор простого числа p . Так как число p близко к машинному слову, но при этом меньше, чем машинное слово, то его выбор можно проводить простым перебором. Учитывая, что $p = 2^m - c$, можно изменять c от 1 до какого-либо предела с проверкой на простоту каждого полученного числа p .

2) Факторизация числа.

Эта процедура не позволит получить исчерпывающий список всех OEF, скорее она построена для быстрого определения порядка OEF вида II. Кроме того, такая процедура рассматривает только простые числа вида $2^m - c$, хотя простые вида $2^m + c$ также подходят для OEF. В этом случае в качестве исходной точки можно взять степень m , которая на 1 меньше, чем машинное слово, и вместо вычитания числа c проводить его прибавление. Остальные операции аналогичны.

Алгоритм, реализующий указанные выше шаги, будет иметь такой вид. Алгоритм требует на входе параметр n – битовую длину требуемого p ; параметры low , $high$ – границы размеров требуемого расширенного поля (алгоритм представлен на языке псевдо Си) [4].

```
for (c = 1; log2 c < ⌊ $\frac{1}{2}$ n⌋; c = c + 2)
{
  p = 2n - c;
  if (p – простое)
  {
    выполнить факторизацию p - 1;
    for (m = low; m < high; m++)
    {
      if (2 – примитив в GF(p))
      {
        if (m ≡ 0(mod 4))
        {
          I f (p ≡ 1(mod 4))
          return p, m;
        }
        else
          return p, m;
      }
    }
  }
}
```

Процедура определения, является ли 2 примитивом в $GF(p)$, описана в [5] и не требует больших временных затрат.

Указанный алгоритм является вероятностным, поэтому указать точную сложность для него не представляется возможным, но если оценивать некоторую усредненную сложность, то можно получить следующие результаты.

Количество простых чисел на интервале определяется согласно теореме Чебышева и в усредненном варианте для интервала $(2^n - c, 2^n)$ составляет $k = \frac{2^n}{\ln 2^n} - \frac{2^n - c}{\ln(2^n - c)}$.

Так как операции факторизации, проверки на простоту и деления являются трудоемкими, то при выполнении алгоритма построения ОЕФ можно не учитывать вычислительные затраты на цикловые операции и операции сравнения. В общем случае, для прохождения первого условия понадобится порядка $\frac{c}{2k}$ проверок, того, что число p является простым. В случае прохождения этого условия необходимо выполнить факторизацию числа $p-1$. В принципе, для ускорения выполнения процедуры построения поля, факторизация может выполняться не в полном объеме, а только для определения количества двоек в числе $p-1$. На основании данных, полученных при факторизации, можно определить, существуют ли расширения поля, которые укладываются в указанный диапазон.

В общем случае сложность построения расширенного поля может быть описана формулой:

$$I_{construct} = xI_{test\ prime} + yI_{part\ factor} + zI_{mask}, \quad (1)$$

здесь x – количество проверок числа на простоту (зависит от дальнейшего выполнения алгоритма, при удачном попадании $x \approx 3$);

y – количество частичных факторизаций (соответствует количеству выполнений условия p – простое). Сложность такой операции зависит от требуемой степени расширения m , не более $5m$ простейших делений, то есть $I_{part\ factor} < 5m \cdot I_{div}$;

z – количество операций наложения маски на два младших бита чисел m и p , $z < 1,5(high-low)$.

2 Арифметика над оптимально расширенными полями

Оптимально расширенные поля могут быть представлены в стандартном каноническом (полиномиальном) виде. Если $A \in GF(P^m)$, то $A(x) = a_{m-1}x^{m-1} + \dots + a_1x + a_0$, где $a_i \in GF(p)$. Так как мы выбираем p таким образом, чтобы его размер был меньше размера машинного слова, то такое число займет в памяти ровно m слов.

Сложение над оптимально расширенными полями [4]

Сложение и вычитание в оптимально расширенном поле выполняются классическим способом – путем сложения или вычитания коэффициентов полиномиального представления с приведением по модулю, в случае необходимости, коэффициента при каждой степени. Приведение по модулю выполняется вычитанием (прибавлением) числа p . Алгоритм является общеизвестным и описан в [5,8]. Кроме того, в [8] приведена оценка сложности алгоритма, но с учетом использования блока предсказания для сложности выполнения операции сложения может быть приведена несколько иная оценка:

$$I_{ADDSF} = mI_{add} + \frac{1}{2}mI_{sub}. \quad (2)$$

Операция вычитания в оптимально расширенном поле будет иметь такую же сложность.

Умножение над оптимально расширенными полями

Умножение выполняется в два этапа: это умножение полиномов (формирование коэффициентов c_i' при степенях x^i , где $i = 0, 1, \dots, 2m - 2$) и приведение по модулю в расширенном поле. Методы приведения по модулю также будут рассмотрены в этой статье. В качестве одного из методов формирования коэффициентов возможно использование классического умножения в столбик. Такой метод требует m^2 умножений и $(m - 1)^2$ сложений в подполе $GF(p)$.

Методы умножения в подполях

Умножение в подполе не требует специального алгоритма, так как выполняется стандартной операцией умножения двух слов. В результате умножения двух элементов, размер которых не превышает размера машинного слова, в общем случае получаем произведение размером более одного слова. Для эффективного выполнения приведения по модулю в случае, когда размер приводимого числа превышает размер машинного слова, существуют несколько методов. Один из наиболее эффективных – метод Баррета. Но такой метод будет высокоэффективным, если применяется для «длинных» чисел. Для чисел, с которыми оперируют алгоритмы умножения в подполях, такой алгоритм не дает выигрыша.

Рассмотрим теперь одно из условий, наложенных при построении оптимально расширенного поля. А именно условие, что число p должно иметь вид $2^n \pm c$, где $\log_2 c \leq \frac{1}{2}n$. Такой выбор числа p позволяет выполнять приведение по модулю, не используя операции деления. Алгоритм, который выполняет операцию приведения по модулю, если число p имеет вид $2^n - c$ представлен в [5].

Алгоритму передаются следующие данные $p = 2^n - c$, $\log_2 c < \frac{1}{2}n$, $x < p^2$, где x - число, которое необходимо привести по модулю.

Результатом работы алгоритма является число r , которое удовлетворяет условию $r \equiv x \pmod{p}$.

В среднем, алгоритм потребует не более двух проходов цикла, то есть на выполнение всего алгоритма приведения по модулю, по подсчетам авторов алгоритма [6], будет затрачено 2 умножения на c , 6 сдвигов на n и 6 сложений и вычитаний, то есть сложность алгоритма можно определить формулой:

$$I_{MRSF} = 2I_{mul} + 6I_{shlong} + 6I_{add}. \quad (3)$$

Если $c = 1$, то алгоритм потребует только одного прохода цикла, кроме того, алгоритм при таких входных значениях не потребует ни одного умножения. В среднем алгоритм будет выполнен за 2 сдвига и 2 сложения, то есть:

$$I_{MRSF} = 2I_{shlong} + 2I_{add}. \quad (4)$$

В [8] показано, что некоторые операции сдвига могут быть заменены операциями присвоения. Однако, было также замечено, что кроме операций сдвига, операциями присвоения можно заменить и некоторые операций вычитания. В таком случае в среднем сложность выполнения алгоритма составит:

$$I_{MRSF} = 2I_{mul} + 4I_{add}, \quad (5)$$

а для случая, когда $c = 1$:

$$I_{MRSF} = 2I_{add} \quad (6)$$

Метод приведения по модулю в оптимально расширенном поле

После выполнения операции умножения получаем промежуточный результат $C'(x)$. В общем случае степень полинома $C'(x)$ будет больше, чем m , или, по крайней мере, равна m . Поэтому необходимо выполнить операцию приведения по модулю. Известно [5], что для выполнения операции приведения в поле необходимо применять умножение в подполе, причем количество таких умножений пропорционально числу термов в полиноме, образующем поле. Таким образом, если мы построим полеобразующий полином так, что его коэффициенты будут иметь небольшой вес, приведение по модулю потребует небольшого количества умножений в подполе. Так как моном вида x^m очевидно всегда приводим, рассмотрим неприводимый бином вида $P(x) = x^m - \omega$.

Существует теорема [7], согласно которой любой полином $C'(x)$ над $GF(p)$, степень которого меньше либо равна $2m - 2$, может быть приведен по модулю $P(x) = x^m - \omega$.

Эта теорема базируется на том, что $c'_{m+i} x^{m+i} \equiv \omega c'_{m+i} x^i \bmod P(x)$, для $i = 0, 1, \dots, m-2$. Кроме того, ясно, что только термы $c'_{m+i} x^{m+i}$, $i \geq 0$ должны быть приведены по модулю $P(x)$. Следовательно, общая формула операции приведения будет выглядеть так: $C(x) \equiv c'_{m-1} x^{m-1} + [\omega c'_{2m-2} + c'_{m-2}] x^{m-2} + \dots + [\omega c'_{m+1} + c'_1] x + [\omega c'_m + c'_0] \bmod P(x)$ [8].

Сложность выполнения данной операции:

$$I_{MREF} = (m-1)I_{mul} + (m-1)I_{MRSF} + (m-1)I_{ADD,SF}. \quad (7)$$

Для такого метода возможно значительное уменьшение сложности выполнения в том случае, когда поле обладает неприводимым биномом вида $x^m - 2$. Тогда реализацию умножения на ω можно заменить командой сдвига, и сложность выполнения данной операции составит:

$$I_{MREF} = (m-1)I_{shl} + (m-1)I_{sub} + (m-1)I_{ADD,SF}. \quad (8)$$

Возведение в квадрат над оптимально расширенными полями

Возведение в квадрат можно выполнить при помощи классического умножения в столбик. Кроме того, для каждой определенной степени m возможно разворачивание формулы умножения с целью ускорения выполнения операции. Например, при $m = 3$ и использовании классического умножения для выполнения операции возведения в квадрат понадобится 9 операций умножения и 4 операции сложения, а при использовании развернутой формулы возведения в квадрат понадобится 6 операций умножения в подполе, 1 операция сложения в подполе и 2 операции сдвига.

Инвертирование в подполе и расширенном поле

Операции инвертирования в подполе и расширенном поле являются не менее важными и необходимыми задачами при работе с оптимально расширенными полями, чем приведенные выше. Например, при выполнении операций на эллиптических кривых в аффинном базисе (сложение, вычитание) необходимо одно инвертирование в расширенном поле, что влечет за собой инвертирование в подполе. Следовательно, перечень алгоритмов, приведенных в обзоре, был бы не полным без алгоритмов инвертирования.

Для полиномов могут быть применены следующие алгоритмы инвертирования. Расширенный алгоритм Эвклида [9], алгоритм «почти» инверсии, [10] модифицированный алгоритм «почти» инверсии [9].

Эти алгоритмы достаточно эффективны при работе с полиномами, однако они не оптимизированы для работы с оптимально расширенными полями. Для вычисления инверсии

в оптимально расширенных полях можно использовать алгоритм, известный как алгоритм Итоха и Тсуи, описанный Бейли в [7]. Рассмотрим этот алгоритм.

На входе полином $A(x) \in GF(p^m)$:

- 1) $B(x) = A(x)$;
- 2) использовать алгоритм суммирующих цепей для вычисления $B(x) = B(x)^{r-1}$;
- 3) $c_0 = B(x)A(x)$, формула для вычисления c_0 имеет следующий вид:

$$c_0 = \omega(a_1b_{m-1} + \dots + a_{m-1}b_1) + (a_0b_0); \tag{9}$$

- 4) $c = c_0^{-1}$;
- 5) $B(x) = B(x)c$;

На выходе: полином $B(x) = \sum b_i x^i$, такой, что $A(x)B(x) \equiv 1 \bmod P(x)$

Этот алгоритм базируется на формуле $A^{-1}(x) = (A^r)^{-1}(x) \cdot A^{r-1}(x)$, $r = \frac{p^m - 1}{p - 1}$.

Сложность вычисления элемента c_0 по формуле (9)

$$I_c = m(I_{mul} + I_{div}) + (m - 1)I_{ADDSF}. \tag{10}$$

А общая сложность алгоритма инвертирования в расширенном поле

$$I_{INVEF} = I_c + I_{INVSF} + I_{MULEF} \tag{11}$$

Проведем теоретический разбор алгоритма.

Пусть $\alpha \in GF(p^m)$. Тогда отображение $\alpha \rightarrow \alpha^p$ – автоморфизм, известный как Фробениус.

Как известно, i -я итерация Фробениуса $\alpha \rightarrow \alpha^{p^i}$ также является автоморфизмом. Рассмотрим произвольный элемент $GF(p^m)$: $A(x)^{p^i} = \sum a_j^{p^i} x^{jp^i}$ для всех $a_j \in GF(p)$. Согласно малой теореме Ферма $a_j^p \equiv a_j \bmod p$. Таким образом:

$$A^{p^i}(x) = a_{m-1}x^{(m-1)p^i} + \dots + a_1x^{p^i} + a_0 \bmod P(x) \tag{12}$$

Теперь рассмотрим элементы x^{jp} , которые не являются фиксированными в автоморфизме.

Теорема [7] гласит: Пусть $P(x)$ – неприводимый полином вида $P(x) = x^m - \omega$ над $GF(p)$, e – целое число, $x \in GF(p)[x]$. Тогда $x^e \equiv \omega^q x^s \bmod P(x)$, где $s \equiv e \bmod m$ с $q \equiv \frac{e - s}{m}$.

Так как величины x^{jp} всегда фиксированы для определенного поля, то можно легко выполнить предвычисления и использовать в дальнейшем уже готовую таблицу. В этом случае при вычислении $B(x) = B(x)^{r-1}$ в алгоритме инвертирования в расширенном поле отпадает необходимость выполнения вычислительно сложных операций возведения в степень по модулю, что позволяет повысить скорость выполнения операции инвертирования. Приведем некоторые таблицы предвычислений для полей разной длины (см. табл. 1 - 3). Предвычисленные константы инверсий для $GF((2^{32} - 5)^5)$ приведены в табл.1.

Таблица 1

$x^p \bmod P(x) \equiv 149005400x$	$x^{p^1} \bmod P(x) \equiv 3451601923x$
$x^{2p} \bmod P(x) \equiv 3451601923x^2$	$x^{2p^1} \bmod P(x) \equiv 1998511013x^2$
$x^{3p} \bmod P(x) \equiv 2990816245x^3$	$x^{3p^1} \bmod P(x) \equiv 149005400x^3$
$x^{4p} \bmod P(x) \equiv 1998511013x^4$	$x^{4p^1} \bmod P(x) \equiv 2990816254x^4$

Предвычисленные константы инверсий для $GF((2^{32} - 387)^6)$ приведены в табл. 2

Таблица 2

$x^p \bmod P(x) \equiv 2630997634x$	$x^{p^2} \bmod P(x) \equiv 2630997633x$
$x^{2p} \bmod P(x) \equiv 2630997633x^2$	$x^{2p^2} \bmod P(x) \equiv 1663969275x^2$
$x^{3p} \bmod P(x) \equiv 1663969275x^3$	$x^{3p^2} \bmod P(x) \equiv 1x^3$
$x^{4p} \bmod P(x) \equiv 1998511013x^4$	$x^{4p^2} \bmod P(x) \equiv 2630997633x^4$
$x^{5p} \bmod P(x) \equiv 1663969276x^5$	$x^{5p^2} \bmod P(x) \equiv 1663969275x^5$

Предвычисленные константы инверсий для $GF((2^{32} - 1053)^7)$ приведены в табл. 3

Таблица 3

$x^p \bmod P(x) \equiv 3008035609x$	$x^{p^2} \bmod P(x) \equiv 2551385844x$
$x^{2p} \bmod P(x) \equiv 2551385844x^2$	$x^{2p^2} \bmod P(x) \equiv 2407186096x^2$
$x^{3p} \bmod P(x) \equiv 2420335890x^3$	$x^{3p^2} \bmod P(x) \equiv 3028229892x^3$
$x^{4p} \bmod P(x) \equiv 2407186096x^4$	$x^{4p^2} \bmod P(x) \equiv 3008035609x^4$
$x^{5p} \bmod P(x) \equiv 3764691640x^5$	$x^{5p^2} \bmod P(x) \equiv 1663969275x^5$
$x^{6p} \bmod P(x) \equiv 3028229892x^6$	$x^{6p^2} \bmod P(x) \equiv 2420335890x^6$

3 Применение оптимально расширенных полей для выполнения операций в группах точек эллиптических кривых

Возможность использования оптимально расширенных полей в группах точек эллиптических кривых в криптографии пока не опровергнута. Специально для скалярного умножения в группах точек эллиптических кривых был адаптирован алгоритм скалярного умножения на эллиптических кривых, использующий эндоморфизм Фробениуса. Базовый алгоритм представлен в [6]. Однако существует возможность использования и обычных алгоритмов скалярного умножения.

Проанализируем сложностные (скоростные) показатели выполнения операций в оптимально расширенных полях. Скоростные показатели реализации математики в оптимально расширенных полях и их сравнение с результатами подобной реализации на DEC Alpha 500MHz (64) [5], а также сравнение со скоростью выполнения операций в полях $GF(p^k)$ и $GF(2^m)$ приведены в табл. 4.

Таблица 4

Размер поля	Процессор	$p = 2^n - c$	Бином $x^m - \omega$	GF умножение (мкс)	ЕС сложение (мкс)		ЕС удвоение (мкс)	
					$GF(p^m)$	$GF(2^m)$	$GF(p^m)$	$GF(2^m)$
160	DEC Alpha 500MHz (64)	$2^{32} - 5$	$x^5 - 2$	1,35	20	-	16.2	-
	Pentium III 800MHz (32)	$2^{32} - 5$	$x^5 - 2$	1,58	13	2100	14	2100
192	DEC Alpha 500MHz (64)	$2^{32} - 387$	$x^6 - 2$	2,13	32	-	26	-
	Pentium III 800MHz (32)	$2^{32} - 387$	$x^6 - 2$	1,75	26	5300	27	5300
224	DEC Alpha 500MHz (64)	$2^{32} - 1053$	$x^7 - 2$	3,00	45	-	36	-
	Pentium III 800MHz (32)	$2^{32} - 1053$	$x^7 - 2$	1,93	32	9800	37	9800

Данные, приведенные в табл. 4, позволяют сделать следующие выводы. Производительность рабочих станций типа DEC Alpha 500MHz выше, чем производительность рабочих станций на основе процессоров Intel Pentium. Кроме того, в алгоритмах выполняется большое количество умножений, которые для 32-разрядных процессоров Intel Pentium выходят за разрядную сетку, что, увеличивает время выполнения инструкции. В то же время таких сложностей у процессоров DEC Alpha нет, так как их разрядная сетка позволяет выполнить любую операцию в подполе, не выходя за разрядную сетку. Поэтому результаты сложения и умножения в группах точек эллиптических кривых, представленные в табл. 4, можно считать сравнимыми. Можно также отметить, что скорость вычислений в оптимально расширенных полях при программной реализации на несколько порядков выше скорости выполнения тех же операций с такими же длинами в полях вида $GF(2^m)$. Скорость вычислений в случае аппаратной реализации будет во многом зависеть от размеров слова аппаратного средства и размеров выбранного подполя.

Что касается безопасности математики такого типа, то в открытых источниках пока не было информации, что она менее стойка, чем криптография в группах точек эллиптических кривых в простых и расширенных полях.

Недостатком OEF является меньшее количество полей по сравнению с полями других типов. Однако их количество все-таки достаточно для применения в криптографии.

Заключение

Были рассмотрены основные понятия и определения для оптимально расширенных полей. Проведена систематизация информации по алгоритмам в этих полях. В ходе работы выполнения реализация основных математических операций. Показано, что свойства OEF позволяют изменить алгоритмы вычисления некоторых операций в оптимально расширенных полях, что дает возможность значительно ускорить выполнение элементарных. Для классических алгоритмов сложения и удвоения в аффинных координатах выигрыш по производительности может достигать сотни и более раз.

Кроме того, в ходе работ были проведены исследования по определению теоретической сложности элементарных операций в OEF и определены особенности, которые позволяют в некоторых случаях снизить сложность выполняемых операций по сравнению с предложенными в [5 - 7].

Список литературы: 1. *De Win E., Bosselaers A., Vandenberghe S., De Gersen P., Vandewalle J.* A fast software implementation for arithmetic operation in $GF(2^n)$. Asiacrypt'96, Springer Lecture Notes in Computer Science, 1996. 2. *Reyhani-Masoleh A., Hasan M.A.* Efficient multiplication beyond optimal normal bases, 2002. 3. *Tim Woo* Efficient Software Technical report of Department of Electrical and Computer Engineering Faculty of Engineering University of Waterloo Waterloo, Ontario, Canada. 2002. 4. *Peng Ning, Yiqun Lisa Yin* Efficient Software Implementation for Finite Field Multiplication in Normal Basis, 2002. 5. *Daniel V. Bailey* Computation in Optimal Extension Fields, Faculty of the Worcester Polytechnic Institute, 2000. 6. *Tetsuiaro Kobayashi* Base O method for elliptic curves over OEF. IEICE TRANS. Fundamentals Vol. E83-A. #4. 2000. 7. *Bailey D. V., Paar C.* Optimal Extension Fields for Fast Arithmetic in Public - Key Algorithms, Advances in Cryptology, CRYPTO '98, Lecture Notes in Computer Science 1462, p.472-485. Springer, 1998. 8. *Д.И. Лавриненко.* Оптимально расширенные поля в алгоритмах для эллиптических кривых // Радиотехника: Все-укр.межвед. науч. техн. сб. 2001. Вып. 119. С. 56-61. 9. *Eun Jeong Lee, Duk Soo Kim, Pil Joong Lee.* Speed-up of Fpm Arithmetic for Elliptic Curve Cryptosystems. In Proceedings of ICICS '98, Berlin. 1998. Springer Lecture Notes in Computer Science. 10. *Schroeppel R., Orman H., O'Malley S., Spatscheck O.* Fast key exchange with elliptic curve systems. Advances in Cryptology, CRYPTO '95, p. 43-56. 1995.

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ ДИСКРЕТНОГО ЛОГАРИФИРОВАНИЯ В ЯКОБИАНЕ ГИПЕРЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Введение

Большинство современных криптографических стандартов основываются на преобразованиях в группе точек эллиптической кривой (ЭК). Постоянный рост мощностей современных вычислительных систем, популярность GRID вычислений приводит к ужесточению требований к существующим криптографическим преобразованиям. Такое развитие событий заставляет искать уязвимости, эффективные атаки, исследовать перспективные трудно-решаемых задач, которые в будущем смогли бы быть положены в основу криптографических стандартов. На сегодняшний день большое количество исследовательских работ посвящено криптографическим преобразованиям на гиперэллиптической кривой (ГЭК), как наиболее перспективным.

Целью данной работы является проведение классификации, изучения и оценки сложности известных методов криптоанализа преобразований на ГЭК, что позволит правильно расставить акценты для последующего повышения их эффективности, выявить потенциально уязвимые классы кривых, которые не следует применять в криптографических целях.

Для понимания излагаемого далее материала напомним некоторые основные определения.

1. Некоторые теоретические сведения

Теоретические положения лежащие в основе криптографии на ГЭК [27, 36, 40] отличаются от положений криптографии на ЭК, поэтому будет уместным напомнить некоторые базовые определения.

Пусть k – поле и $\bar{k}$ – его алгебраическое замыкание. Гиперэллиптической кривой C рода g над k ($g \geq 1$) называется множество точек (u, v) удовлетворяющих уравнению:

$$C: v^2 + h(u)v = f(u) \text{ в } k[u, v], \quad (1)$$

где $h(u) \in k[u]$ – полином степени g , $f(u) \in k[u]$ – монический полином степени $2g+1$, причем исключаются точки $(u, v) \in \bar{k} \times \bar{k}$, которые одновременно удовлетворяют и (1) и частным производным (1) по соответствующим переменным $2u + h(u) = 0$, $h'(u)v - f'(u) = 0$.

Дивизором D называется формальная сумма точек кривой C вида:

$$D = \sum_{P \in C} m_P P, \quad m_P \in \mathbb{Z}, \quad (2)$$

где только конечное число m_P – кратностей точек отлично от нуля.

Степенью дивизора D , $\deg(D)$, называется целое число равное $\sum_{P \in C} m_P$. Через D^0 обозначим множество дивизоров степени 0.

Порядком дивизора D в точке P называется целое число m_P , $\text{ord}_P(D) = m_P$.

Основным дивизором называется дивизор $D = \text{div}(R)$, $D \in D^0$ некоторой рациональной функции $R \in \bar{k}(C)$. Множество основных дивизоров кривой C над полем $k < L < \bar{k}$, обозначается $P_C(L) = \{\text{div}(R) : R \in L(C)\}$, $P_C(L) \subset D^0$.

Группа основных дивизоров кривой C обозначается P или $P(C)$, причем $P(C) = P_C(\bar{k})$. Якобианом кривой C называется фактор группа $J = D^0/P$ или $J(C) = \text{Div}^0(C)/P(C)$. Если $D_1, D_2 \in D^0$ принадлежат одному классу эквивалентности дивизоров $[D]$ в $J_C(L)$, в этом случае D_1 и D_2 являются эквивалентными дивизорами $D_1 \sim D_2$, причем $D_1 - D_2 \in P$.

Дивизором рациональной функции $R = G/H$ является основной дивизор вида $\text{div}(G(u, v)) - \text{div}(H(u, v))$.

Носителем дивизора $D = \sum_{P \in C} m_P P$ называется множество $\text{supp}(D) = \{P \in C \mid m_P \neq 0\}$.

Почти приведенным дивизором называется дивизор вида $D = \sum m_i P_i - (\sum m_i) \infty$, где все $m_i \geq 0$, P_i – конечные точки, $P_i \in \text{supp}(D)$, причем $\tilde{P}_i \notin \text{supp}(D)$, за исключением $P_i = \tilde{P}_i$, т.к. $m_i = 1$.

Приведенным дивизором называется почти приведенный дивизор $D = \sum m_i P_i - (\sum m_i) \infty$, для которого верно $\sum m_i \leq g$.

Для каждого дивизора $D \in D^0$ существует почти приведенный дивизор $D_1 \in D^0$, такой, что $D \sim D_1$. Для удобства, почти приведенные дивизоры $D = \sum m_i P_i - (\sum m_i) \infty$, $P_i = (x_i, y_i) \in C_f$ представляются в предложенной Мамфордом форме $D = \text{div}(a, b) = \text{gcd}(\text{div}(a(u)), \text{div}(b(u) - v))$, где $a(u) = \prod (u - x_i)^{m_i}$ и $b(u)$ – уникальный полином удовлетворяющий условиям: (i) $\deg_u(b) < \deg_u(a)$; (ii) $b(x_i) = y_i$, для всех i таких, что $m_i \neq 0$; (iii) $a(u) \mid (b(u)^2 + b(u)h(u) - f(u))$.

Степенью дивизора D в этом случае является $\deg_u(a)$.

Задача дискретного логарифмирования на гиперэллиптической кривой (ЗДЛГЭК) формулируется для ГЭК $C(F_q)$ рода g и приведенных дивизоров $D_1, D_2 \in J(F_q)$ следующим образом: необходимо найти решение уравнения $D_2 = lD_1$ относительно l .

Решение ЗДЛГЭК является нетривиальной задачей и принадлежит классу $NP \cap co-AM$, где AM -класс алгоритмов, который имеют постоянное число итераций в игре Arthur-Merlin, в то время как ЗДЛЭК принадлежит четко классу $NP \cap co-NP$ [29, 48].

Общий алгоритм индексного исчисления (ИИ) считается одним из наиболее универсальных, на сегодняшний день, алгоритмом решения задачи дискретного логарифма (ЗДЛ), который обладает субэкспоненциальной сложностью [26].

Алгоритм 1 Индексного исчисления в общем виде.

Вход: $a, b \in G$, причем $a^k = b$.

Выход: $\lambda \in \mathbb{Z}$.

1. Выбор базы разложения: $S = \{g_1, \dots, g_k\} \subset G$, где $g_1 = a$, $g_2 = b$.
2. Генерация соотношений вида: $g_1^{e_{i1}} \dots g_k^{e_{ik}} = 1$, $1 \leq i \leq l$, до тех пор, пока $\text{rank}(\|e_{ij}\|) = k - 1$.
3. Решение системы линейных уравнений (СЛУ): $A\bar{x} \equiv 0 \pmod{|G|}$, $A = \|e_{ij}\|$.
4. Выделение решения: Зная $\bar{x} = (x_1, \dots, x_k)$, находится $\lambda \cdot \bar{x} \equiv (\log g_1, \dots, \log g_k) \pmod{|G|}$, для некоторого $\lambda \in \mathbb{Z}$. Дискретный логарифм соответствующий g_1 , вычисляется как $\lambda \equiv x_1^{-1} \pmod{|G|}$, при условии его существования.

Дальнейший анализ показывает, что величина k является предметом оптимизации алгоритма, т.е., если k маленькое, сложность поиска соотношений является высокой, с другой стороны, если k большое, то сложность решения СЛУ является высокой.

Этап генерации соотношений обладает внутренним параллелизмом, т.е. интервал, которому принадлежит решение, может быть разделен и просеян независимо. Параметром оптимизации данного подхода является алгоритм координации просеивания между подинтервалами.

Решение СЛУ над $\mathbb{F}_p$, $\log_2 p > 160$, для решения ЗДЛ стандартными методами, как известно, требует значительного количества памяти и вычислительных ресурсов. Поэтому были в [31] были адаптированы метод Ланкзоса решения разреженных СЛУ и сопряженный градиентный метод.

Широкое применение получили следующие методы решения СЛУ [25]:

- метод Гаусса;
- структурированный метод Гаусса;
- метод Вейдмана [32];
- точный итеративный метод Lanczos'a;
- точный итеративный сопряженный градиентный метод;
- метод bidiagonalization (Голуб, Кахан) для поля общего вида;
- bidiagonalization (Голуб, Кахан) для простого поля;
- оптимизированный метод Ланкзоса для простого поля;
- сопряженный метод;
- ортогональный метод.

Верхняя граница сложности алгоритмов решения СЛУ составляет $O(k^2 + k\omega)$, где ω - количество отличных от нуля элементов матрицы $A = \|e_{ij}\|$.

Подходы к решению ЗДЛГЭК

Для непосредственного применения алгоритма ИИ к ЗДЛГЭК, требуется учесть специфику группы (якобиана), в которой ищется решение. С этой целью проведем аналогию между понятиями поля и якобиана, являющимися базовыми для алгоритма ИИ.

Простым дивизором называют дивизор $D = \text{div}(a, b)$, состоящий лишь из одной точки, у которого полином $a = u - p$ является неприводимой полиномиальной функцией из $\bar{k}(u)$. Уравнение $v^2 + h(u)v - f(u) \equiv 0 \pmod{a} \Leftrightarrow v^2 + h(p)v - f(p) = 0$ в точке $u = p$ имеет два решения $b_1 \in \bar{K} \equiv \bar{K}[u]/\text{div}(a)$ и $b_2 = -b_1 - h(p)$. Решения соответствуют простому дивизору $P = (p, b_1)$ и его сопряжению $\bar{P} = (p, -b_1 - h(p))$, соответственно. Дивизоры, P и $\bar{P}$ называют *расщепленными*, если $P \neq \bar{P}$, иначе *разветвленными*.

Дивизор D является t -гладким, если его можно представить в виде $D = \sum_{i=1}^L e_i \text{div}(a_i, b_i)$ для $e_i \in \mathbb{Z}$ и $\max\{\deg a_i\} \leq t$, т.е. для него справедливо разложение $a = a_1^{e_1} a_2^{e_2} \dots a_L^{e_L}$, $b_i = b \bmod a_i$ на монические неприводимые полиномиальные функции над k .

В основе всех алгоритмов ИИ для решения ЗДЛГЭК лежит одна из трех стратегий [27]:

1. В основе первой стратегии лежит идея, предложенная Hafner'ом и McCurley'ом (НМ) для ЗДЛ в полях мнимых квадратичных чисел. Напомним, что структура $J_C(k)$ якобиана кривой C , определяется как прямая сумма циклических подгрупп. Представления дивизоров D_1 и D_2 задается посредством прямой суммы циклических подгрупп, тогда решение ЗДЛ сводится к решению СЛУ, для найденных представлений дивизоров D_1 и D_2 , составленной с использованием обобщенной китайской теоремы об остатках.

База разложения $S = \{P_1, P_2, \dots, P_n\}$ включает все расщепленные и разветвленные простые дивизоры $P_i = \text{div}(a_i, b_i)$, где $\deg a_i \leq t$ для некоторой границы гладкости t . Если неприводимый полином a_i расщепляемый, тогда только один из двух простых дивизоров над a_i : $\text{div}(a_i, b_i)$ или $\text{div}(a_i, -h - b_i)$ включаются в S . Генерация m , $m > n$ соотношений $\sum_j e_j P_j \sim 0$ осуществляется посредством поиска t -гладких основных дивизоров.

Если S образует $J_c(k)$, то отображение $\phi: \mathbb{Z}^n \rightarrow J_c(k)$, где $\phi(e_1, e_2, \dots, e_n) \mapsto \sum_j e_j P_j$, является сюръективным гомоморфизмом.

Каждое соотношение дает элемент $\vec{e}_i = (e_{i1}, e_{i2}, \dots, e_{in}) \in \ker(\phi)$ и, если множество из m отношений формирует полную систему $\ker(\phi)$, то согласно теореме Лагранжа [28] верно отношение эквивалентности $J_c(k) \cong \mathbb{Z}/d_1\mathbb{Z} \oplus \dots \oplus \mathbb{Z}/d_n\mathbb{Z}$, где $(d_1, d_2, \dots, d_n)$ – диагональные элементы нормальной формы Смита (SNF) матрицы соотношений $A = (\vec{e}_1 \vec{e}_2 \dots \vec{e}_n)$ (отношения записаны как столбцы матрицы A). Вычисление образующего элемента $X_i = \sum_{j=1}^n p_{ji} P_j$ для каждой циклической подгруппы $\mathbb{Z}/d_i\mathbb{Z}$ осуществляется посредством отыскания матриц преобразований с общим модулем $P = (p_{ij})$ и $Q = (q_{ij})$, таких, что $P^{-1}AQ = \text{SNF}(A)$.

Далее для дивизоров $D_1 \sim \sum \alpha'_i P_i$, $D_2 \sim \sum \beta'_i P_i$ ищется представление $(\alpha'_1 \dots \alpha'_n) = P^{-1}(\alpha_1 \dots \alpha_n)^T$ и $(\beta'_1 \dots \beta'_n) = P^{-1}(\beta_1 \dots \beta_n)^T$ в якобиане $J_c(k) \cong \mathbb{Z}/d_1\mathbb{Z} \oplus \dots \oplus \mathbb{Z}/d_n\mathbb{Z}$ посредством разложения над S . ЗДЛ состоит в решении СЛУ $\alpha'_i \equiv \beta'_i \pmod{d_i}$, $1 \leq i \leq n$, для нахождения $l \in \mathbb{Z}$.

2. Отличие второй стратегии от первой состоит в том, что считается известным порядок якобиана $\#J_c(k)$. После того, как будет найдено $n+1$ различных соотношений, это отличие позволяет заменить вычислительно сложное нахождение SNF, решением СЛУ по модулю $\#J_c(k)$ для нахождения нетривиальных линейных комбинаций $\sum_{i=1}^{n+1} \gamma_i \vec{e}_i = (0, 0, \dots, 0)$, т.е. $\sum_{i=1}^{n+1} \gamma_i R_i = 0$. На их основе составляется $\sum_{i=1}^{n+1} \gamma_i (\alpha_i D_1 + \beta_i D_2) = 0$ и решением ЗДЛ будет $\log_{D_1} D_2 = -\frac{\sum \gamma_i \alpha_i}{\sum \gamma_i \beta_i} \pmod{\#J_c(k)}$.

3. Гибридная стратегия. Эта стратегия была предложена в работе [1] для поля квадратных чисел, и может быть легко адаптирована к ЗДЛГЭК. Она является комбинацией двух предыдущих стратегий. Аналогично второй стратегии производится замена вычисления SNF на решение СЛУ над полем целых чисел по модулю $\#J_c(k)$. Ее отличие от предыдущих состоит в том, что генерируется n соотношений вида $\sum_j e_{ij} P_j = 0$, и два дополнительных соотношения $-D_1 + \sum_j e_{(n+1)j} P_j = 0$, $D_2 + \sum_j e_{(n+2)j} P_j = 0$. На этих соотношениях основе формируется матрица соотношений

$$B = \begin{bmatrix} A & \vec{e}_{n+1} & \vec{e}_{n+2} \\ \vec{0} & 1 & 0 \\ \vec{0} & 0 & 1 \end{bmatrix},$$

где $A = (\vec{e}_1 \vec{e}_2 \dots \vec{e}_n)$ – матрица, состоящая из первых n соотношений записанных в столбцах. Таким образом, равенство $D_1 = lD_2$ справедливо, если и только если существует решение $\vec{x} \in \mathbb{Z}^{n+2}$ матричного уравнения $B\vec{x} = (0, \dots, 0, 1, l)$. Решение ЗДЛ l (не обязательно наименьшее) получается из уравнения $l = \vec{b} \cdot \vec{y}$, где $B = \begin{bmatrix} B' \\ \vec{b} \end{bmatrix}$ и $\vec{y} \in \mathbb{Z}^{n+2}$ – решение уравнения

$B'\vec{y} = (0, \dots, 0, 1)$. Эффективность этой стратегии определяется существованием метода генерации соотношений вида $\sum_j e_{ij} P_j = 0$, который быстрее чем метод для генерации $\sum_j e_{ij} P_j = \alpha D_1 + \beta D_2$.

Укажем условия существования, для описанных выше стратегий [27], субэкспоненциального алгоритма со сложностью $O(L_n[c]) = \exp((c+o(1))\sqrt{\log n \log \log n})$:

1) дивизоры из S должны образовывать $J_c(k)$ и их количество составляет $n \in O(L_{q^s}[\rho])$ для некоторой константы $\rho > 0$;

2) количество случайно выбранных дивизоров будет равно $O(L_{q^s}[\frac{\rho}{\alpha}])$ для некоторой константы $\alpha \in (0, 1)$, перед тем, как будет найден t -гладкий дивизор (допущение, что размер базы разложения составит $O(L_{q^s}[\rho])$).

Оптимизация алгоритма ИИ заключается в поиске значения ρ , такого, что решение СЛУ и генерация соотношений будет обладать сравнительно одинаковой сложностью. Согласно первому условию $n \in O(L_{q^s}[\rho])$, сложность решения СЛУ составит $O(L_{q^s}[\rho])$, для некоторой константы $l > 0$, причем непосредственно решение СЛУ размером $n \times n$ составит $O(n^l)$. Сложность генерации соотношений зависит и от второго предположения о гладкости и, в частности:

- от сложности генерации дивизора и тестирования одного дивизора на гладкость;
- от количества гладких дивизоров необходимых для решения ЗДЛ.

Алгоритм решение ЗДЛ, построенный по описанным выше стратегиям, является вероятностным субэкспоненциальным алгоритмом типа Лас-Вегас (если решение получено, то оно гарантировано корректное). Тест ЗДЛ на разрешимость осуществляется проверкой образует ли множество соотношений ядро $\ker(\phi)$. Для этого проверяется равен ли определитель h' матрицы соотношений порядку якобиана $\#J_c(k)$. Если $\#J_c(k)$ – неизвестен, то проверяется неравенство $N^* < h' < 2N^*$, где $N^* < \#J_c(k) < 2N^*$. Определитель h' может быть эффективно вычислен посредством точной и эвристической оценки границы количества классов эквивалентности в поле гиперэллиптических функций [2].

Практические реализации алгоритмов ЗДЛГЭК

Для перехода к реальным оценкам сложности решения ЗДЛГЭК потребуется более детально изложить известные по литературе алгоритмы решения, согласно приведенным выше стратегиям.

Алгоритм ADN (Adleman, DeMarrais, Huang)

Алгоритм ADN впервые был предложен как алгоритм ИИ для решения ЗДЛГЭК. Он был описан для случая, когда q – нечетное простое. Соответствует стратегии 1. Генерация соотношений осуществляется посредством поиска разложения D_1 и D_2 в S на случайные почти приведенные основные дивизоры, образованные рациональными функциями вида $Av + B$, где $A, B \in k[u]$. Каждый такой гладкий дивизор получается из соотношения $(Av + B_i) = \sum_j e_{ij} P_j \sim 0$. Необходимо сгенерировать такое количество соотношений, при котором ранг матрицы соотношений будет равен n , это в свою очередь приведет к успешному решению ЗДЛГЭК согласно стратегии 1.

Разложение основного дивизора $(Av + B)$ на простые дивизоры $(Av + B) = \sum_j e_j P_j$ выполняется посредством разложения нормы функции $N(Av + B)$, т.е. $N(Av + B) = B^2 + ABh - A^2 = \prod_{j=1}^n a_j^{e_j'}$, где $P_j = \text{div}(a_j, b_j)$ – простые дивизоры из S . Известно, что $e_j = e_j'$, если $Ab_j + B \equiv 0 \pmod{a_j}$, иначе $e_j = -e_j'$.

Если $\log q \leq (2g+1)^{0.98}$, то алгоритм АДН как изложено в [6], имеет сложность $O(L_{q^{2g+1}}[c])$ для константы $0 < c < 2.313$. При получении этого результата в [6] сделаны следующие предположения:

- $J_c(k)$ образован из всех простых дивизоров степени не более $\log_q L_{q^{2g+1}}[c]$, для некоторой константы $c > 0$;

- $D = \text{div}(a, b)$ считается случайным дивизором. Вероятность, что D является l -гладким равна вероятности, что случайный полином степени $\deg a$ над k раскладывается на множители степени не более l .

В [6] сделан вывод, что количество попыток тестирования соотношения на пригодность составит $O(L_{q^{2g+1}}[\frac{9}{16c}])$ при условии, что $\log q \leq (2g+1)^{0.98}$.

Известно, что сложность решения СЛУ и генерации соотношений составляет $O(L_{q^{2g+1}}[lc])$ и $O(L_{q^{2g+1}}[\frac{9}{16c}])$, соответственно, где оптимальное значение было достигнуто при $c = \frac{3}{4\sqrt{l}}$. Количество итераций при которых выполняется решение ЗДЛГЭК (генерация соотношений и решение СЛУ) получится равным $O(L_{q^{2g+1}}[c])$, откуда общая сложность алгоритма составил $O(L_{q^{2g+1}}[c])$ при $c = \frac{3(l+1)}{4\sqrt{l}}$ [6]. Полезно будет отметить, выводы из результатов анализа [7] этого алгоритма свидетельствуют о том, что для решения СЛУ необходимо $l < 7.376$ и $c = 2.313$.

Алгоритм MST (Muller, Stein, Thiel)

Алгоритм MST [8] является развитием идей АДН, который применим к полям нечетной характеристики. В этом алгоритме не учитывается влияние отношения $g/\log q$ на сложность алгоритма [15]. Алгоритм соответствует стратегии 1. Для поля нечетной характеристики [9] и поля четной характеристики поля [10] было получено, что ЗДЛЭК полиномиально сводится к так называемой инфраструктуре ЗДЛ в действительном квадратичном поле функций. В свою очередь, ЗДЛГЭК эквивалентна инфраструктуре ЗДЛ над константным расширением поля [11].

Сделаем некоторые пояснения связанные с рассматриваемым подходом. Рассматриваем конечное поле нечетной характеристики $k = \mathbb{F}_q$ и кривую $C: v^2 = f$, где $f \in k[u]$ – монический, $\deg_u(f(u)) = 2g+2$, тогда $L = k(C) = k(u)(\sqrt{f})$ – действительное квадратичное поле функций. Инфраструктурой ЗДЛ (ИЗДЛ) называется задача нахождения $\deg(\alpha)$ приведенного основного идеала $I_a = aK[u] + (b+v)K[u]$ и $I_a = (\alpha)$ для заданных полиномов $a, b \in k[u]$, т.е. необходимо определить расстояние $\delta(I_a, O)$. Заметим, что существует уникальный, вплоть до константы, образующий элемент для I_a , такой, что $0 \leq \deg(\alpha) \leq R$.

На первом этапе формируется база разложения $S = \{I_{p_1}, I_{p_2}, \dots, I_{p_n}\}$ из всех расчлененных и разветвленных простых идеалов P_i с $\deg P_i \leq l$ для некоторой границы гладкости l . Отыскиваются m , $m > n$ l -гладких основных идеалов, которые образуют соотношения вида $I_b = \prod_j I_{p_j}^{e_j} = (\beta)$. Каждое такое соотношение дает элемент $\vec{v}_j = (e_{j1}, e_{j2}, \dots, e_{jn}, \deg(\beta_j)) \in \mathbb{Z}^{n+1}$, Γ_l – множество всех элементов $\vec{v}_j \in \mathbb{Z}^{n+1}$ и Γ'_l – множество $\vec{e}_j = (e_{j1}, e_{j2}, \dots, e_{jn}) \in \mathbb{Z}^n$. Известно, если множество S образует группу классов идеалов, то Γ_l является $(n+1)$ размерной решеткой с дискриминантом $\hat{h} = \hat{h}'R$ и Γ'_l является n размерной решеткой с дискриминантом $\hat{h}'$. Детально остановимся на процессе генерации соотношений:

- случайным образом выбирается $(e_1, e_2, \dots, e_n) \in \{0, \dots, q^{2g+2}\}^n$ и вычисляется $I_c = \prod_j I_{p_j}^{e_j}$;

- находится приведенный идеал $I_b \sim I_c$ и $\deg(\beta)$, где $\beta \in L$ такой, что $I_b = (\beta)I_c$;

- если I_b может быть разложено в базе разложения S , как $I_b = \prod_j I_{p_j}^{v_j}$, тогда $I_b I_c^{-1} = \prod_j I_{p_j}^{(v_j - e_j)} = (\beta)$ дает искомое соотношение;

- если была получена полная образующая система $\{\bar{v}_1, \dots, \bar{v}_m\}$ для решетки Γ_c , то удаляя последнюю координату в каждом векторе $\bar{v}_j$, получается полная образующая система $\{\bar{v}_1', \dots, \bar{v}_m'\}$ для решетки Γ_c' . Зададим матрицу $A = (\bar{v}_1 \bar{v}_2 \dots \bar{v}_m)$, где векторы $\bar{v}_j$ записаны в виде столбцов матрицы, аналогично $A' = (\bar{v}_1' \bar{v}_2' \dots \bar{v}_m')$;

- если I_a разлагается в S как $I_a = \prod_j I_{p_j}^{a_j}$, то $\bar{a} = (a_1, a_2, \dots, a_n) \in \Gamma_c'$, т.к. I_a – принципал. Следовательно, существует решение $\bar{x} = (x_1, x_2, \dots, x_m) \in \mathbb{Z}^m$ для СЛУ $A'\bar{x} = \bar{a}$. Таким образом, образующий элемент для I_a является $\alpha = \prod_j \beta_j^{x_j}$ и $\deg(\alpha) = \sum_j x_j \deg(\beta_j)$;

- если I_a не разлагается в S , необходимо выполнить нахождение приведенного, основного идеала $I_b \sim I_a$, который разлагается в базе. Для вычисления $\deg(\alpha \pmod R)$, необходимо определить R посредством вычисления $\det \Gamma_c$ и аналогично $\det \Gamma_c'$, а затем вычислить $R = \det \Gamma_c / \det \Gamma_c'$.

Отличие алгоритм MST от алгоритма АДН состоит в том, что алгоритм MST не используются эвристические предположения. В него заложено, что существует образующее множество для группы классов идеалов, чья размерность является полиномом в q^g , обычно множество простых идеалов степени $\lfloor 2 \log_q(4g-2) \rfloor$ является образующим множеством. В дополнение доказана точная оценка вероятности t -гладкости случайного приведенного дивизора.

Строгое описание алгоритма MST рассматривается в [8], там же показано, что если $2g+2 \geq \log q$, то при $n \in O(L_{q^{2g+2}}[\rho])$ количество попыток, прежде чем соотношение будет найдено, составит $O(L_{q^{2g+2}}[\frac{1}{4\rho}])$. Если решения СЛУ потребует $O(L_{q^{2g+2}}[5\rho])$, то генерация соотношений потребует $O(L_{q^{2g+2}}[2\rho + \frac{1}{4\rho}])$. Например, при количестве соотношений $m \in O(L_{q^{2g+2}}[\rho])$, выполняется $O(L_{q^{2g+2}}[\rho])$ произведений идеалов со случайной степенью. оптимальное значение $\rho = \frac{5}{2\sqrt{3}}$ и общая сложность составит $O(L_{q^{2g+2}}[1.44])$. Исходя из полученной сложности, делаем вывод, что алгоритм MST обладает меньшей сложностью, чем АДН.

Алгоритм FP (Flassenberg, Paulus)

В работе [21] впервые предложен подход НМ [13] по аналогии с полем квадратичных функций и полем квадратичных чисел для ЗДЛ над полями нечетной характеристики, а также применение алгоритмов Полларда, Похлинга – Хелмана, АДН для решения ЗДЛ в мнимых квадратичных полях (или ГЭК). Алгоритм соответствует стратегии 1. Процесс генерации соотношений состоит в следующем.

1. Случайным образом выбирается $(e_1, e_2, \dots, e_n) \in \mathbb{Z}^n$ и вычисляется $D = \sum_j e_j P_j$;
2. Выполняется приведение дивизор D к D' ;

3. Если D' является t -гладким с $D' = \sum_j v_j P_j$, то соотношением будет $D' - D = \sum_j (v_j - e_j) P_j = 0$.

Отсечение не перспективных путей решения является основным методом уменьшения сложности большинства алгоритмов решения NP -полных задач. Наиболее ценным в предложенном алгоритме является отсеивание, основанное на свойстве: для всех $f(X)$ с коэффициентами из кольца R , в том случае если $p, x_0 \in R$ и $p \mid f(x_0)$, то $p \mid f(x_0 + ip)$, для всех $i \in R$. Заметим, что в случае $R = \mathbb{Z}$, все $x \in [-M, M]$, для которых функция $f(x)$ является гладкой, могут быть найдены со сложностью соизмеримой со сложностью тестирования функции $f(x)$ используя «решето Эратосфена».

Суть отсеивания для ЗДЛГЭК состоит в том, что поиск произвольного t -гладкого дивизора $D' \sim D$ порождает искомое соотношение. Рассмотрим нормальную форму $f(X, Y) \in (k[u])[X, Y]$ дивизора $D = \text{div}(a, b)$. Для $\forall x, y \in k[u]$ существует дивизор $D' = \text{div}(f(x, y), b') \sim D$. Следовательно, можно отбросить полином $f(X) = f(X, 1)$ или даже $f(X, Y)$, как предложено в [14] для поиска x , таких, что $f(x)$ является t -гладким. Каждый такой полином x порождает необходимое соотношение.

В [14] показано, что сложность алгоритма FP [14] является значительно ниже сложности АДН [3].

Авторами [14] реализованы вычисления в $J_c(k)$ для кривых $v^2 = u^{2k+1} + 2u + 1$ над полями $\mathbb{F}_{11}, \mathbb{F}_{101}, \mathbb{F}_{1009}, \mathbb{F}_{10007}, \mathbb{F}_{100003}$ и рода $g \in [1, 12]$. Ими исследовались несколько вариантов варианты алгоритмов: без отсеивания, с пробным делением (версия НМ), с отсеиванием, алгоритм больших и малых шагов. Наибольшим быстродействием среди рассмотренных алгоритмов обладала реализация с отсеиванием. С возрастанием рода, алгоритм больших и малых шагов показал наибольшую производительность.

Алгоритм Smart

Работа [12] посвящена реализации варианта алгоритма АДН с использованием FP решета (на основе подхода Полларда), которое аналогично алгоритму факторизации решета для поля чисел.

Приведем основные этапы алгоритма Smart для кривой вида $C: v^2 = f(u)$ над полем $\mathbb{F}_q$, q – нечетное простое:

1. Строится база разложения S . В базу S заносятся все простые дивизоры из носителей дивизоров D_1 и D_2 . Затем, в отличие от АДН, заносятся все ветвящиеся дивизоры из поля квадратичных функций K . Следующими добавляется множество расщепленных простых дивизоров, степень которых не превышает g . Посредством параметра B производится равномерная регулировка количества элементов в S . Т.е. в базе разложения находится B простых дивизоров степени 1, $B/2$ дивизоров степени 2, $B/3$ дивизоров степени 3 и т.д.

2. Образование матрицы соотношений M . M содержит $n+2$ строки, причем первые 2 с соотношениями для D_1 и D_2 . Матрица заполняется из следующих соображений:

- а) $m_{1,1} = -1$, в остальные заносятся простые дивизоры из базы разложения, образующие D_1 ;
- б) $m_{2,2} = -1$, в остальные заносятся простые дивизоры из базы разложения, образующие D_2 ;
- в) в третью строку заносятся простые дивизоры из базы разложения, которые образуют $\text{div}(f)$, это дает соотношения между ветвящимися дивизорами.

3. Отсечение. Необходимо найти полиномы $a, b \in \mathbb{F}_q[u]$, такие, что дивизор функции $f = av + b$ имеет носитель, в который входят лишь элементы из базы разложения. Каждой такой

полиномиальной функции $f \in F_q[u]$ ставится в соответствие уникальный идентификатор (образ) $e \in \mathbb{N}$ для индексирования массива отсечения. Массив отсечения – двумерный массив, индексирующий полиномиальные функции согласно их образам. Каждый элемент инициализируется значением $\deg(N_{K/F_q[u]}(av+b)) = \deg(b^2 - a^2 f)$ согласно предположению [12], где a и b полиномиальные функции, чьи идентификаторы задают строку и столбец индекса массива.

4. Отсечение. Производится понижение сложности алгоритма решета поля чисел посредством замены стандартного решета на решето Полларда [39].

5. Решение СЛУ $Mx = 0 \pmod{n}$ в поле F_n . Размеры матрицы предлагается уменьшать за счет перехода к нормальной форме Эрмита или предварительном применении структурированного метода Гаусса. В результате получается верхняя треугольная матрица с первым столбцом, отличным от нуля.

6. Выделение решения. Выполняется параллельное тестирование нескольких основных дивизоров вида $(Av+B)$ на гладкость. Проведенные эксперименты показали, что алгоритм АДН с отсечениями является не эффективным на практике.

Результаты применения предложенного алгоритма для решения ЗДЛГЭК для кривых над полями F_{11} с родами 1-9, F_{101} с родами 1-3 и F_{1009} с родом 1 описаны в [12].

Алгоритм Pollard

Метод ρ -Полларда [3, 4] может быть эффективно реализован, для ГЭК имеющих нетривиальный эффективно вычислимый автоморфизм. Для автоморфизма σ порядка m в $J_C(k)$, G_i обозначает некоторое представление дивизора R . Если R разлагается как $R = P_1 + P_2 + \dots + P_r$ на простые дивизоры P_i степени меньшей либо равной t , то R можно записать в виде $R = \theta^1 G_1 + \dots + \theta^t G_t$, $0 \leq t \leq m$. Этот прием позволяет уменьшить размер базы разложения в m раз, и тем самым уменьшить сложность решения СЛУ в m^2 раз [5].

Алгоритм больших и малых шагов

Подход описанный в [30] позволяет понизить сложность всех алгоритмов решения ЗДЛГЭК, благодаря информации об остатке по модулю m характеристического полинома эндоморфизма Фробениуса для якобиана ГЭК, т.е. группа классов почти приведенных дивизоров разбивается на m классов смежности. Так, сложность классического алгоритма составляет $O(q^{(2g-1)/4}/\sqrt{m})$, в то время, как алгоритм больших и малых шагов обладает сложностью $O(q^{g(g-1)/8}/m^{g/2})$, где $m = \Omega(q^{g+(g-2)/4})$, $\forall \varepsilon > 0$.

Алгоритм Бауер и Енге для произвольных конечных полей

Алгоритм АДН был одновременно обобщен для ГЭК над произвольными конечными полями в работах [6] и [15] независимо. Алгоритм [6] схож с АДН и имеет соизмеримую сложность при одинаковых эвристических допущениях. Аналогично FP, алгоритм [15, 16] аналогичен методу НМ [13], где соотношения образуются без отсеивания. В отличие от алгоритма [6], алгоритм [15] основан на строго доказанных утверждениях. База разложения образует $J_C(k)$ согласно обоснованию [8]. Доказательство положения о гладкости производится в [17]. Сложность образования базы разложения составляет $O((g^2 \log^2 q) L_{q^g} \lfloor \rho + \frac{1}{\nu} \rfloor) \subseteq O(L_{q^g} \lfloor \rho + \frac{1}{\nu} \rfloor)$ битовых операций. Причем известно, что простые дивизоры из $J_C(k)$, имеют степень $\lfloor 2 \log_q (4g-2) \rfloor$. В [15] впервые показана зависимость сложности от соотношения между g и $\log q$, что если выполняется условие $g \geq \nu \log q$ (условие субэкспоненциальной сложности алгоритма) для положительной константы ν , то количество соотношений составит $n \in O(L_{q^g} \lfloor \rho + \frac{1}{\nu} \rfloor)$ и количество попыток поиска соотношения огра-

ничено $\frac{5}{4}\sqrt[4]{e}(2g+1)L_{q^*}(\frac{1}{4\rho}) \in O(L_{q^*}[\frac{1}{2\rho} + o(1)])$. Сложность решения СЛУ составляет $O(L_{q^*}[l\rho])$ или $O(L_{q^*}[2(\rho + \frac{1}{\sqrt[4]{\rho}}) + o(1)])$, откуда сложность генерации соотношений составит $O(n^2 L_{q^*}[\frac{1}{4\rho} + o(1)]) \subseteq O(L_{q^*}[2\rho + \frac{2}{\sqrt[4]{\rho}} + \frac{1}{2\rho} + o(1)])$, так как требуется n соотношений. Общая сложность алгоритма составляет $O(L_{q^*}[\max\{5(\rho + \frac{1}{\sqrt[4]{\rho}}), 2(\rho + \frac{1}{\sqrt[4]{\rho}}) + \frac{1}{4\rho}\} + o(1)])$ [16]. Оптимизацию параметра ρ производят посредством задания функции $f: \rho \mapsto \max\{5(\rho + \frac{1}{\sqrt[4]{\rho}}), 2(\rho + \frac{1}{\sqrt[4]{\rho}}) + \frac{1}{4\rho}\}$, которая имеет единственный глобальный минимум. Полагая, что $l=4$ для этапа решения СЛУ получается оптимальное значение $\rho = \frac{1}{2}(\sqrt{1+\frac{1}{v}} - \frac{1}{\sqrt[4]{v}})$ и общая сложность составит $O(L_{q^*}[c])$ при $c = 2(\sqrt{1+\frac{1}{v}} + \frac{1}{\sqrt[4]{v}})$. Для алгоритма [15] были получены упрощенные теоретические оценки сложности алгоритма решения ЗДЛГЭК в $O(\exp(\rho\sqrt{(g \log q)\log(g \log q)}))$ битовых операциях, где $\rho = \frac{5}{2\sqrt{3}}(\sqrt{1+\frac{1}{v}} + \sqrt{\frac{2}{v}}) + o(1)$ для константы $v > 0$, $o(1) \rightarrow 0$ при $g \log q \rightarrow \infty$. Программная реализация алгоритмов не производилась, поэтому отсутствуют результаты экспериментов.

Алгоритм Gaudry для небольшого рода ГЭК

Среди приведенных выше алгоритмов, отсутствуют алгоритмы, спроектированные с известным порядком якобиана $\#J_C(k)$. Такой алгоритм был впервые предложен в работе [5]. Он соответствует стратегии 2, и спроектирован с учетом знания порядка $\#J_C(k)$. После образования базы разложения S , куда включаются только простые дивизоры степени 1, выполняются случайные аддитивные шаги на множестве приведенных дивизоров эквивалентных $\alpha D_1 + \beta D_2$ (подход Теске [18]). Каждый встречающийся 1-гладкий дивизор порождает соотношение. Шаги выполняются, пока не будет сгенерировано необходимое количество соотношений.

Эффективность процесса получения соотношений, обосновывается фактом, что каждый новый кандидат образуется посредством случайного аддитивного шага, требующего единственное сложение в $J_C(k)$. Для тестирования дивизора $\text{div}(a, b)$ на 1-гладкость, проверяется равенство $\gcd(u^q - u, a) = \deg a$ (доказывается, что a не имеет кратных множителей), т.к. $\gcd(u^q - u, a)$ является произведением различных полиномов степени 1, являющихся делителями a . На практике, вероятность наличия у a кратных множителей или проигнорируют или обнаруживают, вычисляя степень $\gcd((u^q - u)^e, a)$.

Приведем алгоритм в общем виде.

1. Построение базы разложения. Для каждого u_i над F_q степени 1, ищется v_i , такой, что $\text{div}(u_i, v_i) \in J_C(k)$. В случае его существования, $\text{div}(u_i, v_i)$ сохраняется в базе S лишь один из двух $\text{div}(u_i, v_i)$ или $-\text{div}(u_i, v_i)$.

2. Инициализация случайных шагов. Для $j = \overline{1, r}$ выбираются случайные $\alpha_j, \beta_j \in [1, n]$ и вычисляются шаги $T_j = \alpha_j D_1 + \beta_j D_2$. Начальная точка $R_0 = \alpha_0 D_1 + \beta_0 D_2$.

3. Вычисление $j = H(R_0)$, $R_0 = R_0 + T_j$, $\alpha_0 = \alpha_0 + \alpha_j \bmod n$ и $\beta_0 = \beta_0 + \beta_j \bmod n$, до тех пор, пока $R_0 = \text{div}(u_0(z), v_0(z))$ является гладким.

4. Разложение R_0 в базе S посредством факторизации $u_0(z)$ над F_q и определение позиций множителей в S . Результат разложения $R_k = \sum m_{ik} g_i$ заносится как строка в матрицу $M = \|m_{ik}\|$. Пересопределение $\alpha_k = \alpha_0$ и $\beta_k = \beta_0$.

5. Если $k < \#S + 1$, то $k = k + 1$ и переход к п.3.

6. Решение СЛУ в поле F_n . Поиск не нулевого вектора $\|\gamma_k\|$ из ядра матрицы M^T .

7. Если $\sum \beta_k \gamma_k \neq 0$, то $\lambda = (\sum \alpha_k \gamma_k) / (\sum \beta_k \gamma_k) \bmod n$ иначе переход к п.2.

8. Вернуть λ .

Обозначим через I_J – сложность операций в $J_C(F_q)$, I_q – сложность операций в F_q , I_{gq} – сложность операций в кольце полиномиальных функций $k[C]$, I_n – сложность операций в F_n , $n \approx q^2$.

Согласно введенным обозначениям сложность алгоритма по пунктам составит:

1. $O(qI_q)$;
2. $O(I_J \log n)$;
3. $O(g!(I_J + I_n + I_{gq}))$;
4. $O(qI_{gq})$.

Количество итерации цикла 3-5 составит $\#S = O(q)$, сложность всего цикла $O(qg!(I_J + I_n + I_{gq})) + O(q^2 I_{gq})$;

6. Размер матрицы составляет $O(q)$, количество отличных от нуля элементов матрицы M составит $O(gq)$. Сложность решения СЛУ алгоритмом Ланкзоса составит $O(gq^2 I_n)$;

7. $O(qI_n)$.

Общая сложность алгоритма составляет: $O(g!qI_J) + O((g!q + gq^2)(I_n + I_{gq})) + O(qI_q)$.

Зафиксировав g и варьируя величиной q , в [5] показано, что алгоритм имеет сложность $O(g^3 q^2 \log^2 q + g^2 g! q \log^2 q)$ битовых операций. При аналогичных условиях алгоритмы общего вида решения ЗДЛ имеют сложность $O(q^2)$. Результаты анализа [5] показывают, что для $g > 4$, ЗДЛГЭК может быть решена эффективнее, чем позволяют алгоритмы общего вида. Исследования [5] уточняют в отношении какого рода допускается более эффективный алгоритм решения ЗДЛ, чем общий алгоритм.

В работе [5] рассматривается два примера с порядком $\#J_C(k) = 10^{40}$, не обладающие эффективно вычислимым групповым автоморфизмом в $J_C(k)$, ГЭК рода 6 над полем $F_{5026243}$ и $F_{2^{20}}$. Практическое применение алгоритма ρ -Полларда для заданных $J_C(k)$ оказалось не целесообразным по причине достаточно большого $\#J_C(k)$.

Алгоритм DGM (Duursma, Gaudry, Morain)

В алгоритме DGM [41], для снижения сложности получения решения ЗДЛГЭК в $\sqrt{m}$, предложено использовать автоморфизм порядка t , который имеется у некоторых кривых, раз. Кривые, для которых известен эффективно вычисляемый автоморфизм и к которым был применен подход [41], приведены в табл. 1.

Таблица 1

Автор	Кривая	По- ле	Автоморфизм	Порядок автомор- физма t
Коблиц	$v^2 + v = u^5 + u^3 (*)$	F_{2^n}	Фробениус $\begin{cases} u \mapsto u + 1 \\ v \mapsto v + u^2 \end{cases}$	$4n$
	$v^2 + v = u^5 + u^3 + u (*)$	F_{2^n}	Фробениус	$2n$

	$v^2 + v = u^{2g+1} + u$ $v^2 + v = u^{2g+1}$	F_{2^n} F_{2^n}	Фробениус Фробениус	$2n$ $2n$
Бахлер, Коблиц, Чао и др.	$v^2 + v = u^{2g+1} (+), p \equiv 1(2g+1)$	F_p	Комплексное умножение в поле $\mathcal{Q}(\zeta_{2g+1})$	$2(2g+1)$
Сакаи, Сакураи	$v^2 + v = u^{13} + u^{11} + u^9 + u^5 + 1 (+)$	$F_{2^{29}}$	Фробениус + $\begin{cases} u \mapsto u+1 \\ v \mapsto v+u^6+u^5+ \\ +u^4+u^3+u^2 \end{cases}$	$4 \cdot 29$
Дуур- сма, Сакураи	$v^2 = u^p - u + 1 (+)$	F_{p^n}	Фробениус + $(u, v) \mapsto (u+1, v)$	$2np$

где (*) – кривые, которые были взломаны методом Fray-Ruck [42] эффективность которого гораздо выше метода ρ -Полларда даже при наличии автоморфизма: (+) – кривые большого рода, которые были взломаны алгоритмом ИИ [5]; ζ – корень пятой степени из -1.

Практическая реализация DGM приведена в работе [41] для двух кривых:

- $v^2 = u^5 - 1$ над полем F_{31^3} , где порядок дивизора образующего группу $n = 778201$;

- $v^2 + v = u^5 + u^3 + u$ над полем F_{2^n} , где порядок дивизора образующего группу $n = 599479$.

Возможно уменьшение сложности за счет уменьшения размера базы разложения [19]. при соотношении «пригодных» дивизоров $1/l$, сложность генерации соотношений увеличится на множитель l^g , но потребуются в l раз меньше соотношений и сложность решения СЛУ $O(L_{q^{2g+1}}[lc])$ уменьшится в l^2 раз. При оптимальном значении $l = O((q/g!)^{1/(g+1)})$, общая сложность составит $O\left(q^{\frac{2g}{g+1}+c}\right)$ для $q \rightarrow \infty$.

Алгоритм GHS (Gaudry, Hess, Smart)

В [19, 43, 51] рассматривается модифицированный алгоритм [5] для решения 3ДЛГЭК для ГЭК рода 4 над полем $F_{2^{21}}$. Суть модификации состоит в использовании усеченной базы разложения, которая включает лишь простые дивизоры степени 1 обладающие нормой $v + \alpha$, причем $\alpha \equiv 0(\text{mod } x^3)$. К тому же, для уменьшения количества соотношений и размера матрицы соотношений, большинство гладких дивизоров $\text{div}(a, b)$ отбрасываются, посредством быстрого теста $\alpha \equiv 0(\text{mod } x^3)$. Общая сложность алгоритма оказалась ниже на 25% чем у ρ -Полларда: $O(N \cdot L_{q^{21}}[\frac{1}{2}, 3.54\sqrt{\log q}])$ групповых операций, при $g \rightarrow \infty$, $N = \deg, \mathbb{C}(x, y)$. Анализ сложности алгоритма показывает, что для ГЭК рода 4 можно предложить алгоритм, который будет эффективнее общего.

Алгоритм EG (Enge, Gaudry)

Авторы [20] предложили свой вариант подхода [5] для ГЭК большого рода, в которых известен порядок группы и сохраняется предположение о гладкости. Он соответствует стратегии 2. Как и в алгоритме [16], база разложения содержит все простые дивизоры степени меньшей t и тестирование на гладкость расширилось проверкой равенства $a = \text{lcm}(\{\gcd(u^i - u, a) \mid i \in [1, t]\})$ (проверка на наличие кратных множителей a может быть проведена как и описывалось в [5]).

В [38] используются случайные шаги $\alpha D_1 + \beta D_2$ [39] в $J_C(k)$ и для получения t -гладких дивизоров (все полученные гладкие дивизоры сохраняются), которые образуют соотношения

дивизоров в базе разложения, решается СЛУ $\sum_{i=1}^{k+1} \gamma_i (e_{i1}, e_{i2}, \dots, e_{ik}) = (0, 0, \dots, 0)$ или $\sum_{i=1}^{k+1} \gamma_i R_i = 0$, одним из известных методов. На выходе имеем $\sum \gamma_i (\alpha_i D_1 + \beta_i D_2) = 0$, откуда $\log_{D_2} D_2 = -(\sum \gamma_i \alpha_i) / (\sum \gamma_i \beta_i) \bmod r$.

Используя эффект гладкости [20], в [16] доказывается утверждение, что сложность составит $O(L_{q^c}[c])$. $c = \sqrt{2} \left(\sqrt{1 + \frac{1}{2\vartheta}} + \frac{1}{\sqrt{2\vartheta}} \right)$ при $g \geq \vartheta \log q$ для некоторой константы $\vartheta > 0$. Доказательство следует из факта, что зная порядок $J_c(k)$, возможно использование рандомизированного метода Ланкзоса для решения СЛУ вместо вычисления SNF матрицы целых чисел. Следовательно, сложность решения СЛУ $O(L_{q^{2+l}}[lc])$ будет минимизирована при $l = 2$, которое предпочтительнее чем 4, и может быть использовано для получения заявленной в [41, 44, 52] сложности.

Алгоритм JMS (Jacobson, Menezes, Stein)

В [21] реализована улучшенная версия алгоритма EG [20] для решения ЗДЛГЭК рода 31 над полями $F_{2^2}, F_{2^3}, F_{2^4}, F_{2^5}$. Особенностями предложенного алгоритма являются: оптимизация стратегии проверки на гладкость и выбор оптимальной границы гладкости t . Общая сложность алгоритма составляет $O(\exp((1+o(1))\sqrt{(2g+1)\log q \log((2g+1)\log q)}))$.

Алгоритм Theriault'a

Из [22] известна улучшенная версия алгоритма [5] для небольшого рода. Авторы [22] предлагают несколько модификаций. Первая модификация состоит в использовании лишь части простых дивизоров степени 1 в базе разложения, это в свою очередь снижает вероятность поиска гладких дивизоров, и сложность решения СЛУ снижается (имеет меньшую размерность). Решив задачу балансировки сложности алгоритмов генерации соотношений и решения СЛУ [22], было получено оптимальное количество простых дивизоров степени 1. В этом случае общая сложность алгоритма составит $O\left(g^5 q^{\frac{2-2}{2+1}+\epsilon}\right)$. Вторая модификация заключается в хранении следа полностью разлагаемых в базе разложения дивизоров, за исключением единственного «большого простого» дивизора (метод «больших простых»).

В случае нахождения двух соотношений, которые или содержат одинаковые простые множители или содержат большие простые множители, являющиеся отрицанием друг друга, то такие соотношения комбинируются для формирования соотношения. Во втором случае общая сложность составит $O\left(g^5 q^{\frac{2-2}{2+1}+\epsilon}\right)$.

Оба алгоритма являются асимптотически быстрее алгоритма [5], при условии, что $q > (g-1)!$ для первого случая и $q > (g-1)!/g$ для второго и являются асимптотически быстрее общего алгоритма для $g \geq 3$. Следовательно ГЭК рода 3 не являются стойкими, как считалось ранее.

Спуск Вейля

В работе [23] показана возможность применение спуска Вейля [37] для атаки на ЗДЛГЭК небольшого рода над F_{2^m} с составной степенью расширения. Впервые в [37] сводят ЗДЛГЭК над полем F_{q^n} к ЗДЛГЭК в якобиане ГЭК большего рода над некоторым подполем F_q поля F_{q^n} . Приведем условия, применения атаки:

1. Числа l и n положительные целые.
2. Причем $q = 2^l$, $k = F_q$ и $K = F_{q^n}$.

2. Причем $q = 2^l$, $k = F_q$ и $K = F_{q^n}$.

3. Не суперсингулярная ЭК E над K задана уравнением $E: y^2 + xy = x^3 + ax^2 + b$, $a \in K$, $b \in K^*$.

4. Порядок ЭК $\#E(K) = hr$, где h небольшое целое 2 или 4, r - простое. Следовательно $n \approx q^n$. Пусть $b_i = b^{q^i}$ и $m(b) = \dim_{F_2}(\text{Span}_{F_2}\{(1, b_0^{1/2}), (1, b_1^{1/2}), \dots, (1, b_{n-1}^{1/2})\})$. Положим, что или n является нечетным или $m(b) = 0$ или $\text{Tr}_{K/F_2}(a) = 0$.

5. Построение ограничения Вейля $W_{E/K}$ скалярных величин $E(K)$, которое является n -размерным абелевым многообразием над k .

6. В результате пересечения $W_{E/K}$ с $n-1$ гиперплоскостью, получается ГЭК C , рода $g = 2^{m-1}$ или $g = 2^{m-1} - 1$, где $m = m(b)$.

Обычно $m \approx n$, т.е. $g \approx 2^{n-1}$. Известно, что $\#J_C(k) \approx q^g \approx q^{2^{n-1}}$. Далее решение ЗДЛГЭК возможно одним из описанных выше методов ИИ.

В [19, 43, 44] показано применение спуска Вейля для сведения ЗДЛЭК в подгруппе точек $E(K)$ порядка r к подгруппе $J_C(k)$ порядка r ГЭК $C(k)$ рода g . Атака спуском Вейля практически не применима к кривым заданным над простым полем и двоичным расширением, когда степень расширения больше 40 и является простым числом [23].

Применение спуска Вейля к большому классу ЭК, заданных над полем характеристики отличной от 2, благодаря эффективному поиску кривых изогенных для заданной (взломанной) кривой, рассмотрено в [45-47].

Сравнение

Результатирующие оценки сложности различных алгоритмов ЗДЛГЭК обобщена в табл.2.

Таблица 2

№	Название алгоритма	Условие применения	Сложность
1	Adleman, DeMarrais, Huang	$C(F_q)$, q – нечетное простое, g – большое	$O(L_{q^{2g+1}}[c])$, $c = 2.313$
2	Muller, Stain, Theil	$C(F_q)$, q – нечетное простое, g – большое	$O(L_{q^{2g+2}}[1.44])$
3	Flassenberg, Paulus	$C(F_q)$, q – нечетное простое, g – большое	?
4	Smart	$C(F_q)$, q – нечетное простое, g – большое	?
5	ρ -Pollard	$J_C(F_q)$ содержит m классов смежности	$O(q^{(2g-1)/4}/\sqrt{m})$
6	Больших и малых шагов	$C(F_q)$	$O(q^{g(g+1)/8}/m^{g/2})$, $m = \Omega(q^{\varepsilon+(g-2)/4})$, $\forall \varepsilon > 0$
7	Bauer & Enge	$C(F_q)$	$O(\exp(\rho\sqrt{(g \log q) \log(g \log q)}))$, $\rho = \frac{5}{2\sqrt{3}}(\sqrt{1+\frac{3}{v}} + \sqrt{\frac{3}{v}}) + o(1)$, $v > 0$, $o(1) \rightarrow 0$ при $g \log q \rightarrow \infty$
8	Gaudry	$C(F_q)$, известен $\#J_C(F_q)$, g – небольшое	$O(g^3 q^2 \log^2 q - g^2 g! q \log^2 q)$

9	Duursma, Gaudry, Morain	$C(F_q)$, известен порядок автоморфизма Фробениуса	$O\left(q^{\frac{2g}{g+1}+\epsilon}\right)$ при $q \rightarrow \infty$
10	Gaudry, Hess, Smart	ГЭК рода 4 над $F_{2^{21}}$	$O\left(N \cdot L_{e^*}\left[\frac{1}{2}, 3.54\sqrt{\log q}\right]\right)$, при $g \rightarrow \infty$, $N = \deg, C(x, y)$
11	Enge, Gaudry	g – большое	$O\left(L_{q^*}[c]\right)$, $c = \sqrt{2}\left(\sqrt{1+\frac{1}{2g}} + \frac{1}{\sqrt{2g}}\right)$
12	Jacobson, Menezes, Stein	$g = 31, F_{2^m}$	$O\left(\exp\left((1+o(1))\sqrt{(2g+1)\log q \log((2g+1)\log q)}\right)\right)$
13	Theriault	$C(F_q)$, g – небольшое	$O\left(g^5 q^{2-2/(g+1)+\epsilon}\right)$ при $q > (g-1)!$ и $O\left(g^5 q^{2-4/(2g+1)+\epsilon}\right)$ при $q > (g-1)/g$

Из данных, приведенных в табл. 2, следует, что алгоритмы 1-4, 11 и 12 эффективно решают ЗДЛГЭК большого рода, следовательно, в криптографических целях имеет смысл применять лишь кривые небольшого рода. В то же время, алгоритмы 8, 10, 13 являются эффективными для ГЭК рода более 4. Причем алгоритм 13 уменьшает границу до 3, в этом случае он имеет сложность меньшую, чем сложности универсальных алгоритмов.

Как известно [18], что для противостояния универсальным алгоритмам криптоанализа (полного перебора, Полларда, Похлинга-Хелмана, больших и малых шагов) предлагается использовать ГЭК $\#J_c(F_q) = q^g \geq 2^{160}$. Исследования [3, 25] показывают, что кривые с эффективно вычислимым эндоморфизмом Фробениуса являются уязвимыми. Наличие эндоморфизма позволяет повысить эффективность универсальных алгоритмов, а также позволяет применить специализированные алгоритмы, например алгоритм 9 в табл. 2.

Выводы

В работе рассмотрены наиболее эффективные алгоритмы решения ЗДЛГЭК, из известных на сегодняшний день. Полученные результаты подтверждают факт, что дальнейшее развитие асимметричной криптографии лежит в увеличении сложности кривой (рода), т.е. перехода к ГЭК. Повышая род, повышается и сложность решения ЗДЛ, причем возможно уменьшение размера базового поля без ущерба стойкости. Другими словами, ГЭК позволяют обеспечить аналогичную сложность ЗДЛ при соизмеримых параметрах с ЭК, в тоже время дальнейшее повышение сложности кривой (род > 3) – не целесообразно, т.к. возможно эффективное применение алгоритма 13 (табл. 2). Уменьшенный размер базового поля открывает «новые горизонты» для ГЭК, которыми являются мобильные устройства.

Анализ алгоритмов позволил сформулировать подходы к реализации скрытых резервов эффективности посредством:

- уменьшения размера базы разложения;
- прореживания (отсеивание) соотношений на этапе их генерации (FP метод, Smart метод);
- решения, как разреженных, так и плотных СЛУ над полем;
- распараллеливания (например генерации соотношений, решения СЛУ);
- использования повторного логарифмирования [53, 54];
- гармонизации подходов к ЗДЛГЭК, используемых для решения ЗДЛ над полем и задачи факторизации.

Остается открытым вопрос о существовании алгоритма криптоанализа преобразований на ГЭК со сложностью соизмеримой с $O\left(\exp\left((c+o(1))m^{1/3}(\log m)^{2/3}\right)\right)$ для алгоритма криптоанализа Адлеманна-Копперсмита в поле [49, 50]. Дальнейшего исследования требует классификация кривых уязвимых к атаке спуском Вейля.

Список литературы: 1. *Vollmer U.* Asymptotically fast discrete logarithms in quadratic number fields. Algorithmic number theory-ANTS-IV, LNCS 1838, 200, pp.581-594. 2. *Stein A., Teske E.* Explicit bounds and heuristics on class numbers in hyperelliptic function fields. Mathematics of computation, 71 (2002), pp.837-861. 3. *Gallant R., Lambert R., Vanstone S.* Improved the parallelized Pollard lambda search on anomalous binary curves. Mathematics of computation, 69 (2000), pp.1699-1705. 4. *Wiener M., Zuccherato R.* Faster attacks on elliptic curve cryptosystems. Selected areas in cryptography, LNCS 1556, 1999, pp.190-200. 5. *Gaudry P.* An algorithm for solving the discrete log problem on hyperelliptic curves. Advances in cryptology-EUROCRYPT'2000, LNCS 1807, 2000, pp.19-34. 6. *Bauer M.* A subexponential algorithm for solving the discrete logarithm problem in the Jacobian of high genus hyperelliptic curves over arbitrary finite fields. Preprint, 1999. 7. *Adleman L., DeMarrais J., Huang M.* A subexponential discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields. Algorithmic number theory-ANTS-II, LNCS 877, 1994, pp.28-40. 8. *Muller V., Stain A., Thiel C.* Computing discrete logarithms in real quadratic congruence function fields of large genus. Mathematics of computation, 68 (1999), pp.807-822. 9. *Stein A.* Equivalences between elliptic curves and real quadratic congruence function fields. Journal de Theorie des nombres de Bordeaux, 9 (1997), pp.75-95. 10. *Zuccherato R.* The equivalence between elliptic curve and quadratic function field discrete logarithms in characteristic 2. Algorithmic number theory-ANTS-III, LNCS 1423, 1998, pp.621-638. 11. *Paulus S., Ruck H.-G.* Real and imaginary quadratic representations of hyperelliptic function fields. Mathematics of computation, 68 (1999), pp.1233-1241. 12. *Smart N.* Experiments using an analogy of the number field sieve algorithm to solve the discrete logarithm problem in the Jacobians of hyperelliptic curves. HP Laboratories Bristol, Technical report HPL-97-130, 1997. 13. *Hafner J., McCurley K.* A rigorous subexponential algorithm for computation of class groups. Journal of the American Mathematical Society, 2 (1989), pp.837-850. 14. *Flassenberg R., Paulus S.* Sieving in function fields. Experimental mathematics, 8 (1999), pp.229-249. Available at: <http://ftp.informatik.tu-darmstadt.de/pub/TI/TR/TI-97-13.rafla.ps.gz>. 15. *Enge A.* Computing discrete logarithms in high-genus hyperelliptic jacobians in provably subexponential time. Mathematics of computation, 71 (2001), pp.729-742. 16. *Enge A.* A general framework for subexponential discrete logarithm algorithms in groups of unknown order. Finite geometries, Developments in mathematics vol. 3, Kluwer Academic publishers, Dordrecht 2001, pp.133-146. 17. *Enge A., Stein A.* Smooth ideals in hyperelliptic function fields. Mathematics of computation, 71 (2001), pp.1219-1230. 18. *Teske E.* Speeding up Pollard's rho method for computing discrete logarithms. Algorithmic number theory, LNCS 1423, 1998, pp.541-554. 19. *Gaudry P., Hess F., Smart N.* Constructive and destructive facets of Weil descent on elliptic curves. Journal of cryptology, 15 (2002), pp.19-46. 20. *Enge A., Gaudry P.* A general framework for subexponential discrete logarithm algorithms. Acta arithmetica, 102 (2002), pp.83-103. 21. *Jacobson M., Menezes A., Stein A.* Solving elliptic curve discrete logarithm problems using Weil descent. Journal of the Ramanujan mathematical society, 16 (2001), pp.231-260. 22. *Theriault N.* Index calculus attack for hyperelliptic curves of small genus. Preprint, 2003. 23. *Galbraith S.* Weil descent of jacobians. Discrete applied mathematics, 128 (2003), pp.165-180. 24. *Shoup V.* Lower bounds for discrete logarithms and related problems. Advances in cryptology – EUROCRYPT'97, LNCS 1233, pp.256-266. 25. *Lambert R.* Computation aspects of discrete logarithms. A thesis for the degree of DP in electrical engineering. Waterloo, Ontario, Canada, 1996. 26. *Buchmann J., Weber D.* Discrete logarithms: recent progress. 1998. 27. *Jacobson M. Jr., Menezes A., Stein A.* Hyperelliptic curves and cryptography. 28. *Тезяшев А.Д.* Алгебра и геометрия. Алгебраические структуры. Категории. – Харьков: ХТУРЕ, 1998. – 388с. 29. *Shizuya H., Itoh T., Sakurai K.* On the complexity of hyperelliptic discrete logarithm problem. 30. *Matsuo K., Chao J., Tsuijii S.* Baby step gaint step algorithms in point counting of hyperelliptic curves. Special section on discrete mathematics and its applications of IEICE. Vol. E86-A, No. 5, May 2003, pp.1127-1134. Japan. 31. *Odlyzko A.M.* Discrete logarithms in finite fields and their cryptographic significance. Advances in cryptology – EUROCRYPT'84 (T. Beth, N. Cot, I. Ingemarsson, eds.), LNCS 209, Springer-Verlag, 1984, pp.224-314. 32. *Wiedemann D.H.* Solving sparse linear equations over finite fields. IEEE Trans. Info. Theory IT-32 (1986), No. 1, pp.54-62. 33. *Pomerance C., Smith J.W.* Reduction of huge, sparse matrices over finite fields via created catastrophes. Experimental mathematics 1 (1992), pp.89-94. 34. *LaMacchia B.A., Odlyzko A.M.* Solving large sparse linear systems over finite fields. Advances in cryptology-CRYPTO'90. (S. Vanstone ed.), LNCS 537, Springer-Verlag, 1990, pp.109-133. 35. *Gallant R., Lambert R., Vanstone S.* Improving the parallelized Pollard lambda search on binary anomalous curves. Research Report CORR 98-15, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 1998. 36. *Wagner L.* Algebro-geometric attack methods in elliptic curve cryptography. Honours thesis. November 18, 2002. 37. *G. Frey.* Applications of arithmetical geometry to cryptographic constructions. Proceedings of the Fifth International Conference on Finite Fields

9	Duursma, Gaudry, Morain	$C(F_q)$, известен порядок автоморфизма Фробениуса	$O\left(q^{\frac{2g}{x+1}+c}\right)$ при $q \rightarrow \infty$
10	Gaudry, Hess, Smart	ГЭК рода 4 над $F_{2^{21}}$	$O\left(N \cdot L_{\frac{1}{2}}\left[\frac{1}{2}, 3.54\sqrt{\log q}\right]\right)$, при $g \rightarrow \infty$, $N = \deg_y C(x, y)$
11	Enge, Gaudry	g – большое	$O\left(L_{q^*}[c]\right)$, $c = \sqrt{2\left(\sqrt{1 + \frac{1}{2g}} + \frac{1}{\sqrt{2g}}\right)}$
12	Jacobson, Menezes, Stein	$g = 31$, $F_{2^{2^n}}$	$O\left(\exp\left((1+o(1))\sqrt{(2g+1)\log q \log((2g+1)\log q)}\right)\right)$
13	Theriault	$C(F_q)$, g – небольшое	$O\left(g^5 q^{2-2/(g+1)+c}\right)$ при $q > (g-1)!$ и $O\left(g^5 q^{2-4/(2g+1)+c}\right)$ при $q > (g-1)!/g$

Из данных, приведенных в табл. 2, следует, что алгоритмы 1-4, 11 и 12 эффективно решают ЗДЛГЭК большого рода, следовательно, в криптографических целях имеет смысл применять лишь кривые небольшого рода. В то же время, алгоритмы 8, 10, 13 являются эффективными для ГЭК рода более 4. Причем алгоритм 13 уменьшает границу до 3, в этом случае он имеет сложность меньшую, чем сложности универсальных алгоритмов.

Как известно [18], что для противостояния универсальным алгоритмам криптоанализа (полного перебора, Полларда, Похлинга-Хелмана, больших и малых шагов) предлагается использовать ГЭК $\#J_C(F_q) = q^g \geq 2^{160}$. Исследования [3, 25] показывают, что кривые с эффективно вычислимым эндоморфизмом Фробениуса являются уязвимыми. Наличие эндоморфизма позволяет повысить эффективность универсальных алгоритмов, а также позволяет применить специализированные алгоритмы, например алгоритм 9 в табл. 2.

Выводы

В работе рассмотрены наиболее эффективные алгоритмы решения ЗДЛГЭК, из известных на сегодняшний день. Полученные результаты подтверждают факт, что дальнейшее развитие асимметричной криптографии лежит в увеличении сложности кривой (рода), т.е. перехода к ГЭК. Повышая род, повышается и сложность решения ЗДЛ, причем возможно уменьшение размера базового поля без ущерба стойкости. Другими словами, ГЭК позволяют обеспечить аналогичную сложность ЗДЛ при соизмеримых параметрах с ЭК, в тоже время дальнейшее повышение сложности кривой (род > 3) – не целесообразно, т.к. возможно эффективное применение алгоритма 13 (табл. 2). Уменьшенный размер базового поля открывает «новые горизонты» для ГЭК, которыми являются мобильные устройства.

Анализ алгоритмов позволил сформулировать подходы к реализации скрытых резервов эффективности посредством:

- уменьшения размера базы разложения;
- прореживания (отсеивание) соотношений на этапе их генерации (FP метод, Smart метод);
- решения, как разреженных, так и плотных СЛУ над полем;
- распараллеливания (например генерации соотношений, решения СЛУ);
- использования повторного логарифмирования [53, 54];
- гармонизации подходов к ЗДЛГЭК, используемых для решения ЗДЛ над полем и задачи факторизации.

Остается открытым вопрос о существовании алгоритма криптоанализа преобразований на ГЭК со сложностью соизмеримой с $O\left(\exp\left((c+o(1))m^{1/3}(\log m)^{2/3}\right)\right)$ для алгоритма криптоанализа Адлеманна-Копперсмита в поле [49, 50]. Дальнейшего исследования требует классификация кривых уязвимых к атаке спуском Вейля.

Список литературы: 1. *Vollmer U.* Asymptotically fast discrete logarithms in quadratic number fields. Algorithmic number theory-ANTS-IV, LNCS 1838, 2000, pp.581-594. 2. *Stein A., Teske E.* Explicit bounds and heuristics on class numbers in hyperelliptic function fields. Mathematics of computation, 71 (2002), pp.837-861. 3. *Gallant R., Lambert R., Vanstone S.* Improved the parallelized Pollard lambda search on anomalous binary curves. Mathematics of computation, 69 (2000), pp.1699-1705. 4. *Wiener M., Zuccherato R.* Faster attacks on elliptic curve cryptosystems. Selected areas in cryptography, LNCS 1556, 1999, pp.190-200. 5. *Gaudry P.* An algorithm for solving the discrete log problem on hyperelliptic curves. Advances in cryptology-EUROCRYPT'2000, LNCS 1807, 2000, pp.19-34. 6. *Bauer M.* A subexponential algorithm for solving the discrete logarithm problem in the Jacobian of high genus hyperelliptic curves over arbitrary finite fields. Preprint, 1999. 7. *Adleman L., DeMarrais J., Huang M.* A subexponential discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields. Algorithmic number theory-ANTS-II, LNCS 877, 1994, pp.28-40. 8. *Muller V., Stain A., Thiel C.* Computing discrete logarithms in real quadratic congruence function fields of large genus. Mathematics of computation, 68 (1999), pp.807-822. 9. *Stein A.* Equivalences between elliptic curves and real quadratic congruence function fields. Journal de Theorie des nombres de Bordeaux, 9 (1997), pp.75-95. 10. *Zuccherato R.* The equivalence between elliptic curve and quadratic function field discrete logarithms in characteristic 2. Algorithmic number theory-ANTS-III, LNCS 1423, 1998, pp.621-638. 11. *Paulus S., Ruck H.-G.* Real and imaginary quadratic representations of hyperelliptic function fields. Mathematics of computation, 68 (1999), pp.1233-1241. 12. *Smart N.* Experiments using an analogy of the number field sieve algorithm to solve the discrete logarithm problem in the Jacobians of hyperelliptic curves. HP Laboratories Bristol, Technical report HPL-97-130, 1997. 13. *Hafner J., McCurley K.* A rigorous subexponential algorithm for computation of class groups. Journal of the American Mathematical Society, 2 (1989), pp.837-850. 14. *Flassenberg R., Paulus S.* Sieving in function fields. Experimental mathematics, 8 (1999), pp.229-249. Available at: <http://ftp.informatik.tu-darmstadt.de/pub/TI/TR/TI-97-13.rafla.ps.gz>. 15. *Enge A.* Computing discrete logarithms in high-genus hyperelliptic jacobians in provably subexponential time. Mathematics of computation, 71 (2001), pp.729-742. 16. *Enge A.* A general framework for subexponential discrete logarithm algorithms in groups of unknown order. Finite geometries, Developments in mathematics vol. 3, Kluwer Academic publishers, Dordrecht 2001, pp.133-146. 17. *Enge A., Stein A.* Smooth ideals in hyperelliptic function fields. Mathematics of computation, 71 (2001), pp.1219-1230. 18. *Teske E.* Speeding up Pollard's rho method for computing discrete logarithms. Algorithmic number theory, LNCS 1423, 1998, pp.541-554. 19. *Gaudry P., Hess F., Smart N.* Constructive and destructive facets of Weil descent on elliptic curves. Journal of cryptology, 15 (2002), pp.19-46. 20. *Enge A., Gaudry P.* A general framework for subexponential discrete logarithm algorithms. Acta arithmetica, 102 (2002), pp.83-103. 21. *Jacobson M., Menezes A., Stein A.* Solving elliptic curve discrete logarithm problems using Weil descent. Journal of the Ramanujan mathematical society, 16 (2001), pp.231-260. 22. *Theriault N.* Index calculus attack for hyperelliptic curves of small genus. Preprint, 2003. 23. *Galbraith S.* Weil descent of jacobians. Discrete applied mathematics, 128 (2003), pp.165-180. 24. *Shoup V.* Lower bounds for discrete logarithms and related problems. Advances in cryptology – EUROCRYPT'97, LNCS 1233, pp.256-266. 25. *Lambert R.* Computation aspects of discrete logarithms. A thesis for the degree of DP in electrical engineering. Waterloo, Ontario, Canada, 1996. 26. *Buchmann J., Weber D.* Discrete logarithms: recent progress. 1998. 27. *Jacobson M. Jr, Menezes A., Stein A.* Hyperelliptic curves and cryptography. 28. *Тевяшев А.Д.* Алгебра и геометрия. Алгебраические структуры. Категории. – Харьков: ХТУПЕ, 1998. – 388с. 29. *Shizuya H., Itoh T., Sakurai K.* On the complexity of hyperelliptic discrete logarithm problem. 30. *Matsuo K., Chao J., Tsuijii S.* Baby step gaint step algorithms in point counting of hyperelliptic curves. Special section on discrete mathematics and its applications of IEICE. Vol. E86-A, No. 5, May 2003. pp.1127-1134. Japan. 31. *Odlyzko A.M.* Discrete logarithms in finite fields and their cryptographic significance. Advances in cryptology – EUROCRYPT'84 (T. Beth, N. Cot, I. Ingemarsson, eds.), LNCS 209, Springer-Verlag, 1984, pp.224-314. 32. *Wiedemann D.H.* Solving sparse linear equations over finite fields. IEEE Trans. Info. Theory IT-32 (1986), No. 1, pp.54-62. 33. *Pomerance C., Smith J.W.* Reduction of huge, sparse matrices over finite fields via created catastrophes. Experimental mathematics I (1992), pp.89-94. 34. *LaMacchia B.A., Odlyzko A.M.* Solving large sparse linear systems over finite fields. Advances in cryptology-CRYPTO'90. (S. Vanstone ed.). LNCS 537, Springer-Verlag, 1990, pp.109-133. 35. *Gallant R., Lambert R., Vanstone S.* Improving the parallelized Pollard lambda search on binary anomalous curves. Research Report CORR 98-15, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 1998. 36. *Wagner L.* Algebro-geometric attack methods in elliptic curve cryptography. Honours thesis. November 18, 2002. 37. *G. Frey.* Applications of arithmetical geometry to cryptographic constructions. Proceedings of the Fifth International Conference on Finite Fields

and Applications. Springer-Verlag, 2001, pp.128-161. 38. *Jacobson M., Menezes A., Stein A.* Solving elliptic curve discrete logarithm problems using Weil descent. May 16, 2001. 39. *Pollard J.M.* The lattice sieve. In A.K. Lenstra, H.W. Lenstra, ed. The development of the number field sieve. Springer-Verlag, LNM 1554, 1993. at pp.43-49. 40. *Menezes A.J., Wu Y., Zuccherrato R.J.* An elementary introduction to hyperelliptic curves. Technical report CORR96-19, Department of combinatorics and optimization, University of Waterloo, Waterloo, Ontario, 1996. In: Koblitz, N.: Algebraic aspects of cryptography. Springer-Verlag, Berlin Heidelberg New York. 1998. 41. *Duursma I., Gaudry P., Morain F.* Speeding up the discrete log computation on curves with automorphisms. 42. *Frey G., Ruck H.-G.* A remark concerning m -divisibility and discrete logarithm in the divisor class group of curves. Math. Comp., 62 (206), pp.865-874, April 1994. 43. *Menezes A., Qu M.* Analysis of the Weil descent attack of Gaudry, Hess and Smart. Research Report CORR 00-48, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2000. 10 pages. Revised April 22, 2001. 44. *Maurer M., Menezes A., Teske E.* Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree. Research Report CORR 01-59, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2001. 42 pages. 45. *Galbraith S.D., Hess F., Smart N.P.* Extending the GHS Weil descent attack. Available: <http://eprint.iacr.org> 46. *Teske E.* An elliptic curve trapdoor system. Research Report CORR 03-07, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2003. 47. *A. Menezes, E. Teske, Weng A.* Weak fields for ECC. Research Report CORR 03-15, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 2003. 48. *Okamoto T., Sakurai K.* Efficient algorithms for the construction of hyperelliptic cryptosystems. 49. *Thome E.* Computation of Discrete Logarithms in F_{2^m} . Laboratoire d'Informatique (LIX) Ecole polytechnique, Essen, France, 2001. 50. *Thome E.* Computing discrete logs in large characteristic 2 finite fields. Laboratoire d'Informatique (LIX) Ecole polytechnique, Essen, France, 25 September 2002. 51. *Galbraith S.D., Smart N.P.* A cryptographic application of Weil descent. Cryptography and coding, 7th IMA Conference, Springer-Verlag, LNCS 1746, pp. 191-200, 1999. The full version of the paper is HP Labs Technical report, HPL-1999-70. 52. *Miyaji A., Nakabayashi M., Takano S.* New explicit conditions of elliptic curve traces for FR-reduction. 53. *Johnson D., Menezes A., Vanstone S.* The Elliptic curve digital signature algorithm. Certicom research. Canada. 54. *Pollard J. M.* Kangaroos, Monopoly and discrete logarithms. Journal of Cryptology, 13: 437-447, 2000.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 01.12.2004

ВИБІР ЕЛІПТИЧНИХ КРИВИХ ДО КРИПТОГРАФІЧНИХ ЗАСТОСУВАНЬ

Зацікавленість математичними конструкціями у вигляді групи раціональних точок еліптичних кривих в криптографії виникла в 1985 р. у двох напрямках: для розв'язання задач факторизації великих чисел і для побудовання криптографічних протоколів [1,2]. Ця зацікавленість актуалізувалась у теперішньому часі і обумовлюється тим, що, з одного боку еліптичні криві виявилися джерелом скінченних абелевих груп, що володіють корисними структурними властивостями, а з другого – тим, що на основі їх використання вдалося забезпечити ті ж криптографічні властивості по стійкості, котрими володіють числові та поліноміальні криптосистеми, але відповідні показники перших були отримані при суттєво меншому розмірі ключа.

Безперечно, що однією з основних показників криптоалгоритмів виступає рівень їх стійкості. Як показує аналіз, не всі еліптичні криві у цьому сенсі однаково корисні для побудовання криптографічних протоколів. Мета цієї роботи заключається в тому, щоб обговорити умови, котрим повинні задовольняти еліптичні криві, для того, щоб вони дозволяли забезпечити високі показники стійкості в криптографічних прикладеннях.

1 Умова несингулярності кривій

Отже, наша увага зосереджується на еліптичних кривих E , одним з зображень котрих є так звана канонічна форма Вейєрштрасса:

$$E: y^2 = x^3 + ax^2 + bx + c. \quad (1)$$

Кубічне рівняння загального виду подають і таким чином:

$$y^2 = f(x), \quad f(x) = x^3 + ax^2 + bx + c, \quad (2)$$

тобто права частина рівності розглядається як звичайний многочлен третього степеня.

Надалі будемо вважати коефіцієнти a, b, c функції $f(x)$ раціональними, зокрема, дійсними числами, і значить поліном $f(x)$ степеня 3 має, щонайменше, один дійсний корінь. В дійсних числах ми можемо факторизувати його як $f(x) = (x - \alpha)(x^2 + \beta x + \gamma)$ з α, β, γ дійсними. Звичайно, він може мати всі три дійсних кореня. Якщо він має один дійсний корінь, то крива має вигляд, подібний рис. 1, тому що $y = 0$, коли $x = \alpha$. Якщо $f(x)$ має три дійсні корені, то крива має вигляд, подібний показаному на рис. 2. У цьому разі дійсні точки утворюють дві компоненти. Останнє справедливе при умові, що корені рівняння $f(x) = 0$ різні. Саме цей випадок нас і буде цікавити. Що суттєво для цієї умови?

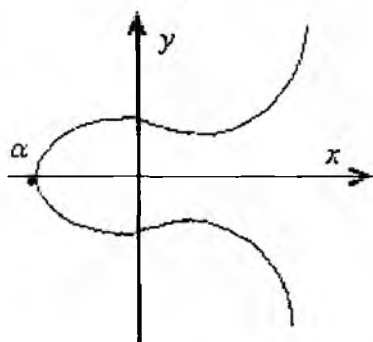


Рис 1

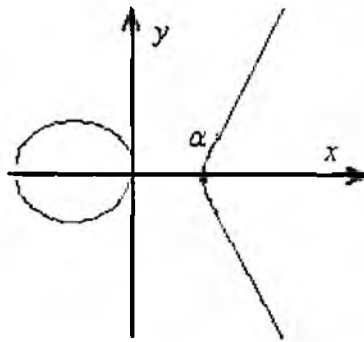


Рис 2

Нагадаємо, що мова йде про використання еліптичних кривих для утворення груп з їх точок. Як відомо [3,4], операція в групі точок еліптичної кривої пов'язана з геометричними побудовами (для будь-яких точок P і Q точка $P * Q$ – це третя точка перетину прямої

$g = \overline{PQ}$ з кривою E , симетрично відбита стосовно осі Ox), і тому перш за все нас повинні цікавити умови існування відповідних геометричних елементів (хорд та дотичних) для довільних точок кривої.

Запишемо рівняння кривої у вигляді $F(x, y) = y^2 - f(x) = 0$ і візьмемо частинні похідні $\frac{dF}{dx} = -f'(x)$, $\frac{dF}{dy} = 2y$. Якщо частинні похідні прямують у деякій точці (x_0, y_0) кривої до нуля, то $y_0 = 0$ і, отже, многочлени $f(x)$ та $f'(x)$ мають спільний корінь x_0 , тобто x_0 є подвійним коренем функції $f(x)$, і, навпаки, якщо f має подвійний корінь x_0 , то $(x_0, 0)$ буде так званою *сингулярною* точкою на кривій. За означенням під сингулярною розуміють точку, у котрій похідна дорівнює нулю або не існує. Якщо еліптична крива має сингулярну точку, то і сама крива називається *сингулярною*. Відповідно, ми будемо мати *несингулярну криву* при умові, що не існує точки на кривій, у якій частинні похідні одночасно зникають. Це і буде означати, що кожна точка на кривій має цілком визначену дотичну.

Скористаємося міркуваннями роботи [3], щоб показати більш точно, що ми дійсно повинні сфокусуватися саме тільки на несингулярних кубах. Це одразу не так зрозуміло. Розглянемо сингулярний випадок більш докладніше.

Можливі дві картини сингулярності. Яка з них виникає залежить від того, чи має функція $f(x)$ подвійний корінь або потрійний. У випадку, якщо така функція має подвійний корінь, типовим буде рівняння

$$y^2 = x^2(x+1),$$

і крива має сингулярну точку з різними напрямками дотичних, як показано на рис.3, тобто дотична у цій точці визначається неоднозначно.

Якщо $f(x)$ має потрійний корінь, то рівняння може бути перетворено до виду $y^2 = x^3$. Це напівкубічна парабола з вершиною у початку координат (рис.4). Існують і інші приклади сингулярних кубів у формі Вейєрштрасса, але вони також приводяться до загального виду після зміни координат.

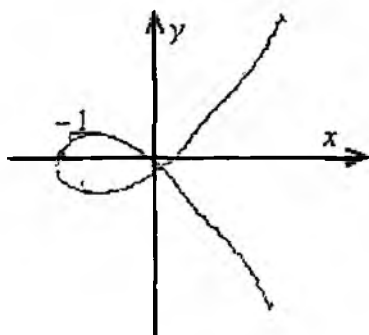


Рис. 3

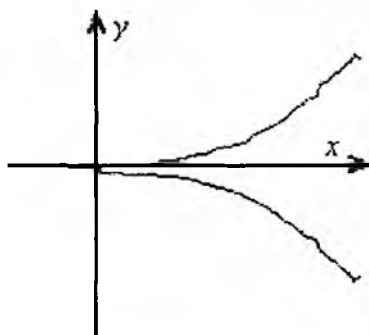


Рис. 4

Справа в тому, що раціональним точкам на сингулярному кубі можуть бути поставлені у взаємно однозначну відповідність раціональні точки на прямій (якщо ми здійснимо проєкцію сингулярної точки на деяку пряму, то ми побачимо, що пряма, яка проходить через сингулярну точку, перетинає куб подвійно, і значить вона перетинає куб ще тільки один раз. Таким чином, проєкція кубічної кривої на пряму є взаємно однозначною.

Фактично це легко зробити явно за допомогою формули. Дійсно, якщо ми приймемо $r = \frac{y}{x}$, то рівняння $y^2 = x^2(x+1)$ перетворюється в $r^2 = x+1$ і значить $x = r^2 - 1$ і $y = r^3 - r$. Для кожного раціонального числа r можна визначити x та y , тобто одержати ра-

ціональну точку на кубі; а якщо ми почнемо з раціональної точки (x, y) на кубі, то отримаємо раціональне число r . Ці операції є інверсними одна до одної і визначаються для всіх раціональних точок, окрім сингулярної точки $(0,0)$ на кривій. Отже, виходить, що за таким способом можна одержати всі раціональні точки на кривій.

Крива $y^2 = x^3$ ще простіше. Ми тільки прийнемо $x = t^2$ та $y = t^3$.

Таким чином, у випадку сингулярної кривої ми приходимо до того, що замість множини її раціональних точок можна розглядати множину точок деякої прямої (за рахунок існування відзначеної взаємно однозначної відповідності між ними).

Отже, сингулярні куби є тривіальними для аналізу, як і раціональні точки на них. Групу можна отримати, якщо уникати сингулярності¹.

Отже, надалі нас будуть цікавити несингулярні еліптичні криві, тобто криві, що не мають точок, у котрих похідні $\frac{dF}{dx} = -f'(x)$, $\frac{dF}{dy} = 2y$ дорівнюють нулю. Такі криві називають також гладкими.

Зазначимо, що з гладкістю кривої часто пов'язується поняття дискримінанта $D(f)$.

Це поняття з'являється при вивченні алгебраїчних рівнянь. Дискримінантом $D(f)$ алгебраїчного рівняння $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$, $a_n \neq 0$ є добуток a_n^{2n-2} і квадратів усіх різниць $x_i - x_k$ ($i > k$) коренів x_i рівняння (кратні корені порядку m розглядаються як m різних коренів з різними індексами):

$$D(f) = a_n^{2n-2} \prod_{i>k} (x_i - x_k)^2 = a_n^{2n-2} [W(x_1, x_2, \dots, x_n)]^2,$$

$$\text{де } W(x_1, x_2, \dots, x_n) = \begin{vmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{vmatrix} = \prod_{i>k} (x_i - x_k)$$

– визначник Вандермонда.

Дискримінант $D(f)$ є симетричною² функцією коренів $x_1, x_2, \dots, x_n$, яка обертається в нуль у тому і тільки у тому випадку, коли $f(x)$ має щонайменше один кратний корінь (який необхідно виявляється коренем $f(x)$ та $f'(x)$).

Обчислимо, наприклад, дискримінант кубічного тричлена $f(x) = x^3 + bx + c$, коренями якого є $\alpha_1, \alpha_2, \alpha_3$. Користуючись наведеною вище формулою, отримуємо

$$D(f) = (-1)^{\frac{3 \cdot 2}{2}} f'(\alpha_1) f'(\alpha_2) f'(\alpha_3) = -(3\alpha_1^2 + b)(3\alpha_2^2 + b)(3\alpha_3^2 + b) = \\ = -[27\alpha_1^2 \alpha_2^2 \alpha_3^2 + 9p(\alpha_1^2 \alpha_2^2 + \alpha_1^2 \alpha_3^2 + \alpha_2^2 \alpha_3^2) + 3b^2(\alpha_1^2 + \alpha_2^2 + \alpha_3^2) + b^3].$$

$$\text{Оскільки } \alpha_1^2 \alpha_2^2 + \alpha_1^2 \alpha_3^2 + \alpha_2^2 \alpha_3^2 = (\alpha_1 \alpha_2 + \alpha_1 \alpha_3 + \alpha_2 \alpha_3)^2 - 2\alpha_1 \alpha_2 \alpha_3 (\alpha_1 + \alpha_2 + \alpha_3), \\ \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = (\alpha_1 + \alpha_2 + \alpha_3)^2 - 2(\alpha_1 \alpha_2 + \alpha_1 \alpha_3 + \alpha_2 \alpha_3),$$

і окрім того, $\alpha_1 + \alpha_2 + \alpha_3 = 0$, $\alpha_1 \alpha_2 + \alpha_1 \alpha_3 + \alpha_2 \alpha_3 = b$, $\alpha_1 \alpha_2 \alpha_3 = -c$,

то $D(f) = -4b^3 - 27c^2$.

¹ Раціональні точки на сингулярній і несингулярній кубічних кривих ведуть себе зовсім по-різному: група раціональних точок на сингулярній кривій не є скінченно породженою, в той час як група раціональних точок на несингулярній кривій – скінченно породжена.

² Симетрична функція – функція, яка не змінюється при будь-якій перестановці своїх аргументів.

Для кривої у нормальній формі (2) дискримінантом функції $f(x)$ є величина

$$D(f) = -4a^3c + a^2b^2 + 18abc - 4b^3 - 27c^2.$$

Більш знайомою є форма $D(f) = -4b^3 - 27c^2$, що відповідає випадку $a = 0$ в рівнянні (1) (див. приклад, наведений вище). Згідно з відомим твердженням [5] (многочлен $f(x)$ має кратний корінь тоді і тільки тоді, коли його дискримінант дорівнює нулю), якщо дискримінант не обертається в нуль, то це свідчить про те, що корені $f(x)$ є різними. Таким чином, еліптична крива з раціональними коефіцієнтами (тобто над полем R) з ненульовим дискримінантом $D(f) \neq 0$ являє собою гладку криву, у кожній точці якої можна провести дотичну.

2 Спарування Вейля і дискретний логарифм

Обговорюючи умови відбору еліптичних кривих для криптографічних застосувань, треба підкреслити, що сьогодні криптографія на еліптичних кривих вже вийшла на рівень прийняття стандартів, тобто виконані дослідження і розробки вже дозволили перейти до практичної реалізації алгоритмів перетворень в групах точок еліптичних кривих. В процесі виконання досліджень виявилось додаткове обмеження до вибору еліптичних кривих, пов'язане з поняттям *суперсингулярності*, на якому ми зараз зосередимо увагу.

Для подальшого розуміння потрібно буде спочатку коротко висвітлити деякі принципи побудування криптографічних алгоритмів та забезпечення їх стійкості. Почнемо з того, що одним з потужних криптографічних інструментів є так звані *важкооборотні (одношляхові) функції*. Загальна концепція побудування одношляхових функцій – конструювання великої скінченної циклічної групи $G = \langle G; \circ \rangle$ разом з породжуючим елементом (генератором) $g \in G$ такої, щоб було “легко” обчислити кратне $m \cdot g := g \circ \dots \circ g$ для всіх $m \in N$, але щоб було “важко” обновити m для деякого довільного елемента $h \in G$, такого що $m \cdot g = h$. Тут “легке” обчислення означає, що воно може бути виконано за поліноміальний час, де поліном має степінь порядку $\log|G|$, а “важке” обчислення означає складність, набагато більшу, ніж поліноміальна, принаймні, для кращого з відомих алгоритмів. Однією з реалізацій цієї конструкції є зведення в степінь (експоненціювання) у великому скінченному полі, запропоноване Діффі і Хеллманом [7]. Цей принцип використовується зокрема для побудування криптосистем з публічними (відкритими) ключами і на еліптичних кривих.

В криптографічних застосуваннях еліптична крива розглядається над скінченним полем $GF(q)$, тобто кубічна крива E визначається як множина всіх точок $(x, y) \in GF(q) \times GF(q)$, котрі задовольняють рівнянню (1) з коефіцієнтами $a, b, c \in GF(q)$. Для виконання криптоперетворень використовуються точки $P \in E(GF(q))$ ³ еліптичних кривих E над скінченними полями $GF(q)$, які утворюють групу стосовно операції додавання точок (див. Л-3, ТГК). Важкооборотне перетворення в цьому випадку полягає в обчисленні точки, що є кратною деякому породжуючому елементу групи точок еліптичної кривої. Воно має саме такий вигляд, про який ми казали вище:

$$Q = kP, Q \in \langle P \rangle.$$

Тут P називається базовою точкою еліптичної кривої⁴ (її називають також генератором); k – деяке ціле число; $1 < k < n$. Ця базова точка повинна мати великий порядок. Тоді порівняно легко обчислити скалярний добуток kP ($kP = P + P + \dots + P$ k -разів), але важко обчислити k за відомими значеннями Q і P . Остання задача здобула назву дискретного логарифмування (якщо операцію трактувати як множення, то очевидна відповідність між ади-

³ $E(GF(q))$ – позначення еліптичної кривої E над полем $GF(q)$.

⁴ Символом $\langle P \rangle$ позначена циклічна група, породжена елементом P .

тивною та мультиплікативною групами $P + P + \dots + P = kP \rightarrow P \circ P \circ \dots \circ P = P^k$, що й виправдовує використання поняття логарифмування у випадку адитивної групи). Отже, стійкість цього криптографічного перетворення визначається складністю рішення задачі дискретного логарифмування в циклічній групі $\langle P \rangle$ великого простого порядку n ; в нашому випадку порядку групи точок еліптичної кривої E , що породжена базовою точкою цієї кривої P .

Тепер можна повернутися до чергового обмеження щодо вибору еліптичних кривих. Нагадаємо спочатку, що ще у 1987 році Кобліц [8] запропонував для криптосистем з відкритими ключами використати так звані суперсингулярні криві з скінченним числом $\#E(GF(q)) = q + 1$ елементів (точок на кривій). Однак пізніше було встановлено, що такі криві не придатні для криптографічних застосувань. Криптосистеми, які базуються на кривих такого типу, можуть бути атаковані алгоритмом дискретного логарифмування, запропонованим ще у 1993 А. Менезісом, Т. Окомото та С. Венстоном [9]. Цей алгоритм використовує так зване спарування Вейля для приведення групи точок еліптичної кривої над полем $GF(2^m)$ до мультиплікативної групи деякого розширення $GF(2^m)$ вихідного скінченного поля, що дозволяє звести проблему дискретного логарифмування на кривій до проблеми дискретного логарифмування у скінченному полі. Пізніше були зроблені спроби замінити еліптичні криві, запропоновані Кобліцем, деякими іншими суперсингулярними кривими, які вважалися більш надійними до застосування. Однак і ці криві допускали приведення до деякого скінченного поля великого розміру.

Отже, коротко нагадаємо сутність алгоритму зведення, запропонованого А. Менезісом, Т. Окомото та С. Венстоном⁵. Як вже було вказано, головна ідея підходу полягає в тому, щоб провести обчислення дискретного логарифма в групі точок еліптичної кривої $E(K)$, що розглядається над полем K , до проблеми дискретного логарифмування в скінченному полі GF , шляхом зведення групи точок еліптичної кривої $E(K)$ до мультиплікативної групи скінченного поля GF .

Нам будуть потрібні ще декілька фактів з теорії еліптичних кривих. Нагадаємо, що точки еліптичної кривої з введеною на множині точок "геометричною" операцією додавання (множення) точок разом з точкою на нескінченності O утворюють групу [3,4]. Згідно з положеннями теорії груп, будь-яка точка P скінченного порядку n утворює підгрупу $E[n]$ групи точок еліптичної кривої. Така підгрупа називається групою кручення еліптичної кривої⁶. Якщо n – непарне число, то порядок групи кручення дорівнює n^2 ($P = (x_p, y_p) \in K \times K$). Криптографічні алгоритми будуються саме в циклічній групі простого порядку n . Ця циклічна група є підгрупою групи кручення. Решта точок групи кручення не належить основному полю (підполю з n елементів). Оскільки група кручення скінченна, то існує розширення $GF(2^{m'})$ вихідного поля⁷, у якому лежать усі точки групи кручення. Мультиплікативна група цього розширення має порядок $2^{m'} - 1$, тому, відповідно до теореми Лагранжа [5], вкладення групи кручення у мультиплікативну групу цього розширення можливе тільки у тому випадку, якщо n ділить $2^{m'} - 1$. Виявилось, що цієї умови досить. Вказаний факт і лежить в основі методу зведення задачі дискретного логарифмування в групі точок еліптичної кривої до задачі дискретного логарифмування в мультиплікативній групі розширення вихідного скінченного поля, для якої, існують алгоритми рішення задачі дискретного логарифмування субекспоненціальної складності (останній факт вам буде доводитися в теорії складності). Фактично це зведення виконується за допомогою так званого спарування Вейля, що є відображенням виду

⁵ Зараз вже запропонований ряд інших алгоритмів зведення (спарування Frey – Rück (1994), Mstsunari-Sakai-Kasahara (1999) та інші).

⁶ Кручення скінченно породженої абелевої групи G – група T , яка складається із усіх елементів скінченного порядку v групи G . Числа $v > 1$ можуть бути однозначно з точністю до перестановки вибрані у вигляді степенів простих чисел, і тоді вони називаються коефіцієнтами кручення групи G .

⁷ Тут мова йде про розширення двійкового поля.

$$e_N: E[n] \times E[n] \rightarrow \mu_n,$$

де μ_n – група коренів степеня n з одиниці, котра при наведеній умові є підгрупою мультиплікативної групи скінченного поля $GF(2^m)$. Якщо $Q = kP$ і T – точка з $E[n]$, яка не належить $\langle P \rangle$, то для $e_N(P, T) = \alpha \in \mu_n$ і $e_N(Q, T) = \beta \in \mu_n$ при умові, що відображення e_N є білінійним⁸ та не виродженим⁹, виходить, що $\beta = \alpha^k$. Цей й означає, що задача дискретного логарифмування виразу $Q = kP$ у циклічній групі $\langle P \rangle$ зводиться до задачі дискретного логарифмування $\beta = \alpha^k$ у скінченному полі $GF(2^m)$.

Якщо степінь розширення l мала, то для вихідної задачі дискретного логарифмування існує субекспоненціальний алгоритм. Наприклад, у випадку дуже цікавих з обчислювальної точки зору суперсингулярних кривих вийшло так, що $l \leq 6$, тому при відносно малих розмірах основного поля довелося відмовитися від застосування таких кривих у криптографії.

Таким чином, ми прийшли ще до однієї умови відбору кривих для використання в криптографії, яка здобула назву MOV (умова Менезіса-Окомато-Венстона): *порядок базових точок еліптичних кривих, які дозволені до криптографічного використання, повинен задовольняти обмеженню $2^{mi} \neq 1 \pmod{n}$, для $1 \leq i \leq 32$.*

Це означає, що циклічна група, що породжується базовою точкою, повинна мати великий порядок (міркування, наведені вище стосуються розширень полів характеристики 2, тобто у більш загальному випадку мова повинна йти про скінченне розширення $GF(p^k)$).

Зазначимо, що дискретний логарифм у скінченному полі $GF(2^n)$ можна обчислити за допомогою алгоритму Коopersміта [10] за складністю

$$O(\exp(cn^{\frac{1}{3}} (\ln n)^{\frac{2}{3}})).$$

У подальшому виявилось, що MOV умова може бути виконана, якщо скористатися не суперсингулярними еліптичними кривими.

Отже, зупинимося на поняттях суперсингулярності та не суперсингулярності, а також на умовах виконання вимоги не суперсингулярності більш докладніше.

3 Умови не суперсингулярності кривої

Зосередимо увагу на теоретичному обірунтуванні MOV умови. Для цього доведеться познайомитися з додатковими більш складними узагальненнями й побудуваннями з теорії полів Галуа.

Почнемо з обговорення важливих наслідків, що випливають з теореми Мордела [3, 4]. Нагадаємо, що відповідно до теореми Мордела група $E(\mathbb{Q})$ раціональних точок на не сингулярній кривій (1) є скінченно породжуваною абелевою групою, а з фундаментальної теореми по абелевих групах [4]: *усяка скінченно породжувана абелева група розкладається в пряму суму скінченного числа нерозкладних циклічних підгруп, з яких частина – скінченно примарні¹⁰, частина – нескінченні* випливає, що $E(\mathbb{Q})$, як абстрактна група, ізоморфна прямій сумі нескінченних циклічних груп та скінченних циклічних груп простого порядку¹¹.

⁸ Білінійність спарування $e_N([n]P, Q) = e_N(P, [n]Q) = e_N(P, Q)^n$.

⁹ Невиродженість спарування $e_N(P, P) \neq 1$.

¹⁰ Примарна група – періодична абелева група, порядки усіх елементів якої є степенем фіксованого простого числа.

¹¹ Зокрема, скінченна абелева група розкладається у пряму суму примарних циклічних груп. Такий розклад, взагалі кажучи, не єдиний, але будь-які два розклади абелевої групи з скінченним числом твірних ізоморфні між собою і, таким чином, число нескінченних циклічних доданків й сукупність порядків примарних циклічних доданків не залежать від вибору розкладу. Ці числа називаються і н в а р і я н т а м и с к і н ч е н н о п о р о д ж у в а н о ї а б е л е в о ї г р у п и. Усякі підгрупи цієї групи, для яких ці інваріанти збігаються, ізоморфні.

Пояснимо цей момент більш докладніше (тут ми скористаємося результатами роботи [3]). Нехай Z позначає адитивну групу цілих чисел, та нехай Z_m позначає циклічну групу Z/mZ цілих чисел за модулем m (фактор-кілеце цілих чисел за модулем m). Тоді структурна теорема свідчить про те, що група $E(\mathbf{Q})$ має вигляд

$$E(\mathbf{Q}) \cong \underbrace{Z \oplus Z \oplus \dots \oplus Z}_{\text{гр } \in K_{\text{м}}^{\text{м}}} \oplus Z_{p_1}^{v_1} \oplus Z_{p_2}^{v_2} \oplus \dots \oplus Z_{p_s}^{v_s}. \quad (3)$$

Якщо казати простіше, наведений запис свідчить, що існують генератори (породжуючі елементи):

$$P_1, \dots, P_r, Q_1, \dots, Q_s \in E(\mathbf{Q}),$$

які дозволяють кожну точку $P \in E(\mathbf{Q})$ еліптичної кривої записати у вигляді

$$P = n_1 P_1 + \dots + n_r P_r + m_1 Q_1 + \dots + m_s Q_s. \quad (4)$$

Тут цілі числа n_i унікально визначені точкою P , в той час як цілі числа m_j визначаються за відповідними модулями $p_j^{v_j}$, $j = 1, 2, \dots, s$. Ціле число r називається рангом групи $E(\mathbf{Q})$. Група $E(\mathbf{Q})$ може бути скінченною, якщо і тільки якщо вона має ранг $r = 0$. Підгрупа $Z_{p_1}^{v_1} \oplus Z_{p_2}^{v_2} \oplus \dots \oplus Z_{p_s}^{v_s}$ відповідає елементам скінченного порядку в $E(\mathbf{Q})$; вона має порядок $p_1^{v_1} p_2^{v_2} \dots p_s^{v_s}$.

Звичайно, точки $P_1, \dots, P_r, Q_1, \dots, Q_s$ не є унікальними. Існує багато можливих варіантів генераторів для $E(\mathbf{Q})$.

Ми поки що казали про раціональні точки, але фактично рівняння кубічної кривої (1) визначає декілька множин точок. Ми можемо записати $E(\mathbf{Q})$ для множини точок кривої, чії координати виявляються раціональними. $E(\mathbf{R})$ для точок, що мають дійсні координати і $E(\mathbf{C})$ для множини пар комплексних чисел (x, y) , що задовольняють рівнянню (1). Існує також точка O на нескінченності, яку ми вважаємо присутньою у всіх таких множинах.

Користуючись законом додавання, легко впевнитись, що точки кривої з комплексними координатами теж утворюють групу. Точки з дійсними координатами утворюють підгрупу цієї групи. А тому що ми допускаємо, що коефіцієнти рівняння кривої a, b, c є раціональними числами, то вірно те, що її раціональні точки утворюють групу дійсних чисел. Отже, ми маємо групу й декілька підгруп:

$$\{\mathbf{Q}\} \subset E(\mathbf{Q}) \subset E(\mathbf{R}) \subset E(\mathbf{C}).$$

В остаточному підсумку нас будуть цікавити точки еліптичних кривих, координати яких розглядаються як елементи полів та їх розширень. Тому надалі корисно буде нагадати деякі факти, що стосуються загальної теорії розширень полів й теорії Галуа. Отже, розглянемо поле комплексних чисел зі своїми підполлями $\mathbf{Q} \subset K \subset \mathbf{C}$. В даному випадку поле K як розширення поля $\mathbf{Q}$ є $\mathbf{Q}$ -векторним простором, зі степенем розширення K над $\mathbf{Q}$, що є розмірністю лінійного простору [4]: кожне розширення Ω можна розглядати як лінійний простір над полем Δ . Нагадаємо, що степінь розширення позначається як $[K:\mathbf{Q}]$.

Якщо степінь розширення $[K:\mathbf{Q}]$ скінченна (ступінь скінченного розширення Ω над полем Δ дорівнює максимальному числу елементів поля Ω , котрі в змозі утворити лінійно незалежну систему), то будемо казати, що K є числовим полем.

Важливим методом вивчення числових полів є розгляд множини їх гомоморфізмів, тобто множини можливих відображень одного в одне:

$$\sigma: K \rightarrow \mathbf{C}.$$

Підкреслимо, що гомоморфізм полів завжди є взаємно однозначним відображенням (ізоморфізмом), тому що поле не має нестривіальних ідеалів. Зауважимо також, що відповідно до властивостей гомоморфізмів кілець $\sigma(1) = 1$, і тому усякий гомоморфізм σ автоматично буде задовольняти умові $\sigma(a) = a$ для всіх $a \in \mathbb{Q}$. Це свідчить про те, що число гомоморфізмів $K \rightarrow \mathbb{C}$ завжди точно дорівнюватиме степені розширення $|K:\mathbb{Q}|$ (гомоморфізм полів це вкладення полів).

Іноді трапляється так, що образ $\sigma(K)$ дорівнює вихідному полю K . Тоді σ дійсно є гомоморфізмом K на самого себе, і в такому випадку ми називатимемо σ *автоморфізмом* поля K . Зазначимо, однак, що це не означає рівності $\sigma(\alpha) = \alpha$ для кожного $\alpha \in K$, а просто те, що $\sigma(\alpha) \in K$. Якщо це грапляється для кожного відображення σ , то K називається *Галуа розширенням* $\mathbb{Q}$. Більш узагальнено визначимо:

$$\text{Aut}(K) = \{\text{автоморфізми } \sigma: K \rightarrow K\}.$$

Ясно, що $\text{Aut}(K)$ є групою, тому що якщо $\sigma, \tau \in \text{Aut}(K)$, то тоді ми можемо сформулювати елемент $\sigma\tau \in \text{Aut}(K)$, шляхом визначення композиції $(\sigma\tau)\alpha = \sigma(\tau(\alpha))$. Числове поле K є Галуа розширенням $\mathbb{Q}$, якщо і тільки якщо $\#\text{Aut}(K) = |K:\mathbb{Q}|$.

В цьому випадку будемо записувати $\text{Gal}(K/\mathbb{Q})$ замість $\text{Aut}(K)$ і називати $\text{Gal}(K/\mathbb{Q})$ *групою Галуа*¹² $K/\mathbb{Q}$. Нас цікавитиме розгляд полів саме як полів гомоморфізмів.

Наведені міркування дуже абстрактні. Як насправді знайти числові поля, котрі є полями Галуа над $\mathbb{Q}$? Відповідь буде простою. Візьмемо будь-який многочлен з раціональними коефіцієнтами $f(X) \in \mathbb{Q}[X]$. Факторизуємо $f(X)$ над комплексними числами:

$$f(X) = a(X - \alpha_1)(X - \alpha_2)\cdots(X - \alpha_n)$$

і нехай

$$K = \mathbb{Q}(\alpha_1, \alpha_2, \dots, \alpha_n)$$

буде найменшим¹³ підполем $\mathbb{C}$, що містить усі α_i . Тоді будь-який гомоморфізм $\sigma: K \rightarrow \mathbb{C}$ визначається значеннями $\sigma(\alpha_1), \dots, \sigma(\alpha_n)$ й кожне $\sigma(\alpha_i)$ повинне бути корнем $f(X)$ і, значить, повинне дорівнювати α_i . Зокрема, $\sigma(\alpha_i) \in K$, так що $\sigma(K) \in K$ (включення $\sigma(K) \subset K$ є очевидним, і тоді рівність випливає з порівняння степенів $\sigma(K)$ і K над $\mathbb{Q}$). Поле K , очевидно, є *полем розкладу* $f(X)$ над $\mathbb{Q}$, і можна переконатися в тому, що таке поле є розширенням Галуа. І навпаки, можна довести, що, якщо числове поле K є розширенням Галуа $\mathbb{Q}$, то воно є полем розкладу деякого полінома $f(X) \in \mathbb{Q}[X]$.

Повернемося тепер до еліптичних кривих. Як вже зазначалося раніше, у криптографічних прикладеннях еліптична крива розглядається над деяким скінченим полем K (полем Галуа $GF(p^n)$), тобто крива $E(K)$ над полем K визначається як множина точок $(x, y) \in K \times K$, що є розв'язками рівняння (1), в якому $a, b, c \in K$ є константами, разом з додатною точкою O – “точкою на нескінченності”. Відповідно до умови 1 щодо відбору еліптичних кривих, розглянутій вище, константи a, b, c повинні бути вибраними таким чином, щоб крива E була несингулярною.

Нас цікавитимуть точки еліптичної кривої порядку n , які утворюють групу кручення $E[n]$ (підгрупу групи комплексних точок кривої E), тобто $E[n] = \{P \in E(\overline{K}) : nP = O\}$ (нагадаємо, що $\overline{K}$ позначає замкнення поля K).

¹² Група Галуа – група автоморфізмів Галуа розширення L поля k , тобто група, що складається зі всіх автоморфізмів поля L , які залишають усі елементи підполя k нерухомими. Група Галуа позначається $G(L/k)$.

¹³ Мінімальним полем $P(M)$, котре містить дану числову множину M , називається поле, котре є перетином всіх числових полів, які містять множину M .

Будемо розглядати $E[n]$ як абстрактну групу. Справедливе таке твердження [3]

Твердження. Як абстрактна група $E[n] \cong \frac{\mathbb{Z}}{n\mathbb{Z}} \oplus \frac{\mathbb{Z}}{n\mathbb{Z}}$ є прямою сумою двох циклічних груп порядку n .

Це твердження, як бачимо, є частковим виглядом подання (3), і воно означає, що $E[n]$ породжується двома "базисними" елементами, скажімо, P_1 і P_2 . Отже, відповідно до виразу (4) елементи $E[n]$ можуть бути подані у вигляді множини

$$\left\{ a_1 P_1 + a_2 P_2 : a_1, a_2 \in \frac{\mathbb{Z}}{n\mathbb{Z}} \right\},$$

і значить кожний елемент $E[n]$ може бути записаним для унікальної пари елементів $a_1, a_2 \in \mathbb{Z}/n\mathbb{Z}$ як $a_1 P_1 + a_2 P_2$.

Згідно з [3] розглянемо тепер більш докладніше так званий мультиплікативний n -гомоморфізм

$$\phi: E(\mathbb{C}) \xrightarrow[n]{\text{множення}} E(\mathbb{C})P \mapsto nP. \quad (5)$$

Очевидно, що ядром гомоморфізму $\phi(P) = nP$ є саме підгрупа $E[n]$, тобто $E[n] = \text{Ker}(\phi) = \{P \in E(\mathbb{C}) : nP = O\}$.

Тому що P має порядок, що дорівнює точно n , то це буде множина точок, порядки яких діляться на n .

Підкреслимо, і це принципово, що мультиплікативний n -гомоморфізм на $E(\mathbb{C})$ має специфічну властивість, а саме – це відображення визначається за допомогою раціональних функцій, тобто x, y координати точки nP є раціональними функціями x, y координат точки P . Наприклад, якщо $P = (x, y)$ є точкою на еліптичній кривій

$$y^2 = x^3 + ax^2 + bx + c,$$

то після виконання обчислень можна виявити, що

$$2P = \left(\frac{g(x)}{4y^2}, \frac{h(x)}{8y^2} \right),$$

де

$$\begin{aligned} g(x) &= x^4 - 2bx^2 - 8cx + b^2 - 4ac, \\ h(x) &= x^6 + 2ax^5 + 5bx^4 + 20cx^3 + 5(4ac - b^2)x^2 + \\ &\quad + 2(4a^2c - ab^2 - 2bc)x + 4abc - b^3 - 8c^2. \end{aligned}$$

У загальному випадку гомоморфізм, який визначається за допомогою раціональних функцій, називається *ізогенією*. Таким чином, ізогенія є гомоморфізмом $\phi: E(\mathbb{C}) \rightarrow E(\mathbb{C})$,

який має форму $\phi(x, y) = \left(\frac{\text{поліном від } x \text{ та } y}{\text{поліном від } x \text{ та } y}, \frac{\text{поліном від } x \text{ та } y}{\text{поліном від } x \text{ та } y} \right)$.

Криптографічні перетворення виконуються на основі саме відображення (5), тобто вихідні точки еліптичної кривої перетворюються у точки цієї ж групи. Тому нас повинні цікавити ізогенії еліптичних кривих в собі. Такі ізогенії називаються *ендоморфізмами*¹⁴ (або іноді *алгебраїчними ендоморфізмами*, щоб підкреслити той факт, що вони визначаються через раціональні функції). Кожна еліптична крива має мультиплікативний n -ендоморфізм

¹⁴ Під ендоморфізмом алгебраїчної системи A розуміється відображення алгебраїчної системи в собі, що узгоджене з її структурою. В даному випадку ізогенія й визначає узгодженість гомоморфізму, який вона визначає за допомогою раціональної функції, із структурою групи точок еліптичної кривої. Поняття ендоморфізму є частковим випадком поняття гомоморфізму двох алгебраїчних систем.

(5), один для кожного цілого n . Аналіз показує, що більшість еліптичних кривих не має інших ендоморфізмів. Однак є деякі еліптичні криві з додатковими ендоморфізмами, які не є мультиплікативними n -ендоморфізмами.

Сфокусуємо увагу саме на цих особливих еліптичних кривих, котрі, як ми побачимо далі, заслуговують на те, щоб дати їм спеціальне ім'я.

Означення. Нехай E буде еліптичною кривою. Будемо вважати, що E має комплексне множення, якщо існує ендоморфізм $\phi: E \rightarrow E$, який не є мультиплікативним n відображенням.

Буде корисним зараз же навести приклади еліптичних кривих, які мають комплексне множення.

П р и к л а д 1. Еліптична крива $C: y^2 = x^3 + x$ має комплексне множення $\phi(x, y) = (-x, iy)$, тому що, якщо $y^2 = x^3 + x$, то $(iy)^2 = -y^2 = -x^3 - x = (-x)^3 + (-x)$. (Тут $i = \sqrt{-1}$, як звичайно).

П р и к л а д 2. Еліптична крива $E: y^2 = x^3 + 1$ має комплексне множення $\phi(x, y) = \left(\frac{-1 + \sqrt{-3}}{2} x, -y \right)$, тому що, якщо $\left(\frac{-1 + \sqrt{-3}}{2} \right)^3 = 1$, і легко побачити, що $\phi(x, y)$ дійсно є точкою на E .

У більш загальному випадку, якщо ϕ_1 та ϕ_2 є ендоморфізмами E , то ми можемо визначити новий ендоморфізм $\phi_1 + \phi_2$ через

$$(\phi_1 + \phi_2): E \rightarrow E, (\phi_1 + \phi_2)(P) = \phi_1(P) + \phi_2(P).$$

Новий ендоморфізм одержимо, якщо візьмемо композицію

$$(\phi_1 \cdot \phi_2): E \rightarrow E, (\phi_1 \cdot \phi_2)(P) = \phi_1(\phi_2(P)).$$

З цим "додаванням" та "множенням" множина ендоморфізмів кривої E стає кільцем $\text{End}(E)$. Якщо еліптична крива E не має комплексного множення, то це кільце обов'язково ізоморфне $\mathbb{Z}$, тобто звичайному кільцю цілих чисел (у такому кільці існують тільки ендоморфізми $\phi: E \rightarrow E$, які є мультиплікативними n відображеннями). Особливістю кільця ендоморфізмів у цьому випадку є його некомутативність.

Ендоморфізм $\phi: E \rightarrow E$ дає гомоморфізм $\phi: E[n] \rightarrow E[n]$ (так, якщо $P \in E[n]$, то $n\phi(P) = \phi(nP) = \phi(O) = O$ і, отже, $\phi(P) \in E[n]$). Це означає, що існують елементи $a, b, c, d \in \mathbb{Z}/n\mathbb{Z}$ такі, що $\phi(P_1) = a(P_1) + c(P_2)$, $\phi(P_2) = b(P_1) + d(P_2)$, тобто гомоморфізм $\phi: E[n] \rightarrow E[n]$ відповідає матриці $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Кільце таких матриць не комутативне. Наприклад, матриці

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ та } \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ не комутативні.}$$

І саме в цьому випадку можна застосувати спарування Вейля. Але, якщо E має комплексне множення, то кільце ендоморфізмів є точно більшим ніж, $\mathbb{Z}$. Розглянемо цей випадок докладніше.

Нехай $K/\mathbb{Q}$ буде будь-яким Галуа розширенням з уявною одиницею $i \in K$, і нехай $\sigma \in \text{Gal}(K/\mathbb{Q})$. Тоді для будь-якої точки $P \in E(K)$ еліптичної кривої з комплексним множенням маємо два шляхи дістати нову точку в $E(K)$, а саме, можна застосувати ендоморфізм ϕ до P або елемент розширення Галуа σ до P . Запитаємо, де σ і ϕ комутативні. Іншими словами, чи вірно, що

$$\sigma(\phi(P)) = \phi(\sigma(P)) \text{ для кожної точки } P \in E(K)?$$

Використовуючи введені вище означення, можна побачити, що для $\phi(x, y) = (-x, iy)$:

$$\sigma(\phi(P)) = \sigma(-x, iy) = (\sigma(-x), \sigma(iy)) = (-\sigma(x), \sigma(i)\sigma(y)).$$

$$\phi(\sigma(P)) = \phi(\sigma(x), \sigma(y)) = (-\sigma(x), i\sigma(y)).$$

Отже, σ і ϕ комутативні за умови, що $\sigma(i) = i$: іншими словами, вони будуть комутативні, якщо $\sigma \in \text{Gal}(K/\mathbb{Q}(i))$. Тому, якщо йти до розгляду відображення ϕ , щоб вивчати групи Галуа, то ми будемо це робити у сенсі розгляду Галуа розширень $\mathbb{Q}(i)$ скоріше, ніж $\mathbb{Q}$. Саме еліптичні криві з комутативним ендоморфізмом $\text{End}(E)$ і є тими, що потрібні для криптографічних застосувань. Ці криві були названі *несуперсингулярними* на відміну від *суперсингулярних* еліптичних кривих, у яких кільце ендоморфізмів некомутативне. Зафіксуємо цей факт у формі означення.

Означення. Еліптична крива E називається *суперсингулярною*, якщо кільце ендоморфізмів $\text{End}(E)$ є некомутативним.

Зауважимо, що поряд з визначенням суперсингулярності через властивості кільця ендоморфізмів використовуються також і інші (еквівалентні) означення суперсингулярності [11].

Наприклад, для "суперсингулярності" кривої E , що задана рівнянням

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

можна скористатись виконанням будь-якої з наступних умов: (i) $a_1 = 0$, (ii) $|E(K)|$ є парним ($K = F(p^m)$), (iii) j -інваріант E є нулем.

Залишається помітити, що непридатними для криптографічних застосувань вважаються також так звані *аномальні криві*. Еліптична крива E , визначена над $F(p)$, називається *аномальною*, якщо $\#(E) = p$ (p – просте число), тобто це така крива, група точок якої збігається з звичайною групою простого порядку.

В якості загального висновку можна визначити, що саме несуперсингулярні еліптичні криві при побудуванні криптосистем відкривають можливість широкого вибору між багатьма різними групами різного порядку, які можна отримати за допомогою еліптичних кривих. Ця різниця виявляється додатковою їх перевагою в порівнянні з використанням груп скінченних полів, де є тільки один кандидат для кожного поля [6].

Список літератури: 1. Miller V.C. Use of Elliptic Curves in Cryptography; Advances in Cryptology: Proceedings of Crypto 85, Springer LNCS 218, 1986. p. 417-426. 2. Lenstra H.W., Jr Factoring integers with elliptic curves. Ann Of Math, (2) 126 (1987). p. 674-745. 3. Silverman J. The Arifmetic of Ellipic Curves. Springer-Verlag. 1986. 4. Долгов В.І., Лисицька І.В. Теорія груп та кільць: Конспект лекцій з дисципліни "Спеціальні розділи математики", {Харків: ХТУРЕ, 2000. 5. Завало С.Т. Алгебра и теория чисел. К.: Вища шк., 1980. Ч.2. 402 с. 6. Beth Y., Schaefer F. Non Supersingular Elliptic Curves for Publik Key Cruptosystems. 1998, Springer-Verlag. 7. Koblitz N. Elliptic Curve Cryptosystems; Mathematics of Computation, V. 48, № 177. 1987. P. 203-209. 8. Menezes A., Okamoto T. and Vanstone. S. Reducing Elliptic Curve Logarithms to a Finite Field, IEEE Trans. Info. Theory, 39, 1993. P. 1693-1646. 9. Coppersmith D. Fast evaluation of logarithms in fields of characteristic two; IEEE Trans. Inform. Theory, IT 30. 1984. p. 587-594. 10. Koblitz N. Constructing Elliptic Curve Cryptosystems in Characteristic 2. Springer-Verlag, 1998.

Харківський національний
університет радіоелектроніки

Надійшла до редколегії 12.11.2004

ОСНОВНІ ПОЛОЖЕННЯ МАТЕМАТИЧНОГО АПАРАТУ СУБ'ЕКТИВНОЇ ЛОГІКИ ТА ЙОГО ЗАСТОСУВАННЯ ДЛЯ ОЦІНКИ РІВНЯ ЗРІЛОСТІ СИСТЕМ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ІНФОРМАЦІЇ

Вступ

У відповідності до сучасних поглядів, забезпечення безпеки інформації в організаціях, відомствах, та підприємствах та у корпоративних інформаційно-телекомунікаційних системах (ІТС) здійснюється на чотирьох рівнях:

- законодавчому (нормативно-правовому);
- адміністративному;
- процедурному (організаційно-технічному);
- технологічному (програмно-технічному).

На кожному з цих рівнів застосовують відповідні підходи (концепції, стратегії) забезпечення безпеки інформації, заходи безпеки, методи та засоби захисту інформації.

Суть однієї з актуальних проблем, що сформулювались на сучасному етапі розвитку систем забезпечення безпеки інформації на процедурному рівні полягає у наступному.

Основним підходом захисту інформації на процедурному рівні є здійснення ефективної практичної діяльності щодо впровадження політики безпеки (концепції, стратегії) організації або ІТС. Основою організації практичної діяльності є застосування так званих "добрих практик безпеки", тобто тих практичних рішень у різних напрямках діяльності із забезпечення безпеки інформації, які добре себе зарекомендували. На цей час розроблено вже декілька стандартів, які у відповідності до моделей управління безпеки визначають моделі практичної діяльності, наприклад ISO/IEC 17779, ISO/IEC 21827, NIST SP 800-26, OCTAVE [1-4] тощо.

Але виникає проблема, як на процедурному рівні здійснити комплексну оцінку ефективності практичної діяльності, "якість" організації та виконання тих або інших практичних заходів безпеки. За допомогою такої оцінки керівник організації, відповідальні посадові особи отримають деяку вивченість, гарантії того, що стан безпеки на підпорядкованих об'єктах відповідає визначеному рівню.

Одним із підходів щодо вирішення такої проблеми є застосування моделей зрілості організації систем та процесів [2]. Основними методами оцінки рівня зрілості є методи неформального системного аналізу на основі експертиз.

У даній роботі наводить огляд основних положень теорії суб'єктивної логіки, яка запропонована Аудуном Джосангом [6-9]. На думку авторів статті апарат суб'єктивної логіки є більш зручним, у порівнянні з іншими для здійснення експертної оцінки рівня зрілості виконання практик безпеки [10-12]. Це припущення перевірено на практиці у ході розробки та застосуванні експертної системи "Радник" [13-14].

У огляді проводиться визначення сутності невизначених імовірностей, надаються базові визначення та терміни суб'єктивної логіки, визначається простір думок та надаються практичні приклади застосування суб'єктивної логіки для розв'язання парадоксу Елсберга.

У третій частині розглядаються основні оператори суб'єктивної логіки, які дозволяють формалізувати обробку думок експертів.

1 Логіка для невизначених імовірностей

Стандартна логіка оперує двома значеннями "ІСТИНА" та "БРЕХНЯ". Але явища реального світу складно оцінювати за допомогою цих категорій. До того ж не можна заперечити

чувати, що будь-яка оцінка надана людиною, з самого початку носить відбиток суб'єктивності. Отже стандартна логіка може застосовуватися лише для ідеалізованого світу.

З метою вирішення цієї проблеми було запропоновано декілька математичних апаратів та логік, які в деякій мірі враховували ступень невизначеності чи незнання. Вони переважно застосовувались для вирішення практичних ситуацій у яких мають місце недостатні спостереження, що зумовлюють невизначеність [15,16]. Введення невизначеності було великим кроком до реалізації процесу людської довіри, але поза розглядом залишився той факт, що думку висловлює певна людина (індивід) і отже вона суб'єктивна за своєю суттю.

Наступним кроком до моделювання процесів довіри є розробка математичного апарату суб'єктивної логіки (СЛ), що оперує суб'єктивною довірою до сутностей світу. Для вираження суб'єктивної довіри в рамках цього апарату використовується термін "думка". Для здійснення обчислень над думками вводяться спеціальні оператори СЛ. Деякі з них мають аналог у стандартній логіці, але існують оператори, що залежать від того, хто висловлює довіру. Думка може розглядатися як міра імовірності, що містить вторинну невизначеність, а власне СЛ може розглядатися як розширення та поєднання імовірнісної моделі та бінарної логіки.

Суб'єктивну логіку необхідно відрізнити від теорії нечітких множин. Оскільки теорія нечітких множин оперує з чіткими та визначеними значеннями відносно розпливчастих та нечітких твердженнях, в той час як СЛ оперує мірами невизначеності о чітко сформульованих твердженнях.

2 Сутність невизначених імовірностей

2.1 Модель довіри

Для пояснення сутності невизначених імовірностей буде використовуватися модель довіри, що схожа на модель довіри в теорії доказів Демпстера-Шафера [17]. Теорія Демпстера-Шафера була розроблена в 1960 році у якості праці по визначенню верхніх та нижніх границь імовірності. Подальшого розвитку вона набула у 1970 році у праці Шафера «Mathematical Theory of Evidence».

Перший крок у застосуванні моделі довіри Демпстера-Шафера – це визначення усього набору можливих ситуацій, що має назву **фрейм розрізнення**. Фрейм розрізнення задає множину станів даної системи, але при цьому існує умова: лише один стан може бути істинним у певний момент часу. На рис. 1 подано фрейм розрізнення, що складається з чотирьох елементарних станів. $x_1, x_2, x_3, x_4 \in \theta$

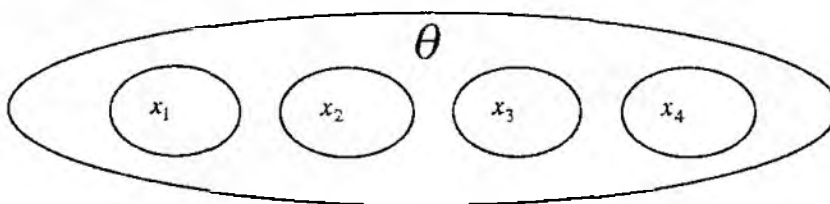


Рис. 1

З метою опису фрейма розрізнення, скористуємося положеннями стандартної теорії множин. Але з огляду на область застосування будемо замість терміну "набір" використовувати термін "стан". Як було вище зазначено, система може знаходитися водночас лише в одному елементарному стані, тобто лише один елементарний стан може бути істинним у певний момент часу. У цьому разі постає питання: "Яку частину істинності буде містити суперпозиція, що включає цей стан?". Наприклад, якщо x_2 істинно, виходить $x_2 \cup x_3$, а також усі інші суперпозиції, що містять x_2 також істинні. Слід замітити, що θ – завжди частково істин у силу свого визначення. Але без призначення ваги довіри, все ці спостереження не дуже корисні.

Атомарний стан – елементарний стан у фреймі розрізнення, тобто такий, що не містить підстанів. Ступенева множина θ позначається через 2^θ і містить усі атомарні стани, та усі їх сполучення, включаючи і θ . Ступенева множина може бути як кінцевою так і нескінченною в залежності від того, чи є кінцевим фрейм розрізнення.

Спостерігач що виказує думку в те, що один або декілька станів у ступеневій множині може бути істинним має призначити вагу довіри. Вага довіри атомарного стану $x \in 2^\theta$, це фактично віра в те, що саме цей стан істинний. Вага довіри в неатомарний стан $x \in 2^\theta$, це віра в те, що один із атомарних станів, що входять до нього істинний, але спостерігач невпевнений який саме. Нижче наведене визначення є центральним у теорії Демпстера-Шафера [17].

Визначення 1 (Призначена вага довіри)

Нехай θ - фрейм розрізнення. Якщо до кожного стану $x \in 2^\theta$ призначено таке значення $m_\theta(x)$, що виконуються наступні умови:

1. $m_\theta(x) \geq 0$
2. $m_\theta(\emptyset) = 0$
3. $\sum_{x \in 2^\theta} m_\theta(x) = 1$

то $m_\theta(x)$ – призначена вага довіри (ПВД). Для кожного підстану $x \in 2^\theta$ число $m_\theta(x)$ має назву вага довіри до x .

На рис.2 показана частина ступеневої множини, у фреймі розрізнення рис.1, із атомарними станами x_1, x_2, x_3, x_4 та неатомарними станами x_5, x_6 та фреймом розрізнення θ . Всі ці стани є елементами 2^θ , а ваги довіри встановлені у відповідності до визначення 1.

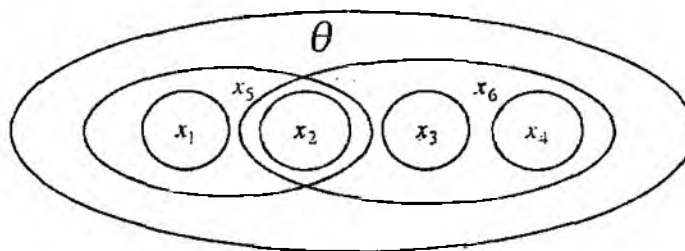


Рис.2

Вага довіри $m_\theta(x)$ позначає лише віру в істинність стану x у цілому, тобто не містить ніякої віри в стани, що входять до нього. Наприклад у разі, коли вага довіри призначена для x_5 , то це позначає віру спостерігача в те, що x_1 або x_2 істинно, але що саме невідомо.

На противагу вазі довіри, довіра до стану має розглядатися як абсолютна впевненість спостерігача, що конкретний стан істинний. Роздивимось визначення з теорії Демпстера-Шафера, що вказує на те, що довіра залежить не лише від ваги довіри призначеної до x , але й ще від значень ваг довіри до його підстанів.

Визначення 2 (Функція довіри). Нехай θ - фрейм розрізнення, а $m_\theta(x)$ – призначена вага довіри до θ . В такому випадку, функція довіри, що відповідає $m_\theta(x)$ це функція $b: 2^\theta \rightarrow [0,1]$, що визначена наступним чином [5]:

$$b(x) = \sum_{y \subseteq x} m_\theta(y), x, y \in 2^\theta.$$

За аналогією недовіра спостерігача має визначатися, як абсолютна впевненість у тому, що стан не є істинним.

Визначення 3 (Функція недовіри) Нехай θ - фрейм розрізнення, а $m_\theta(x)$ – призначена вага довіри до θ . В такому випадку, функція недовіри, що відповідає $m_\theta(x)$ це функція $d: 2^\theta \rightarrow [0,1]$, що визначена наступним чином [5]:

$$d(x) = \sum_{y \cap x = \emptyset} m_\theta(y), x, y \in 2^\theta.$$

Наприклад недовіра до стану x_5 (рис. 2) – це є сума ваг довіри до станів x_3 та x_4 тобто для усіх, що не перетинаються з x_5 . У книзі Шафера терміну “недовіра” до x відповідає термін “сумнів” відносно x . Але в теорії суб’єктивної логіки використовується термін “недовіра”, тому що, коли достовірно відомо, що стан неістинний – термін “абсолютна недовіра” набагато краще відбиває ситуацію ніж термін “абсолютний сумнів”.

У суб’єктивній логіці невизначеність поточного стану, це сума ваг довіри суперпозицій цього стану або станів, що частково перекриваються з даним.

Визначення 4 (Функція невизначеності) Нехай θ - фрейм розрізнення, а $m_\theta(x)$ – призначена вага довіри до θ . В такому випадку, функція невизначеності, що відповідає $m_\theta(x)$ це функція $u: 2^\theta \rightarrow [0,1]$ що визначена наступним чином [5]:

$$u(x) = \sum_{\substack{y \cap x = \emptyset \\ y \not\subseteq x}} m_\theta(y), x, y \in 2^\theta.$$

У нашому прикладі (рис. 2) невизначеність для x_5 це сума ваг довіри до x_6 та θ .

Абсолютна невизначеність може бути представлена випадком, коли для θ призначена вага довіри рівна одиниці. Функція довіри в такому разі має назву вакуумної функції довіри.

Якщо ПВД для θ дорівнює нулю то ПВД називається догматичною. У наступних розділах доводиться, що догматична ПВД є штучною за своєю природою і може існувати лише для гіпотетичних ситуацій.

Теорема 1 (Закон нормування для функцій довіри) [5]

$$b(x) + d(x) + u(x) = 1, x \in 2^\theta, x \neq \emptyset \quad (1)$$

Доказ 1.

Сума функцій довіри, недовіри та невизначеності дорівнює сумі вагів у ПВД, що у відповідності до визначення 1 дорівнює одиниці.

Вираз (1) є центральним у моделі довіри для суб’єктивної логіки. Функція невизначеності позначає невизначеність спостерігача відносно істинності даного стану та може бути пояснена, як дещо що заповнює відстань між довірою та недовірою.

Для пояснення невизначених імовірностей важливим також є спільна кількість атомарних станів. Припустимо, що вага довіри до $m_\theta(\theta) = 1$ призначено для θ (рис.2). Інтуїтивно здається, що імовірність того, що x_1 буде істинно дорівнює $\frac{1}{4}$, в наслідок того, що кожний з чотирьох атомарних станів може бути істинним і при цьому немає підстав для того, щоб стверджувати що вони не рівноімовірні. Тепер зробимо припущення, що вага довіри

$m_\theta(x_5) = 1$ призначено до x_5 . Виходячи з тих же міркувань, тепер імовірність того, що стан x_1 буде істинним дорівнює $\frac{1}{2}$.

Для будь якого стану x його атомарність це кількість станів, що він містить. Атомарність позначається як $|x|$. Наприклад для рис.2 $|x_5| = 2$. Якщо θ – фрейм розрізнення, то його атомарність дорівнює загальній кількості станів, що містяться в ньому. Тобто, якщо для $x, y \in 2^\theta$, то перетинання x по відношенню до y може бути виражено з використанням кількості атомарних станів.

Визначення 5 (Відносна атомарність). Нехай θ – фрейм розрізнення, а $x, y \in 2^\theta$. У такому випадку відносна атомарність, це функція $a: 2^\theta \rightarrow [0,1]$ що визначена наступним чином [5]:

$$a(x/y) = \frac{|x \cap y|}{|y|}, x, y \in 2^\theta$$

Неважко помітити, що $x \cap y = \emptyset \Rightarrow a(x/y) = 0$ і що $y \subseteq x \Rightarrow a(x/y) = 1$. Для усіх інших випадків відносна атомарність змінюється у проміжку від нуля до одиниці. Для нашого прикладу (рис. 2) відносна атомарність x_6 до x_5 дорівнює:

$$a(x_6/x_5) = |x_6 \cap x_5|/|x_5| = \frac{1}{2}$$

Відносна атомарність будь якого атомарного стану по відношенню до його фрейму розрізнення записується як $a(x)$

Фрейм розрізнення із відповідною ПВД може використовуватися для знаходження математичного очікування імовірності для будь якого стану. Невизначеність оказує вплив на математичне очікування імовірності, але буде мати різну вагу в залежності від відносної атомарності. Наприклад, якщо, розглядається вага довіри x_1 по відношенню до x_5 та θ , то обидва вони обчислюються як невизначеність, але вага довіри до θ буде меншою, внаслідок того, що відносна атомарність x_1 менше по відношенню до θ ніж до x_5 .

Визначення 6 (Математичне очікування імовірності). Нехай θ – фрейм розрізнення, а $m_\theta(x)$ – призначена вага довіри до θ . В такому випадку, функція математичного очікування імовірності, що відповідає $m_\theta(x)$ це функція $E: 2^\theta \rightarrow [0,1]$, що визначена наступним чином [5]:

$$E(x) = \sum_y m_\theta(y) a(x/y), y \in 2^\theta. \quad (2)$$

Подане визначення еквівалентне (pignistic probability) поданому у [18], яка заснована на принципі недостатніх спостережень. Вага довіри призначена для об'єднання із n атомарних станів поділяється рівномірно серед них.

Таким чином математичне очікування імовірності для даного стану визначається ПВД та кількістю атомарних станів. Треба замітити, що одному значенню функції математичного очікування імовірності може відповідати нескінченна кількість ПВД.

Функції довіри Шафера та можливість вимірювань були інтерпретовані відповідно, як верхня та нижня границя [19]. З точки зору СЛ функції довіри можуть розглядатися тільки для оцінки значень імовірності, а не для встановлення границь, оскільки імовірність реальних подій ніколи не може бути визначена з впевненістю, отже не можуть бути визначені границі.

2.2 Сфокусований фрейм розрізнення

Визначення 7 (Сфокусований фрейм розрізнення). Нехай θ - фрейм розрізнення, а $x \in 2^\theta$. Фрейм розрізнення позначений як $\tilde{\theta}^x$ до якого входять два атомарних стани x та $-x$, де $-x$ є таким, що доповнює (компліментарним) до x в θ має назву сфокусований фрейм розрізнення з фокусом у x .

Перехід від початкового фрейму розрізнення (рис.1) до сфокусованого фрейму розрізнення з фокусом у $x_7 = (x_2 \cup x_3)$ представлено на рис. 3. Це може бути пояснено наступним чином – вага довіри, що назначена усім станам, що нарисовані сполушною лінією у лівій частині рисунку, а x_7 (пунктирна лінія) визначено як один з двох станів у сфокусованому фреймі розрізнення.

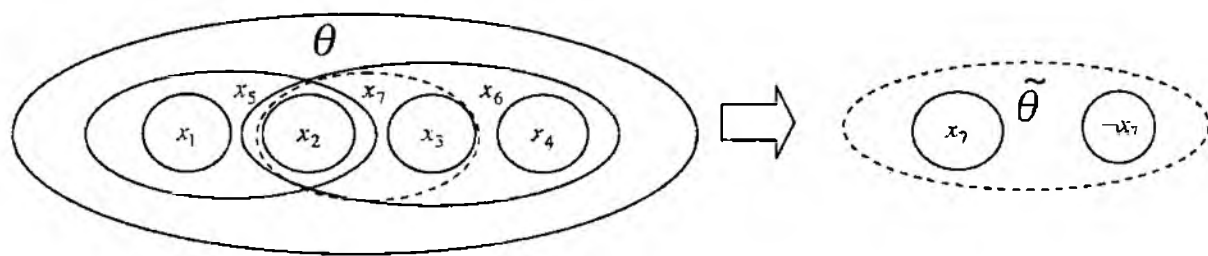


Рис. 3

Визначення 8 (сфокусована ПВД та відносна атомарність). Нехай θ - фрейм розрізнення, $m_\theta(x)$ – призначення вага довіри до θ , а $b(x)$, $d(x)$, $u(x)$ - функції довіри, недовіри та невизначеності до x у степеневій множині 2^θ . Також $\tilde{\theta}^x$ - буде сфокусованим фреймом розрізнення з фокусом у x . В такому випадку сфокусована ПВД $m_{\tilde{\theta}^x}$ на $\tilde{\theta}^x$ визначається к:

$$\begin{aligned} m_{\tilde{\theta}^x}(x) &= b(x) \\ m_{\tilde{\theta}^x}(-x) &= d(x) \\ m_{\tilde{\theta}^x}(\tilde{\theta}^x) &= u(x) \end{aligned} \quad (3)$$

а відносна атомарність до x буде дорівнювати:

$$a_{\tilde{\theta}^x}(x) = [E(x) - b(x)] / u(x) \quad (4)$$

Не складно помітити, що функції довіри, недовіри та невизначеності для x однакові як 2^θ так і для $2^{\tilde{\theta}^x}$. Сфокусована відносна атомарність визначена таким чином, що математичне очікування значення імовірності стану x однакова для θ та $\tilde{\theta}^x$, а вираз для $a_{\tilde{\theta}^x}$ у визначені 8 може буде визначений з використанням визначення 6.

У загальному випадку сфокусована відносна атомарність буде відрізнятися від $\frac{1}{2}$ не дивлячись на те, що $\tilde{\theta}^x$ містить лише два стани. Це виведене значення, що позначає зважене середнє значення відносних атомарностей x до усіх інших станів 2^θ у якості функції їх невизначених мас.

Використання сфокусованого фрейму розрізнення, сфокусованої ПВД та сфокусованої відносної атомарності надає можливість працювати з бінарними фреймами розрізнення замість використання повного простору станів, що є незаперечною перевагою.

2.3 Приклад А: Призначення ваги довіри

У якості прикладу використаємо фрейм розрізнення, що зображено у лівій частині рис. 3 із наступною ПВД

$$m_{\theta} : \begin{cases} m_{\theta}(x_1) = 0,10 \\ m_{\theta}(x_2) = 0,20 \\ m_{\theta}(x_3) = 0,20 \\ m_{\theta}(x_4) = 0,00 \\ m_{\theta}(x_5) = 0,10 \\ m_{\theta}(x_6) = 0,30 \\ m_{\theta}(\theta) = 0,10 \end{cases}$$

Обчислимо функції довіри, недовіри та невизначеності для стану x_7

$$\begin{aligned} b(x_7) &= m_{\theta}(x_2) + m_{\theta}(x_3) = 0,40 \\ d(x_7) &= m_{\theta}(x_1) + m_{\theta}(x_4) = 0,10 \\ u(x_7) &= m_{\theta}(x_5) + m_{\theta}(x_6) + m_{\theta}(\theta) = 0,50 \end{aligned} \quad (5)$$

Згідно з визначенням 6 розрахуємо математичне очікування імовірності для x_7 .

$$E(x_7) = m_{\theta}(x_1) \cdot 0 + m_{\theta}(x_2) \cdot 1 + m_{\theta}(x_3) \cdot 1 + m_{\theta}(x_4) \cdot 0 + m_{\theta}(x_5) \cdot \frac{1}{2} + m_{\theta}(x_6) \cdot \frac{2}{3} + m_{\theta}(\theta) \cdot \frac{1}{2} = 0,7$$

Згідно до виразів (3) та (5) сфокусований фрейм розрізнення має наступний вигляд:

$$m_{\tilde{\theta}^{x_7}} : \begin{cases} m_{\tilde{\theta}^{x_7}}(x_7) = 0,40 \\ m_{\tilde{\theta}^{x_7}}(-x_7) = 0,10 \\ m_{\tilde{\theta}^{x_7}}(\tilde{\theta}^{x_7}) = 0,50 \end{cases}$$

Згідно до виразу (4) сфокусована відносна атомарність для x_7 дорівнює

$$a_{\tilde{\theta}^{x_7}}(x) = [0,7 - 0,4] / 0,5 = 0,60 \quad (6)$$

Перевагою сфокусованого ПВД є його більша компактність, яка пояснюється вираженням довіри лише до стану, що знаходиться у фокусі.

2.4 Простір думок

Після розгляду фундаментальних концепцій спробуємо уявити невизначені імовірності у почуттєвому досвіді. З цією метою визначимо 3-хмірну метрику, що отримало у СЛ назву "думка". Для більш простого виконання логічних операцій думка буде містити, ще один четвертий додатковий параметр.

Визначення 9 (Думка) [5]. Нехай θ - двійковий фрейм розрізнення з двома атомарними станами x та $-x$, $m_{\theta}(x)$ - це призначена вага довіри до θ , а $b(x)$, $d(x)$, $u(x)$, $a(x)$ - функції довіри, недовіри, невизначеності та відносної атомарності до x у ступеневій множині 2^{θ} ,

відповідно. У такому випадку думка відносно x , що позначається як ω_x , являє собою групу пов'язаних величин.

$$\omega_x \equiv (b(x), d(x), u(x), a(x)). \tag{7}$$

З метою компактності довіру (belief), недовіру (disbelief), невизначеність (uncertainty) та відносну атомарність у подальшому будемо позначати як b_x , d_x , u_x та a_x відповідно. Позначення $E(\omega_x)$ буде рівнозначним до запису $E(x)$ і буде позначати математичне очікування імовірності думки.

У виразі (1) були визначені три залежні координати (b, d, u) , таким чином один з параметрів думки є надлишковим. В цілому трійка параметрів позначає не більше ніж традиційна пара (Довіра, Правдоподібність) у теорії довіри Шафера. Однак, три координати дозволяють використовувати оператори СЛ. Вираз (1) має графічне представлення (рис. 4).

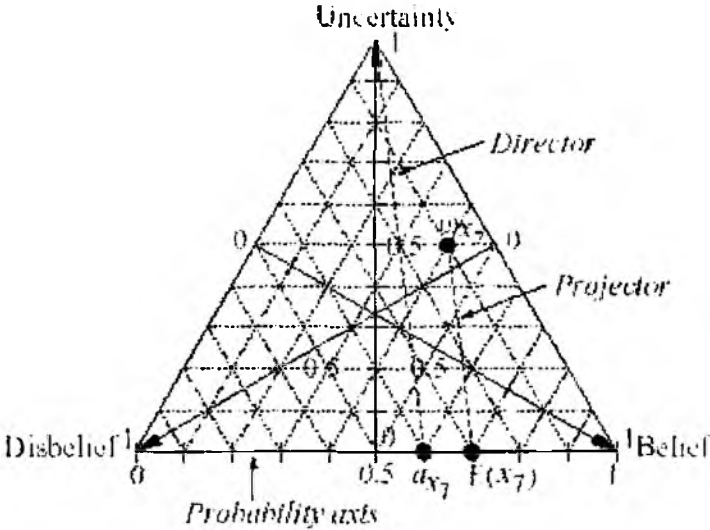


Рис. 4

Думку з прикладу розділу 2.3 $\omega_{x_7} = (0.40, 0.10, 0.50, 0.60)$ представлено як точку у просторі трикутника думок. Також на рисунку показані математичне очікування імовірності $E(x_7)$ та відносна атомарність a_{x_7} .

Горизонтальна лінія, що єднає кути довіри та недовіри (рис.4) називається *віссю імовірності (probability axis)*. Відносна атомарність може бути представлена графічно як точка на осі імовірності. Лінія, що єднає верхній кут трикутника та точку відносної атомарності називається *директор (director)*.

Лінія, що проходить через точку думки рівнобіжно директору має назву *прожектор (projector)*. Точка його перетинання з віссю імовірності визначає значення математичного очікування імовірності, яке обчислюється за виразом (6).

Думки, що розташовані на осі імовірності називаються *догматичними думками*. Вони представляють ситуацію відсутності ступеню невизначеності та відповідають традиційним частотним імовірностям. Відстань між точкою думки та віссю імовірності може бути інтерпретовано як ступінь невизначеності.

Думки, що розташовані у лівому або правому куті, тобто або $(b=1; d=0; u=0)$ або $(b=0; d=1; u=0)$ мають назву *абсолютні думки*. Вони представляють ситуацію, абсолютної впевненості у тому, що стан істинний або навпаки і відповідають поняттям “БРЕХНЯ” та “ІСТИНА” у бінарній логіці. У відповідності до зроблених визначень були дані фундаментальні аксіоми Колмогорова із традиційної теорії імовірності у якості теорем.

Теорема 2 (Аксіоми Колмогорова)

У заданому фреймі розрізнення – θ , з призначеною до нього вагою довіри $m_\theta(x)$, функція математичного очікування імовірності E в області 2^θ задовольняє наступним виразами:

1. $E(x) \geq 0$ для усіх $x \in 2^\theta$.

2. $E(\theta) = 1$.

3. Якщо $x_1, x_2 \dots \in 2^\theta$ попарно не перетинаються, то $E\left(\bigcup_{i=1}^{2^\theta} x_i\right) = \sum_{i=1}^{2^\theta} E(x_i)$.

Доказ 2. Докажемо кожен властивість окремо.

1. Згідно з виразами (1) та (2) $b_x \geq 0$, $d_x \geq 0$ та $a_x \geq 0$ для усіх x . Таким чином, будь яке математичне очікування імовірності у відповідності до визначення 6 буде задовольняти нерівності $0 \leq E(x) \leq 1$.

2. Згідно визначенню 1 $b_\theta = 1$ та $n_\theta = 0$, ось чому $E(\theta) = 1$

3. Нехай $x_1, x_2 \dots \in 2^\theta$ множини, що попарно не перетинаються, тобто $x_i \cap x_j = \emptyset$ для $i \neq j$. Відповідно до визначення 6 маємо:

$$E(x_i \cup x_j) = \sum_y m_\theta(y) a((x_i \cup x_j) / y), y \in 2^\theta, \quad (8)$$

а в силу того, що x_i та x_j не перетинаються:

$$a((x_i \cup x_j) / y) = a(x_i / y) + a(x_j / y). \quad (9)$$

Сума (8) може бути розбита на такий спосіб:

$$E(x_i \cup x_j) = \sum_y m_\theta(y) a(x_i / y) + \sum_y m_\theta(y) a(x_j / y), y \in 2^\theta = E(x_i) + E(x_j) \quad (10)$$

Даний результат можна узагальнити для будь якої кількості множин, що не перетинаються.

Упорядкування думок проводиться за математичним очікуванням імовірності, але у разі їх рівності необхідні додаткові критерії.

Визначення 10 (Упорядкування думок). Нехай ω_x та ω_y – дві думки. Вони можуть бути упорядковані у відповідності до нижче наведених критеріїв, що розташовані згідно пріоритетів у порядку спадання.

1. Думці з найбільшим математичним очікуванням імовірності – надається перевага.
2. Думці з найменшою невизначеністю – надається перевага.
3. Думці з найменшою відносно атомарністю надається перевага – надається перевага.

Перший критерій є очевидним. Другий критерій менш очевидний, але підтверджується дослідженнями Елсберга (див. Приклад Б). Третій критерій більше наочний. Таким чином пріоритет між другим та третім критерієм може бути встановлено читачем на основі запропонованих прикладів.

2.5 Приклад Б: Парадокс Елсберга

Парадокс Елсберга [20] є класичним прикладом неспроможності класичної теорії імовірності виразити невизначеність. Припустимо, що вам показують корзину з 90 кулями та говорять, що 30 з них червоні, а інші (60) чорного або жовтого кольору. Тепер випадковим чином буде обрано одну з куль. Перед вами постає вибір одного із можливих варіантів відповіді.

Вибір 1 – Ви отримаєте 100 \$ у випадку, якщо буде обрано червону кулю, та нічого у протилежному випадку.

Вибір 2 – Ви отримаєте 100 \$ якщо буде обрано чорна куля и нічого у протилежному випадку. У таблиці 1 наведені вищезазначені умови.

Таблиця 1

	Червона	Чорна	Жовта
Вибір 1	100 \$	0	0
Вибір 2	0	100 \$	0

Зробіть вибір, а тепер оберіть один з двох інших варіантів. Ці варіанти також засновані на випадковому обиранні кульок з скриньки.

Таблиця 2

	Червона	Чорна	Жовта
Вибір 3	100 \$	0	100 \$
Вибір 4	0	100 \$	100 \$

Зробіть вибір ще раз.

Елсберг стверджує, що у випадку запропонованих альтернатив більшість людей робить вибір 1 та вибір 4. Цей факт вказує на корисність теорії і цей факт зовсім несумісний з імовірнісною оцінкою. Коли людина надає перевагу вибору 1 перед вибором 2, вона обирає більш високу суб'єктивну оцінку "Червона", ніж "Чорна". Однак, коли та сама людина, надає перевагу вибір 4 перед вибором 3, вона вважає, що його суб'єктивна оцінка "Чорна або Жовта" вище ніж "Червона або Жовта", надивляючись на те, що "Чорна" має більш високу суб'єктивну оцінку ніж "Червона".

У випадку коли ми використовуємо невизначені імовірності у якості думок, вибір більшості стає найкращою логікою. На рис. 5 показано частину множини парадоксу Елсберга із відповідним ПВД.



Рис. 5

Практичність виборів 1 та 2 ґрунтується на думці y_1 "Червона" та y_2 «Чорна»

$$\text{Вибір 1: } \omega_{y_1} = \left(\frac{1}{3}, \frac{2}{3}, 0, \frac{1}{3} \right) \quad \text{Вибір 2: } \omega_{y_2} = \left(0, \frac{1}{3}, \frac{2}{3}, \frac{1}{2} \right)$$

Використавши визначення 6 маємо

$$\begin{cases} E(\omega_{y_1}) = \frac{1}{3} \\ E(\omega_{y_2}) = \frac{1}{3} \end{cases} \Rightarrow E(\omega_{y_1}) = E(\omega_{y_2})$$

Математичне очікування імовірності однакове для обох випадків, але ω_{y_2} містить на протилегу від ω_{y_1} невизначеність. На рис. 6 чітко показана різниця у рівні невизначеності між цими двома думками.

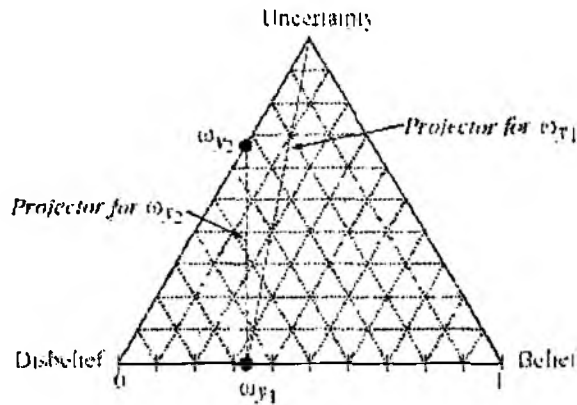


Рис. 6

Таким чином можна зробити висновок, що вибір 1 кращий за вибір 2 саме тому, що його імовірність менш невизначена.

Тепер розглянемо наступну пару можливих виборів 3 та 4. Це все рівно, що робити вибір між станами $y_1 \cup y_3$ – “Червона або Жовта” та y_4 – “Чорна або Жовта”, відповідно.

$$\text{Вибір 3: } \omega_{y_1 \cup y_3} = \left(\frac{1}{3}, 0, \frac{2}{3}, \frac{1}{2} \right) \quad \text{Вибір 4: } \omega_{y_4} = \left(\frac{2}{3}, \frac{1}{3}, 0, \frac{2}{3} \right)$$

Використавши визначення 6 маємо:

$$\begin{cases} E(\omega_{y_1 \cup y_3}) = \frac{2}{3} \\ E(\omega_{y_4}) = \frac{2}{3} \end{cases} \Rightarrow E(\omega_{y_1 \cup y_3}) = E(\omega_{y_4})$$

Математичне очікування імовірності однакове для обох випадків, але $\omega_{y_1 \cup y_3}$ містить невизначеність на противагу від ω_{y_4} . На рис. 7 чітко показана різниця у рівні невизначеності для цих двох думок.

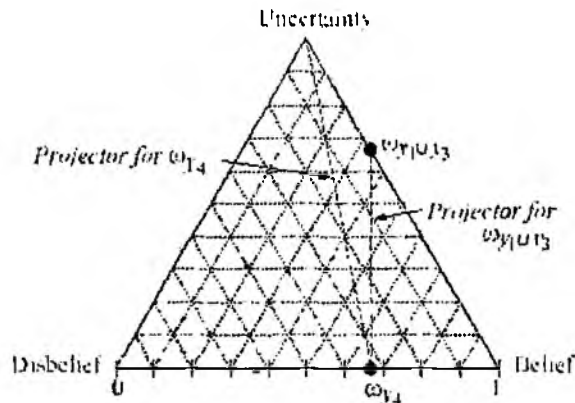


Рис. 7

Виходячи з вищезазначеного можна зробити висновок, що ω_{y_4} краще від $\omega_{y_1 \cup y_3}$ у наслідок того, що імовірність його виникнення має менше невизначеності.

Таким чином, зроблені переваги вибору 1 перед вибором 2 та вибору 4 перед вибором 3, цілком раціональні і власно кажучи не містять парадоксу у рамках моделі довіри, що розгля-

дається. Деякі інші моделі для невизначених імовірностей також можуть пояснити парадокс Елсберга. Але наступний приклад не може пояснити жодна з них.

2.6 Приклад В: Упорядкування думок

У цьому прикладі як і у парадоксі Елсберга, буде розглядатись заповнення скриньки кульками, але цього разу, це будуть кульки 9 різних кольорів.

Припустимо Вам показують скриньку з 80 кулями та говорять, що її заповнювали в три етапи. На першому етапі в неї було покладено 20 червоних кульок, та 10 із кольорової гами (червоний, чорний, жовтий). На другому етапі поклали 20 куль з кольорової гами (блакитний, білий, зелений, рожевий, коричневий та оранжевий). На третьому етапі 30 куль з усіма вище зазначеними кольорами. На рис. 8 подано фрейм розрізнення для цієї ситуації із відповідними ПВД.

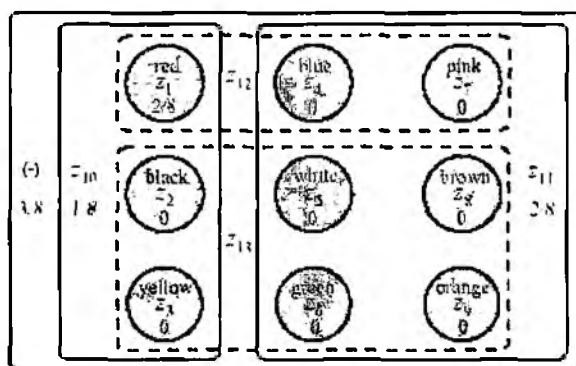


Рис. 8

За умовою, із скриньки має бути випадковим чином обрана одна куля. Порівняємо думки відносно станів $z_{10}, z_{11}, z_{12}, z_{13} \in 2^\theta$:

z_{10} "червона, чорна або жовта"

$$\omega_{z_{10}} = \left(\frac{3}{8}, \frac{2}{8}, \frac{3}{8}, \frac{1}{3} \right)$$

z_{11} "блакитна, біла, зелена, рожева, коричнева або оранжева"

$$\omega_{z_{11}} = \left(\frac{2}{8}, \frac{3}{8}, \frac{3}{8}, \frac{2}{3} \right)$$

z_{12} "червона, блакитна або рожева"

$$\omega_{z_{12}} = \left(\frac{2}{8}, 0, \frac{6}{8}, \frac{1}{3} \right)$$

z_{13} "чорна, жовта, біла, зелена, коричнева або оранжева"

$$\omega_{z_{13}} = \left(0, \frac{2}{8}, \frac{6}{8}, \frac{2}{3} \right)$$

Обчислимо математичне очікування імовірності:

$$E(\omega_{z_{10}}) = E(\omega_{z_{11}}) = E(\omega_{z_{12}}) = E(\omega_{z_{13}}) = 1/2$$

Проблема полягає в упорядкуванні думок з однаковими математичними очікуваннями імовірності. На рис. 9 можна побачити, що не дивлячись на рівність математичних очікувань імовірності, рівні невизначеності та відносні атомарності – різні.

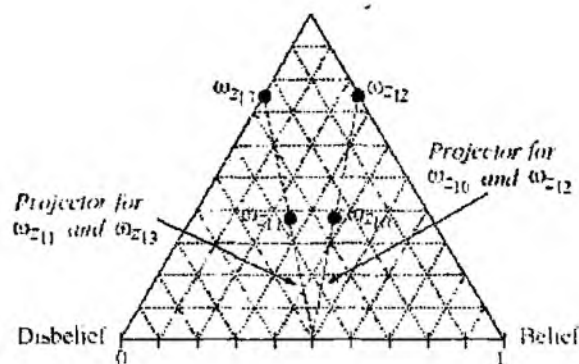


Рис. 9

У відповідності до другого критерію визначення 10, перевага надається точці, що розташована більш низько у трикутнику думок. У відповідності до третього критерію цього ж визначення перевагу треба надати тому, що знаходиться правіше. Таким чином, ми можемо провести упорядкування наступним чином $\omega_{z10} \succ \omega_{z11} \succ \omega_{z12} \succ \omega_{z13}$. Цей факт надає можливість побудови різних ранжирувань, в залежності від переваг експерта відносно сформульованих критеріїв.

3 Основні оператори суб'єктивної логіки

Стандартна логіка оперує на двійкових змінних, які можуть приймати дискретні значення «ІСТИНА» та «БРЕХНЯ». Суб'єктивна логіка оперує з думками, представленими у вигляді вектору $\omega_p = \{b_p, d_p, u_p, a_p\}$.

Думки, як і свідчення, розглядаються індивідуально і таким чином, вони обов'язково належать комусь і виносяться до чогось (чи когось). Думка експерта А про істинність (правдивості, вірогідності) твердження p позначається як ω_p^A .

У суб'єктивній логіці визначено 7 операторів [5]:

- кон'юнкція (об'єднання);
- диз'юнкція (роз'єднання);
- заперечення;
- незалежний консенсус;
- залежний консенсус;
- частково залежний консенсус;
- рекомендація.

3.1 Кон'юнкція

Формування кон'юнкції двох тверджень з сфокусованих фреймів розрізнення полягає у виробленні з двох думок стосовно кожного твердження, нової думки, що буде виражати віру в істинність обох тверджень одночасно. Цей оператор є аналогом "AND" у бінарній логіці.

Теорема 3 (Кон'юнкція тверджень)

Нехай θ_x та θ_y це два бінарних сфокусованих фрейма розрізнення, x та y - відповідно твердження стосовно станів в θ_x та θ_y . Тоді $\omega_x = (b_x, d_x, u_x, a_x)$ та $\omega_y = (b_y, d_y, u_y, a_y)$, це думки відносно цих станів, нехай $\omega_{x \wedge y} = (b_{x \wedge y}, d_{x \wedge y}, u_{x \wedge y}, a_{x \wedge y})$ така думка, що

1. $b_{x \wedge y} = b_x b_y$
2. $d_{x \wedge y} = d_x + d_y - d_x d_y$
3. $u_{x \wedge y} = b_x u_y + u_x b_y + u_x u_y$

$$4. a_{x \wedge y} = \frac{b_x u_y a_y + u_x a_x b_y + u_x a_x u_y a_y}{b_x u_y + u_x b_y + u_x u_y}$$

Тоді $\omega_{x \wedge y}$ має назву кон'юнкція тверджень думок ω_x та ω_y , що виражають думку, особи що приймає рішення, стосовно того що обидва твердження x та y є істинними. Для позначення цього оператора використовується символ $\wedge$ таким чином $\omega_{x \wedge y} \equiv \omega_x \wedge \omega_y$

3.2 Диз'юнкція

Формування диз'юнкції двох тверджень з сфокусованих фреймів розрізнення полягає у визначенні, з двох думок стосовно кожного твердження, нової думки, що буде виражати віру в істинність або одного тверджень або їх одночасну істинність. Цей оператор є аналогом "OR" у бінарній логіці.

Теорема 4 (Диз'юнкція тверджень)

Нехай θ_x та θ_y пс два бінарних сфокусованих фрейма розрізнення, x та y - відповідно твердження стосовно станів в θ_x та θ_y . Тоді $\omega_x = (b_x, d_x, u_x, a_x)$ та $\omega_y = (b_y, d_y, u_y, a_y)$, це думки відносно цих станів, нехай $\omega_{x \vee y} = (b_{x \vee y}, d_{x \vee y}, u_{x \vee y}, a_{x \vee y})$ така думка, що

1. $b_{x \vee y} = b_x + b_y - b_x b_y$
2. $d_{x \vee y} = d_x d_y$
3. $u_{x \vee y} = d_x u_y + u_x d_y + u_x u_y$
4. $a_{x \vee y} = \frac{u_x a_x + u_y a_y - b_x u_y a_y - u_x a_x b_y - u_x a_x u_y a_y}{u_x + u_y - b_x u_y - u_x b_y - u_x u_y}$

Тоді $\omega_{x \vee y}$ має назву кон'юнкція тверджень думок ω_x та ω_y , що виражають думку, особи що приймає рішення, стосовно того що або обидва одне з тверджень x чи y є істинними або вони істинні разом. Для позначення цього оператора використовується символ $\vee$ таким чином $\omega_{x \vee y} \equiv \omega_x \vee \omega_y$

3.3 Заперечення

Заперечення думки щодо твердження x визначає думку що x буде неістинне. Ця операція є аналогом заперечення "NOT" у бінарній логіці.

Теорема 5 (Заперечення)

Нехай $\omega_x = (b_x, d_x, u_x, a_x)$ це думка відносно x тоді $\omega_{\neg x} = (b_{\neg x}, d_{\neg x}, u_{\neg x}, a_{\neg x})$ це заперечення ω_x де:

1. $b_{\neg x} = d_x$
2. $d_{\neg x} = b_x$
3. $u_{\neg x} = u_x$
4. $a_{\neg x} = 1 - a_x$

Заперечення є інволютивним, тобто $\neg(\neg \omega_p) = \omega$ вірно для будь-якої думки ω .

3.4 Консенсус між незалежними думками

Консенсус між двома думками об'єднує їх, вважаючи, що вони незалежні та мають однакове значення. Наприклад якщо два експерта спостерігали за певним явищем у два проміжки часу, що не перекриваються, вони можуть мати про нього різні думки. Консенсус дозво-

ляє сформулювати нову думку, що повинна дорівнювати думці третім експертом, що спостерігав це явище у обидва проміжки часу.

Теорема 6 (Заперечення)

Нехай $\omega_x^A = \{b_x^A, d_x^A, u_x^A\}$ і $\omega_x^B = \{b_x^B, d_x^B, u_x^B\}$ думки, що належать відповідно експертам A і B , відносно твердження p . Нехай $\omega_x^{A,B} = \{b_x^{A,B}, d_x^{A,B}, u_x^{A,B}\}$ буде думкою, такою, що

1. $b_x^{A,B} = (b_x^A u_x^B + b_x^B u_x^A) / k$;
2. $d_x^{A,B} = (d_x^A u_x^B + d_x^B u_x^A) / k$;
3. $u_x^{A,B} = (u_x^A u_x^B) / k$.
4. $a_x^{A,B} = \frac{a_x^B u_x^B + a_x^A u_x^B - (a_x^A + a_x^B) u_x^A u_x^B}{u_x^A + u_x^B - 2u_x^A u_x^B}$

де $k = u_x^A + u_x^B - u_x^A u_x^B$ причому $k \neq 0$ та $a_x^{A,B} = (a_x^A + a_x^B) / 2$.

Операція консенсусу є комутативною й асоціативною, отже порядок комбінування думок не важливий. Передбачається, що думки незалежні. Не можна врахувати думки більш ніж оди раз.

Дві думки що містять нульову невизначеність, відповідно до теореми 1 не можуть комбінуватися. Це можна пояснити через ітерацію невизначеності "як кімната для впливу", тобто виникає можливість удлинути на думки, що ще не були висловлені ні за, ні проти твердження. Ситуація з незалежними абсолютно визначеними думками з філософської точки зору безглузда, по-перше тому, що думки не можуть бути абсолютно визначеними і по-друге, вони тоді повинні бути обов'язково однаковими.

3.5 Консенсус між залежними думками

Припустимо, що два експерти A і B одночасно спостерігають об'єкт, що породжує бінарні результати визначену кількість разів. Оскільки їхні спостереження ідентичні, то їхні думки будуть залежними і консенсус у змісті Теореми 6 не має сенсу.

Теорема 7 Нехай $\omega_x^{A_i} = \{b_x^{A_i}, d_x^{A_i}, u_x^{A_i}\}$ де $i \in \{1, \dots, n\}$ це n залежних думок експертів $A_1, \dots, A_n$ відносно твердження x . Нехай $\omega_x^{A_1, \dots, A_n} = \{b_x^{A_1, \dots, A_n}, d_x^{A_1, \dots, A_n}, u_x^{A_1, \dots, A_n}\}$ це така думка, що

$$b_h^{A_1, \dots, A_n} = \frac{\sum_{i=1}^n (b_p^{A_i} / u_p^{A_i})}{\sum_{i=1}^n (b_p^{A_i} / u_p^{A_i}) + \sum_{i=1}^n (d_p^{A_i} / u_p^{A_i}) + n};$$

$$d_h^{A_1, \dots, A_n} = \frac{\sum_{i=1}^n (d_p^{A_i} / u_p^{A_i})}{\sum_{i=1}^n (b_p^{A_i} / u_p^{A_i}) + \sum_{i=1}^n (d_p^{A_i} / u_p^{A_i}) + n};$$

$$u_h^{A_1, \dots, A_n} = \frac{n}{\sum_{i=1}^n (b_p^{A_i} / u_p^{A_i}) + \sum_{i=1}^n (d_p^{A_i} / u_p^{A_i}) + n}.$$

Тоді $\omega_x^{A_1, \dots, A_n}$ називається залежним консенсусом між усіма n думками. Для операції використовується символ $\oplus$, тобто $\omega_x^{A_1, \dots, A_n} = \omega_x^{A_1} \oplus \dots \oplus \omega_x^{A_n}$.

Операція $\oplus$ є і комутативною, і асоціативною, отже порядок комбінування думок не важливий. Думки, що не містять невизначеності не можуть бути скомбіновані.

3.6 Консенсус між частково залежними думками

Якщо два експерти A і B спостерігають ті самі події протягом двох частково непересічних проміжків часу, то їх відповідні думки будуть частково залежними. Консенсус між частково залежними думками експертів A і B можна записати у вигляді

$$1. \omega_x^{Ai(B)} : \begin{cases} b_x^{Ai(B)} = b_x^A \mu_x^{AB}; \\ d_x^{Ai(B)} = d_x^A \mu_x^{AB}; \\ u_x^{Ai(B)} = u_x^A \mu_x^{AB} / (1 - \gamma_x^{AB}) \end{cases} \quad \text{де } \mu_x^{AB} = \frac{1 - \gamma_x^{AB}}{(1 - \gamma_x^{AB})(b_x^A + d_x^A) + u_x^A}.$$

Для позначення консенсусу між частково залежними думками використовується символ $\tilde{\oplus}$. Консенсус між думками експертів A і B може бути записаний у вигляді

$$\omega_x^A \tilde{\oplus} \omega_x^B = \omega_x^{A.B} = \omega_x^{\overline{Ad(B).BdA.Ai(B).Bi(A)}} = (\omega_x^{Ad(B)} \tilde{\oplus} \omega_x^{Bd(A)}) \oplus \omega_x^{Ai(B)} \oplus \omega_x^{Bi(A)}.$$

Для будь-якої думки ω_x^A з коефіцієнтом залежності γ_x^{AB} з будь-якою іншою думкою ω_x^B справедливе співвідношення

$$\omega_x^A = \omega_x^{Ai(B)} \oplus \omega_x^{Ad(B)}.$$

3.7 Рекомендація

Припустимо, існують два експерти A і B , причому експерт A має деяку думку про експерта B , а експерт B має думку щодо твердження x .

Рекомендація цих двох думок полягає в комбінуванні думки експерта A відносно B з думкою експерта B відносно твердження x .

Не існує такої речі як рекомендація в чистому виді. Рекомендація думок у загальному випадку піддається різним інтерпретаціям. Основними труднощами є опис результату невір'я експерта A в те, що експерт B буде давати добру пораду. Що насправді це означає? Існує три різних інтерпретації.

1. Схильність до невизначеності. Недовіра експерта A в рекомендації експерта B полягає в тому, що A вважає, що B має невизначеність (не знання) щодо широго значення x . У результаті A також невизначений щодо широго значення x .

2. Схильність до обману. Недовіра експерта A в рекомендації експерта B спирається на те, що A думає про те, що B послідовно намагається (пробує) зменшити інформованість A через злий намір. У результаті експерт A не вірить експерту B .

3. Схильність до припущення про погані поради. Невір'я експерта A в рекомендації експерта B означає, що A вважає, що B послідовно дає ради зворотні від його реальної думки щодо широго значення x . У результаті експерт A не тільки не вірить у x зі ступенем, з яким B рекомендує вірити, але також і вірить у x зі ступенем, з яким B рекомендує не вірити в x . Таким чином, комбінація двох результатів приводить до довіри.

Теорема 8 Нехай A і B два експерти, а $\omega_B^A = \{b_B^A, d_B^A, u_B^A\}$ думка експерта A щодо рекомендацій експерта B . Нехай x є твердження $\omega_x^B = \{b_x^B, d_x^B, u_x^B\}$ і думка експерта B відносно твердження x , виражене у формі рекомендації для експерта A . Нехай $\omega_x^{AB} = \{b_x^{AB}, d_x^{AB}, u_x^{AB}\}$ буде думка така, що $b_x^{AB} = b_B^A b_x^B$; $d_x^{AB} = b_B^A d_x^B$; $u_x^{AB} = d_B^A + u_B^A + b_B^A u_x^B$.

Тоді ω_x^{AB} називається думкою що виражає, думку експерта A щодо твердження x яка сформована в результаті рекомендацій, отриманих від B . Для позначення цієї операції буде-мо використовувється символ $\otimes$, тобто

$$\omega_x^{AB} = \omega_B^A \otimes \omega_x^B.$$

Операція $\otimes$ є асоціативною, але не комутативною. Отже, комбінування думок не може починатися з якого-небудь кінця ланцюга, порядок, у якому комбінуються думки, важливий. У ланцюгу, що містить більш одного рекомендуемого об'єкта, думки передбачаються незалежними. Це значить, що в ланцюжку не може бути об'єкта, що дає більш ніж одну рекомендацію.

Рекомендація B повинна інтерпретуватися як те, що експерт B рекомендує експерту A , але не обов'язково відбиває реальну думку U відносно твердження x .

Правило рекомендації може бути виправданим тільки в припущенні того, що рекомендація транзитивна, тобто експерти в ланцюзі не можуть змінювати свого поведіння (тобто того, що вони рекомендують) у залежності від того, які об'єкти з ними взаємодіють. Однак це не завжди вірно, оскільки свідомо невірні рекомендації можуть бути мотивовані, наприклад, протиріччями (конфліктами) між експертами. Правило рекомендації необхідно застосовувати з обережністю і тільки тоді, коли передбачається інваріантне поведіння експертів.

Список літератури: 1. *ISO/IEC 17779:2000 Code of practice for information security management* 2. *ISO/IEC 21827: 2002 Information technology – Systems Security Engineering – Capability Maturity Model 3. NIST SP 800-26, Security Self-Assessment Guide for Information Technology Systems.* 4. *Managing information security risks. Operationally Critical Threat, Asset and Vulnerability.* Christopher Alberts and Audrey Dorofee, James Stevens, August, 2003 5. A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001. 6. A. Jøsang., S.J. Knapskog. A Metric for Trusted Systems. In Reinhard Posh, editor, *Proceedings of the 15th IFIP/SEC International Information Security Conference*. IFIP, 1998 7. A. Jøsang An Algebra for Assessing Trust in Certification Chains. In J. Kochmar, editor, *Proceedings of the Network and Distributed Systems Security Symposium (NDSS'99)*. The Internet Society, 1999. 8. A. Jøsang. The Consensus Operator for Combining Beliefs, Distributed System Technology Centre, email: ajosang@dstc.edu.au 9. A. Jøsang. Prospectives for Modelling Trust in Information Security, Distributed System Technology Centre, email: ajosang@dstc.edu.au. 10. *Ленишин А.В.* Применение аппарата субъективной логики для оценки безопасности банковских ИТ-систем // Актуальні проблеми та перспективи розвитку фінансово-кредитної системи України: Збірник наукових статей. Харків: Фінарт, 2002. С.410-412. 11. *Ленишин А.В., Потий А.В.* Применение оператора попарной усредненной конъюнкции для оценки уровня защищенности ИТ-систем // Збірник наукових статей за матеріалами VI міжнародної науково-практичної конференції “Безпека інформації в інформаційно-телекомунікаційних системах” Київ, 2003. С. 57-58. 12. *Потий А.В., Ленишин А.В.* Оценка защищенности информационно-телекоммуникационных систем с использованием математического аппарата субъективной логики // 7-я Научно-практическая конференция «Безопасность информации в информационно-телекоммуникационных системах», Киев, 2004. 13. *Потий А.В., Ленишин А.В.* Принципы построения системы экспертной оценки защищенности ИТ-систем «Советник» // Збірник наукових статей за матеріалами VI міжнародної науково-практичної конференції “Безпека інформації в інформаційно-телекомунікаційних системах”. Київ, 2003. С. 25-26. 14. *Потий О.В., Ленишин А.В.* Практичні рекомендації по використанню системи “Радник” при оцінці рівня організаційного захисту інформації в ІТС // 7-я Научно-практическая конференция «Безопасность информации в информационно-телекоммуникационных системах», Киев, 2004 р. 15. A.Hunter. *Uncertainty in information systems* McGraw-Hill, London, 1996. 16. A. Motro and Ph. Smets. *Uncertainty management in information systems: from needs to solutions.* Kluwer, Boston, 1997. 17. Shafer G. *A Mathematical Theory of Evidence.* Princeton University Press, 1976. 18. Ph. Smets, R. Kennis The transferable belief model. *Artificial Intelligence*, 66:191-1-234, 1994. 19. G. de Cooman, D. Ayles. *Supremum preserving upper probabilities.* Information Sciences, 1998 20. Daniel Ellsberg, Risk, ambiguity, and the Savage axioms. *Quarterly Journal of Economics*.

Харьковский национальный
университет радиоэлектроники

Поступила в редакцию 11.09.2004

TWO-TRACK-MAC АЛГОРИТМ ВЫСОКОЙ СЕКРЕТНОСТИ

Обеспечение информационной безопасности возможно на основе решения взаимосвязанной совокупности задач защиты информации. Качество решения задач защиты в значительной мере определяется используемыми криптографическими алгоритмами шифрования, цифровой подписи, хеширования и формирования кодов аутентификации.

Коды аутентификации сообщений (MAC коды), известные также как коды подлинности сообщений, являются криптографическими примитивами, которые используются для обеспечения целостности и аутентичности данных. Можно выделить три основных подхода к построению кодов аутентификации сообщения:

- 1) построенные с применением блочных шифров;
- 2) построенные на основе без ключевых хэш-функций;
- 3) построенные с использованием семейства универсальных хэш-функций.

Стандарт ISO/IEC 9797-1 [1] определяет алгоритмы для кодов подлинности сообщений, использующих блочный шифр в режиме сцепления по шифр тексту, EMAC и RMAC. Во вторую часть ISO/IEC 9797-2 [1] включены алгоритмы HMAC и MD_x-MAC кодов, построенные на основе без ключевых хэш-функций. Для изучения основных характеристик MAC кодов и их сравнительной оценки в научно-исследовательском проекте NESSIE (IST-1999-12324) были рассмотрены основные практические алгоритмы кодов подлинности сообщений и отобраны четыре лучших:

- **Two-Track-MAC**: K.U.Leuven, Бельгия и debis AG, Германия;
- **UMAC**: Intel корпорации, США, Университета из штата Невада в Рено, США, Научно-исследовательской лаборатории IBM, США, Technion, Израиль и Университета из Калифорнии в Девисе, США;
- **CBC-MAC (ISO/IEC 9797-1)**;
- **HMAC (ISO/IEC 9797-1)**.

TTMAC (Two-Track-MAC) имеет самый высокий уровень защиты для MAC примитивов, представленных NESSIE, и по итоговым показателям поставлен на первое место среди отобранных лучших примитивов.

Задачей данной статьи является анализ основных характеристик Two-Track-MAC кодов, изложение алгоритма построения и оценка секретности. С этой целью в разделе 1 приводятся требования к кодам аутентификации сообщений. В разд. 2 рассматривается алгоритм построения TTMAC с использованием хэш-функции RIPEMD-160. В разд. 3 оценена секретность TTMAC-кодов.

1 Определения и требования безопасности к кодам аутентификации сообщений

Основными характеристиками MAC алгоритмов, по которым выполняется их сравнительная оценка в соответствии с рекомендациями проекта NESSIE [NES] являются:

- уровень защищенности MAC кодов от общих атак;
- быстродействие MAC алгоритмов;
- статистические свойства распределений MAC кодов.

В представлении Принеля [2], MAC код есть функция отображения $h: K \times M \rightarrow R$, где пространство ключей $K = \{0,1\}^k$, пространство сообщений $M = \{0,1\}^*$ и пространство MAC значений $R = \{0,1\}^n$ для $k, n \geq 1$. Для заданных значений ключа $K \in K$ и сообщения $X \in M$ функция производит MAC значение $Y \in R$.

Противник подделает сообщение для MAC кода h , если, не зная случайный ключ K , он способен произвести новое сообщение X и MAC значение Y такое, что $h(K, X) = Y$.

Определение 1. MAC код $h: K \times M \rightarrow R$ является $(t; \epsilon; q)$ -секретным, если, при случайно взятом ключе K , противник не может подделывать новое сообщение за время t с вероятностью лучше, чем ϵ , даже если ему представлены значения q MAC кодов других сообщений по его выбору.

Большинство MAC кодов – это итеративные конструкции, которые используют функцию сжатия f , предварительное разбиение данных X на подблоки X_i и связку по обратному входу промежуточных результатов вычислений хеш значений.

Обобщенная модель итеративного MAC кода для t подблоков определяется следующим алгоритмом итеративных вычислений:

$$\begin{aligned} H_0 &= IV_K, \\ H_i &= f_K(H_{i-1}, X_i), \quad 1 \leq i \leq t, \\ h(K, X) &= g_K(H_t). \end{aligned}$$

Секретный ключ K может использоваться в векторе инициализации IV , в функции сжатия f и в выходном преобразовании g .

В зависимости от информации, доступной противнику, различают следующие типы атак на коды подлинности сообщений:

Атака с известным текстом. Злоумышленник имеет возможность исследовать некоторые открытые тексты и их соответствующие значения кода подлинности сообщений.

Атака с выбранным текстом. Нарушитель имеет возможность выбирать наборы текстов и впоследствии получает список значений кода подлинности сообщений, соответствующих этим текстам.

Атака с адаптивным выбором текста. Это является наиболее общей атакой, когда злоумышленник выбирает текст и немедленно получает соответствующие значения кода подлинности сообщений. Выбор текста может зависеть от результата предыдущих попыток.

"Взлом" алгоритма MAC кода может иметь различные последствия:

Экзистенциальная подделка. Нарушитель может определить значение кода подлинности сообщений, по крайней мере, для одного текста. Поскольку он не имеет контроля над текстом, он это может сделать без смысла и случайно.

Избирательная подделка. Злоумышленник может определить значение MAC кода для специфического текста, выбранного априорно. Практические атаки часто требуют, чтобы подделка была поддающейся проверке. Это означает, что нарушитель знает, что подделанный код подлинности сообщений правилен с вероятностью близкой к единице.

Ключевое восстановление. Нарушитель может определить секретный ключ K . Такой взлом более опасный, чем подделка, так как это позволяет создавать произвольные избирательные подделки.

Как правило, адаптивная атака с выбором текста на практике трудно выполнима. Тем не менее, следует требовать, чтобы алгоритмы кода подлинности сообщений были устойчивыми к самым сильным из возможных атак.

Лучшие известные атаки на коды аутентификации сообщения рассмотрены в [2].

Угадывание кода подлинности сообщений (Guessing of the MAC).

Прямая атака на алгоритм MAC код состоит в выборе произвольного нового сообщения, и впоследствии угадывании значения кода подлинности сообщений. Это может быть сделано двумя способами: или угадывание MAC кода непосредственно с вероятностью успеха 2^{-n} , или угадывание ключа и затем вычисление значения MAC кода с вероятностью успеха 2^{-k} . Здесь n обозначает размер в битах значения MAC кода и k размер в битах секретного ключа. Это – атака, не поддающаяся проверке: нарушитель не знает априорно, было ли его угадывание правильным. Выполнимость атаки зависит от числа испытаний, которые могут быть выполнены, и ожидаемого значения успешной атаки.

Исчерпывающий поиск ключа (Exhaustive Key Search).

Это – другая прямая атака, которая может применяться к любому алгоритму. Атака требует приблизительно k/n известных пар текст – MAC для данного ключа; пытаюсь определить ключ, пробуя один за другим все ключи. Ожидаемое число испытаний равно 2^{k-1} . В отличие от предыдущей атаки, эту атаку выполняют вне сеанса связи (off-line), и это приводит к взлому MAC алгоритма.

Подделка, основанная на внутренней коллизии (Internal Collision Based Forgery)

Последствие этой атаки состоит в том, что, если обнаружить внутреннюю коллизию (совпадение промежуточных результатов при вычислении значений MAC кодов), её можно использовать для создания подделки MAC кода на отдельном выбранном тексте.

Preneel и van Oorschot [3] показали, что внутреннюю коллизию для h можно найти, используя u известных пар текст – MAC и v выбранных текстов, где ожидаемые значения для u и v являются следующими (здесь l обозначает размер в битах связанной переменной): $u = 2^{(l+1)/2}$ и $v = 0$, если выход преобразования MAC кода является перестановкой; иначе v равно приблизительно $2[2^{l-n} + \lfloor (1-n)/(n-1) \rfloor + 1]$.

Дальнейшие оптимизации этой атаки возможны, если множество пар текст – MAC имеет общую последовательность из s последних блоков.

Ключевое восстановление на основе внутренней коллизии (Internal Collision Based Key Recovery)

Для некоторых функций сжатия, можно расширить внутреннюю атаку коллизии к ключевой атаке восстановления [4]. Идея состоит в том, чтобы идентифицировать одну или более внутренних коллизий; например, если f не является перестановкой для фиксированного H_0 , и внутренняя коллизия после первого блока сообщения определяется уравнением $f_K(H_0, X_i) = f_K(H_0, X_i')$, в котором K и, возможно, H_0 являются неизвестными (предполагается, что $H_0 = IV$ зависимы от ключа). Для некоторых функций сжатия f можно получить информацию относительно секретного ключа, основанного на таких отношениях.

Атака методом декомпозиции (Divide-and-Conquer Attack).

Эта атака является специальным случаем ключевого восстановления на основе внутренней коллизии. Для некоторых функций сжатия, которые используют два отдельных ключа, можно использовать внутренние коллизии для ключевого восстановления [3]. Пусть ключи обозначены K_1, K_2 . Общая идея состоит в том, что нарушитель сначала ищет некоторые внутренние коллизии, затем методом исчерпывания определяет ключ K_1 , который производит эти коллизии. Когда K_1 определен, исчерпывающий поиск используется, чтобы найти K_2 . Поэтому стойкость такого MAC кода зависит от его индивидуальных ключей, а не от их объединенной длины. Эта атака менее практична, чем простой исчерпывающий поиск ключа, поскольку требуется большое количество известных пар текст – MAC.

Подделка Exor (Exor Forgery)

Этот тип подделки возможен, если значение H_i вычисляется как функция $H_{i-1} \oplus X_i$ и если нет никакого последующего преобразования хеш выхода. Самый легкий вариант атаки требует только единственной известной пары текст – MAC. Предположим, что вход X и его дополняемая версия X' состоят из единственного блока. Если известно $h(K, X)$, немедленно следует, что

$$h(K, X \parallel (X \oplus h(K, X))) = h(K, X).$$

Таким образом можно создать новое сообщение с тем же значением MAC, которое является подделкой. Базовая CBC-конструкция использует вычисление $H_i \oplus X_i$ и оказывается уязвима к атаке exor подделки.

2 Two-Track-MAC: описание алгоритма

Two-Track-MAC (также известный как ТТМАС) – код аутентификации сообщения, разработанный Бертом Воег и Бартом Ваном Ромпейом [5] и представленный проекту NESSIE. Алгоритм работает на блоках 512 бит, разделенных на слова по 32 бита, использует секретный ключ 160 бит и производит выход до 160 бит.

Проект Two-Track-MAC основан на хэш-функции RIPEMD-160 с модификациями. Сначала, сообщение, которое будет заверено дополняется 1 битной частью и затем 0-битами, пока его длина не станет равной 448 по модулю 512. Двоичное представление длины первоначального сообщения ($\text{mod } 2^{64}$) добавляется в конец, так что длина сообщения становится кратной 512. Каждый блок с 512 битами разбивается на шестнадцать слов по 32 бита $W_0, \dots, W_{15}$. Секретный ключ – это множество из пяти слов с 32 битами $K_0, \dots, K_4$. Алгоритм работает путем итерационного использования функции сжатия следующим образом.

Два множества из пяти слов с 32 битами $X_0, \dots, X_4$ и $Y_0, \dots, Y_4$ и множество из 16 слов сообщения являются входными к двум различным функциям f_L и f_R , которые производят пять 32 битовых слов каждый:

$$A_0, \dots, A_4 = f_L(X_0, \dots, X_4; W_0, \dots, W_{15}), B_0, \dots, B_4 = f_R(Y_0, \dots, Y_4; W_0, \dots, W_{15}).$$

Эти две функции идентичны используемым в RIPEMD-160 (детали даются ниже). Затем вычисляются два новых множества из пяти слов каждый, вычитая слова входа от результатов предыдущего шага:

$$C_i = A_i - X_i \text{ mod } 2^{32}; 0 \leq i \leq 4, D_i = B_i - Y_i \text{ mod } 2^{32}; 0 \leq i \leq 4.$$

Далее, десять слов C_i и D_i перемешиваются в двух линейных преобразованиях g_L и g_R :

$$E_0, \dots, E_4 = g_L(C_0, \dots, C_4; D_0, \dots, D_4), F_0, \dots, F_4 = g_R(C_0, \dots, C_4; D_0, \dots, D_4).$$

E_i и F_i вместе со следующим множеством слов сообщения формируют вход к следующему применению функции сжатия. На первом шаге итераций пять слов секрета ключа K_i являются входом и определяют значения переменных X_i и Y_i , $0 \leq i \leq 4$.

На последней итерации, функции f_L и f_R меняются местами. После вычитания входных слов $C_i = A_i - X_i$ и $D_i = B_i - Y_i$, вместо выполнения линейных преобразований g_L и g_R вычисляются:

$$E_i = C_i - D_i \text{ mod } 2^{32}; 0 \leq i \leq 4.$$

Последнее вычисление E_i , $0 \leq i \leq 4$ формирует выход Two-Track-MAC.

Необязательное преобразование выхода определено для вычисления более коротких значений MAC кода.

Функции f_L и f_R

Функции f_L и f_R , называемые левым и правым следом функции сжатия, являются идентичными, используемыми в функции сжатия RIPEMD-160. Они состоят из 80 последовательных шагов. Константы и функции определяются следующим образом.

Дополнительные константы:

$$\begin{aligned} k_i &= 00000000_x, k'_i = 50a28be6_x, 0 \leq i \leq 15, k_i = 5a827999_x, k'_i = 5c4dd124_x, 16 \leq i \leq 31, \\ k_i &= 6ed9eba1_x, k'_i = 6d703ef3_x, 32 \leq i \leq 47, k_i = 8f1bbcdc_x, k'_i = 7a6d76e9_x, 48 \leq i \leq 63, \\ k_i &= a953fd4e_x, k'_i = 00000000_x, 64 \leq i \leq 79. \end{aligned}$$

Нелинейные функции битового уровня:

$$\begin{aligned} f_i(x, y, z) &= x \oplus y \oplus z, 0 \leq i \leq 15, f_i(x, y, z) = (x \& y) \vee (\bar{x} \& z), 16 \leq i \leq 31, \\ f_i(x, y, z) &= (x \mid \bar{y}) \oplus z, 32 \leq i \leq 47, f_i(x, y, z) = (x \& z) \mid (y \& \bar{z}), 48 \leq i \leq 63, \\ f_i(x, y, z) &= x \oplus (y \mid \bar{z}), 64 \leq i \leq 79. \end{aligned}$$

Выбор слова сообщения:

$r[i] = i, 0 \leq i \leq 15$
 $r[i] = 7; 4; 13; 1; 10; 6; 15; 3; 12; 0; 9; 5; 2; 14; 11; 8; 16 \leq i \leq 31$
 $r[i] = 3; 10; 14; 4; 9; 15; 8; 1; 2; 7; 0; 6; 13; 11; 5; 12; 32 \leq i \leq 47$
 $r[i] = 1; 9; 11; 10; 0; 8; 12; 4; 13; 3; 7; 15; 14; 5; 6; 2; 48 \leq i \leq 63$
 $r[i] = 4; 0; 5; 9; 7; 12; 2; 10; 14; 1; 3; 8; 11; 6; 15; 13; 64 \leq i \leq 79$
 $r'[i] = 5; 14; 7; 0; 9; 2; 11; 4; 13; 6; 15; 8; 1; 10; 3; 12; 0 \leq i \leq 15$
 $r'[i] = 6; 11; 3; 7; 0; 13; 5; 10; 14; 15; 8; 12; 4; 9; 1; 2; 16 \leq i \leq 31$
 $r'[i] = 15; 5; 1; 3; 7; 14; 6; 9; 11; 8; 12; 2; 10; 0; 4; 13; 32 \leq i \leq 47$
 $r'[i] = 8; 6; 4; 1; 3; 11; 15; 0; 5; 12; 2; 13; 9; 7; 10; 14; 48 \leq i \leq 63$
 $r'[i] = 12; 15; 10; 4; 1; 5; 8; 7; 6; 2; 13; 14; 0; 3; 9; 11; 64 \leq i \leq 79$

Вращение констант:

$s[i] = 11; 14; 15; 12; 5; 8; 7; 9; 11; 13; 14; 15; 6; 7; 9; 8; 0 \leq i \leq 15$
 $s[i] = 7; 6; 8; 13; 11; 9; 7; 15; 7; 12; 15; 9; 11; 7; 13; 12; 16 \leq i \leq 31$
 $s[i] = 11; 13; 6; 7; 14; 9; 13; 15; 14; 8; 13; 6; 5; 12; 7; 5; 32 \leq i \leq 47$
 $s[i] = 11; 12; 14; 15; 14; 15; 9; 8; 9; 14; 5; 6; 8; 6; 5; 12; 48 \leq i \leq 63$
 $s[i] = 9; 15; 5; 11; 6; 8; 13; 12; 5; 12; 13; 14; 11; 8; 5; 6; 64 \leq i \leq 79$
 $s'[i] = 8; 9; 9; 11; 13; 15; 15; 5; 7; 7; 8; 11; 14; 14; 12; 6; 0 \leq i \leq 15$
 $s'[i] = 9; 13; 15; 7; 12; 8; 9; 11; 7; 7; 12; 7; 6; 15; 13; 11; 16 \leq i \leq 31$
 $s'[i] = 9; 7; 15; 11; 8; 6; 6; 14; 12; 13; 5; 14; 13; 13; 7; 5; 32 \leq i \leq 47$
 $s'[i] = 15; 5; 8; 11; 14; 14; 6; 14; 6; 9; 12; 9; 12; 5; 15; 8; 48 \leq i \leq 63$
 $s'[i] = 8; 5; 12; 9; 12; 5; 14; 6; 8; 13; 6; 5; 15; 13; 11; 11; 64 \leq i \leq 79$

Предположим, что заданы начальные значения $a_0; b_0; c_0; d_0; e_0$, функция f_L (левый след функции сжатия) состоит из следующих шагов для $0 \leq i \leq 79$ (сложение по $\text{mod } 2^{32}$):

$a_{i+1} = e_i,$
 $b_{i+1} = (a_i + f_i(b_i, c_i, d_i) + W_{f_L i} + k_i)^{<<s[i]} + e_i,$
 $c_{i+1} = b_i,$
 $d_{i+1} = c_i^{<<10},$
 $e_{i+1} = .$

Точно так же, когда заданы начальные значения $a_0; b_0; c_0; d_0; e_0$, функция f_R (правый след функции сжатия) состоит из следующих шагов для $0 \leq i \leq 79$ (сложение по $\text{mod } 2^{32}$):

$a_{i+1} = e_i,$
 $b_{i+1} = (a_i + f_{79-i}(b_i, c_i, d_i) + W_{f_R i} + k_i)^{<<s'[i]} + e_i,$
 $c_{i+1} = b_i,$
 $d_{i+1} = c_i^{<<10},$
 $e_{i+1} = d_i.$

Функции g_L и g_R

Линейные преобразования g_L и g_R используются, чтобы перемешать результаты двух следов функции сжатия. Для входов $C_0, \dots, C_4$ и $D_0, \dots, D_4$ функция g_L вычисляет пять слов $E_0, \dots, E_4$ следующим образом (операции по $\text{mod } 2^{32}$):

$$E_0 = (C_1 + C_4) - D_3; \quad E_1 = C_2 - D_4; \quad E_2 = C_3 - D_0; \quad E_3 = C_4 - D_1; \quad E_4 = C_0 - D_2;$$

Для входов $C_0, \dots, C_4$ и $D_0, \dots, D_4$, функция g_R вычисляет пять слов $F_0, \dots, F_4$ следующим образом (операции по $\text{mod } 2^{32}$):

$$F_0 = C_3 - D_4; \quad F_1 = (C_4 + C_2) - D_0; \quad F_2 = C_0 - D_1; \quad F_3 = C_1 - D_2; \quad F_4 = C_2 - D_3;$$

В табл. 1 представлены основные результаты по оценке быстродействия Two-Track-MAC алгоритма для различных операционных платформ [6]. Скорость вычислений определяется количеством циклов процессора, затрачиваемых на один байт обрабатываемого сообщения.

Таблица 1

Алгоритм	Длина MAC кода (бит)	Длина ключа (бит)	Тип ПЭВМ				
			Pentium 2	РПМ/Linux	Pentium 4	Xeon	AMD
Ttmac	160	160	21	21	40	37	21
Umac-16	64	128	6.1	6.0	6.2	6.1	6.2
Umac-32	64	128	2.5	2.9	6.7	6.6	1.9
HMAC-Whirlpool	512	512	86	72	98	103	100
HMAC-MD4	128	512	4.7	4.7	6.4	6.4	4.7
HMAC-MD5	128	512	7.2	7.3	9.4	9.4	7.4
HMAC-RIPE-MD	160	512	23	18	27	26	21
HMAC-SHA-0	160	512	16	15	23	23	13
HMAC-SHA-1	160	512	16	15	25	24	12
HMAC-SHA-2	256	512	40	39	40	39	33
	384		84	84	124	132	72
	512		84	84	124	132	72
HMAC-Tiger	192	512	24	21	28	26	20
CBCMAC-Rijndael	128	128	24	26	26	27	31
CBCMAC-DES	64	56	62	61	72	69	54
CBCMAC-Shacal	512	160	31	31	67	74	29

Как следует из анализа табл. 1, Two-Track-MAC уступает по быстродействию только Umac алгоритму, приблизительно в 3-7 раз, но имеет в 2,5 раза больше размер MAC кода.

3 Анализ безопасности Two-Track-MAC алгоритма

Безопасность Two-Track-MAC может быть доказана с условием, что основная функция сжатия псевдослучайная. Эта функция подобна функции сжатия, используемой в RIPEMD-160, которая является хорошо изученным примитивом, для которого не были найдены никакие слабости в доказательстве секретности [7]. Функции f_L и f_R состоят из 80 итеративных шагов, и используют различные операции с битами И, ИЛИ, XOR и битовые преобразования. Шаговые функции включают вращения бит и дополнения по mod 2^{32} , и в течение 80 итераций, оказывается достаточно трудным проследить неизвестные ключевые биты. Никакие слабости в Two-Track-MAC не были обнаружены.

Стоит отметить, что ключевой материал используется только один раз при первой итерации алгоритма. Результат MAC кода в 160 бит является разностью между двумя 160 битовыми значениями, так что знание этой разности дает еще 160 бит неопределенности, относительно двух значений которых их произвели. Другими словами, чтобы угадать эти два значения путем обратного вычисления, потребуется объем вычислений такой же, как при непосредственном угадывании ключа. Большой размер внутреннего состояния (320 бит) в Two-Track-MAC дает алгоритму высокий уровень защиты от атак, основанных на внутренних коллизиях. Если мы обозначим длину MAC результата m (значения для m поддерживаются алгоритмом между 32 и 160 бит), сложность основных атак на этот примитив следующая:

- приблизительно 2^{159} вычислений MAC кода и $160/m$ известных пар текст – MAC необходимы для исчерпывающего поиска ключа.

- угадывание значения MAC кода имеет вероятность успеха 2^{-m} .
- Атаки, основанные на внутренних коллизиях, требуют приблизительно 2^{160} известных пар текста – MAC и приблизительно 2^{320-m} выбранных текстов.

Таким образом, TTMAC (Two-Track-MAC) имеет самый высокий уровень защиты для MAC примитивов, представленных NESSIE, и по итоговым показателям является предпочтительным. Алгоритм TTMAC основан на хэш-функции RIPEMD-160 с небольшими модификациями [7]. Алгоритм имеет определенные преимущества в быстродействии, особенно в случае коротких сообщений, и оптимальную длину ключевых данных.

Список литературы: 1. *International Organization for Standardization. ISO/IEC 9797-2: Information Technology - Security Techniques - Message Authentication Codes (MACs). Part 2: Mechanisms using a dedicated hash-function.* 2002. P. 152, 153. 2. *Preneel B. Cryptographic primitives for information authentication | state of the art.* in *State of the Art in Applied Cryptography* (B. Preneel and V. Rijmen, eds.), no. 1528 in *Lecture Notes in Computer Science*, pp. 50{105, Springer-Verlag, 1998. 3. *Preneel B. and. Oorschot Van P. C. MDx-MAC and building fast MACs from hash functions.* in *Proceedings of Crypto'95* (D. Coppersmith, ed.), no. 963 in *Lecture Notes in Computer Science*, pp. 1{14, Springer-Verlag, 1995. P. 150. 152. 4. *B. Preneel and P. C. van Oorschot, "On the security of two MAC algorithms."* in *Proceedings of Eurocrypt'96* (U. Maurer, ed.), no. 1070 in *Lecture Notes in Computer Science* P 19{32. Springer-Verlag, 1996. P. 150 5. *B. Rompay Van and Den Boer B., TTMAC. Primitive submitted to NESSIE, Sept. 2000.* 6. *Performance of Optimized Implementations of the NESSIE Primitives. NESSIE. Performance Evaluation. V 2.0, Feb 20, 2003.* 7. *Preneel B., Bosselaers A., and Dobbertin H., The cryptographic hash function ripemd-160. Cryptobytes. V. 3, no. 2, P. 9{14. 1997. P. 157*

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 20.10.2004

ФУНКЦИОНАЛЬНОСТЬ И ХАРАКТЕРИСТИКИ КРИПТОБИБЛИОТЕКИ НА ЯЗЫКЕ JAVA

На данный момент в области разработки приложений язык Java занимает ведущие позиции. Отчасти это объясняется не только тем, что Java является удобным и простым языком программирования, но и самостоятельной успешной платформой. Такая платформа имеет как свои плюсы, обусловившие ее широкое распространение, так и характерные минусы, которые являются прямым последствием их реализации.

Java, как язык программирования для Web, занимает ведущее место как инструмент разработки безопасных веб-приложений. Java апплеты обеспечивают возможность доставки выполняемых приложений веб-браузерам. JSP и сервлеты обеспечивают возможность безопасной обработки запросов браузера. Технология JDBC обеспечивает безопасный доступ Web приложений к корпоративным базам данных. Технология EJB обеспечивает основу для безопасной реализации бизнес-приложений. Технологии RMI и CORBA на платформе Java создают основу для развертывания высокоэффективных распределенных объектов. Как и любая успешная платформа имеет свои средства защиты и целостности информации. Интерфейс Java Security API и его расширения дают возможность реализации и выполнения полностью распределенных приложений безопасным путем.

На рис. 1 представлена архитектура системы безопасности Java 2. В саму платформу Java 2 входят лишь два компонента представленных в нижней части схемы. Это само ядро безопасности, куда входят загрузчик классов, процессор обработки исполняемой программы, верификатор байт кода, диспетчер безопасности, управление доступом и политиками безопасности [1]. Также в платформу Java 2 входит JCA – Архитектура шифрования Java, которая обеспечивает существование инфраструктуры

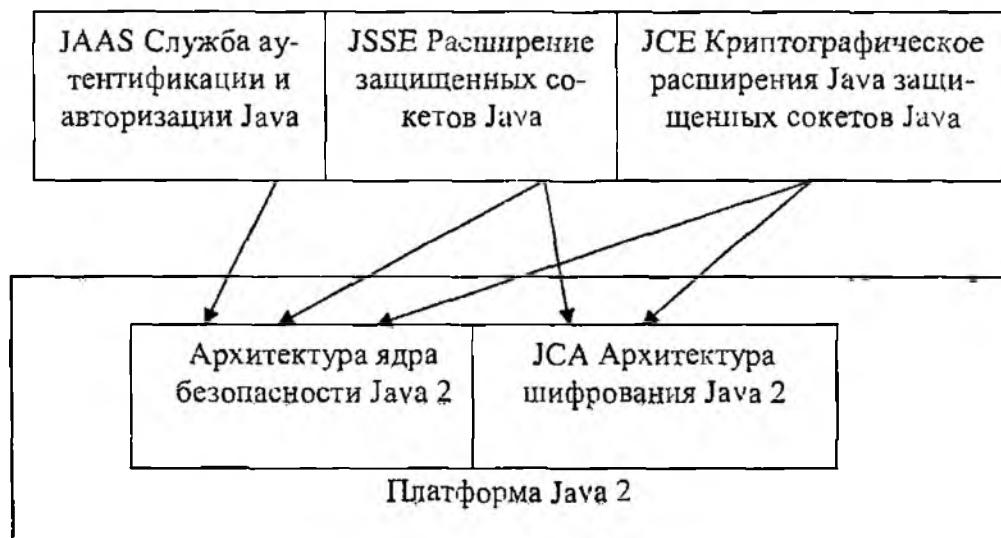


Рис. 1

для выполнения основных шифровальных функций. Сюда входит защита данных от повреждений с целью сохранения их целостности путем использования основных криптографических функций и алгоритмов, криптографические алгоритмы создания подписей, API интерфейсы для работы ключами и сертификатами.

В расширение (JCE – Java Cryptography Extention) Java 2 входят функции, используемые для шифрования блоков данных с целью обеспечить дополнительную защиту их конфиденциальности на время передачи, которые впоследствии расшифровываются получателем.

Данные функции были специально вынесены Sun за пределы платформы, из-за ограничений, которые наложило правительство США на экспорт технологий шифрования. К дополнительным компонентам платформы Java можно отнести JSSE – расширение защищенных сокетов, которое обеспечивает стандартный интерфейс вместе с базовой эталонной реализацией для создания приложений Java, использующих протокол защищенных сокетов SSL. В JSSE также входят протоколы TLS(Transport Layer Security) и WTLS(Wireless Transport Layer Security). Также одним из расширений платформы Java является JAAS – Служба аутентификации и авторизации Java, которая была разработана с целью обеспечить стандартный способ ограничения доступа к ресурсам на основе аутентификации личности пользователя.

Но использование стандартных средств в ряде случаев становится невозможным из-за проблем, связанных с требуемыми условиями гибкости систем их защищенности, или просто из-за невозможности использовать стандартные алгоритмы. Такая ситуация проявляется при проектировании и реализации программных систем в Украине, предназначенных для банковской или государственной сферы. В таких системах доминирующим фактором является надежность защиты и использование национальных стандартов защиты информации.

Построение библиотеки с использованием и интеграцией с системой безопасности Java было бы достаточно правильным. Конфигурируемая модель безопасности, заложенная в платформу Java 2 (рис.2), является тщательно проработанной и безопасной, в отличие от предшествующих моделей безопасности в Java 1.0 и Java 1.1. Для любого кода независимо от того, локальный он или удаленный, может быть назначена своя конфигурируемая политика безопасности, позволяющая коду использовать ресурсы только в определенных назначенных областях. Области доступа и конфигурируемые политики безопасности обеспечивают, значительно большую гибкость, чем Java 1.0 [1].

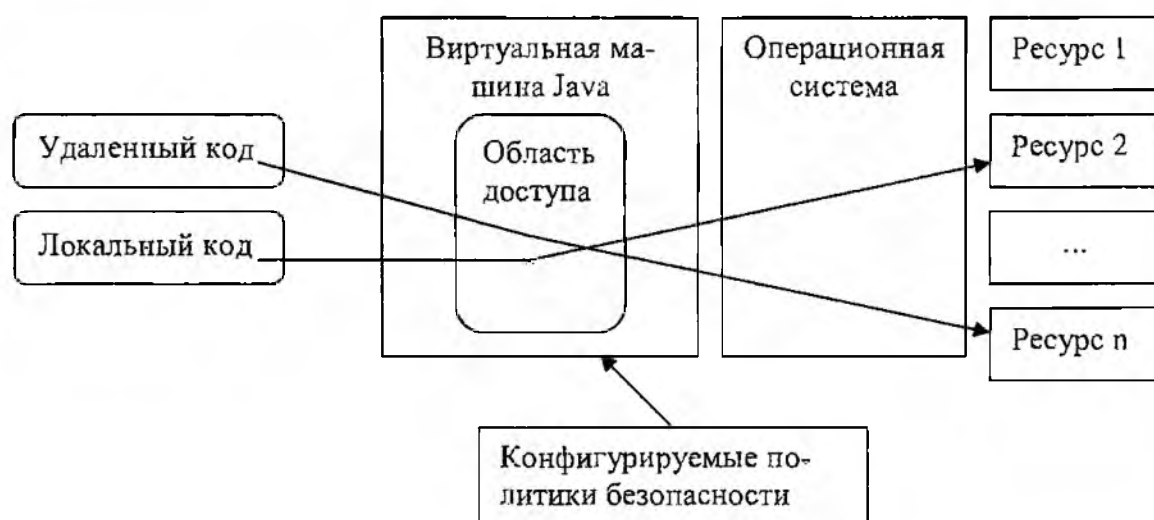


Рис. 2

Однако, несмотря на все преимущества конфигурируемой модели безопасности Java 2. существующее положение дел с виртуальными машинами в браузерах заставляют разработчиков приложений (апплетов), предназначенных для использования в Интернете, опираться на модели безопасности Java 1.1 либо заставлять пользователей дополнительно закачивать Java plug – in, что в большинстве случаев является неприемлемо. Дело в том, что по статистике в более чем 90% случаев используется браузеры Internet Explorer, в которых стандартно установлена виртуальная машина Microsoft Java, в которой поддерживается модель безопасности Java 1.1 (рис.3). Из представленных фактов следует еще одно достаточно строгое требование разработки Интернет приложений – использование JDK1.1.

Целью данной статьи является определение основных требований для создания библиотеки криптографических функций при использовании ее в государственных и частных орга-

низациях Украины, на основе опыта при создании библиотеки JShield (АО ИТ). На наш взгляд, выполнение этих требований при проектировании и создании приложения даст возможность создать адекватную библиотеку криптографических функций.

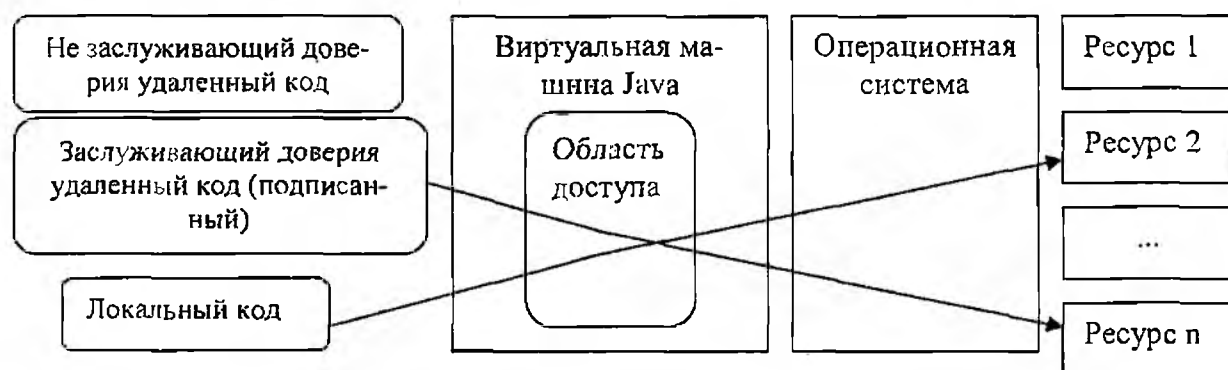


Рис. 3

Рассматривая направленность платформы Java, можно определить несколько основных характеристик для данного типа криптографической библиотеки:

- *Реализация национальных стандартов*
- *Сертификация библиотеки*
- *Скоростные показатели*
- *Размер библиотеки*
- *Соответствие интерфейсов библиотеки интерфейсам JCE*

Рассмотрим каждый пункт более подробно.

Реализация национальных стандартов

Данное требование является основополагающим, и представленные стандарты алгоритмов криптографического преобразования для систем обработки информации в сетях электронных вычислительных машин (ЭВМ) являются обязательными для организаций, предприятий и учреждений, применяющих криптографическую защиту данных, хранимых и передаваемых в сетях ЭВМ, в отдельных вычислительных комплексах или в ЭВМ.

Обязательными для реализации являются:

- **ГОСТ 28147-89** – определяет правила шифрования данных и выработки имитовставки. Алгоритм симметрического криптографического преобразования предназначен для аппаратной или программной реализации, удовлетворяет криптографическим требованиям и по своим возможностям не накладывает ограничений на степень секретности защищаемой информации [4].

- **ГОСТ 34311-95** – определяет алгоритм и процедуру вычисления хэш-функции для любых последовательностей двоичных символов, применяемых в криптографических методах обработки и защиты информации [3]. Данный алгоритм базируется на блочном алгоритме шифрования **ГОСТ 28147-89**.

- **ГОСТ 34310-95** – определяет правила выработки и проверки цифровой подписи [2].

- **ДСТУ 4145-2002** – определяет правила выработки и проверки цифровой подписи на основе эллиптических кривых [5]. Данный стандарт был принят относительно недавно и является достаточно перспективным исходя из преимуществ, предоставляемых данным алгоритмом по сравнению с **ГОСТ 34310-95**. Но одним из препятствий, которые препятствуют повсеместному внедрению, является отсутствие надежной и четкой инфраструктуры ключей, инфраструктуры взаимодействий алгоритма.

Сертификация библиотеки

Сертификация программного средства ДСТСЗИ СБУ и УкрСЕПРО дает гарантию пользователям гарантию правильности реализации:

- точное соответствие заявленным производителем стандартам алгоритмов и рекомендаций;
- отсутствие дополнительной функциональности, которая может, каким либо образом нарушить работоспособность алгоритма или нанести вред пользователю программного средства;

Скоростные показатели

Данный вопрос является краеугольным камнем для криптографических библиотек и не только, на основе платформы Java и является прямым следствием преимущества мультиплатформенности Java. На данный момент решения на основе Java в основном проигрывают по скорости выполнения, а в некоторых случаях существенно, аналогичным решениям написанным, например на C/C++. Поэтому вопрос производительности должен закладываться на самых ранних этапах разработки программных средств. По-возможности, еще на этапе анализа необходимо знать, какой уровень производительности должно обеспечивать разрабатываемое приложение.

Для оптимизации работы криптографических программных средств применяются те же правила, что и для обычных приложений на основе платформы Java. Кратко представим способы оптимизации, наиболее характерные для разработки криптографических приложений:

минимизирование операции ввода – вывода жесткими дисками и внешними носителями. Если есть возможность, необходимо предпочесть хранение данных в оперативной памяти, используя класс `ByteArrayStream`;

минимизирование вывода информации на монитор. По возможности, надо убрать все лишние и избыточные выводы данных на экран;

использование буферизированных потоков ввода – вывода. Для ввода необходимо использовать `BufferedReader` или `BufferedInputStream`, а для вывода – соответственно `BufferedWriter` или `BufferedOutputStream`;

правильное использование циклов. Сюда можно отнести несколько предложений.

1. Необходимо избегать внесения в тело цикла операторов, выполняющих общие действия, которые можно вынести за пределы цикла.

2. Если возможно – объединить несколько совместимых последовательных циклов в один цикл.

3. Везде, где возможно, необходимо избегать выполнения операций сравнения в цикле;

повторение использования одних и тех же вычислений. Необходимо стараться избегать повторных действий, выполняя вычисление многократно используемого значения только один раз, и заменить все повторные вычисления этого значения на выборку уже вычисленного значения;

выполнение проверок в порядке убывания вероятности их появления. Если выполняется проверка нескольких разных условий и вероятность выполнения одного из них существенно выше остальных, то первым необходимо выполнять проверку именно этого условия;

замена сложных вычислений выборкой готовых значений из таблицы;

использование динамического заполнения. Если используется достаточно большая таблица данных, которая вычисляется на начальном этапе, можно использовать постепенное ее заполнение по мере необходимости в данных;

поиск по массиву. Если достаточно часто используется обращение к одному из элементов массива, лучше записать значение этого элемента в переменную и использовать ее вместо постоянного обращения к массиву;

поиск по сложной структуре данных. Если часто приходится выполнять поиск по большой и сложной структуре данных более рациональным станет создание индекса для этой структуры;

Использование данных способов оптимизации позволит в некоторых случаях ощутимо повысить производительность. Но в то же время, как показал наш опыт, существуют операции,

для которых оптимизация не дает необходимого результата. Например, при использовании достаточно ресурсоемких операций несимметричного шифрования или распределения ключей. Использование RSA операций с размером ключа более 768 бит становится неприемлемо долгой, в отличие от использования библиотек написанных на С, и тем более при использовании аппаратных средств. Поэтому в этих случаях следует рассмотреть вариант написания особо ресурсоемких частей на других языках программирования. Платформа Java предоставляет возможность подключать к программе с помощью интерфейса JNI библиотеки с машинно-зависимыми программами. Однако распространение разработанного продукта с использованием JNI потребует написания машинно-зависимой версии библиотеки для каждой из платформ.

Размер библиотеки

Данная характеристика является важной при реализации систем, которые функционируют в Интернете. Примером может служить реализация приложений в виде апплетов, которые реализуют все функции защиты и целостности информации с использованием национальных стандартов. Для таких приложений очень важен размер самого приложения, особенно в наших условиях, где большинство пользователей используют низкоскоростные каналы связи. В таких условиях использование библиотек, не оптимизированных по размеру, загрузка приложения занимает много времени, и приложение становится непривлекательным с коммерческой точки зрения.

При разработке библиотеки Jshield одним из обязательных условий поставленных заказчиком являлось разработка оптимизированной по размеру версии библиотеки для использования в апплетах. Для такой версии библиотеки заказчиком был определен предпочтительный размер в 20 – 30 кб. На первый взгляд данное условие было достаточно сложным, если учесть, что стандартная версия библиотеки занимала приблизительно 100 кб.

Использование представленных ниже методов позволило оптимизировать размер библиотеки до 24 кб, для jar версии и 18 кб, для cab версии соответственно. Данные методы применялись последовательно.

Перепроектировка всех интерфейсов и классов с учетом оптимизации по размеру. Данный способ дал возможность уменьшить размер библиотеки до 71 кб (jar).

1. Избавление от избыточности в методах и удаление неиспользуемых переменных классов, методов, переменных (Использовались внутренние средства рефакторинга среды разработки IDEA). Данный способ дал возможность уменьшить размер библиотеки до 54 кб (jar).

2. Использование пакетов для оптимизации профайлеров. (Использовался пакет Optimizelt). Данный способ дал возможность уменьшить размер библиотеки до 48 кб (jar).

3. Компилирование с удалением отладочной информации (ключ – g:none).

Данный способ дал возможность уменьшить размер библиотеки до 35 кб (jar).

4. Использование обфускатора (Proguard). Данный способ дал возможность уменьшить размер библиотеки до 24 кб (jar).

Использование данной криптографической библиотеки в составе апплета сделало приемлемым время загрузки даже при использовании достаточно медленных каналов связи (например, 9600 кбит/с).

Соответствие интерфейсов библиотеки интерфейсам JCE

Эта характеристика важна для разработчиков программного обеспечения, позволяет значительно ускорить ознакомление и подключение библиотеки к программному продукту. При разработке собственных решений в области защиты информации, опирающихся на стандартную модель интерфейсов сервис-провайдеров JCE, разработчик имеет возможность достаточно легко встраивать их в существующие решения, которые поддерживают JCE.

Заключение

Была рассмотрена основная концепция системы безопасности Java. Рассмотрены основные принципы построения криптографической библиотеки на платформе Java. Проведена

систематизация основных требований и критериев библиотек такого типа. В ходе работы выполнена реализация криптографической библиотеки с учетом требований, представленных выше.

Список литературы: 1. Джим Яворски, Пол Дж. Перроун. Система безопасности Java. Руководство разработчика, 2001. 2. *ГОСТ 34.310-95*. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки ЭЦП на базе асимметричного криптографического алгоритма.; Введ. 21.10.1997. К.: Госстандарт Украины, 1998. 3. *ГОСТ 34.311-95*. Информационная технология. Криптографическая защита информации. Функция хеширования. Введ. 21.10. 1997. К.: Госстандарт Украины, 1998. 4. *ГОСТ 28.147-89*. Информационная технология. Криптографическая защита информации. Функция шифрования. К.: Госстандарт Украины. 1989. 5. *ДСТУ 4145-2002* „Інформаційні технології Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка”. К.: Держстандарт України, 2003.

*Харьковский национальный
университет радиозлектроники*

Поступила в редколлегию 10.12.2004

ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННЫХ СИСТЕМ СО ВСТРОЕННЫМ МЕХАНИЗМОМ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

В настоящее время вопрос обеспечения безопасности информации, циркулирующей в электронных средствах ее обработки, является одним из приоритетных направлений на рынке информационных технологий. В то же время, на сегодняшний день, практически все разрабатываемые средства обеспечения безопасности информации ориентированы на угрозы, реализуемые различными программными средствами либо использующие каналы ПЭ-МИН. Однако нельзя не учитывать, что сами аппаратные средства, посредством которых осуществляется обработка информации, могут реализовывать различные угрозы безопасности информации. Реализация этих угроз осуществляется посредством внедренных в структуру штатных электронных устройств аппаратных закладок (АЗ), использующих в качестве каналов утечки информации штатные каналы передачи информации и управления. В связи с этим остро стоит вопрос обеспечения безопасности информации в аппаратных средствах ее обработки.

Одним из направлений обеспечения безопасности информации в аппаратных средствах ее обработки является проектирование электронных систем со встроенным механизмом управления доступом. Данный подход направлен на обеспечение безопасности информации путем разграничения прав доступа различных модулей электронных систем к ресурсам друг друга. Дело в том, что при проектировании электронных систем разработчик имеет возможность заложить в разрабатываемое устройство, кроме функций, заявленных в технической документации, различные неспецифицированные функции, реализующие угрозу безопасности информации.

В качестве базовой модели для описания взаимодействия аппаратных ресурсов КС выбрана субъектно-объектная модель отношений [1,2] как наиболее полно отражающая суть процессов, протекающих в аппаратной системе. Несанкционированный доступ одного субъекта к другому может выражаться не только самим фактом доступа, но и способом доступа, а именно форматом команд и данных, передаваемых посредством штатных каналов передачи информации и управления. Таким образом, проектируемая система должна иметь возможность контролировать формат команд и данных, циркулирующих в системе при помощи некоторого монитора безопасности потоков (МБП), функцией которого является мониторинг потоков информации, циркулирующих в КС, с целью выявления и блокирования не специфицированных потоков. Предварительно введем некоторые определения.

Пара субъектов называется корректной относительно друг друга, если субъекты не могут изменять ассоциированные объекты друг друга.

Монитор безопасности обращений (МБО) – это субъект, который активизируется при возникновении факта обращения субъекта к объекту и разрешает поток, принадлежащий только множеству разрешенного доступа R .

Сформулируем критерии гарантированного выполнения политики безопасности в аппаратной системе.

Критерий 1. Необходимое условие выполнения политики безопасности в КС. В КС гарантированно осуществляется только разрешенное обращение компонент, если все существующие в системе субъекты корректны относительно друг друга и монитора безопасности обращений (МБО).

Однако сформулированный критерий имеет следующие ограничения:

1. Условие корректности всех субъектов системы является весьма жестким и труднореализуемым. Обеспечение этого условия в архитектуре КС приводит к аппаратной избыточности и резкой потере производительности системы. Более того, невозможно обеспечить корректность всех устройств КС относительно друг друга в уже спроектированной системе.

Поскольку само функционирование аппаратной части КС предполагает наличие потоков между ассоциированными объектами субъектов, и, следовательно, множества ассоциированных объектов субъектов КС будут пересекаться. Примером подобной ситуации может быть простая передача данных через Com Port персонального компьютера. При любой передаче данных ICH2 программирует внутренние регистры Super IO, устанавливая скорость, формат передачи и непосредственно обращаясь к регистрам приемо-передатчика. Таким образом, множества ассоциированных объектов ICH2 и Super IO пересекаются.

2. Не гарантируется блокировка инициализации АЗ злоумышленником S_0 , так как последний использует легализованные операции обращения субъекта к объекту, которые не блокируются МБО.

3. На этапах передачи информации третьему лицу АЗ использует легализованные операции обращения субъекта к объекту, которые не блокируются МБО.

Обобщая пункты 2 и 3, находим, что и в первом, и во втором случае характерным является создание неспецифицированных потоков. В связи с этим логично говорить не только и не столько о корректности субъектов, а о корректности потоков между взаимодействующими субъектами и объектами.

Как видим, механизм МБО гарантирует блокирование нелегальных операций инициализации АЗ в архитектуре КС, но при этом допускается существование некорректных потоков.

Введем определение корректности потоков. Поток называется корректным относительно взаимодействующей пары субъекта – объекта, если в любой момент времени его существование не приводит к недокументированному изменению собственных свойств взаимодействующих пар субъектов и объектов. Под недокументированным изменением собственных свойств субъектов понимается активация функций субъекта, не описанных (не специфицированных) в технической документации данного субъекта. При этом активация производится при помощи передачи неспецифицированных команд либо данных посредством штатных каналов передачи информации.

Введем определение монитора безопасности потоков. Монитор безопасности потоков (МБП) – субъект, который активизируется при порождении любого потока между аппаратными ресурсами КС и разрешает порождение только специфицированных потоков, принадлежащих множеству R для фиксированного подмножества пар активизирующих субъектов и порождающих объектов.

МБП функционирует таким образом, что при возникновении потока от ассоциированного объекта O_m любого субъекта S_i ($S_i(O_m)$) к объекту O_j и обратно он участвует в потоке от субъекта к объекту (в том смысле, что информация проходит через его ассоциированные объекты и существует тождественное отображение объекта на какой-либо ассоциированный объект МБП). Таким образом, при возникновении потока МБП получает и обрабатывает информацию, передаваемую данным потоком, посредством собственных функционально ассоциированных объектов. Под обработкой информации МБП понимается процесс идентификации информации, передаваемой в потоке, и принятия решения о ее допустимости для заданной пары субъект – объект.

Сформулируем определение изолированной аппаратной системы (ИАС), в которой в качестве критериев разрешения или запрещения доступа принимается не только факт осуществления доступа, но и информация, передаваемая посредством потока.

Множество аппаратных субъектов КС называется изолированным, если в системе действует МБП и все порождаемые потоки корректны относительно взаимодействующих пар субъектов – объектов и МБП.

Теперь возможно переформулировать необходимый критерий гарантированного выполнения политики безопасности следующим образом.

Критерий 2. Достаточное условие гарантированного выполнения политики безопасности в КС. Если в изолированной КС существует МБП и порождаемые субъекты корректны

относительно МБП, то в такой КС реализуется только разрешенный политикой безопасности доступ.

Легко видеть, что данное утверждение более конструктивно относительно предыдущего достаточного условия гарантированной защищенности, поскольку ранее требовалась корректность МБО относительно произвольного субъекта, что практически выполнить очень трудно. В данном же случае вместо механизма МБО предлагается применять МБП, который значительно усиливает защитные механизмы КС, а также накладывает менее жесткие требования на структурные свойства системы в целом.

Решение задачи проектирования электронных систем со встроенным механизмом управления доступом осуществляется путем выполнения следующих этапов при проектировании структуры защищаемой КС:

1. Аппаратная система представляется в виде конечного множества элементов.
2. Анализируется возможность применения АЗ различных типов в рассматриваемой аппаратной системе.
3. Осуществляется декомпозиция множества элементов системы на два подмножества: множество субъектов и множество объектов.
4. Выделяются пары взаимодействующих элементов системы.
5. Для каждой взаимодействующей пары субъектов и объектов определяется множество разрешенных с точки зрения политики безопасности потоков.
6. Для заданного множества элементов системы формируется матрица прав доступа, с учетом возможности применения типов АЗ, определенных на этапе 2.
7. В системе выделяется активный компонент – МБП, реализующий функцию управления доступом.
8. Топология защищаемой системы описывается при помощи оператора взаимодействия.
9. Реализуется возможность аппаратного управления доступом к каналам передачи информации и управления для всех элементов системы с использованием оператора взаимодействия.

Рассмотрим метод построения электронных систем со встроенным механизмом реализации политики безопасности на примере защиты информации, циркулирующей в контроллере Super IO персонального компьютера. Пример структурной схемы контроллера Super IO показан на рис. 1.

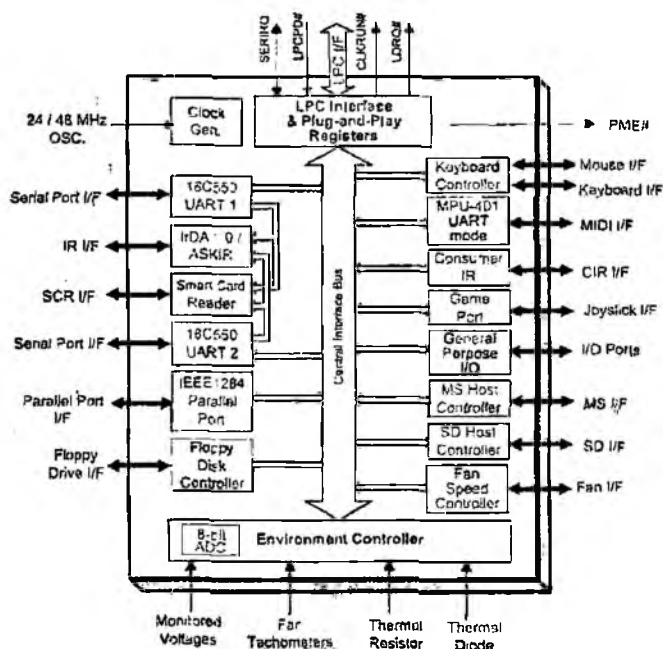


Рис. 1

Как видно из рисунка, данный контроллер содержит в своей структуре практически все периферийные контроллеры ввода/вывода используемые в персональном компьютере. Таким образом, именно в данной ИС сосредоточены все потоки информации, циркулирующие между пользователем системы и самой системой.

На первом этапе необходимо определить множество элементов защищаемой системы. В данном случае искомое множество будет содержать: множество периферийных устройств, подключаемых к соответствующим портам Super IO $\{Peripheral_i\}$ $i=1 \dots 13$ (по числу периферийных устройств); непосредственно сам Super IO; шину данных LPC интерфейса ICH2Bus, посредством которой Super IO обменивается информацией с ICH2. Таким образом множество элементов рассматриваемой аппаратной системы $\{E\} = \{Peripheral_i\} \cup SuperIO \cup ICH2Bus$, где $i=1 \dots 13$.

На втором этапе, необходимо определить возможность применения АЗ различных гипов в рассматриваемом множестве элементов системы. Поскольку объемы и скорость передачи информации, циркулирующей между периферийными устройствами (управляемыми пользователем) и самой системой, не высоки, а ее содержание имеет явно выраженный характер, то, очевидно, что в данном случае возможно применение АЗ накапливающего типа. Кроме того, поскольку в данной ИС сосредоточены практически все средства управления, доступные пользователю, для осуществления управления персональным компьютером, то в случае блокировки работы данной ИС пользователь системы потеряет возможность контролировать функционирование персонального компьютера. Таким образом, применение АЗ разрушающего типа также возможно. Применение АЗ, модифицирующих протокол передачи данных, в данном случае также возможно, однако поскольку в данной ИС циркулирует информация, вводимая либо выводимая пользователем, то АЗ будет реализовывать угрозу целостности информации, что для данного случая практически тождественно угрозе доступности, реализуемой АЗ разрушающего типа. По этой причине АЗ, модифицирующие протокол передачи данных исключаются из дальнейшего рассмотрения.

На третьем этапе, необходимо разбить все множество элементов на два подмножества: множество субъектов и множество объектов. В данном случае, все элементы системы в различные моменты времени могут представлять собой как субъекты так и объекты. По этой причине $\{S\} = \{E\}$, $\{O\} = \{E\}$.

На четвертом этапе, выделяются пары взаимодействующих элементов системы. Для нашего примера можно выделить следующие взаимодействующие пары:

$ICH2Bus \leftrightarrow SuperIO$

$SuperIO \leftrightarrow \{Peripheral_i\}$, где $i=1 \dots 13$.

На пятом этапе, для каждой взаимодействующей пары субъектов и объектов определяется множество разрешенных с точки зрения политики безопасности потоков. В рассматриваемом примере политика безопасности направлена:

- 1) на предотвращение возможности активации АЗ.
- 2) на предотвращение несанкционированной передачи накопленной информации.

В этом случае, запрещенной, с точки зрения ПБ, является любая передача данных по шине управления, не определенная в технической спецификации SuperIO, а также любая передача данных между $SuperIO \rightarrow ICH2Bus$, при отсутствии активности субъектов из множества $\{Peripheral_i\}$ и любая передача данных между $SuperIO \rightarrow \{Peripheral_i\}$, при отсутствии активности субъекта ICH2Bus.

Рассмотрим более подробно вопрос определения множества разрешенных с точки зрения политики безопасности потоков на примере взаимодействия SuperIO и периферийного контроллера клавиатуры персонального компьютера, подключенной непосредственно к SuperIO.

В технической спецификации контроллера SuperIO определены следующие возможные команды на шине данных со стороны контроллера SuperIO: Edh (00h-07h), Eeh, F0h (01,02,03), Efh-F1h, F2h, F3h (00h-7Fh), F4h, F5h, F6h, F7h-Fah, FBh-FDh, FEh, FFh.

При каждой передаче команды или данных от системы на клавиатуру система требует подтверждения от клавиатуры. Если только система не запретит вывод данных с клавиатуры, последняя даст ответ в течение 20 мс.

Если ответ клавиатуры недействителен или содержит ошибку паритета, система повторно посылает ей команду или данные. Однако двухбайтовые команды требуют особой обработки. Если были переданы и подтверждены команды F3h (установить скорость/задержку повторения), F0h (выбрать альтернативные коды сканирования) или Edh (установить/сбросить индикаторы режима), а также передан байт данных, но ответ оказался недействительным или с ошибкой паритета, то система передаст повторно и команду, и байт данных.

Клавиатура отвечает системе кодами, описанными ниже:

FAh- Подтверждение (ACK) в ответ на любую правильную команду, кроме команд "Эхо" RESEND; AAh – Код завершения включения питания; EEh – Клавиатура передает этот код в ответ на команду "Эхо"; 83Abh – ID клавиатуры.

Как после нажатия, так и после отжатия клавиши, контроллер клавиатуры посылает в систему соответствующий клавише скан-код. При этом код отжатия клавиши формируется путем прибавления префикса F0h к скан-коду нажатия.

В технической спецификации стандартной клавиатуры определены следующие скан-коды: 0x05, 0x06, 0x0e, 0x12, 0x15, 0x16, 0x1a-0x1e, 0x21-0x26, 0x29-0x2e, 0x31-0x36, 0x39-0x3e, 0x41-0x46, 0x49-0x4e, 0x52, 0x54, 0x55, 0x59, 0x5a, 0x5b, 0x5d, 0x61, 0x66, 0x69, 0x6b, 0x6c, 0x70-0x75, 0x79-0x7d.

Некоторые клавиши имеют расширенные скан-коды, начинающиеся с префикса E0h либо E1: E011, E014, E04A, E05A, E06B, E06C, E070, E071, E072, E075, E07A, E07D, E012E07C, E11477E1F014F077.

Таким образом, передача по шине данных команд либо данных, не описанных выше, является неспецифицированной и, следовательно, должна быть запрещена.

На шестом этапе, на основании информации, полученной на предыдущих этапах, формируется трехмерная матрица прав доступа M_R .

Матрица прав доступа для любой аппаратной системы, в общем случае, представляет собой трехмерную матрицу $M_R[S,O,F]$, где F – функция управления доступом. Таким образом, если по осям матрицы расположить множество субъектов, объектов и потоков, то содержимое любой ячейки будет не что иное, как значение функции управления доступом для элементов перечисленных множеств.

Присвоим контроллеру SuperIO индекс – 1, а контроллеру клавиатуры индекс – 2. Тогда матрица прав доступа для данной пары элементов будет иметь вид, представленный на рис. 2:

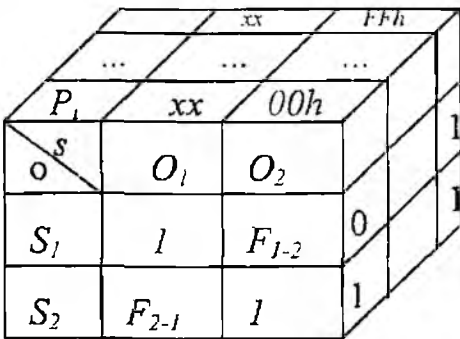


Рис. 2

На практике подобные матрицы оказываются достаточно громоздкими. По этой причине предлагается осуществлять декомпозицию данной матрицы на множество двумерных матриц, каждая из которых описывает функцию доступа, для заданной пары субъект – объект. Таким образом, для пары S_1O_2 матрица прав доступа будет иметь следующий вид (табл. 1).

Таблица 1

I	Oh	Ch	E D00h	E D07h	E DFFh	Eh	F 000h	F 001h
			1	1	0	1	0	1

В свою очередь, из табл. 1 видно, что она является избыточной, поскольку для задания функции управления доступом достаточно описать только разрешенные либо запрещенные команды для заданной пары субъект – объект (очевидно, что необходимо выбирать более короткий перечень команд), обозначим данное множество команд как $S_i O_j P$. Таким образом функция управления доступом может быть описана следующим образом:

$$\begin{aligned} \text{if } P_i \in \{S_i O_j P\} \text{ then } F_i(P_j) = 1 \\ \text{else } F_i(P_j) = 0 \end{aligned} \quad (1)$$

где $i = 1, \dots, n$; $\{S_i O_j P\}$ – множество разрешенных потоков (команд) для пары $S_i O_j$

На седьмом этапе в структуру аппаратной системы вносится новый элемент системы – МБП. Данный субъект системы представляет собой микропроцессорное устройство, обладающее разветвленной структурой портов ввода/вывода и реализующее функции управления доступом. Как уже отмечалось ранее, МБП представляет собой активный субъект системы, реализующий функцию управления доступом. Таким образом, алгоритм работы МБП может быть описан в виде следующего выражения:

$$\begin{aligned} S_i \text{ request (Command } x_i(P)) \text{ for } O_j \{ \\ \text{if } P_i \in \{S_i O_j P\} \text{ then } F_i(P_j) = 1 \\ \text{else } F_i(P_j) = 0 \\ \text{if } P_i = 1 \text{ then Command } x_i(P) \\ \} , i = 1, \dots, n; j = 1, \dots, n. \end{aligned} \quad (2)$$

На восьмом этапе топология защищаемой системы описывается при помощи оператора сопряжения и определяется перечень каналов, управление которыми необходимо осуществлять МБП. Интерфейс передачи данных контролера клавиатуры имеет вид, показанный на рис. 3.

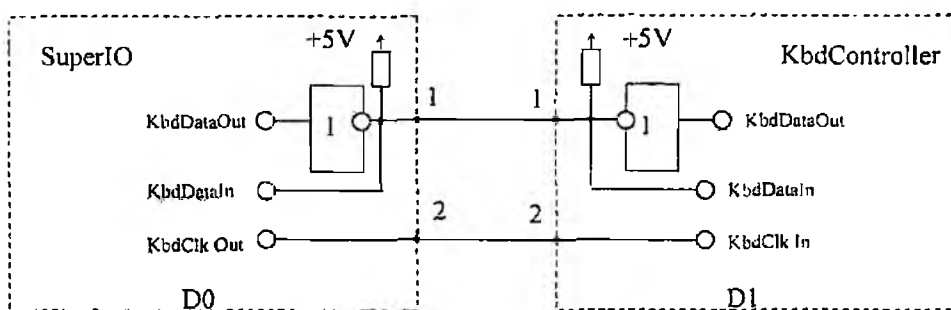


Рис. 3

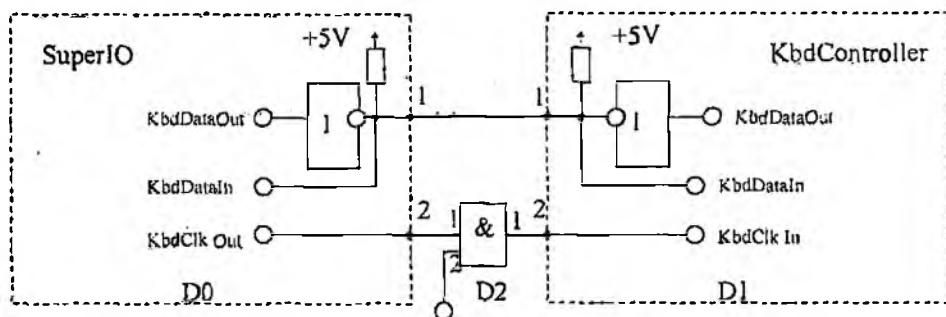
Оператор сопряжения для элементов показанных на рис. 3 зададим таблично:

Таблица 2

j i	1	2
0	1,1	
1	0,1	0,2

Поскольку интерфейс клавиатуры является синхронным, то для осуществления управления каналом передачи данных достаточно иметь возможность управления каналом синхронизации. В этом случае, при осуществлении факта запрещения доступа, для рассматриваемой пары элементов, достаточно удалить из табл. 2 элемент, стоящий на пересечении строки 1 и столбца 2.

На девятом этапе с целью возможности реализации МБП функции управления доступом, структура связей между элементами аппаратной системы модифицируется таким образом, чтобы МБП имел возможность управлять ими. Для этого между всеми элементами рассматриваемой системы устанавливаются управляемые буферы ввода/вывода, при помощи которых и производится процесс управления доступом. В рассматриваемом примере, исходная схема сопряжения элементов может быть модифицирована (рис. 4).



Управление от МБП

Рис. 4

Таким образом, данный пример показывает применимость предлагаемого метода для реализации заданной политики безопасности в аппаратных средствах обработки информации.

Список литературы: 1. Девянин П. Н., Михальский О. О., Правиков Д. И., Щербаков А. Ю. и др. Теоретические основы компьютерной информации. М.: Радио и связь, 2000. 189 с. 2. Зегжда Д. П., Ивашко А. М. Основы безопасности информационных систем. М.: Горячая линия – Телеком, 2000. 452 с., ил.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 11.10.2004

ВЕРоятностные характеристики схем генерации ключей с неравновероятным распределением

Основными требованиями, предъявляемыми к случайным и псевдослучайным генераторам, используемым в криптографических целях, являются независимость и равновероятность выходных символов. Так, например, если существует некоторое отклонение вероятностного распределения символов выходной последовательности генератора от равновероятного, то тогда может быть построен метод направленного перебора, эффективность которого будет значительно выше метода полного перебора. В связи с этим представляет интерес задача о нахождении оценок эффективности применения методов направленного перебора ключей, соответственно, о достаточных требованиях к статистическим свойствам генераторов ключевых данных. В общем случае количественными мерами эффективности применения методов криптоанализа являются трудоемкость и вероятность их применения.

Решаемая в рамках статьи задача – это оценка вероятности применения методов направленного перебора ключей при известном статистическом распределении символов выходной последовательности генераторов ключевых данных. В работе рассматривается вопрос о зависимости между количеством попыток и вероятностью успеха при известном методе направленного перебора ключей (в порядке убывания вероятности ключа).

Пусть G – случайный или псевдослучайный битовый генератор; $\xi_1, \xi_2, \dots, \xi_s, \dots$ – последовательность его выходов. Пусть выполнены следующие условия:

1) $\{\xi_i\}_{i=1}^{\infty}$ – последовательность независимых, одинаково распределённых случайных величин;

$$2) \forall i \geq 1 \quad P(\xi_i = 1) = p = \frac{1}{2} + \delta, \quad P(\xi_i = 0) = q = \frac{1}{2} - \delta, \quad 0 \leq \delta < \frac{1}{2}.$$

Здесь и далее будем обозначать $\eta_1, \eta_2, \dots, \eta_s, \dots$ последовательность независимых, одинаково распределённых случайных величин, для которых $\forall i \geq 1 \quad P(\eta_i = 1) = P(\eta_i = 0) = \frac{1}{2}$.

В данной работе будут рассмотрены следующие вероятностные характеристики генератора G :

- 1) Энтропия векторов $\xi_1, \xi_2, \dots, \xi_n$; определение такого числа m , что $H(\xi_1, \dots, \xi_n) = H(\eta_1, \dots, \eta_m)$.
- 2) Для заданного $0 < \gamma < 1$ нахождение такого (минимального) числа $N(\gamma)$, что $P_{N(\gamma)} \geq \gamma$, где $P_{N(\gamma)}$ – вероятность угадать некоторую фиксированную реализацию $a_1, \dots, a_n$ последовательности $\xi_1, \xi_2, \dots, \xi_n$ за $N(\gamma)$ попыток при оптимальном методе опробований (методе направленного перебора).
- 3) Нахождение математического ожидания числа попыток до успеха (т.е. угадывания вектора $(a_1, \dots, a_n)$) при направленном переборе.

1 Энтропийные характеристики векторов $(\xi_1, \xi_2, \dots, \xi_n)$

Введём следующие обозначения:

$H_i(p)$ – энтропия вероятностного распределения случайной величины ξ_i : $H_i(p) = -p \log p - q \log q = H_1(p) \quad \forall i \geq 1$;

- $H^{(n)}(p)$ – энтропия вероятностного распределения случайного вектора $(\xi_1, \xi_2, \dots, \xi_n)$;
- $H^{(m)} = H^{(m)}\left(\frac{1}{2}\right)$ – энтропия вероятностного распределения случайного вектора $(\eta_1, \eta_2, \dots, \eta_n)$.

Здесь и далее $\log$ обозначает логарифм по основанию 2. Тогда справедлива

Теорема 1

$$H^{(m_1)} \leq H^{(n)}(p) \leq H^{(m_2)}, \quad (1)$$

$$\text{где } m_1 = \left\lceil -n \log \left(\left(\frac{1+2\delta}{1-2\delta} \right)^\delta \sqrt{\frac{1}{4} - \delta^2} \right) \right\rceil, \quad m_2 = \left\lceil -n \log \left(\left(\frac{1+2\delta}{1-2\delta} \right)^\delta \sqrt{\frac{1}{4} - \delta^2} \right) \right\rceil.$$

Доказательство

Во-первых, заметим, что, по свойству энтропии, $H^{(n)}(p) = H_1(p) + \dots + H_n(p) = nH_1(p)$, вследствие независимости и одинаковой распределённости случайных величин $\xi_1, \xi_2, \dots, \xi_n$. Заметим также, что в данном случае $H^{(m)} = H^{(m)}\left(\frac{1}{2}\right) = mH^{(1)}\left(\frac{1}{2}\right) = m$.

Условие (1) эквивалентно условию $m_1 \leq nH_1(p) \leq m_2$, а условие $nH_1(p) = H^{(m_1)}$ – условию $m = -n \log \left(\left(\frac{1+2\delta}{1-2\delta} \right)^\delta \sqrt{\frac{1}{4} - \delta^2} \right)$. Последнее выражение, вообще говоря, не является целым

числом и лежит в пределах от $\left\lceil -n \log \left(\left(\frac{1+2\delta}{1-2\delta} \right)^\delta \sqrt{\frac{1}{4} - \delta^2} \right) \right\rceil$ до $\left\lceil -n \log \left(\left(\frac{1+2\delta}{1-2\delta} \right)^\delta \sqrt{\frac{1}{4} - \delta^2} \right) \right\rceil$. Поскольку $H^{(k)}$ как функция аргумента k является возрастающей, отсюда сразу следует утверждение теоремы.

В табл. 1 приведены значения m_1 , соответствующие указанным значениям n и δ . (Аналогичные значения, в соответствии с теоремой (1), могут быть получены и для m_2 ; при этом $m_2 \leq m_1 + 1$).

Таблица 1

$\delta \backslash n$	64	128	256	1024	2048
0,001	63	127	255	1023	2047
0,01	63	127	255	1023	2047
0,05	63	127	254	1016	2033
0,25	51	103	207	830	1661

К примеру, энтропия вектора, состоящего из 128 компонент, полученного при $\delta=0.25$, не менее чем энтропия 112-мерного вектора, имеющего равновероятное распределение.

2 Оценка функции распределения количества попыток до угадывания

Пусть N – случайная величина, равная количеству опробований (методом оптимального перебора) до успеха. Нужно найти функцию распределения случайной величины N в зависимости от n и δ , или хотя бы дать оценки для этой функции. Похожие вопросы были рассмотрены в работах [1-3]. В частности, наиболее близкая постановка задачи встречается в [3],

п.11. Однако там рассматривается случай двуступенчатого вероятностного распределения, тогда как в нашем случае количество «ступенек» равно n .

Теорема 2

Пусть последовательность $(\xi_1, \xi_2, \dots, \xi_n)$ получена с использованием генератора G .

Тогда

$$\lim_{n \rightarrow \infty} \left[P \left(N \leq \sum_{i=0}^k C_n^i \right) - \left(\Phi \left(\frac{2k - n(1-2\delta)}{\sqrt{n(1-4\delta^2)}} \right) - \Phi \left(-\sqrt{\frac{n(1-2\delta)}{1+2\delta}} \right) \right) \right] = 0, \quad (2)$$

где $0 \leq k \leq n$, $\Phi(x)$ – функция стандартного нормального распределения.

Доказательство

Согласно методу направленного перебора, сначала опробуем ключ, полностью состоящий из единиц (его вероятность максимальна и равна $p^n = (\frac{1}{2} + \delta)^n$); далее – ключи, в которых в точности $n-1$ единица, и т.д. Обозначим A_i , $i = \overline{0, n}$ – множество ключей, в которых ровно i нулевых координат. Очевидно, что $|A_i| = C_n^i$, $P(A_i) = C_n^i p^{n-i} q^i$. Тогда вероятность того, что искомым ключ содержится в множестве $\bigcup_{i=0}^k A_i$, равна $\sum_{i=0}^k C_n^i p^{n-i} q^i$, т.е., применив метод направленного перебора, мы определим нужный ключ не более чем за $\left| \bigcup_{i=0}^k A_i \right| = \sum_{i=0}^k C_n^i$ шагов.

Поскольку (см., например, [4])

$$\lim_{n \rightarrow \infty} \sum_{i=0}^k C_n^i p^{n-i} q^i = \lim_{n \rightarrow \infty} P \{ n-k \leq S_n \leq n \} = \lim_{n \rightarrow \infty} \left[\Phi \left(\frac{2k - n(1-2\delta)}{\sqrt{n(1-4\delta^2)}} \right) - \Phi \left(-\sqrt{\frac{n(1-2\delta)}{1+2\delta}} \right) \right],$$

то утверждение теоремы выполнено.

Замечания

1) Утверждение теоремы можно переформулировать следующим образом:

$P \left(N \leq \sum_{i=0}^k C_n^i \right) \approx \Phi \left(\frac{2k - n(1-2\delta)}{\sqrt{n(1-4\delta^2)}} \right) - \Phi \left(-\sqrt{\frac{n(1-2\delta)}{1+2\delta}} \right)$, причём модуль разности левой и правой части, по теореме Бери-Эссеена [4], не превосходит $\frac{p^2 + q^2}{\sqrt{npq}} \leq \frac{0,5}{\sqrt{0,25n}} = \frac{1}{\sqrt{n}}$. Отсюда

видно, что с ростом n приближение становится всё более точным. Например, при $n=64$ погрешность между реальной функцией распределения и её приближенным нормальным распределением не превосходит 0,125; при $n=256$ – 0,0625, при $n=2048$ – $9,8 \times 10^{-4}$, и т.д.

2) Если $T \in N$ такое, что $\sum_{i=0}^k C_n^i \leq T \leq \sum_{i=0}^{k+1} C_n^i$ и $L = T - \sum_{i=0}^k C_n^i$,

$$\text{то } P(N \leq T) \approx \Phi \left(\frac{2k - n(1-2\delta)}{\sqrt{n(1-4\delta^2)}} \right) - \Phi \left(-\sqrt{\frac{n(1-2\delta)}{1+2\delta}} \right) + L p^{k+1} q^{n-k-1},$$

и справедлива такая же оценка погрешности.

В табл. 2 – 5 приведены значения вероятности успеха для различных n и δ .

Таблица 2

$n = 64$ ($2^{64} \approx 1.8446 \times 10^{19}$), $\delta=0.25$					
T	$P_1(N \leq T)$	$P_2(N \leq T)$	T	$P_1(N \leq T)$	$P_2(N \leq T)$
1	1.928×10^{-6}	6.221×10^{-16}	2.092×10^{15}	0.61358	0.00023
65	7.451×10^{-6}	4.594×10^{-15}	5.693×10^{15}	0.71814	0.00057
2081	2.656×10^{-10}	3.19×10^{-14}	1.441×10^{16}	0.80676	0.00135
43745	8.744×10^{-9}	2.083×10^{-13}	3.403×10^{16}	0.87589	0.00298
679121	0.00026	1.279×10^{-12}	7.514×10^{16}	0.92554	0.00621
8.303×10^6	0.00074	7.392×10^{-12}	1.554×10^{17}	0.95836	0.01222
8.327×10^7	0.00194	4.016×10^{-11}	3.022×10^{17}	0.97834	0.02275
7.044×10^8	0.00468	2.052×10^{-10}	5.528×10^{17}	0.98953	0.04
5.13×10^9	0.01046	9.865×10^{-10}	9.538×10^{17}	0.99531	0.06687
3.267×10^{10}	0.02165	4.462×10^{-9}	1.555×10^{18}	0.99805	0.10565
1.841×10^{11}	0.04163	1.898×10^{-8}	2.402×10^{18}	0.99925	0.15865
9.277×10^{11}	0.07445	7.604×10^{-8}	3.52×10^{18}	0.99973	0.22662
4.211×10^{12}	0.1241	2.866×10^{-7}	4.909×10^{18}	0.99991	0.30853
1.734×10^{13}	0.19323	1.017×10^{-6}	6.529×10^{18}	0.99997	0.40129
6.52×10^{13}	0.28185	3.397×10^{-6}	8.307×10^{18}	0.999993	0.40129
2.247×10^{14}	0.38641	1.068×10^{-5}	1.013×10^{19}	0.999998	0.5
7.132×10^{14}	0.5	8.841×10^{-5}			

Таблица 3

$n = 64$ ($2^{64} \approx 1.8446 \times 10^{19}$), $\delta=0.1$					
T	$P_1(N \leq T)$	$P_2(N \leq T)$	T	$P_1(N \leq T)$	$P_2(N \leq T)$
1	3.2454×10^{-11}	6.221×10^{-16}	2.092×10^{15}	0.01410	0.00023
65	1.7278×10^{-10}	4.5946×10^{-15}	5.693×10^{15}	0.02624	0.00057
2081	8.6317×10^{-10}	3.1908×10^{-14}	1.441×10^{16}	0.04608	0.00135
43745	4.0465×10^{-9}	2.0838×10^{-13}	3.403×10^{16}	0.07652	0.00298
679121	1.7804×10^{-8}	1.2798×10^{-12}	7.514×10^{16}	0.12025	0.00621
8.303×10^6	7.3532×10^{-8}	7.3922×10^{-12}	1.554×10^{17}	0.17916	0.01222
8.327×10^7	2.8510×10^{-7}	4.016×10^{-11}	3.022×10^{17}	0.25353	0.02275
7.044×10^8	1.0379×10^{-6}	2.0522×10^{-10}	5.528×10^{17}	0.34154	0.04
5.13×10^9	3.5489×10^{-6}	9.8658×10^{-10}	9.538×10^{17}	0.43916	0.06687
3.267×10^{10}	1.1398×10^{-5}	4.4621×10^{-9}	1.555×10^{18}	0.54064	0.10565
1.841×10^{11}	3.4396×10^{-5}	1.8989×10^{-8}	2.402×10^{18}	0.63953	0.15865
9.277×10^{11}	9.7554×10^{-5}	7.6049×10^{-8}	3.52×10^{18}	0.72985	0.22662
4.211×10^{12}	0.00026	2.8665×10^{-7}	4.909×10^{18}	0.80717	0.30853
1.734×10^{13}	0.00065	1.017×10^{-6}	6.529×10^{18}	0.86921	0.40129
6.52×10^{13}	0.00153	3.3976×10^{-6}	8.307×10^{18}	0.91587	0.40129
2.247×10^{14}	0.00341	1.0688×10^{-5}	1.013×10^{19}	0.94876	0.5
7.132×10^{14}	0.00715	8.8417×10^{-5}			

Таблица 4

$n = 64$ ($2^{64} \approx 1.8446 \times 10^{19}$), $\delta=0.01$					
T	$P_1(N \leq T)$	$P_2(N \leq T)$	T	$P_1(N \leq T)$	$P_2(N \leq T)$
1	2.2247×10^{-15}	6.221×10^{-16}	2.092×10^{15}	0.00016	0.00023
65	1.5808×10^{-14}	4.5946×10^{-15}	5.693×10^{15}	0.00041	0.00057
2081	1.0563×10^{-13}	3.1908×10^{-14}	1.441×10^{16}	0.00099	0.00135
43745	6.6372×10^{-13}	2.0838×10^{-13}	3.403×10^{16}	0.00225	0.00298
679121	3.922×10^{-12}	1.2798×10^{-12}	7.514×10^{16}	0.00479	0.00621
8.303×10^6	2.1796×10^{-11}	7.3922×10^{-12}	1.554×10^{17}	0.00963	0.01222
8.327×10^7	1.1394×10^{-10}	4.016×10^{-11}	3.022×10^{17}	0.01829	0.02275
7.044×10^8	5.6027×10^{-10}	2.0522×10^{-10}	5.528×10^{17}	0.03285	0.04
5.13×10^9	2.5917×10^{-9}	9.8658×10^{-10}	9.538×10^{17}	0.05588	0.06687
3.267×10^{10}	1.128×10^{-8}	4.4621×10^{-9}	1.555×10^{18}	0.09007	0.10565
1.841×10^{11}	4.62×10^{-8}	1.8989×10^{-8}	2.402×10^{18}	0.1378	0.15865
9.277×10^{11}	1.7807×10^{-7}	7.6049×10^{-8}	3.52×10^{18}	0.2004	0.22662
4.211×10^{12}	6.4604×10^{-7}	2.8665×10^{-7}	4.909×10^{18}	0.27755	0.30853
1.734×10^{13}	2.2065×10^{-6}	1.017×10^{-6}	6.529×10^{18}	0.3669	0.40129
6.52×10^{13}	7.096×10^{-6}	3.3976×10^{-6}	8.307×10^{18}	0.46413	0.40129
2.247×10^{14}	2.1492×10^{-5}	1.0688×10^{-5}	1.013×10^{19}	0.56357	0.5
7.132×10^{14}	6.1324×10^{-5}	8.8417×10^{-5}			

Таблица 5

$n = 64$ ($2^{64} \approx 1.8446 \times 10^{19}$), $\delta=0.01$					
T	$P_1(N \leq T)$	$P_2(N \leq T)$	T	$P_1(N \leq T)$	$P_2(N \leq T)$
1	7.0824×10^{-16}	6.221×10^{-16}	2.092×10^{15}	9.4228×10^{-5}	0.00023
65	5.2103×10^{-15}	4.5946×10^{-15}	5.693×10^{15}	0.00024	0.00057
2081	3.6042×10^{-14}	3.1908×10^{-14}	1.441×10^{16}	0.00061	0.00135
43745	2.3446×10^{-13}	2.0838×10^{-13}	3.403×10^{16}	0.00142	0.00298
679121	1.4343×10^{-12}	1.2798×10^{-12}	7.514×10^{16}	0.00312	0.00621
8.303×10^6	8.2522×10^{-12}	7.3922×10^{-12}	1.554×10^{17}	0.00649	0.01222
8.327×10^7	4.4657×10^{-11}	4.016×10^{-11}	3.022×10^{17}	0.01274	0.02275
7.044×10^8	2.2731×10^{-10}	2.0522×10^{-10}	5.528×10^{17}	0.02362	0.04
5.13×10^9	1.0885×10^{-9}	9.8658×10^{-10}	9.538×10^{17}	0.04145	0.06687
3.267×10^{10}	4.9041×10^{-9}	4.4621×10^{-9}	1.555×10^{18}	0.0689	0.10565
1.841×10^{11}	2.0789×10^{-8}	1.8989×10^{-8}	2.402×10^{18}	0.1086	0.15865
9.277×10^{11}	8.2935×10^{-8}	7.6049×10^{-8}	3.52×10^{18}	0.16255	0.22662
4.211×10^{12}	3.1139×10^{-7}	2.8665×10^{-7}	4.909×10^{18}	0.23147	0.30853
1.734×10^{13}	1.1006×10^{-6}	1.017×10^{-6}	6.529×10^{18}	0.31419	0.40129
6.52×10^{13}	3.6626×10^{-6}	3.3976×10^{-6}	8.307×10^{18}	0.40749	0.40129
2.247×10^{14}	1.1478×10^{-5}	1.0688×10^{-5}	1.013×10^{19}	0.50638	0.5
7.132×10^{14}	3.3881×10^{-5}	8.8417×10^{-5}			

3 Математическое ожидание числа попыток до успеха при направленном переборе

В предыдущем разделе была определена случайная величина N , равная количеству опробований (методом оптимального перебора) до успеха. Найдем математическое ожидание этой величины. Для этого введем следующие обозначения:

$P\left(N \leq \sum_{i=0}^k C_n^i\right) \approx \Phi\left(\frac{2k - n(1 - 2\delta)}{\sqrt{n(1 - 4\delta^2)}}\right) - \Phi\left(-\sqrt{\frac{n(1 - 2\delta)}{1 + 2\delta}}\right) = u(k)$ – вероятность того, что число опробований не будет превосходить $\sum_{i=0}^k C_n^i$, а также $s_k = \sum_{i=0}^k C_n^i$ – для упрощения вида дальнейших формул.

Теорема 3

Математическое ожидание количества опробований методом оптимального перебора до успеха:

$$M(N) = p^n + \sum_{k=1}^n \frac{s_{k-1} + 1 + s_k}{2} \cdot C_n^k \cdot p^{n-k} \cdot q^k \approx \sum_{k=1}^n \frac{s_{k-1} + 1 + s_k}{2} \cdot (u(k) - u(k-1))$$

Доказательство

Согласно определениям, данным в разделе I, вероятность

$$P(s_{k-1} < N \leq s_k) = C_n^k \cdot p^{n-k} \cdot q^k \approx u(k) - u(k-1)$$

при $k = \overline{1, n}$ равна вероятности попасть во множество A_k – множество ключей, в которых ровно k нулевых координат. Все ключи из этого множества равновероятны, и для $\forall i: s_{k-1} + 1 \leq i \leq s_k$ справедливо равенство: $P(N = i) = p^{n-k} \cdot q^k \approx \frac{u(k) - u(k-1)}{C_n^k}$. Таким образом, вероятности угадывания ключа, когда перебор идет в фиксированном множестве ключей A_k , равны между собой.

В таком случае, математическое ожидание опробований методом оптимального перебора до угадывания ключа:

$$\begin{aligned} M(N) &= \sum_{j=1}^n j \cdot P(N = j) = 1 \cdot P(N = 1) + \sum_{k=1}^n \left[((s_{k-1} + 1) + (s_{k-1} + 2) + \dots + s_k) \cdot p^{n-k} \cdot q^k \right] = \\ &= p^n + \sum_{k=1}^n \frac{s_{k-1} + 1 + s_k}{2} \cdot C_n^k \cdot p^{n-k} \cdot q^k \end{aligned}$$

Если использовать приближения для $P(N = 1) = P(N \leq 1) = P(N \leq s_0) \approx u(0) = 0$, а для

$$C_n^k \cdot p^{n-k} \cdot q^k \approx u(k) - u(k-1), \text{ то } M(N) \approx \sum_{k=1}^n \frac{s_{k-1} + 1 + s_k}{2} \cdot (u(k) - u(k-1)).$$

Доказательство окончено.

В табл. 6 приведены значения $M(N)$, среднего числа опробований при случайном выборе ключа $M = 2^{n-1}$ и $M - M(N)$, подсчитанные с помощью пакета Maple 6 для разных n и δ :

Таблица 6

$\delta = 0,001$

n	$M(N)$	M	$M - M(N)$
64	9.176911×10^{18}	9.223372×10^{18}	4.646115×10^{16}
128	1.686446×10^{38}	1.701412×10^{38}	1.496606×10^{36}
256	5.707925×10^{76}	5.789604×10^{76}	8.167992×10^{74}

512	6.558992×10^{153}	6.703904×10^{153}	1.449117×10^{152}
1024	8.698785×10^{307}	8.988466×10^{307}	2.896809×10^{306}
2048	1.539545×10^{616}	1.615850×10^{616}	7.630563×10^{614}

$\delta=0,01$

n	$M(N)$	M	$M-M(N)$
64	8.736717×10^{18}	9.223372×10^{18}	4.866551×10^{17}
128	1.545949×10^{38}	1.701412×10^{38}	1.554629×10^{37}
256	4.947643×10^{76}	5.789604×10^{76}	8.419613×10^{75}
512	5.226237×10^{153}	6.703904×10^{153}	1.477667×10^{153}
1024	6.089292×10^{307}	8.988466×10^{307}	2.899173×10^{307}
2048	8.781819×10^{615}	1.615850×10^{616}	7.376684×10^{615}

$\delta=0,05$

N	$M(N)$	M	$M-M(N)$
64	6.444089×10^{18}	9.223372×10^{18}	2.779283×10^{18}
128	8.802398×10^{37}	1.701412×10^{38}	8.211721×10^{37}
256	1.820256×10^{76}	5.789604×10^{76}	3.969349×10^{76}
512	8.932599×10^{152}	6.703904×10^{153}	5.810644×10^{153}
1024	2.571467×10^{306}	8.988466×10^{307}	8.731319×10^{307}
2048	2.659694×10^{613}	1.615850×10^{616}	1.613191×10^{616}

$\delta=0,25$

N	$M(N)$	M	$M-M(N)$
64	8.174593×10^{16}	9.223372×10^{18}	9.141626×10^{18}
128	1.312809×10^{34}	1.701412×10^{38}	1.701281×10^{38}
256	4.52034×10^{68}	5.789604×10^{76}	5.789604×10^{76}
512	7.328172×10^{137}	6.703904×10^{153}	6.703904×10^{153}
1024	2.673735×10^{276}	8.988466×10^{307}	8.988466×10^{307}
2048	4.984681×10^{553}	1.61585×10^{616}	1.61585×10^{616}

Представленные результаты позволяют оценивать эффективность применения методов направленного перебора при соответствующих значениях n и δ . В частности, показано, что с ростом n увеличивается допустимое значение отклонения δ . Кроме того, данные результаты позволяют оценить необходимое количество шагов, при котором вероятность применения методов направленного перебора не опускается ниже некоторого наперед заданного значения.

Список литературы: 1. Wozencraft J.M., Reiffen B. Sequential Decoding. Cambridge, MA: Technical M.I.T., 1960. 2. Maurer U.M. A Universal Statistical Test for Random Bit Generators// Journal of Cryptology. 1992. №5. P.89-105. 3. Савчук М.Н. Асимптотический анализ вариационного ряда вероятностей серий различных исходов в полиномиальной схеме испытаний // Доповіді Національної академії наук України. 1999. №3 С.101-105. 4. Ширяев А.Н. Вероятность. М.: Наука, 1989. 640 с.

Департамент специальных
телекоммуникационных
систем и защиты информации

Поступила в редколлегию 07.11.2004

МЕТОД ФОРМИРОВАНИЯ ПРОФИЛЕЙ СТАТИСТИЧЕСКОГО ТЕСТИРОВАНИЯ ДЛЯ ОЦЕНКИ СВОЙСТВ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Развитие систем обработки информации приводит к необходимости совершенствования механизмов защиты информации. В зависимости от назначения и условий применения в этих системах могут применяться средства криптографической защиты информации, в основу построения которых положены симметричные криптографические алгоритмы. Для систем больших размеров одной из важных проблем становится создание подсистемы управления ключами.

Рассматривая различные варианты построения подсистем управления ключами можно отметить, что одним из основных элементов данной подсистемы являются средства формирования ключевых документов¹. Основными требованиями, которые выдвигаются к данным средствам, являются требования по производительности и типам формируемых ключевых документов. Очевидно, что производительность средств должна быть такой, чтобы полностью обеспечить потребности системы во всех типах ключевых документов.

В состав средств формирования ключевых документов, как правило, входят:

- генератор случайной последовательности (ГСП) задачей которого является формирование случайной последовательности с заданными характеристиками;
- устройство формирования ключевого документа, которое на базе случайной последовательности формирует ключевые данные (преобразует последовательность в формат воспринимаемый конкретным средством КЗИ) и записывает их на соответствующий носитель.

Из анализа данной структуры следует, что наиболее важным ее элементом является генератор случайных последовательностей. Именно его характеристиками определяется качество ключевых данных и эффективность функционирования системы в целом. Выбор и обоснование характеристик генераторов случайных последовательностей в каждом конкретном случае осуществляется отдельно. Сложность формирования требований к генераторам заключается в том, что тяжело формализовать и проконтролировать выполнение требований, выдвигаемых к случайным последовательностям.

В данной статье предлагается метод формализации требований к случайным последовательностям и методика формирования соответствующих требований к структуре ГСП.

Генератор случайных чисел рассматривается как совокупность датчика случайных чисел (ДСЧ) и схемы тестирования – системы проверки претендентов (СПП) [1] - [5] (рис. 1). Будем полагать, что на выходе ДСЧ формируются последовательности вероятность появления которых подчиняется биномиальному закону при $p=0.5$.

Обобщенный алгоритм работы такого ГСП следующий:

- 1) Датчик случайных чисел формирует множество бит, которые объединяются в последовательности – претенденты R_p (далее претенденты).
- 2) Каждый претендент проверяется системой проверки претендентов.
- 3) По результатам проверки принимается решение о пригодности каждого претендента.

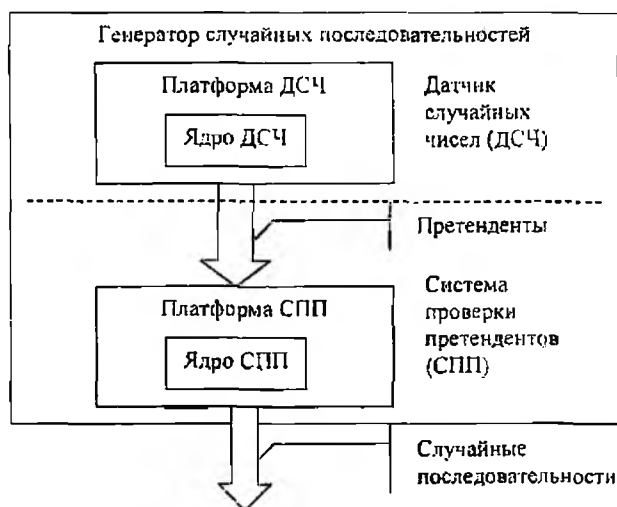


Рис. 1

¹ Под ключевым документом понимается носитель с нанесенными на него соответствующим образом ключевыми данными.

Если претендент отвечает заданным требованиям, считается, что он получил подтверждение системы проверки претендентов и может считаться случайной последовательностью R . В противном случае он отбраковывается.

Анализ структуры ГСП показывает, что качество случайной последовательности полностью определяется правилами отбора последовательностей, реализованными в СПП. Производительность зависит от производительности ДСЧ и производительности СПП и определяется наименьшей из них.

Для построения системы проверки претендентов предлагается использовать набор статистических тестов, которые реализуют правила отбора претендентов. Множество статистических тестов, объединенных правилами их применения, будем называть профилем статистического тестирования. Будем полагать, что статистический тест имеет конкретную сложность O_i , измеряемую количеством тактов ЭВМ, необходимых для обработки одного бита информации (такт/бит). Очевидно, что различные реализации одного и того же теста могут иметь различную сложность. Сложность реализации СПП будет равна сумме сложностей применяемых тестов.

Для построения профилей используются статистические тесты, которые можно использовать при применении вероятностного подхода оценки случайности последовательности. Суть подхода заключается в проверке нулевой гипотезы H_0 , заключающейся в том, что тестируемая последовательность является случайной. Альтернативной гипотезой H_a является гипотеза о том, что тестируемая последовательность не случайна. По результатам применения теста нулевая гипотеза либо принимается, либо отвергается.

Порядок тестирования отдельной последовательности выглядит следующим образом.

1. Выдвигается нулевая гипотеза H_0 - предположение о том, что данная двоичная последовательность S случайна.
2. По последовательности S вычисляется статистика теста $c(S)$.
3. С использованием специальной функции и статистики теста вычисляется значение вероятности $P = f(c(S))$, $P \in [0, 1]$.
4. Значение вероятности P сравнивается с уровнем значимости α , $\alpha \in [0.001, 0.01]$. Если, $P \geq \alpha$ то гипотеза H_0 принимается. В противном случае принимается гипотеза H_a .

Статистика теста строится таким образом, чтобы выявить некий дефект в последовательности.

Значение вероятности P может определяться с использованием [9]:

- функции стандартного нормального распределения

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{u^2}{2}} du; \quad (1)$$

- дополнительной функции ошибок

$$\operatorname{erfc} = \frac{2}{\sqrt{\pi}} \int_{-z}^{\infty} e^{-u^2} du; \quad (2)$$

- неполной гамма-функции

$$Q(a, x) \equiv 1 - P(a, x) \equiv \frac{\Gamma(a, x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)} \int_x^{\infty} e^{-t} t^{a-1} dt, \quad (3)$$

где $Q(a, 0) = 1$ и $Q(a, \infty) = 0$.

Анализ показал [6] - [9], что в основу наиболее мощных библиотек статистического тестирования ГПСЧ, к которым можно отнести пакет DIEHARD, Crypt-SX и пакет NIST 800-22, заложен именно такой критерий принятия решения.

Основываясь на том факте, что статистика теста строится таким образом, чтобы ее значения указывали на какой-либо дефект случайной последовательности, можно проверить реальную чувствительность каждого теста на данные дефекты. Для этого введем понятие искажения

D как множество неслучайных элементов в алфавите последовательности $\mathcal{A}$, распределенных по некоторому закону и привнесенных в случайную последовательность с заменой в последней оригинальных элементов. Основными характеристиками искажения являются:

- абсолютная длина искажения l_d находится в пределах $0 \leq l_d \leq l_r$, где l_r – длина случайной последовательности;
- закон распределения символов внутри.

Каждое искажение, привнесенное в последовательность, представляет собой определенный дефект, который должен определяться тестом. Для упрощения дальнейшего изложения введем понятие относительной длины искажения, определяемой как $\rho = l_d / l_r$.

Для определения чувствительности теста воспользуемся следующим подходом. Возьмем n случайных последовательностей $R = \{R_i\} = \{r_{ij}\}$, $i = \overline{1, n}$, $j = \overline{1, l_r}$, $\forall r_{ij} \in \mathcal{A}$. Определим шаг изменения относительной длины искажения Δl_d . Установим начальное значение длины искажения равное $l_{d0} = 0$. Рассчитаем для каждой последовательности значение вероятности статистики теста P_0 . Подсчитаем количество подтвержденных претендентов и определим коэффициент подтверждения претендентов η_0 для l_{d0} . Общее выражение для расчета коэффициента подтверждения претендентов следующее

$$\eta = \frac{N_R}{N_p}, \quad (4)$$

где N_R – количество случайных последовательностей;

N_p – количество претендентов.

В соответствии с выражением $l_d = l_{d-1} + \Delta l_d$ определяем длину очередного искажения и рассчитываем значение η . Увеличение длины искажения следует продолжать до тех пор, пока $N_R > 0$ и $\rho < 1$. Таким образом, мы получим зависимость коэффициента подтверждения претендентов η от размера относительной длины искажения ρ (рис. 2).

В случае, когда тест определяет конкретный дефект, то с увеличением длины искажения количество случайных последовательностей будет уменьшаться и в определенный момент станет равным нулю (кривые t_2 , t_4 , t_5). Будем говорить, что такие тесты имеют положительную чувствительность на данное искажение. Когда количество случайных последовательностей не изменяется (кривая t_3) или увеличивается (кривая t_1) будем говорить, что тест имеет нулевую или отрицательную чувствительность на данное искажение. Другими словами данный тест не распознает данное искажение. Графики чувствительности тестов (рис. 2) построены на основании расчетов приведенных в [1].

Для дальнейшего анализа будем рассматривать только те тесты, кото-

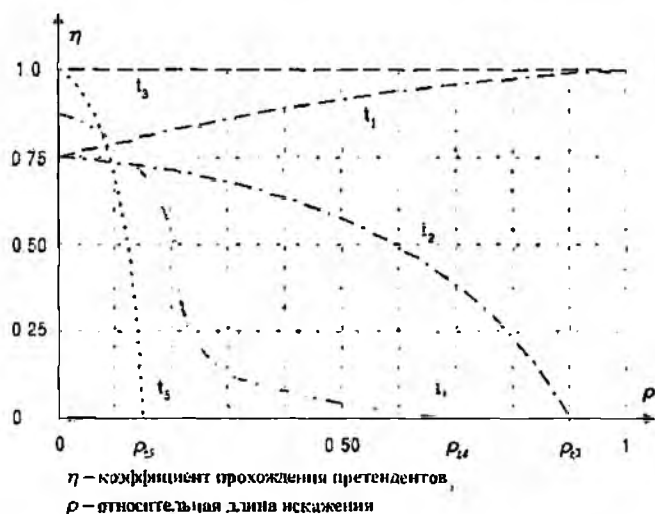


Рис. 2

рые имеют положительную чувствительность. Наиболее важной характеристикой для таких тестов является значение относительной длины искажения, при котором коэффициент подтверждения равняется нулю. Данное значение относительной длины искажения будем называть пределом чувствительности теста и обозначать ρ_c . Следует заметить, что оценка чувствительности теста должна происходить при определенном уровне значимости теста. Исходя из практических соображений уровень значимости теста был выбран 0,001.

Кроме предела чувствительности теста, для построения системы проверки претендентов необходимо ввести следующие характеристики:

- чувствительность теста $\xi = 1/\rho_i$;
- добротность теста $G=(\rho_1\alpha_1)/(\rho_2\alpha_2)$, $\alpha_1 < \alpha_2$;
- область применимости теста $L_i = (l_{i\min}, l_{i\max})$, где $l_{i\min}$ – минимальная, а $l_{i\max}$ – максимальная длина последовательности, при которой может применяться тест;
- сложность теста O_i .

Исходными данными для построения профиля статистического тестирования являются перечень недопустимых дефектов и их длины. Методика построения профиля заключается в выборе тестов и определении параметров тестирования для выявления каждого искажения. Порядок выбора заключается в выполнении следующих действий:

1. Определяется абсолютная минимальная длина искажения $l_{d\max}$, при которой данный профиль должен его распознавать.
2. Выбирается тест, имеющий соответствующую чувствительность на данное искажение.

При больших длинах тестируемых последовательностей чувствительность теста может оказаться не достаточной для выявления искажений малой длины, поэтому необходимо применять несколько тестов, которые тестируют последовательность на разных длинах. Множество тестов, применяемых для выявления одного искажения, будем называть линейкой тестирования. Область применимости тестов должна выбираться таким образом, чтобы любое значение длины последовательности в заданных пределах принадлежало объединению областей применимости множества тестов:

$$l_r \in \bigcup_{i=\overline{1,n}} L_i, l_r = \overline{l_{d\max}, l_r}. \quad (5)$$

Для каждого из тестов определяем минимальное $l_{i\min}$ и максимальное $l_{i\max}$ значения длин тестируемых последовательностей, а также значение предела чувствительности ρ_i . Первым тестом линейки будем называть тест, для которого $l_{i\max} = l_r$, т.е. тест, который тестирует последовательности самой большой длины. Длины последовательностей, тестируемые следующими тестами, определяются из условий:

$$l_{i+1\max} \geq \rho_i l_{i\max}, \quad (6)$$

$$l_{d\max} \geq \rho_{n_e} l_{n_e\max} \quad (7)$$

где n_e – общее количество тестов в одной линейке.

В случае, когда область применимости теста достаточно широка, а чувствительность низкая, допускается применение одного и того же теста на разных длинах. При этом выражение (6) можно переписать в виде

$$l_{i+1\max} \geq \rho_i^{n_i} l_{i\max} \quad (8)$$

где n_i – количество "повторов" одного теста.

Из выражения (8), можно получить количество повторов одного и того же теста

$$n_i = \left\lceil \log_{\rho_i} \frac{l_{i+1\max}}{l_{i\max}} \right\rceil \quad (9)$$

Построенная таким образом линейка тестов является составной частью профиля тестирования. Множество тестов $T = \{t_i\}$, $i = \overline{1,n}$, логически объединенных в k линеек с соответствующими значениями $(\alpha_i, l_{i\min}, l_{i\max})$, представляют собой бета - профиль $\Pi_\beta = \{t_i, \alpha_i, l_{i\min}, l_{i\max} \}$, $i = \overline{1,n}$. Суммарная ошибка первого рода будет рассчитываться в соответствии с выражением

$$A = 1 - \prod_{i=1}^n (1 - \alpha_i). \quad (10)$$

Заметим, что данная оценка справедлива в случае применения независимых тестов.

Анализ выражения (10) показывает, что при увеличении количества тестов будет увеличиваться суммарная ошибка первого рода. Как показывает анализ выражений (8) - (10), одним из возможных путей уменьшения суммарной ошибки первого рода является уменьшение количества тестов в одной линейке за счет увеличения уровня значимости каждого теста.

Рассмотрим условия, при которых такой подход является приемлемым. Очевидно, что с увеличением уровня значимости теста уменьшается его предел чувствительности. Зависимость степени изменения предела чувствительности от изменения уровня значимости определена как добротность теста G . Определим граничные значения добротности теста, при которых целесообразно увеличивать уровень значимости, а не количество тестов в "линейке". Анализ выражений (6)-(9) показывает, что должно выполниться следующее условие:

$$\rho_1^{n_{e1}} = \rho_2^{n_{e2}}, \quad (11)$$

откуда

$$\rho_2 = \sqrt[n_{e2}]{\rho_1^{n_{e1}}}. \quad (12)$$

где n_{e1} – начальное количество тестов в одной "линейке";

n_{e2} – конечное количество тестов в одной "линейке".

Очевидно, что $\rho_{22} < \rho_{21}$ а $\alpha_2 > \alpha_1$. С другой стороны, суммарная ошибка первого рода в обоих случаях должна быть одинаковой, т.е.

$$(1 - \alpha_1)^{n_{e1}} = (1 - \alpha_2)^{n_{e2}}, \quad (13)$$

откуда

$$\alpha_2 = 1 - \sqrt[n_{e2}]{(1 - \alpha_1)^{n_{e1}}}. \quad (14)$$

Подставляя выражения (12) и (14) в формулу, определяющую добротность теста, получим выражение для определения граничного значения добротности теста:

$$G_0 = \frac{\rho}{\sqrt[n_{e1}]{\rho_1^{n_{e1}}}} \cdot \frac{\alpha_1}{1 - \sqrt[n_{e2}]{(1 - \alpha_1)^{n_{e1}}}}. \quad (15)$$

Если реальное значение добротности теста превышает граничное, то целесообразно увеличивать уровень значимости теста, а не количество тестов в одной линейке.

В табл. 1 и 2 приведены рассчитанные граничные значения добротности теста в зависимости от общего количества тестов в "линейке" и уровня значимости теста.

Таблица 1

$\rho = 0,001, n_{e2} = n_{e1} - 1$

n_{e1}	α					
	0.001	0.005	0.01	0.05	0.1	0.5
2	500.25	501.25	502.51	512.82	526.32	666.67
3	21.087	21.108	21.135	21.351	21.632	24.459
4	7.5013	7.5063	7.5125	7.5637	7.6301	8.2898
5	4.4993	4.5015	4.5044	4.5274	4.5572	4.8515
6	3.3179	3.3192	3.3209	3.3345	3.352	3.5248
7	2.7107	2.7117	2.7128	2.722	2.734	2.8512
8	2.3475	2.3482	2.349	2.3559	2.3648	2.4516
9	2.108	2.1085	2.1092	2.1146	2.1216	2.1896
10	1.9391	1.9395	1.9401	1.9445	1.9502	2.0058

$$\rho = 0,1, n_{e2} = n_{e1} - 1$$

n_{e1}	α					
	0.001	0.005	0.01	0.05	0.1	0.5
2	5.0025	5.0125	5.0251	5.1282	5.2632	6.6667
3	2.1087	2.1108	2.1135	2.1351	2.1632	2.4459
4	1.6161	1.6172	1.6185	1.6296	1.6439	1.786
5	1.4228	1.4235	1.4244	1.4317	1.4411	1.5342
6	1.3209	1.3214	1.3221	1.3275	1.3345	1.4032
7	1.2582	1.2586	1.2592	1.2635	1.269	1.3234
8	1.2159	1.2162	1.2167	1.2202	1.2248	1.2698
9	1.1854	1.1857	1.1861	1.1891	1.193	1.2313
10	1.1625	1.1627	1.163	1.1657	1.1691	1.2024

На практике при оценке добротности тестов, входящих в состав пакета статистического тестирования NIST STS 800-22 [9, 10], полученные значения добротности тестов находились в пределах 0 - 3,5.

Анализ требований, выдвигаемых к случайным последовательностям [11 - 15], показывает, что в отдельных случаях полученный профиль статистического тестирования может быть достаточно сложным, что приводит к недопустимо низкой производительности ГСП. Для повышения производительности ГСП можно применять следующие основные подходы:

- повышение производительности платформы СПП;
- оптимизация профиля статистического тестирования;
- снижение требований к свойствам случайных последовательностей.

Повышение производительности ГСП за счет увеличения производительности платформы СПП является достаточно эффективным, т.к. процесс тестирования может быть легко распараллелен на несколько ЭВМ, количество которых, в принципе, не ограничено. Логичным ограничением в этом случае является достигнутая абсолютная производительность СПП, которая не должна превышать абсолютной производительности ДСЧ, т.к. в противном случае система проверки претендентов будет часть времени простаивать.

Оптимизация профиля включает в себя уменьшение количества тестов в линейках тестирования и формирование определенного порядка применения тестов. Уменьшение количества тестов в линейках должно происходить без снижения требований к случайной последовательности, что возможно только при условии достаточно большого значения добротности тестов. Повышение производительности СПП будет осуществляться за счет уменьшения сложности профиля, которая определяется суммой сложностей входящих в него тестов, при этом остальные параметры ГСП остаются без изменений.

Суть формирования определенного порядка применения тестов заключается в установлении четкой очередности. Основным признаком, определяющим порядок применения тестов, является длина тестируемой последовательности – от меньшей к большей. Но в случае, когда встречаются два или более тестов, предназначенных для тестирования последовательностей равной длины, очередность их применения будет играть существенную роль в суммарной производительности СПП. Это обуславливается тем, что каждый из применяемых тестов будет отбраковывать количество последовательностей, не меньшее, чем α . В результате каждый последующий статистический тест будет обрабатывать меньше количество последовательностей, чем предыдущий, что можно представить как эквивалентное уменьшение сложности по отношению к общему количеству входящей информации. Таким образом, суммарную эквивалентную сложность тестов, предназначенных для тестирования последовательностей равной длины, можно представить выражением

$$O_{i+}^{el} = O_{i1}^i + \sum_{j=2}^n \left(\prod_{j=1}^i (1 - \alpha_j) \right) O_{ij}^i, \quad (16)$$

где O_{i+}^l - суммарная эквивалентная сложность тестов, предназначенных для тестирования последовательностей длины l (далее теста длины l);

O_n^l - сложность i -го теста длины l ;

n - количество тестов длины l ;

α_j - уровень значимости j -го теста.

Анализ выражения (16) показывает, что в случае когда все уровни значимости равны между собой ($\alpha_j = \alpha$, $j \neq i$) минимум достигается, когда $O_n^l \leq O_{n+1}^l$. Таким образом, очередность применения статистических тестов, предназначенных для тестирования последовательностей равной длины, определяется сложностью реализации теста - от меньшей к большей.

Обобщая выражение (16), можно говорить, что суммарная эквивалентная сложность ядра системы проверки претендентов определяется выражением

$$O_{\text{СПП}}^e = O_{n1} + \sum_{i=2}^n \left(\prod_{j=1}^i (1 - \alpha_j) \right) O_n^l, \quad (17)$$

где $O_{\text{СПП}}^e$ - суммарная эквивалентная сложность ядра системы проверки претендентов;

O_n^l - сложность i -го теста;

n - количество тестов длины l ;

α_j - уровень значимости j -го теста

при условии, что значения O_n^l и α_j в выражении (17) соответствуют реальной очередности применения статистических тестов и все тесты независимы.

В случае, когда оптимизация профиля и повышение производительности платформы СПП не привели к достижению заданной производительности ГСП, единственным способом достижения цели является снижение требований к свойствам случайных последовательностей. При этом уменьшается не только значение суммарной ошибки первого рода, но и, возможно, общее количество тестов, входящих в состав профиля. Однако такой подход однозначно приводит к снижению требований к качеству случайных последовательностей и к изменению их свойств.

Таким образом, предложенный подход применения искажений позволяет формализовать требования к случайным последовательностям и оценить чувствительность статистических тестов. Метод формирования профилей статистического тестирования случайных последовательностей на основе формализованных требований к ним и с учетом характеристик статистических тестов, таких как чувствительность и сложность реализации, позволяет:

- обосновать множество тестов необходимых и достаточных для построения профиля статистического тестирования;
- определить порядок применения тестов, который обеспечивает выполнение требований, выдвинутых к случайным последовательностям;
- оценить сложность реализации профиля, что создает условия для оценки производительности генератора случайных последовательностей в целом, и позволяет обосновать требования к производительности остальных составных частей ГСП.

Построенный таким образом профиль статистического тестирования представляет собой механизм, позволяющий оценить соответствие свойств случайных последовательностей выдвинутым требованиям.

Список литературы: 1. *Попович Е.В., Потий А.В.* Методика оперативного статистического тестирования случайных последовательностей большой длины // Радиотехника 2003. Вып. 134. С.41-48. 2. *Попович Е.В.* Построение профилей оперативного статистического тестирования и их применение в криптографии // Седьмая международная научно-практическая конференция "Безопасность информации в информационно-телекоммуникационных системах". 12-14 мая 2004: Тез.докл. К.: ЧП "ЕКМО", НИЦ "ТЕЗИС" НТУ "КПИ", 2004. С. 48-50. 3. *Потий А.В. Попович Е.В., Горбенко Ю.И., Вержейко В.* Программно-аппаратный комплекс формирования и исследования случайных и псевдослучайных чисел // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні: Науково-технічний збірник. Вип.2. Київ, 2001. С. 171-175. 4. *Потий А.В., Пестерев А.К.* Принципы системного подхода к сертификации генераторов псевдослу-

чайных чисел в системах защиты информации // Радиотехника. 1997. Вып.104. С.163-172. 5. *Торба А.А., Елаков С.Г., Степченко А.З.* Генерация равновероятных случайных последовательностей на основе физических датчиков // Радиотехника. Всеукр. Межвед. науч.-техн. сб. 2001. Вып. 119. С. 108-113. 6. *Gustafson H.* Statistical test suite Crypt-SX. Available on <http://www.isrc.qut.edu.au/cryptx>. 7. *Marsaglia G.* DIEHARD Statistical Tests. Available on <http://stat.fsu.edu/~geo/diehard.html>. 8. *Потий А.В., Орлова С.Ю., Гриценко Т.А.* Статистическое тестирование случайных и псевдослучайных чисел с использованием набора статистических тестов NIST STS // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. Науково-технічний збірник. Вип.2. Київ, 2001. С. 206-214. 9. *COMPUTER SECURITY. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications.* NIST Special Publication 800-22, 2000. 10. *Попович Е.В., Потий А.В.* Построение профиля оперативного статистического тестирования случайных последовательностей большой длины // Безпека інформації в інформаційно-телекомунікаційних системах: Тези доповідей. К.: Інтерлінк, НДІ "Тезис", КПІ, 2003. С. 58-60. 11. *Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.* Основы криптографии: Учеб. пособие. М.: Гелиос АРВ, 2001. 480 с. 12. *Шеннон К. Э.* Теория связи в секретных системах // Работы по теории информации и кибернетике (Сб. статей). М.: Изд. иностр. лит. 1963. 830с. (с.333-369). 13. *Бабаш А.В., Шанкин Г.П.* Криптография. М.: СОЛОН-Р, 2002. 512с. 14. *Зубов А.Ю.* Совершенные шифры. М.: Гелиос АРВ, 2003. 160 с. 15. *Котенко В.В., Поликарпов С.В.* Новый подход к оценке эффективности защиты информации // Информационное противодействие угрозам терроризма. Науч.-практ. журнал №1. 2003. С.92 – 94.

*Департамент специальных
телекоммуникационных
систем и защиты информации*

Поступила в редколлегию 09.11.2004

КРИТЕРИЙ ОЦЕНКИ ТРЕБОВАНИЙ К ГЕНЕРАТОРАМ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ В КРИПТОГРАФИЧЕСКИХ СИСТЕМАХ

Задача построения надежных систем с функциями криптографической защиты информации, с развитием средств и систем обработки информации становится все более актуальной. Одним из основных элементов таких систем, от характеристик которого зависят характеристики системы в целом, является средство генерации ключей. Главными требованиями, выдвигаемыми к ним, являются [1 - 5]:

- размерность ключевого пространства должна быть максимальной;
- появление каждого отдельного ключа должно быть равновероятным и независимым;
- качество получаемых ключей должно соответствовать заданным требованиям.

На сегодняшний день наиболее широкое распространение получили средства генерации ключей, основанные на генераторах случайных последовательностей (ГСП) которые в большинстве случаев состоят из датчика случайных чисел и схемы проверки пригодности последовательностей к применению [6 - 8]. Датчики случайных чисел (ДСЧ) будем рассматривать как источник потока двоичных данных. Схема проверки является фильтром, исключающим последовательности, не удовлетворяющие заданным требованиям, и, как правило, представляет собой набор статистических тестов, объединенных некоторыми правилами их применения [9 - 11].

Как показал анализ источников [1 - 7, 12] в настоящее время не существует единого критерия, позволяющего дать оценку требованиям, предъявляемым к генераторам случайных последовательностей. Наличие такой оценки должно позволить не только обосновать перечень, но и подобрать оптимальный набор требований, выдвигаемых к ГСП.

В данной статье предлагается в качестве такого критерия использовать обобщенный коэффициент эффективности криптографической атаки Υ , который показывает, во сколько раз уменьшается сложность криптографической атаки по сравнению с атакой "грубая сила" при полном переборе ключей. Следует заметить, что атака типа "грубая сила" является единственной атакой, применимой для любого криптографического алгоритма, сложность которой определяется размерностью ключевого пространства:

$$S_{na} = \sum_{l=0}^n C_n^l, \quad (1)$$

где n – длина ключа; l – количество единиц в ключе.

Лучшим ГСП для конкретной системы является тот, у которого обобщенный коэффициент эффективности криптографической атаки является наименьшим. Данный коэффициент предлагается рассчитывать в соответствии со следующим выражением

$$\Upsilon = \Upsilon_0 \prod_{i=1}^k \Upsilon_i, \quad (2)$$

где Υ_0 – коэффициент эффективности атаки типа "перебор";

k – количество известных криптографических атак, применимых для данного криптоалгоритма, построенных на использовании запрещенных ключей;

Υ_i – коэффициент эффективности i -й атаки.

Под атакой типа "перебор" понимается частный случай атаки "грубая сила", которая заключается в переборе разрешенных ключей для данного криптоалгоритма. Таким образом, коэффициент эффективности атаки "перебор" показывает, во сколько раз уменьшается размерность реально используемого ключевого пространства по отношению к размерности абсолютного ключевого пространства для данного криптоалгоритма.

Среди условий, влияющих на размерность реально используемого ключевого пространства, можно выделить условия, определяемые особенностями криптоалгоритмов и особенностями генераторов случайных последовательностей. К особенностям криптоалгоритмов относится наличие у них запрещенных к использованию ключей, применение которых может привести к понижению стойкости криптоалгоритма. Применение запрещенных ключей приводит к тому, что атака, ориентированная на использование особенностей данных ключей, будет иметь коэффициент эффективности больше единицы. И, наоборот, исключение из применения запрещенных ключей приведет к тому, что коэффициенты эффективности данных атак будут равны единице. Однако исключение отдельных ключей из применения уменьшает размерность реально используемого ключевого пространства, что приводит к увеличению эффективности атаки "перебор".

Из анализа выражения (2) следует, что в условиях, когда на выходе ГСП отсутствуют любые запрещенные ключи, обобщенный коэффициент эффективности криптографической атаки равняется коэффициенту эффективности атаки типа "перебор" и определяется размерностью реально используемого ключевого пространства. В таких условиях задачу оценки обобщенного коэффициента можно свести к задаче определения размерности реально используемого ключевого пространства. Для решения данной задачи рассмотрим свойства ГСП.

Для предложенной структуры ГСП выполнение первого и второго требований, выдвигаемых к средствам генерации ключей, осуществляется за счет свойств датчика случайных чисел, третьего – за счет свойств схемы проверки.

Как было отмечено, ДСЧ будем рассматривать как источник двоичных данных. Следовательно, распределение вероятностей появления последовательностей на выходе ДСЧ, подчиняется биномиальному закону распределения [13]. Идеальным ДСЧ будет тот, у которого вероятности появления "0" и "1" на выходе равны между собой. На практике построить такой ДСЧ является задачей, требующей сложных технических решений [8, 14]. Поэтому реально применяются ДСЧ асимметрия символов "0" и "1" на выходе которых в общем случае равняется ε . Это означает, что вероятность появления символа "0" равняется $0,5 - \varepsilon$, а "1" – $0,5 + \varepsilon$. Очевидно, что $-0,5 \leq \varepsilon \leq +0,5$. Другим словами, когда $\varepsilon < 0$, преобладают нули, когда $\varepsilon > 0$, единицы.

Сформируем задачу следующим образом. Пусть ГСП формирует последовательность независимых символов $R = \{r_i\}$, $r_i \in \{0, 1\}$, $i = \overline{1, n}$ где n – длина последовательности. Символ r_i может принимать значения либо "0", либо "1" в соответствии с вероятностью P_0 и $P_1 = 1 - P_0$. При этом, в общем случае, $P_1 \neq P_0$, $P_1 + P_0 = 1$. Кроме того, $|P_0 - P_1| = 2\varepsilon$.

Необходимо определить значение коэффициента эффективности атаки "перебор" для криптоалгоритма, применяющего ключи длиной n , когда асимметрия появления "0" и "1" на выходе ДСЧ равняется ε . Решение данной задачи позволит определить влияние асимметрии ε на вероятность появления ключей и, как следствие, на снижение стойкости криптографической системы, а также позволит определить допустимые значения ε для разных длин ключей n .

Для рассматриваемой задачи вероятность события, что ДСЧ за n тактов сформирует i символов "0" и $n - i$ символов "1", можно определить как

$$P(i) = C_n^i p_0^i (1 - p_0)^{n-i}, \quad i = \overline{0, n}. \quad (3)$$

В качестве эталонных можно взять значения $P(i)$ для случая, когда $\varepsilon = 0$, т.е. когда вероятности появления символов "0" и "1" равны. В этом случае

$$P^0(i) = C_n^i 0,5^i 0,5^{n-i} = C_n^i 2^{-n}, \quad i = \overline{0, n}. \quad (4)$$

Для случая, когда $\varepsilon \neq 0$ выражение (3) можно представить в виде

$$P^\varepsilon(i) = C_n^i (0,5 - \varepsilon)^i (1 - 0,5 + \varepsilon)^{n-i}, \quad i = \overline{0, n}. \quad (5)$$

Сравнивая выражения (4) и (3), можно определить степень отклонения $P_{\varepsilon}(i)$ от вероятного распределения. Например, в качестве меры отклонения можно взять неравномерность отклонения:

$$|\Delta P_{(i)}| = |P^0(i) - P^{\varepsilon}(i)| = \left| C_n^i \left\{ 0,5^i \cdot 0,5^{n-i} - (0,5 - \varepsilon)^i (1 - 0,5 + \varepsilon)^{n-i} \right\} \right|, \quad i = \overline{0, n}, \quad (6)$$

или

$$\Delta P_{(i)} = P^0(i) - P^{\varepsilon}(i) = C_n^i \left\{ 0,5^i \cdot 0,5^{n-i} - (0,5 - \varepsilon)^i (1 - 0,5 + \varepsilon)^{n-i} \right\}, \quad i = \overline{0, n}. \quad (7)$$

Зная $|\Delta P_{(i)}|$ или $\Delta P_{(i)}$ ряд, в качестве критерия оценки ГСП можно взять критерии:

$$\ln |\Delta P_{(i)}| \text{ и } \max |\Delta P_{(i)}| < P_{\text{дон}} \text{ для всех } i = \overline{0, n} \quad (8)$$

Более грубым подходом определения влияния ε отклонения на неравномерность появления ключей является следующий:

Разобьем интервал $[0, n]$ на три - $\left[0, \frac{n}{2} - 1\right]$, $\left[\frac{n}{2}\right]$ и $\left[\frac{n}{2} + 1, n\right]$, при этом интервал указывает на количество нулей в ключе, который находится в интервале. Далее определим вероятность события, что в ключе соответственно находится:

- 1) не более чем $\frac{n}{2} - 1$ символов "0";
- 2) точно $\frac{n}{2}$ символов "0" и $\frac{n}{2}$ символов "1";
- 3) не менее чем $\frac{n}{2} + 1$ символов "0".

В первом случае

$$P_1 = P\left(i \leq \frac{n}{2} - 1\right) = \sum_{i=0}^{\frac{n}{2}-1} C_n^i P_0^i (1 - P_0)^{n-i}. \quad (9)$$

Во втором случае

$$P_2 = P\left(i = \frac{n}{2}\right) = C_n^{\frac{n}{2}} P_0^{\frac{n}{2}} (1 - P_0)^{\frac{n}{2}}. \quad (10)$$

В третьем случае

$$P_3 = P\left(i \geq \frac{n}{2} + 1\right) = \sum_{i=\frac{n}{2}+1}^n C_n^i P_0^i (1 - P_0)^{n-i}. \quad (11)$$

Учитывая, что для случая, когда $\varepsilon=0$, значения . (9) и . (11) являются одинаковыми, в качестве показателя качества ГСП можно выбрать степень асимметрии:

$$\Delta P_{\pm} = \left| P\left(i \leq \frac{n}{2} - 1\right) - P\left(i \geq \frac{n}{2} + 1\right) \right|. \quad (12)$$

В таком случае идеальным ГСП можно считать тот, у которого $\Delta P_{\pm}=0$.

Полученные оценки дают возможность определить степень отклонения вероятностей появления ключей в зависимости от наличия асимметрии на выходе ДСЧ, но не учитывают реальную размерность ключевого пространства и не дают возможность оценить коэффициенты эффективности атаки типа "перебор".

Введем понятие ключевой набор K_l , под которым будем понимать множество ключей длиной n , в котором находится l единиц:

$$K_l = C_n^l, l = \overline{0, n}. \quad (13)$$

Очевидно, что график распределения величины K_l будет "пропорционален" графику распределения плотности вероятности P для биномиального распределения при $p=0.5$:

$$P = \sum_{l=0}^n P_l = \sum_{l=0}^n C_n^l p^l (1-p)^{n-l}, \quad (14)$$

т.е. при $p=0.5$ максимальным значениям K_l будет соответствовать максимальное значение вероятности P_l – вероятности появления ключей l -го ключевого набора. Обозначим через P_l^0 значения вероятности появления ключей из l -го ключевого набора при $p=0.5$.

Введем характеристику "эффективное ключевое пространство", определяемую как

$$A = \frac{\sum_{l=0}^n P_l K_l}{\sum_{l=0}^n P_l^0 K_l}, \quad (15)$$

где P_l – вероятность появления ключей из l -го ключевого набора.

Очевидно, что максимум $A=1$ достигается при $p=0.5$, т.е. при $\varepsilon = 0$. Данную характеристику можно интерпретировать как эффективность использования реального ключевого пространства, т.е. самому "емкому" ключевому набору соответствует самая большая вероятность появления ключей из данного набора. Такое распределение вероятности обеспечивает полный, исчерпывающий перебор всех ключевых комбинаций из всех ключевых наборов.

Предположим, что на выходе генератора существует некоторая асимметрия (в частном случае преобладают единицы). Обозначим через P_l^ε – значения вероятности появления ключевых наборов при $\varepsilon > 0$. При этом график плотности распределения вероятностей сместится вправо. Очевидно, что максимум вероятности теперь будет совпадать с ключевыми наборами, для которых значение K_l не является максимальным. В то же время ключевым наборам с максимальным значением K_l соответствует меньшее значение вероятности. Распределение плотностей вероятности для различных значений асимметрии представлено на рис. 1.

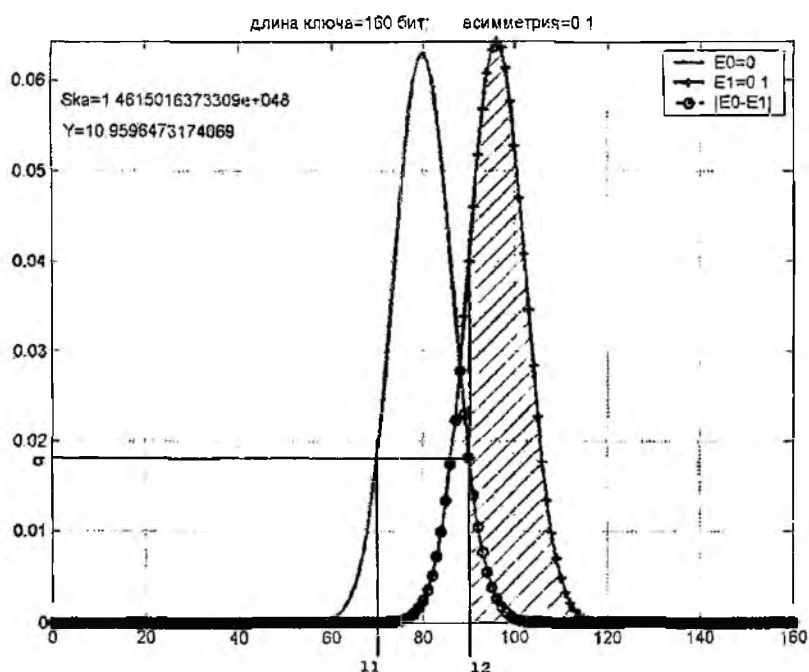


Рис. 1

Такое распределение вероятностей, с одной стороны, не обеспечивает исчерпывающего выбора ключевых комбинаций в случаях, в которых $P_l^\varepsilon < P_l^0$. Это приводит к уменьшению реального ключевого пространства и, как следствие, к упрощению криптографической атаки "перебор". С другой стороны, увеличение вероятностей появления ключей из наборов с малыми значениями K_l обеспечивает многократный выбор ключей из этих наборов. На практике это приводит к тому, что отдельные ключи будут появляться чаще, чем другие, что в свою очередь приводит к увеличению вероятности того, что один и тот же ключ будет применен несколько раз, что также приводит к уменьшению сложности атаки "перебор".

Для оценки уменьшения сложности данной атаки при наличии асимметрии рассмотрим распределение вероятностей P_l^Δ , которое определяется как

$$P_l^\Delta = \begin{cases} P_l^0, & P_l^0 \leq P_l^\varepsilon \\ P_l^\varepsilon, & P_l^0 > P_l^\varepsilon \end{cases} \quad (16)$$

Данное распределение вероятностей определяет эффективное ключевое пространство для случая $\varepsilon \neq 0$ и по сути показывает, какая часть общего ключевого пространства используется реально:

$$A_\Delta = \frac{\sum_{l=0}^n P_l^\Delta K_l}{\sum_{l=0}^n P_l^0 K_l} \quad (17)$$

Для оценки степени повторного выбора ключей введем коэффициент повторного выбора, который определяется как

$$K_p = \frac{A_\varepsilon}{A_\Delta} \quad (18)$$

где

$$A_\varepsilon = \frac{\sum_{l=0}^n P_l^\varepsilon K_l}{\sum_{l=0}^n P_l^0 K_l} \quad (19)$$

Данный коэффициент показывает, во сколько раз ключи из ключевых наборов с малыми значениями K_l будут появляться чаще, чем должны появляться при $\varepsilon = 0$.

Основываясь на вышеизложенном материале, коэффициент эффективности атаки "перебор" можно представить как

$$\Upsilon_0 = \frac{K_p}{A_\Delta} = \frac{A_\varepsilon}{(A_\Delta)^2} \quad (20)$$

Анализ выражения (20) показывает, что существенную роль в уменьшении сложности криптографической атаки играет размерность эффективного ключевого пространства

Рассмотрим влияние фильтров на данную оценку. Применение фильтров, отбраковывающих плохие ключи, обусловлено тем, что наличие слабых или запрещенных ключей может привести к вскрытию системы при помощи атак, не связанных с перебором ключей. Очевидно, что применение фильтра, с одной стороны, является необходимым для исключения применения запрещенных ключей, с другой – уменьшает ключевое пространство S_{kf} .

$$0 \leq S_{kf} \leq S_{na} \quad (21)$$

В соответствии с (1) емкость ключевого пространства при применении фильтра можно представить как

$$S_{\text{эф}} = \sum_{l=0}^n K_l = \sum_{l=0}^n C_n^l \xi_l \mid \forall (0 \leq \xi_l \leq 1), \quad (22)$$

где ξ_l – коэффициент соответствия требованиям ключей l -го набора.

Для упрощения рассмотрения введем модель фильтра, которая определяется как

$$\forall_{l=0}^n K_l = \begin{cases} K_l \xi_l, P_l^0 \leq \sigma \\ K_l \xi_l, P_l^0 > \sigma \end{cases}, \quad (23)$$

где σ – пороговое значение вероятности появления ключей или, другими словами, порог чувствительности фильтра.

Данная модель позволяет ввести три базовые модели фильтров:

1. Широкополосный: $\sigma=0$; $\xi_1=0$; $0<\xi_2<1$. Когда из каждого ключевого набора отбраковывается определенное количество ключей.
2. Узкополосный: $0<\sigma<1$; $\xi_1=0$; $\xi_2=1$. Когда ключи из наборов, вероятность появления которых достаточно мала, отбрасываются полностью.
3. Комбинированный: $0<\sigma<1$; $\xi_1=0$; $0<\xi_2<1$. Когда ключи из наборов, вероятность появления которых достаточно мала, отбрасываются полностью, а для наборов, вероятность появления которых больше заданной, отбраковывается определенное количество ключей.

Следует заметить, что в криптографических приложениях чаще всего применяется комбинированный тип фильтра.

Рассматривая варианты применения фильтров, особый интерес вызывают узкополосный и комбинированный фильтры, т.к. их применение позволяет избежать больших значений коэффициента повторного выбора ключей. На рис. 1 видно, что при наличии асимметрии применение узкополосного фильтра при $\sigma=0,018$ позволяет отсечь все ключи, которые не попадают в интервал (l_1, l_2) , а следовательно, отсечь ключи для которых высока вероятность повторного выбора. Очевидно, что подбором порогового уровня для такого фильтра можно добиться минимального значения коэффициента эффективности атаки "перебор".

Таким образом, с учетом (22) выражения (17) и (19) для расчета коэффициента эффективности криптографической атаки (20) можно представить в виде

$$A_{\Delta} = \frac{\sum_{l=0}^n P_l^{\Delta} K_l \xi_l}{\sum_{l=0}^n P_l^0 K_l}, \quad (24)$$

$$A_{\varepsilon} = \frac{\sum_{l=0}^n P_l^{\varepsilon} K_l \xi_l}{\sum_{l=0}^n P_l^0 K_l}. \quad (25)$$

В табл. 1, 2 приведены значения коэффициента эффективности атаки "перебор" в зависимости от длин ключей, асимметрии на выходе ДСЧ и типа применяемых фильтров.

γ_0 при $\varepsilon = 0.001$

Длина ключа n	Без фильтра $\xi_1=0; \xi_2=1; \sigma=0$	Широкополосный $\xi_1=0; \xi_2=0,7; \sigma=0$	Узкополосный $\xi_1=0; \xi_2=1; \sigma=0,001$	комбинированный $\xi_1=0; \xi_2=0,7; \sigma=0,001$
16	1.00443249294448	1.43490356134926	1.00443333154713	1.43490475935305
32	1.00635128119841	1.4376446874263	1.00635426871539	1.43766324102199
48	1.00781756356374	1.43973937651963	1.00783042366905	1.43975774809865
64	1.00905278677017	1.44150398110024	1.00906013141412	1.44151447344874
80	1.01014125759303	1.44305893941861	1.0101570438999	1.44308149128557
96	1.01112585777779	1.44446551111113	1.01114964762704	1.44449949661006
112	1.01203193443289	1.4457599063327	1.0120613374907	1.445801910701
128	1.0128759471553	1.44696563879328	1.01290810709849	1.44701158156927
144	1.01366930104406	1.44809900149152	1.01370174915325	1.44814535593321
160	1.0144202847761	1.44917183539443	1.01445123829567	1.4492160547081

Таблица 2

- γ_0 при $\varepsilon = 0.1$

Длина ключа n	без фильтра $\xi_1=0; \xi_2=1; \sigma=0$	Широкополосный $\xi_1=0; \xi_2=0,7; \sigma=0$	узкополосный $\xi_1=0; \xi_2=1; \sigma=0,001$	комбинированный $\xi_1=0; \xi_2=0,7; \sigma=0,001$
16	1.70701040383677	2.43858629119539	1.70700197052057	2.43857424360082
32	2.25065812831035	3.21522589758621	2.25028314740016	3.21469021057166
48	2.83565419191044	4.05093455987206	2.83467322698997	4.04953318141425
64	3.49258449750409	4.98940642500585	3.49117805948981	4.98739722784259
80	4.24175482680365	6.05964975257664	4.23685144175642	6.05264491679488
96	5.2113991609037	7.44485594414815	5.19931563752177	7.42759376788824
112	6.35090989271921	9.0727284181703	6.32675680045669	9.03822400065241
128	7.6689610847678	10.9556586925254	7.62674144939085	10.8953449277012
144	9.19417505047016	13.1345357863859	9.12675265891416	13.0382180841631
160	10.9596473174069	15.6566390248671	10.8586587464963	15.5123696378519

Анализ табл. 2 показывает, что для данного соотношения уровня асимметрии и порогового значения вероятности появления ключей при применении узкополосного фильтра имеет место уменьшение эффективности атаки "перебор" по сравнению со случаем отсутствия фильтров. На рис. 2 показаны зависимости коэффициента эффективности атаки "перебор" от значения порога чувствительности фильтра при $\varepsilon = 0.1$ для различных длин ключей. Из анализа рис. 2 следует что:

- коэффициент упрощения атаки типа "перебор" имеет нелинейную зависимость от уровня порога чувствительности узкополосного фильтра;

- существуют такие значения порога чувствительности узкополосного фильтра, при которых значение коэффициента эффективности атаки типа "перебор" достигает своего минимума. Данное значение порога чувствительности фильтра будем называть оптимальным порогом чувствительности σ_0 ;

- для различных значений длин последовательностей значения оптимального порога чувствительности различны.

Исходя из анализа рис. 1 и 2 и выражений (5), (13)-(20) следует, что значения оптимального порога чувствительности зависят не только от длины последовательностей, но и от асимметрии на выходе ДСЧ. Зависимости оптимального порога чувствительности фильтра от значения асимметрии на выходе ДСЧ для различных длин последовательностей приведены на рис. 3.

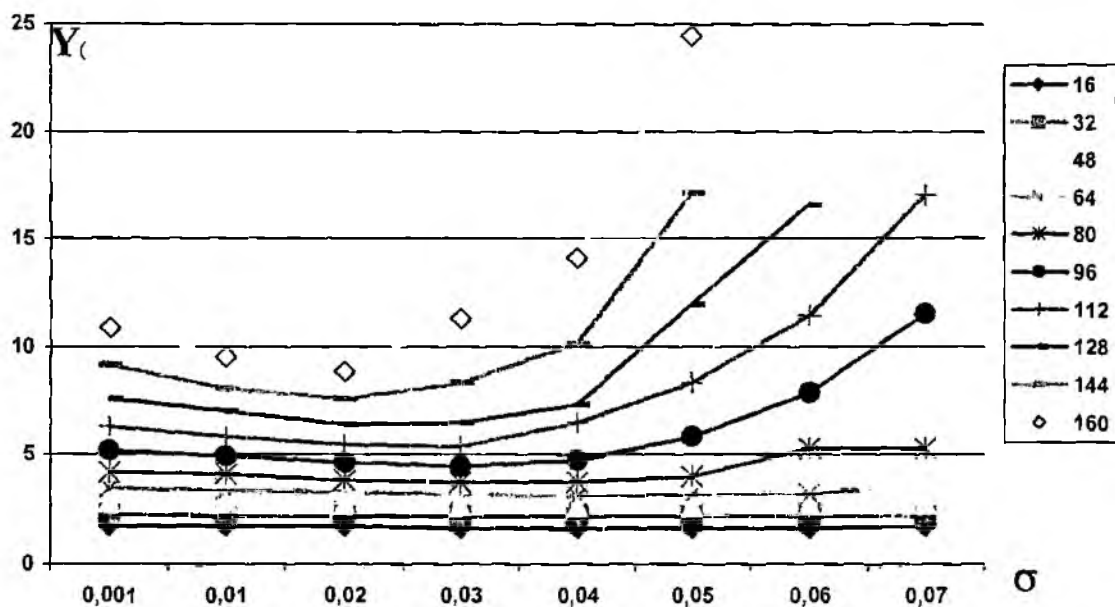


Рис.2

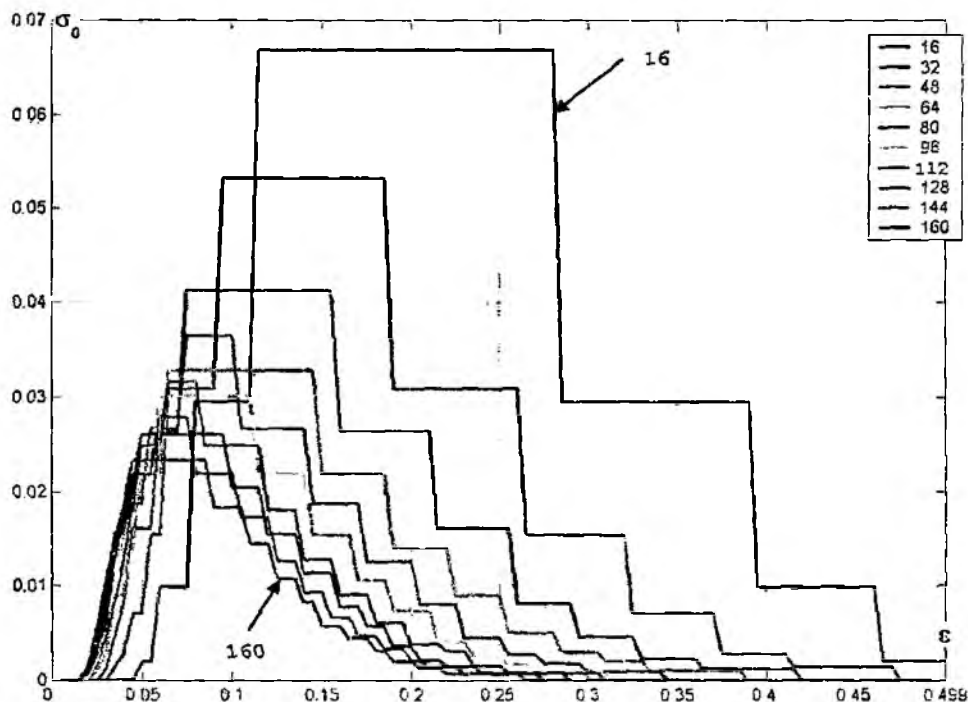


Рис. 3

Характер поведения графиков, приведенных на рис. 3, можно объяснить следующим образом. При увеличении асимметрии на выходе ДСЧ максимум плотности распределения вероятности начинает смещаться вправо. При этом значение вероятности появления ключей наиболее быстро растет в крайних точках. При определенном значении вероятности появления ключей из крайних наборов их отсекание за счет применения узкополосного фильтра приводит к уменьшению повторного выбора ключей.

С дальнейшим ростом асимметрии вероятность повторного выбора ключей постепенно увеличивается в точках, лежащих ближе к центру. Поэтому постепенно сужается ширина полосы пропускания фильтра, т.е. увеличивается значение σ_0 .

Увеличение асимметрии приводит к тому, что при определенном значении ϵ отгибающая плотности распределения вероятности становится четко выделенным, отделенным от стандартного "колоколом", который при дальнейшем увеличении асимметрии смещается в крайнюю точку. При этом узкополосный фильтр должен отсекал практически весь смещенный "колокол", а следовательно, постепенно расширяться, т.е. значение σ постепенно должно уменьшаться.

В качестве иллюстрации рассмотрим кривую, соответствующую длине ключа 160 бит (кривая 160, рис. 3). При значении асимметрии $\epsilon = 0.1$ уровень оптимального порога чувствительности фильтра равняется $\sigma_0 = 0.018$. На рис. 1, на котором приведены распределение плотностей вероятности для ключей длиной 160 бит без асимметрии (кривая $E_0=0$) и при наличии асимметрии, равной 0.1 (кривая $E_1=0.1$). линиями, соответствующими 11 и 12, показана ширина полосы пропускания узкополосного фильтра, пороговое значение которого равняется $\sigma = 0.018$. Как видно из рис. 1, данный фильтр практически полностью отсекает область, в которой находятся ключевые наборы, для которых вероятность повторного выбора ключей достаточно велика. При этом $\gamma_0 = 8.91$, в то время как при отсутствии фильтра $\gamma_0 = 10.96$.

Таким образом, обобщенный коэффициент эффективности криптографической атаки предложенный для оценки требований, выдвигаемых к генераторам случайных последовательностей, позволяет: 1) Дать численную оценку качества требований, выдвигаемых к ГСП, с учетом технических характеристик конкретного ДСЧ. 2) Сравнить между собой различные варианты требований, выдвигаемых к ГСП. 3) Показать реальное изменение свойств ГСП в зависимости от изменения набора требований к нему. 4) Сформировать и обосновать требования к ДСЧ.

Список литературы: 1. Чмора А.Л. Современная прикладная криптография. М.: Гелиос АРВ, 2001. 244 с. 2. Введение в криптографию / Под ред. В.В. Яценко СПб.: Питер, 2001. 288 с. ISBN 5-318-00443-1. 3. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учеб. пособие. М.: Гелиос АРВ, 2001. 480 с. 4. Бабаи А.В., Шанкин Г.П. Криптография. М.: СОЛОН-Р, 2002. 512 с. 5. Анохин М.И., Варновский Н.П., Сидельников В.М., Яценко В.В. Криптография в банковском деле. М.: Изд-во МИФИ, 1997. 6. Потий А.В., Пестерев А.К. Принципы системного подхода к сертификации генераторов псевдослучайных чисел в системах защиты информации // Радиотехника. 1997. Вып. 104. С. 163-172. 7. Потий А.В., Попович Е.В., Горбенко Ю.И., Вервейко В. Программно-аппаратный комплекс формирования и исследования случайных и псевдослучайных чисел // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні: Наук.-техн. збірник. Вип. 2. Київ, 2001. С. 171-175. 8. Торба А.А., Елаков С.Г., Степченко А.З. Генерация равновероятных случайных последовательностей на основе физических датчиков // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С. 108-113. 9. COMPUTER SECURITY. A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications. NIST Special Publication 800-22, 2000. 10. FIPS PUB 140-2 Security Requirements For Cryptographic Modules. Federal Information Processing Standard Publication 140-2 1999. 11. FIPS PUB 197 Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197 November 26, 2001. 12. Потий А.В., Пестерев А.К., Олейников Р.В. Декомпозиция требований, предъявляемых к генераторам случайных и псевдослучайных чисел, на основе классификации специальных данных // Ювілейна науково-технічна конференція. // Правове нормативне та метрологічне забезпечення системи захисту інформації в Україні. Вип. Київ 9-11 червня 1998. 13. Ширяев А.Н. Вероятность. М.: Наука, 1980. 581 с. 14. Торба А.А., Бобух В.А., Елаков С.Г., Степченко А.З. Повышение быстродействия генераторов случайных последовательностей на основе физических датчиков // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2002. Вып. 126. С. 218-221.

Департамент специальных
телекоммуникационных
систем и защиты информации,
Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 09.11.2004

ФАЗОВІ СИСТЕМИ ПЕРЕДАЧІ ІНФОРМАЦІЇ НА ОСНОВІ ЧИСЕЛЬНОЇ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

Сучасні технічні системи та комплекси мають розподілені в просторі структури, елементи яких об'єднані за допомогою каналів зв'язку. В таких системах виникають задачі передачі з високою вірогідністю великих об'ємів інформації. Для передачі даних застосовують цифрові канали, в яких підвищення завадостійкості та вірогідності досягається використанням спеціальних способів кодування. Одним з таких способів є представлення інформації в числовій системі залишкових класів (СЗК) [1].

Одна з основних властивостей СЗК полягає в тому, що всі цілі числа в цій системі відображаються множиною невід'ємних лишків від ділення чисел на інші певні цілі числа (модулі системи), які утворюють групу взаємнопростих чисел. Ця група чисел дістала назву основи СЗК. До основних властивостей СЗК належать наступні: 1) відсутність зв'язків між суміжними розрядами кодового сигналу, 2) можливість організації ефективної системи виявлення та виправлення помилок при обробці і передачі цифрових даних. Зазначені вище властивості СЗК роблять її досить привабливою для використання у обчислювальних системах підвищеної надійності і швидкодії. Значна кількість робіт з тематики СЗК пов'язана саме з вирішенням задач такого роду. В той же час зазначені вище та інші характеристики і властивості СЗК можуть бути запропоновані для використання в каналах зв'язку як спосіб представлення, відновлення та передачі інформації.

В статті наведені результати досліджень із застосування СЗК в цифрових системах передачі даних при комплексному вирішенні ряду актуальних питань передачі інформації, а саме: кодування цифрових сигналів (шифрування повідомлення), виявлення і виправлення помилок при передачі даних, синхронізація прийому сигнальних посилок та інші.

Постановка задачі.

Задано числову інформацію, що підлягає кодуванню та передачі, подану цілим додатним числом $A < A_p$, де A_p – границя робочого діапазону подання цілих чисел. Кодування інформації відбувається в СЗК. Передбачається, що при передачі та прийомі повідомлення відомий ключ кодування, яким є відповідна система модулів.

Необхідно проаналізувати використання СЗК для комплексного вирішення побудови каналу передачі даних на фізичному і каналному рівнях.

Розв'язок

Наведемо результати поетапного розв'язку задачі.

1. *Кодування інформації.* Спочатку для розуміння суті процесів кодування і відновлення повідомлення наведемо основні властивості СЗК. З теорії чисел відомо [1], що задача відновлення цілого числа A в позиційній системі числення з його подання лишками $\alpha_i = A(\bmod n_i)$, $i = \overline{1, m}$ за заданою системою модулів $(n_1, \dots, n_i, \dots, n_m)$ має однозначний розв'язок у випадку, якщо:

1) модулі системи – взаємно прості числа, тобто найменший загальний дільник

$$(n_1, \dots, n_i, \dots, n_m) = 1; \quad (1)$$

2) максимальне відновлюване число $A_{\max}$ задовільняє умові

$$A_{\max} < \prod_{i=1}^m n_i = A_p. \quad (2)$$

Ці вимоги забезпечуються вибором відповідних значень чисел n_i та m .

У разі виконання умов (1), (2) повідомлення відновлюється з його представлення лишками за формулою:

$$A = \sum_{i=1}^m a_i B_i \pmod{A_p}, \quad (3)$$

де $(B_1, \dots, B_i, \dots, B_m)$ – система ортонормованих базисів, яка однозначно визначається обраною системою модулів; B_i – ціле число, яке ділиться без залишку на всі модулі, крім i -го, ділення на який дає залишок, що дорівнює 1, тобто

$$\begin{aligned} \frac{B_i}{n_i} &\equiv 1 \pmod{n_i}, \quad i = \overline{1, m}; \\ \frac{B_i}{n_j} &\equiv 0 \pmod{n_j}, \quad i \neq j, \quad j, i = \overline{1, m}. \end{aligned} \quad (4)$$

Модулі та базиси СЗК входять до системи кодування як ключі. Базиси системи повністю визначаються її модулями і розраховуються до відновлення повідомлення.

Отже, повідомлення передається одержувачу у вигляді послідовності лишків цього числа – $\alpha_1, \dots, \alpha_i, \dots, \alpha_m$. Відновити число A безпосередньо з величин α_i , $i = \overline{1, m}$ без спеціальної обробки неможливо. Ключем до відновлення A є обрана система лишків, тобто їх значення, кількість і порядкове місце кожного модуля в системі модулів. Ця система і становить основу шифрування даних і ускладнює несанкціонований доступ до інформаційного повідомлення.

З метою підвищення скритності передачі операцію подання числа лишками можна застосовувати декілька разів (повторно), тобто лишки α_i , $i = \overline{1, m}$ також можна подавати лишками в іншій системі модулів $n_1, \dots, n_j, \dots, n_l$.

Приклади кодування і відновлення повідомлень з використанням СЗК наведені в [2].

2. *Виявлення і виправлення помилок при передачі.* Застосування СЗК дозволяє виконати пошук і виправлення невірних лишків. Такий контроль може бути виконаний шляхом передачі допоміжних лишків переданого числового повідомлення, які визначаються за додатковими модулями.

Теоретичним підґрунтям, яке доводить властивості кодів в СЗК до самокоригування, є ряд теорем, розглянутих і доведених в [1]. Основна ідея виявлення помилок полягає в тому, що модулі СЗК доповнюються додатковим (одним чи декількома) модулем $n_{m+1} > n_i$, $i \in [1, m]$ взаємнопростим з довільним модулем n_i , $i \in [1, m]$. Надалі будемо називати: робочим

діапазоном – інтервал подання чисел $[0, M_p)$, обмежений значенням $M_p = \prod_{i=1}^m n_i$; цілі числа

з інтервалу $[0, M_p)$ – правильними числами; інтервал $[0, M_n)$, обмежений значенням

$M_n = \prod_{i=1}^{m+1} n_i$, – повним діапазоном; цілі числа з інтервалу $[M_p, M_n)$ – неправильними.

Довільне викривлення одного з лишків у поданому в СЗК числі $A < M_p$ при відновленні в позиційній системі переводить його в інтервал $[M_p, M_n)$. При викривленні k -го лишку існує лише одне єдине значення лишку α_k , яке повертає відновлене в позиційній системі число A в робочий діапазон $[0, M_p)$. Якщо ж $A_s < M_p$, то або помилка відсутня, або вона має складніший характер, що перешкоджає її виявленню.

Структурна схема послідовності перетворення, передачі і відновлення повідомлення із застосуванням СЗК наведена на рис.1.

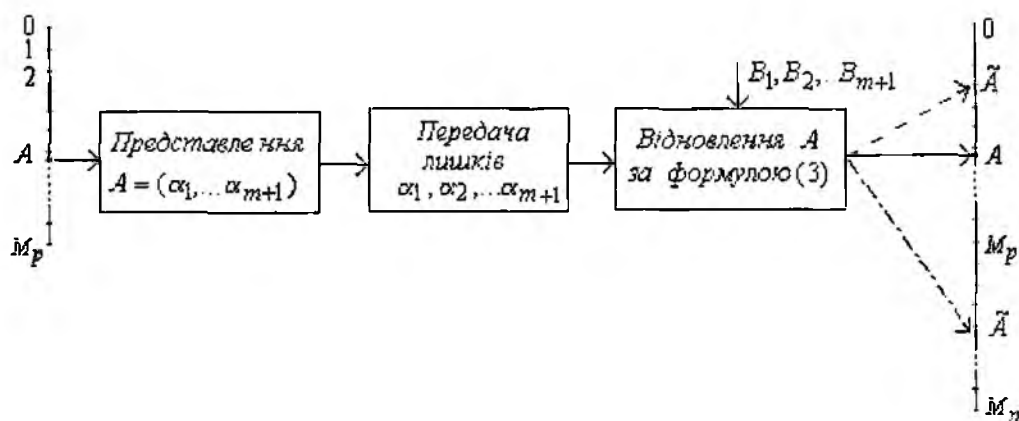


Рис. 1

Виправлення помилок виконується різними способами: нуль-способом, способом проєкцій та способом множення на числа, кратні модулям СЗК, які розглянуті в [1,3]. Слід відзначити, що за здатністю коригувати помилки СЗК у загальному випадку поступається циклічним кодам [4].

Нуль-спосіб полягає у послідовному, розряд за розрядом, переході від вихідного подання числа $(\alpha_1, \dots, \alpha_m, \alpha_{m+1})$ до числа $(0, \dots, \alpha_m, \alpha_{m+1})$, кратного n_1 , потім до числа $(0, 0, \dots, \gamma_{m+1})$, кратного $n_1 n_2$ і т.д. за допомогою послідовного віднімання певних чисел, що називаються константами нуль-способу. Ці константи обираються як найменші можливі для виконання даної операції, числа. Виконання цієї умови необхідно для того, щоб результат не вийшов за границю робочого діапазону. Загальне число констант дорівнює $\sum_{i=1}^m n_i - m$.

Результат перетворень не залежить від порядку застосування констант, і в той же час дає можливість визначити набір можливих помилок $\Delta \alpha_i, i = \overline{1, m}$, які призводять до певного значення γ_{m+1} .

На рис.2. графічно і таблично проілюстровано суть нуль-способу, тобто показаний розподіл помилок залежно від значень γ_{m+1} .

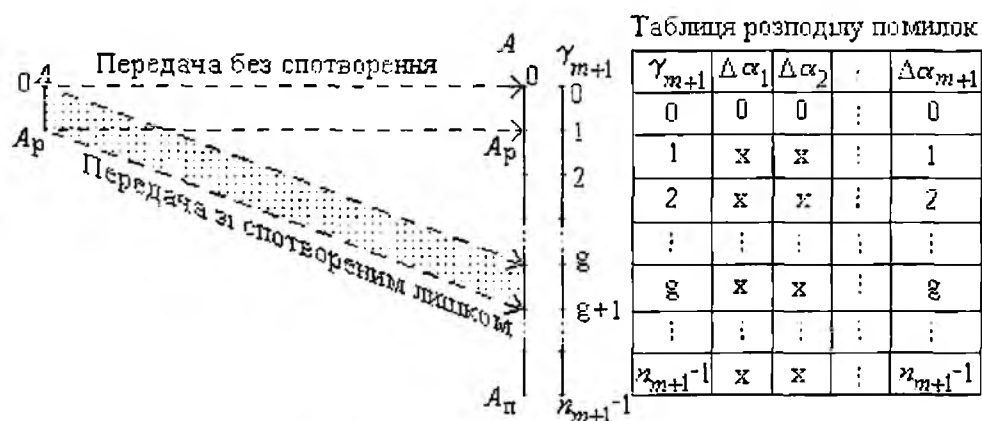


Рис. 2

За відсутності помилки $A \in [0, A_p)$ такій події відповідає значення $\gamma_{m+1} = 0$. Кожна з можливих помилок переводить правильне число A в неправильне $\tilde{A}$, яке належить до одного з інтервалів: $[A_p, 2A_p)$, $[2A_p, 3A_p)$, ..., $[mA_p, (m+1)A_p)$. Номер інтервалу співпадає з γ_{m+1} , що формується за допомогою нуль-способу і дозволяє визначити сукупність модулів з помилками, а також величину цих помилок. Отримане число γ_{m+1} визначає набір можливих помилок, які також розраховуються на етапі проєктування і зберігаються в системі у вигляді таблиці.

В тих випадках, коли нуль-спосіб не дає однозначної відповіді на запитання, який лишок в $A_{CЗК}$ є викривленим, необхідно виконати додаткову обробку одним із наступних способів: способом проєкцій або способом множення на числа, кратні модулям СЗК [1,3].

Розглянемо приклад застосування нуль-способу.

П р и к л а д 1. Дана система модулів $n_1=5$, $n_2=6$, $n_3=7$, $n_4=11$ з робочим діапазоном $A_p = n_1 n_2 n_3 = 210$ і повним діапазоном $A_n = n_4 A_p = 2310$. Ортонормовані базиси для системи модулів (5,6,7,11): $B_1 = 1386$, $B_2 = 385$, $B_3 = 330$, $B_4 = 210$. Приймається число $A = 113$. Замість правильного подання $A_{CЗК} = (3,5,1,3)$ цього числа прийнято $\tilde{A}_{CЗК} = (3,5,6,3)$ з помилкою $\Delta\alpha_3 = 5$ в третьому лишку.

При відновленні числа без коригування маємо:

$$\hat{A} = (3 \cdot 1386 + 5 \cdot 385 + 6 \cdot 330 + 3 \cdot 210) \bmod 2310 = 1763 > A_p.$$

Необхідно знайти і виправити помилку, а також відновити вірне значення числа A .

Розраховані за формулою (5) ваги ортонормованих базисів складають: $m_1=3$, $m_2=1$, $m_3=1$, $m_4=1$.

Розраховані константи нуль-способу наведені в табл.1, а розподіл можливих помилок $\Delta\alpha_1, \Delta\alpha_2, \Delta\alpha_3, \Delta\alpha_4$, у відповідних лишках числа – в табл.2.

Таблиця 1

n_1		n_2		n_3	
A_{10}	$A_{CЗК}$	A_{10}	$A_{CЗК}$	A_{10}	$A_{CЗК}$
	1,1,1, 1	25	0,1,4,3	120	0,0,1,10
2	2,2,2, 2	20	0,2,6,9	30	0,0,2,8
3	3,3,3, 3	15	0,3,1,4	150	0,0,3,7
4	4,4,4, 4	10	0,4,3,10	60	0,0,4,5
		35	0,5,0,2	180	0,0,5,4
				90	0,0,6,2

Таблиця 2

γ_4	$\Delta\alpha_1$	$\Delta\alpha_2$	$\Delta\alpha_3$	$\Delta\alpha_4$
1	-	1	1	1
2	2	1	1	2
3	2	2	2	3
4	4	2	2,3	4
5	4	3	3	5
6	1	3	4	6
7	1	4	4,5	7
8	3	4	5	8
9	3	5	6	9
10	-	5	6	10

Табл.2 розраховувалась за формулою (4) для всіх можливих значень помилок $\Delta\alpha_i \in [1, n_i - 1]$, $i = \overline{1, m}$.

При складанні таблиці враховано, що кожному значенню γ_4 може відповідати помилка і по контрольному модулю n_4 , що дорівнює $\Delta\alpha_4 = \gamma_4$. Застосуємо нуль-спосіб до числа $\tilde{A}_{CЗК}$:

$$\begin{array}{r} (3,5,6,3) \\ -(3,3,3,3) \\ \hline (0,2,3,0) \\ -(0,2,6,9) \\ \hline (0,0,4,2) \\ -(0,0,4,5) \\ \hline (0,0,0,8) \end{array}$$

Отже $\gamma_4 = 8$. З табл.2 видно, що такий результат можна отримати в чотирьох випадках: 1) $\Delta\alpha_1=3$; 2) $\Delta\alpha_2=4$; 3) $\Delta\alpha_3=5$; 4) $\Delta\alpha_4=8$. Спробуємо зменшити кількість альтернатив шляхом обчислення оцінки вірного результату $\hat{A}$ і його порівняння з A_p :

- 1) $\hat{A}_{CЗК} = (3,5,6,3) - (3,0,0,0) = (0,5,6,3)$, відновлення дає $\hat{A} = 2225 > A_p$;
- 2) $\hat{A}_{CЗК} = (3,5,6,3) - (0,4,0,0) = (3,1,6,3)$, відновлення дає $\hat{A} = 223 > A_p$;
- 3) $\hat{A}_{CЗК} = (3,5,6,3) - (0,0,5,0) = (3,5,1,3)$, відновлення дає $\hat{A} = 113 < A_p$;
- 4) $\hat{A}_{CЗК} = (3,5,6,3) - (0,0,0,8) = (3,5,6,6)$, відновлення дає $\hat{A} = 83 < A_p$.

Таким чином, кількість можливих варіантів зменшилась до двох. Отже нуль-спосіб в загальному випадку не дає однозначної відповіді на запитання: який лишок в $A_{CЗК}$ є викривленим.

Для надійної локалізації викривленого лишку необхідно виконати додаткову обробку одним з інших способів: способом проекцій або способом множення на числа, кратні модулям СЗК [1].

Сутність способу проекцій полягає у формуванні проекцій числа, відновленні значень проекцій в позиційній системі числення і порівнянні результату з A_p . Якщо $A > A_p$, а проекція $A_i < A_p$, це свідчить про можливість помилки в i -му лишку. Ця властивість дозволяє за m обрахунків за формулою (3) визначити викривлені лишки $\tilde{\alpha}_i, i = \overline{1, m}$. При обрахунку A_i кожній модифікованій системі модулів $(n_2, n_3, \dots, n_{m+1}), (n_1, n_3, \dots, n_{m+1}), \dots, (n_1, n_2, \dots, n_m)$ відповідає певна система ортонормованих базисів з різними границями робочого діапазону A_p . Розглянемо застосування цього способу на наступному прикладі.

П р и к л а д 2. Для даних прикладу 1 розрахуємо проекції числа $\tilde{A}_{CЗК}$. Результати розрахунків зведені в табл. 3.

Порівняння чисел $\hat{A}$ з A_p дозволяє припустити можливість помилки в третьому чи четвертому лишках. Застосована в цьому прикладі перевірка також зменшила кількість варіантів до двох, але не дала чіткої вказівки на викривлений лишок. Сутність способу множення на числа, кратні модулям СЗК, полягає у множенні $A_{CЗК}$ на числа $K_i, i = \overline{1, m}$ такі, що $\frac{K_i}{n_i} \bmod n_i = 0$,

тобто мають в поданні СЗК нуль в i -му лишку. Тому добуток $A_{CЗК} K_i$ буде мати нуль в i -му лишку. Якщо в числі $A_{CЗК}$ лишок $\tilde{\alpha}_i$ був викривлений, то в добутку він "обнуляється" і результат повертається до робочого діапазону. Природно, що робочий діапазон в цьому випадку повинен бути розширений з урахуванням результату операції множення.

Таблиця 3

Модулі				Базиси	A_i	A_p	$\hat{A}$
n_1	n_2	n_3	n_4				
-	6	7	11	(385,330,210)	(5,6,3)	46	377
5	-	7	11	(231,330,210)	(3,6,3)	38	223
5	6	-	11	(66,55,210)	(3,5,3)	33	113
5	6	7	-	(126,175,120)	(3,5,6)	21	83

Це можна зробити введенням ще одного додаткового модуля n_{m+2} . Тоді повний діапазон буде обмежений значенням $n_{m+1}n_{m+2}A_p$, а розширений робочий діапазон – $n_{m+1}A_p$. Слід зазначити, що така система, окрім властивості виправлення однократних помилок, дозволяє виявляти дво- чи більшої кратності помилки у q лишках, для яких $\prod_q n_q < n_{m+1}n_{m+2}$.

Варто вказати ще на одну особливість виявлення помилок способом множення. Складні помилки (в двох і більше лишках) можуть маскувати одна одну і не виявляться перевіркою відновленого числа за ознакою $A < A_p$. Однак, множення подання числа $\tilde{A}_{СЗК}$ зі викривленими лишками, серед яких є i -й, на константу K_i з нулем в i -му лишку може вивести відновлене число з меншою кількістю помилок з робочого діапазону, що дозволить виявити цю складну помилку.

Розглянуті способи, в залежності від конкретної задачі передачі даних, можуть застосовуватись окремо чи в комбінації з іншими способами і дозволяють з високою вірогідністю [1] виявляти і виправляти складні помилки, хоча їх реалізація вимагає значних обчислювальних потужностей.

3. *Застосування фазоманіпульованих сигналів для передачі.* Для кодування повідомлення на фізичному рівні використовують один з відомих способів маніпуляції. Відомо [4], що багаторівнева фазова маніпуляція дозволяє ефективно використовувати частотну смугу пропускання каналу передачі і має переваги, що обумовлює її використання в цифрових системах передачі даних. Подання інформації в СЗК відповідає циклічній природі сигналів з фазовою маніпуляцією і добре узгоджується з цим видом маніпуляції. Особливості передачі кодів СЗК фазоманіпульованими сигналами розглядалися в роботі [5]. Коротко зупинимось на розгляді цих особливостей.

Лишки α_i , $i = \overline{1, m}$ при застосуванні синусоїдальних сигналів однієї несучої частоти передаються по лінії зв'язку послідовно в часі за допомогою фазової маніпуляції гармонічного коливання $u_0(t)$, $t \in [0, T_n)$. Кожному лишку α_i відповідає свій початковий фазовий зсув

$\varphi_i = 2\pi \frac{\alpha_i}{n_i}$, який використовують для модуляції $u_0(t)$. Розглянемо сигнал вигляду

$$u(t) = \sum_{i=1}^m U_0 \cos(2\pi f t - \varphi_i) [G(t - (i-1)T_n) - G(t - iT_n)], \quad t \in [0, T_n). \quad (5)$$

де $G(x)$ – функція Хевісайда; $T_n = \frac{T_n}{m}$ – інтервал часу передачі одного лишку, реалізація якого графічно зображена на рис.3, де показано інформаційний сигнал передачі $m+1$ лишку.

Сигнал (5) на кожному інтервалі передачі лишку α_i , $i = \overline{1, m}$ має дві складові, отримані за допомогою амплітудної модуляції косинусної і синусної несучих, тобто

$$\begin{aligned}
 u(t) &= U_0 \cos(2\pi ft - \varphi_i) = U_0 \cos(\varphi_i) \cos(2\pi ft) + U_0 \sin(\varphi_i) \sin(2\pi ft) = \\
 &= U_c \cos(2\pi ft) + U_s \sin(2\pi ft), \quad t \in [(i-1)T_{\text{л}}, T_{\text{л}}]
 \end{aligned}
 \tag{6}$$

де U_c, U_s – амплітуди косинусної і синусної компонент.

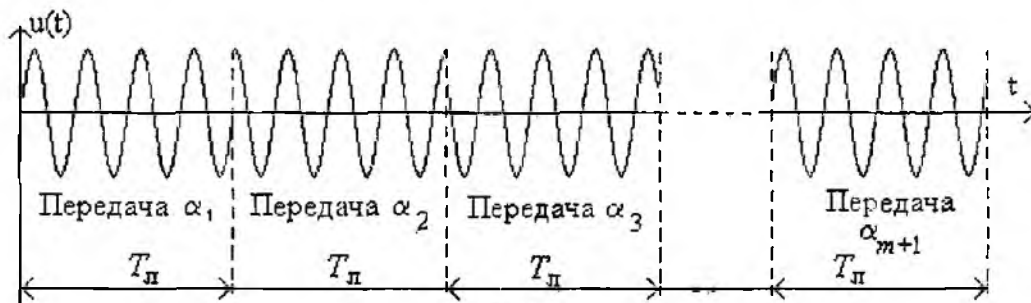


Рис.3

Графічне зображення таких сигналів подають у вигляді характерної геометричної побудови, яку називають *сигнальним сузір'ям*. Дві ортогональні осі сузір'я символізують два гармонічні коливання: вісь x – синусне, вісь y – косинусне). По осі x відкладають значення U_c , а по осі y – U_s . В такому поданні варіантам маніпульованого сигналу відповідають точки відповідного графіка. Три сигнальні сузір'я 38-, 41-, 43-позиційних фазоманіпульованих (ФМ) сигналів подано на рис.4: на рис.4, а зображено сузір'я для передачі першого лишку $\alpha_1 = 0,37$ сигналом ФМ-38, на рис.4, б – другого лишку $\alpha_2 = 0,40$ сигналом ФМ-41, а на рис.4, в – третього лишку $\alpha_3 = 0,42$ сигналом ФМ-43. На сузір'ях, як приклад, жирними лініями позначено вектори, які відповідають лишкам числа $A = 29652$.

Характерна ознака кодування СЗК полягає в тому, що кожний лишок передається фазоманіпульованим сигналом відмінної позиційності і з різними кутами $\Delta\varphi_i$ між суміжними сигналами для передачі різних лишків:

$$\Delta\varphi_i = \frac{2\pi}{n_i}, \quad i = \overline{1, m}.
 \tag{7}$$

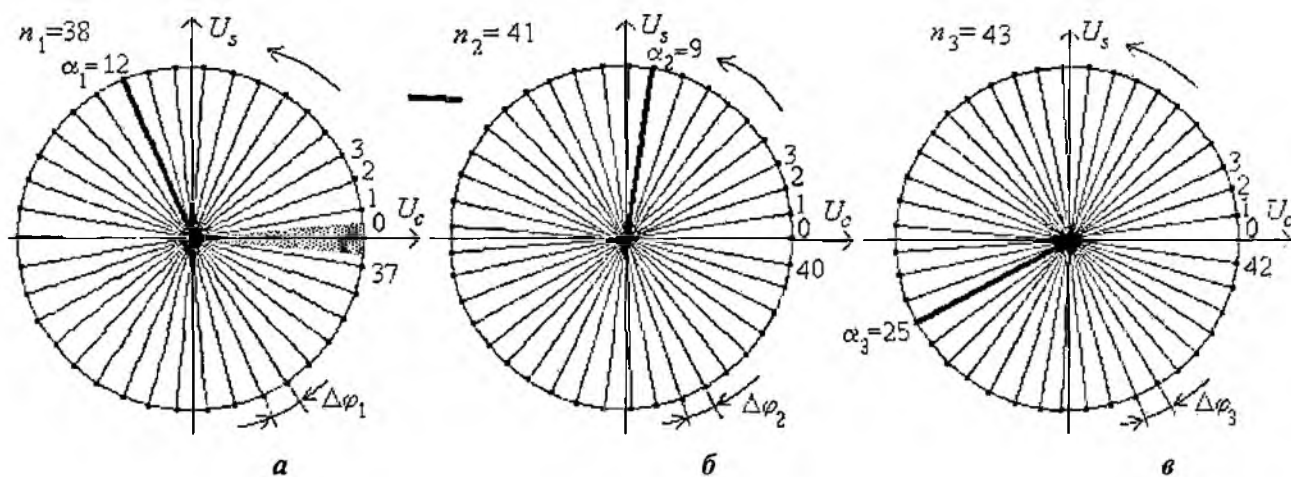


Рис. 4

Внаслідок дії в лінії передачі шумів і перешкод, а також шумів і апаратних похибок приймача, отриманий фазовий зсув буде відрізнятися від сформованого для передачі повідомлення. Як приклад, на рис.4, а заштриховано сектор в околі $\varphi = 0$, в якому може бути значення прийнятого фазового зсуву і належність до якого ідентифікується як передача лиш-

ку $\alpha_i = 0$. З метою зменшення ймовірності появи грубих помилок під час передавання повідомлення доцільно виконати зміщення лишків на величину 0,5, визначаючи фазові зсуви як

$$\varphi_i = 2\pi \frac{(\alpha_i + 0,5)}{n_i}, \quad i = \overline{1, m}. \quad (8)$$

Таке штучне зміщення зсуває значення переданих фазових зсувів в області найбільш віддалених від значень, кратних $\frac{2\pi}{n_i}$, $i = \overline{1, m}$, які розділяють два суміжні значення лишків.

Статистичні оцінки лишків на боці приймання повідомлення визначають за формулою

$$\hat{\alpha}_i = \left[\frac{\hat{\phi}_i}{\Delta\varphi_i} \right]^+ = \left[\frac{\hat{\phi}_i}{2\pi n_i} \right]^+, \quad i = \overline{1, m}, \quad (9)$$

де $[\cdot]^+$ – операція виділення цілої частини числа; $\hat{\phi}_i$ – значення i -го початкового вимірювання фазового зсуву.

Структурну схему каналу послідовного передавання інформації, в якому застосовано кодування в СЗК, зображено на рис.5.

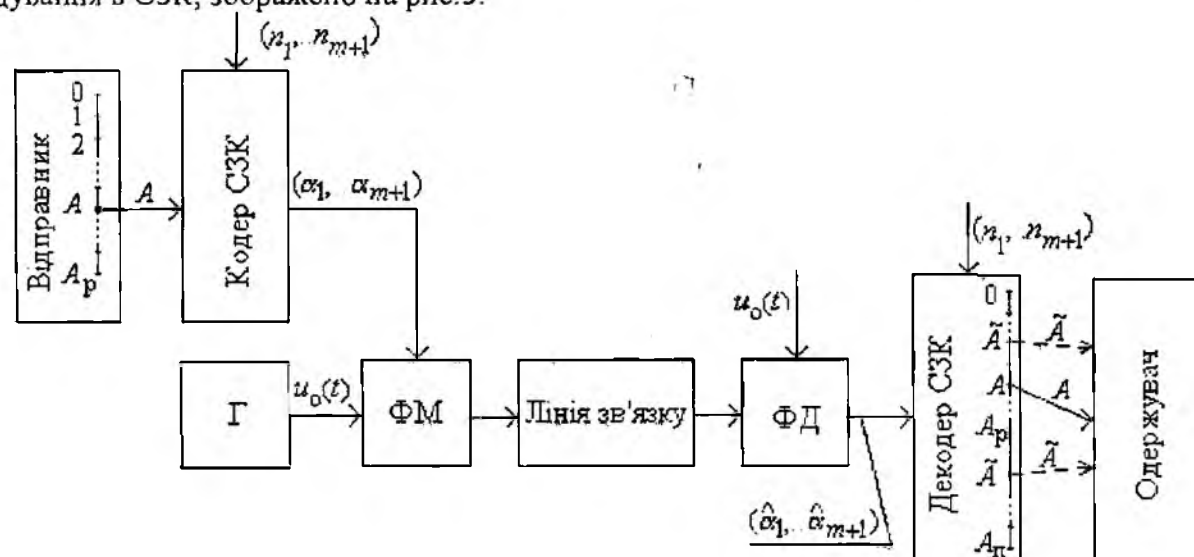


Рис. 5

Числове повідомлення A в декодері СЗК перетворюється на лишки $\alpha_1, \dots, \alpha_i, \dots, \alpha_m$. Ці лишки використовують у фазовому модуляторі ФМ для формування у відповідності до алгоритму (8) початкових фазових зсувів гармонічного сигналу генератора сигналу Г (5). Отриманий таким чином інформаційний сигнал передається по лінії зв'язку. На приймальній частині повідомлення декодується. Передбачається, що у одержувача є опорний сигнал і йому відома система модулів $(n_1, \dots, n_i, \dots, n_m)$. Фазовий детектор ФД забезпечує вимірювання початкових фазових зсувів $\{\hat{\phi}_i, i = \overline{1, m}\}$, які відповідають переданим лишкам, і визначення оцінок лишків за формулою (8). Декодер СЗК відновлює повідомлення з отриманої системи лишків $(\hat{\alpha}_1, \dots, \hat{\alpha}_m)$ за алгоритмом (3).

Таким чином, ключем для декодування повідомлення є система модулів і їх послідовність, яку використовують двічі: першого разу для розрахунку лишків за формулою (9), другого – для розрахунку базисів СЗК, потрібних для визначення повідомлення за алгоритмом (4).

Слід зазначити, що відсутність зв'язків між лишками дозволяє організувати їх паралельну передачу на різних несучих частотах, що значно скорочує час передавання.

4. Синхронізація прийому сигнальних посилок. При передачі цифрових даних на великій відстані суттєвою є проблема синхронізації [6]. Як відомо, сигнали синхронізації в цифрових системах передачі даних визначають моменти часу, коли необхідно виявити з лінії передачі

оновлені дані. Надійність синхронізації суттєво впливає на ймовірність виникнення помилок перелачі, отже на надійність передачі даних. Найчастіше вона вирішується за допомогою передачі сигналів синхронізації по окремому каналу синхронізації або шляхом періодичної синхронізації при передачі по інформаційному каналу заздалегідь визначених кодів або сигналів, які відрізняються своїми характеристиками від інформаційних сигналів. Перший спосіб має суттєву ваду: внаслідок дисперсності каналу передачі груповий час розповсюдження сигналів в каналі даних і каналі синхронізації може бути різним і привести до пропуску або повторного прийому цифрових даних. Тому даний спосіб не використовується для передачі даних на великі відстані. Другий спосіб, пов'язаний з використанням додаткових кодів чи сигналів в інформаційному каналі, веде до додаткових витрат часу на передачу, отже, є неекономічним.

При передачі даних на великі відстані перевагу мають самосинхронізуючі коди, сигнали яких містять і ознаки, за якими відбувається синхронізація прийому даних. До таких кодів належить і СЗК. Проаналізуємо можливість формування синхросигналів в системі з фазовою маніпуляцією безпосередньо з інформаційного сигналу – послідовності сигнальних посилок, які відрізняються різними початковими фазовими зсувами.

Відмінною рисою формування фазоманіпульованого сигналу з кодуванням в СЗК є те, що в суміжних сигнальних посилках початкові фазові зсуви завжди відрізняються. Виключення становить код числа нуль (ця комбінація може бути вилучена з переліку кодових комбінацій, якими передається повідомлення). Ця властивість пояснюється тим, що модулі СЗК складають взаємно прості числа.

Розглянемо інформаційний сигнал (5), приклад реалізації якого наведено на рис.3.

Мінімальний стрибок фази при переході через границю суміжних сигнальних посилок становить:

$$\Delta\varphi = \frac{2\pi}{n_k} - \frac{2\pi}{n_{k-1}}, \quad (10)$$

де $n_k = \max\{n_i, i = \overline{1, m}\}$.

Для виділення стрибків фази сигналу (5) використаємо поняття аналітичного сигналу [7]. Це поняття широко застосовується в радіотехніці і техніці зв'язку при прийманні модульованих і маніпульованих радіосигналів. Аналітичний сигнал є комплексним і утворюється з дійсного сигналу (1) таким чином:

$$\dot{z}(t) = u(t) + i\hat{u}(t) = U(t)\cos[\Phi(t)], \quad (11)$$

де $i = -\sqrt{-1}$; $\hat{u}(t)$ – сигнал, який пов'язаний з $u(t)$ перетворенням Гільберта [8].

$$\hat{u}(t) = \frac{1}{\pi} \int_0^\infty \frac{u(t+\tau) - u(t-\tau)}{\tau} d\tau, \quad (12)$$

$U(t), \Phi(t)$ – відповідно обвідна і фаза аналітичного сигналу. Сигнал $\dot{z}(t)$ можна подати у вигляді вектора на комплексній площині рис.6, де графічно представлено аналітичний сигнал у фіксований момент часу $t = t_g$.

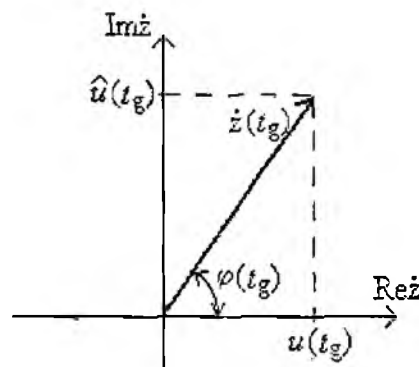


Рис.6

Обвідна і фаза (в межах інтервалу $[0, 2\pi)$) аналітичного сигналу визначаються як

$$U(t) = \sqrt{u^2(t) + \hat{u}^2(t)}, \quad (13)$$

$$\varphi(t) = \arctg\left(\frac{\hat{u}(t)}{u(t)}\right) + \frac{\pi}{2} \{2 - [\text{sign}(\hat{u}(t))] \cdot [1 + \text{sign}(u(t))]\}, \quad (14)$$

де $\text{sign}(\cdot)$ – знакова функція. Якщо ввести числову функцію $K[u(t), \hat{u}(t)]$, яка усуває стрибки фази на 2π , то на інтервалі спостереження можна визначити фазову характеристику сигналу (5):

$$\Phi(t) = \varphi(t) + K[u(t), \hat{u}(t)]. \quad (15)$$

При цьому стрибки фази, менші 2π , пов'язані з процесом фазової маніпуляції, залишаються. За наявності стрибків фази в момент стрибка $t = t_c$ маємо $\left. \frac{d\Phi(t)}{dt} \right|_{t=t_c} \rightarrow \infty$.

З метою перевірки стійкості даного способу виділення границь сигнальних посилок і формування сигналів синхронізації виконаємо моделювання цього процесу.

Результати моделювання у вигляді графіки сигналів $u(q), \hat{u}(q)$ (а), фази $\Phi(q)$ (б) та стрибків фази $\Delta\Phi(q)$ (в) наведені на рис.7. На рисунку позначено: q – номер відліку сигналу, $q = \frac{t}{T_d}$, $t \in [0, T_n]$, де T_d – інтервал дискретизації для випадку, коли інформаційний сигнал складається з трьох сигнальних посилок. Стрибки фази між сигнальними посилками становлять $\Delta\varphi_1 = \frac{\pi}{5}$ рад і $\Delta\varphi_2 = \frac{\pi}{20}$ рад.

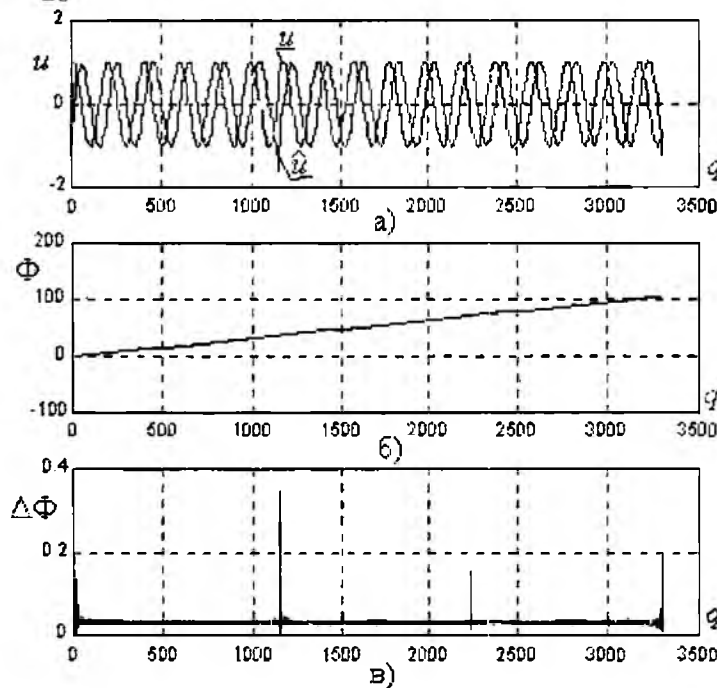


Рис.7

На графіку рис. 7а зображено вибіркові значення вихідного $u(q)$ і спряженого за Гільбертом $\hat{u}(q)$ сигналів, розрахованих для дискретних моментів часу $t_q = qT_d$, $q \in \left[0, \frac{T_n}{T_d}\right]$, на рис.7б – вибіркові значення фази $\Phi(q)$, на рис.7в – вибіркові значення приростів фази $\Delta\Phi(q) = \Phi(q) - \Phi(q-1)$, $q \in \left[0, \frac{T_n}{T_d}\right]$ при $\Phi(0) = 0$.

З рис.7в видно, що при збільшенні n , зменшується крок фази $\Delta\phi$, отже, зменшується і амплітуда синхроімпульсів. Це може ускладнити виділення стрибків фази і формування синхросигналу за наявності в каналі передачі перешкод. Проте, це питання виходить за рамки даної роботи і потребує окремого дослідження.

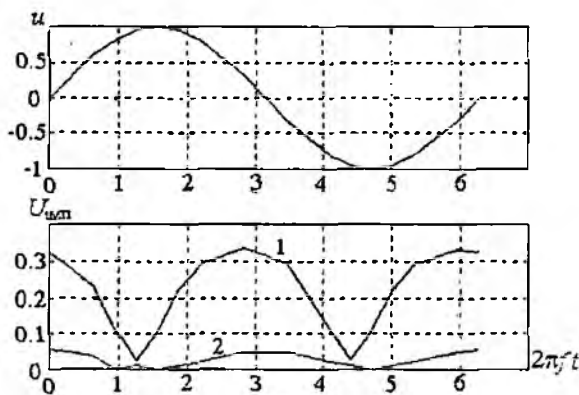


Рис. 8

Амплітуда імпульсів синхронізації залежить від положення стрибка фази в періоді сигналу несучої частоти. Аналіз амплітуд імпульсів синхронізації виконано шляхом моделювання, за результатами якого на рис.8б побудовано графіки амплітуд імпульсів синхронізації для різних величин стрибка: крива 1 відповідає стрибку на $\Delta\phi = \pi/5$ рад випадку, а крива 2 – $\Delta\phi = \pi/50$ рад. Крок дискретизації становив $\pi/100$ рад. На рис.8а зображено сигнал несучої, який дозволяє співставити місце розташування стрибка з його положенням на періоді сигналу.

Отже, на рис. 8 показано графіки одного періоду інформаційного сигналу несучої частоти (а) і амплітуд, виділених з інформаційного сигналу синхроімпульсів (б) в залежності від розташування стрибка фази в періоді інформаційного сигналу.

З аналізу кривих можна зробити такі висновки. По-перше, межа сигнальної посилки найкраще виділяється у випадку, коли вона розташована в тій частині, для якої модуль похідної сигналу в часі максимальний. По-друге, зі зменшенням $\Delta\phi$ в 10 разів амплітуда сигналів синхронізації падає в $\approx 4 \div 5.5$ разів.

Таким чином, утворені в СЗК коди можна віднести до само синхронізуючих кодів. На відміну від інших самосинхронізуючих кодів, наприклад таких, що використовують маніпуляцію амплітуди синусоїдальної несучої, кодування в СЗК не вимагає додаткових енергетичних витрат для формування ознак синхронізації.

Застосування СЗК в цифрових системах передачі даних дозволяє реалізувати системний підхід до організації каналу передачі даних і комплексно вирішити ряд актуальних питань, до числа яких належать: завадостійке кодування, виявлення і виправлення помилок при передачі даних, захист інформації (шифрування повідомлення, ключем до його декодування є система модулів), синхронізація прийому сигнальних посилок.

Подання інформації в СЗК забезпечує надійність і скритність її передачі. Виявлення і виправлення помилок передачі реалізується на основі введення в систему СЗК додаткових модулів за кожний з модулів СЗК. Для виправлення помилок, що одночасно виникають у декількох лишках, необхідно розширити повний діапазон СЗК і застосувати комбінацію різних способів виявлення і виправлення помилок.

Разом з високою здатністю до виправлення помилок кодування СЗК забезпечує скритність передачі інформації, яка може бути підвищена за рахунок збільшення робочого діапазону СЗК та багатократного застосування СЗК. Ключем до відновлення повідомлення є система модулів.

СЗК може бути віднесена до самосинхронізуючих кодів, тому її застосування в системі передачі даних дозволяє отримувати сигнали синхронізації безпосередньо з інформаційних сигналів і за рахунок цього підвищити надійність і ефективність передачі даних. Часове по-

ложення синхросигналів визначається по стрибках фази інформаційного сигналу. Використання інформаційного сигналу для формування синхросигналів дозволяє формувати і передавати сигнальні посилки різної тривалості, що може бути додатковим фактором, який підвищує скритність і ускладнює декодування при передачі інформації по каналу зв'язку.

Кодування інформації, що ґрунтується на СЗК, дозволяє використовувати існуючі системи передачі даних з фазовою маніпуляцією без суттєвих додаткових апаратних витрат шляхом доопрацювання програмної частини системи.

Список літератури: 1. *Акуликин И.Я. Юдинский Д.И.* Машинная арифметика в остаточных классах. М.: Сов. радио, 1968. 438 с. 2. *Куц Ю.В.* Застосування системи залишкових класів в задачах захисту інформації // *Защита информации.* Сб. науч. трудов НАУ. 2003. Вып. 10. С. 137 – 149. 3. *Бабак В.П., Куц Ю.В.* Використання коригуючих властивостей системи залишкових класів у фазометрії // *Вісник НАУ* 2004. №1. С. 3-9. 4. *Столлингс В.* Беспроводные линии связи и сети. Пер. с англ. М.: Издат. Дом «Вильямс», 2003. 640 с. 5. *Куц Ю.В., Щербак Л.М.* Застосування фазових методів в задачах захисту інформації // *Захист інформації.* 2004. №1. С. 43 – 51. 6. *Компьютерные сети. Принципы, технологии, протоколы* / В.Г. Олифер, Н.А. Олифер. СПб: Питер, 1999. 672 с. 7. *Бендат Дж., Пирсол А.* Прикладной анализ случайных данных. Пер. с англ. М.: Мир, 1989. 540 с. 8. *Математическая энциклопедия* / Гл. ред. И.М. Виноградов. М.: Сов. энциклопедия, 1980. Т.1. 1184 с.

*Національний авіаційний університет,
Київ*

Надійшло до редколегії 05.10.2004

УНИВЕРСАЛЬНЫЕ АЛГОРИТМЫ СЖАТИЯ ТАБЛИЧНЫХ ЦИФРОВЫХ ДАННЫХ РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

Введение

Решение широкого круга сложных вычислительных научно-технических задач требует значительных объемов расчетов и часто в реальном времени функционирования системы. В этом аспекте проводимые исследования поиска методов и средств повышения производительности переработки цифровой информации весьма важны и актуальны.

В любых позиционных системах счисления (ПСС) выполнение арифметических операций предполагает последовательную обработку разрядов операндов по правилам данной операции и она не может быть закончена до тех пор, пока не будут определены последовательно результаты всех предыдущих межразрядных операций с учетом всех связей между разрядами. Таким образом, ПСС, используемые в современных вычислительных машинах, в которых представляется и обрабатывается информация, обладают существенным недостатком – наличием межразрядных связей, которые накладывают свой отпечаток на принципы и методы реализации арифметических операций.

Одним из перспективных и действенных направлений повышения пользовательской производительности, надежности и достоверности вычислительных средств является внедрение нетрадиционных методов представления и обработки информации в числовых системах с параллельной структурой, и в частности, в так называемых модулярных системах счисления, обладающих максимальным уровнем внутреннего параллелизма в организации процесса переработки информации. К таким системам счисления относятся и непозиционная система счисления в остаточных классах (СОК).

Известная непозиционная система счисления в остаточных классах, как отмечается в [1-2], обладает ценным свойством независимости друг от друга остатков числа по принятой системе оснований. Это открывает широкие возможности создания не только новой машинной модулярной арифметики, но и принципиально новой схемной реализации архитектуры ЭВМ, которая, в свою очередь, заметно расширяет применение машинной арифметики. Основными преимуществами СОК являются возможность разработки и внедрения высокоэффективных алгоритмических и аппаратных архитектур вычислительных структур параллельно-конвейерного типа. При этом обеспечивается, во-первых, высокая степень интеграции и унификации арифметических блоков и вычислительных узлов и, во-вторых, использования уникальных корректирующих свойств непозиционных кодовых структур при обнаружении и исправлении, а также при контроле ошибок в динамике вычислительного процесса, т.е. в реальном времени без останова вычислений. И, наконец, использования свойств машинной арифметики в СОК позволяет организовать высокопроизводительную реализацию вычислительных процессов, требующих больших объемов вычислений.

Кратко охарактеризуем основные свойства СОК.

1. *Независимость остатков.* Это свойство дает возможность построить ЭВМ в виде набора независимых вычислительных трактов (отдельных ЭВМ, функционирующих по своему определенному модулю m_i в СОК). При этом время реализации арифметических операций определяется временем реализации в вычислительном тракте ЭВМ по наибольшему основанию m_n СОК. Как видно, ошибки, возникающие за счет отказов (сбоев) схем двоичных разрядов в произвольном вычислительном тракте ЭВМ, не «размножаются» в соседние тракты (остаются в пределах одного остатка), что дает возможность повысить достоверность вычислений в СОК.

2. *Равноправность остатков.* Любой остаток a_i числа $A = (a_1, a_2, \dots, a_n)$ несет информацию обо всем исходном числе. Использование этого свойства вместе с первым, а также применяя идею структурного резервирования, можно синтезировать надежностную модель ЭВМ в СОК, соответствующую модели динамического резервирования в ПСС. В этом случае информационные тракты $m_1 \div m_n$ ЭВМ играют роль рабочих элементов, а тракты $m_{n+1} \div m_{n+k}$ – роль резервных элементов (рис.1), где k – количество контрольных (резервных) оснований СОК.

3. *Малоразрядность остатков.* Это свойство позволяет существенно повысить быстродействие выполнения арифметических операций как за счет малоразрядности построения вычислительных трактов ЭВМ, так и за счет возможности применения (в отличие от ПСС) табличной арифметики, где арифметические операции сложения, вычитания и умножения выполняются практически в один такт [3, 4].

Основные свойства СОК обуславливают высокое быстродействие выполнения арифметических операций за счет возможности представления и реализации алгоритмов машинной арифметики в конвейерно-табличной (матричной) форме. Известно, что все основные достоинства СОК более полно проявляются при использовании табличного принципа реализации арифметических операций т.е. при использовании табличных (матричных) коммутаторов, реализующих модульные операции. Поэтому по мере возрастания степени совершенствования и развития интегральной технологии производства запоминающих устройств на БИС и СБИС в виде логических программируемых матриц (ПЛМ) или программируемых логических интегральных схем (ПЛИС), составляющих основу для реализации табличных методов вычислений в СОК, интенсивность исследований в направлении создания табличных алгоритмов реализации модульных операций будет возрастать.

Цель статьи – разработка и дальнейшее совершенствование алгоритмов табличной арифметики в СОК и, в частности, разработка универсальных табличных алгоритмов информационного сжатия цифровых данных результатов выполнения одновременно арифметических операций сложения, вычитания и умножения в модулярной арифметике на основе использования кода табличного умножения (КТУ).

Анализ последних исследований

Операционное устройство (ОпУ) ЭВМ в СОК принципиально может быть выполнено следующим образом: в сумматорном варианте (на базе малоразрядных двоичных сумматоров) [1]; при использовании кольцевых регистров сдвига (КРС) [3]; используя прямой логический метод [4] и в табличном (матричном) варианте [5].

При построении ОпУ на основе использования первых двух вариантов каждый остаток числа в СОК, представленный двоичным кодом, обрабатывается независимо от других, и время выполнения всей операции определяется временем, необходимым для получения результата по наибольшему основанию.

Отметим основные недостатки этих двух вариантов выполнения арифметических операций:

- некоторая сложность синтеза двоичных сумматоров и КРС;
- значительное время преобразования информации, определяемое величиной максимального основания СОК, что существенно для ЭВМ с большой разрядной сеткой;
- сложность технической реализации операции умножения;
- не эффективное использование двоичных элементов разрядной сетки ЭВМ, вследствие возможной избыточности представления максимальных чисел.

Эффективным методом повышения надежности и производительности ЭВМ является использование при реализации арифметических операций матричных схем на основе использования постоянных запоминающих устройств (ПЗУ), ПЛМ, а также ПЛИС. Небольшая потребляемая мощность, повышенные надежностные характеристики и компактность матричных схем открывают широкие перспективы использования их в качестве основных составляющих структуры ОпУ ЭВМ.

Из проведенных исследований [1,2,5] очевидно, что вопросы, связанные с выпол-

нением арифметических операций табличными методами (посредством ПЗУ), целесообразно рассматривать лишь в применении к ЭВМ в СОК.

Кратко отметим достоинства матричного (табличного) варианта построения ЭВМ в СОК:

- матричные схемы имеют высокую конструктивную надежность, так как реализуются в виде компактных ПЗУ; в этом случае ЭВМ строится по блочному принципу, что улучшает ремонтпригодность ЭВМ (в частности, уменьшается среднее время восстановления);
- простота построения матричных схем и дешифраторов, имеющих количество выходов равное величине основания СОК;
- высокое быстродействие: результат операции может быть получен в момент поступления входных операндов, т.е. в один такт; таким образом, время выполнения арифметических операций в СОК сравнимо с тактовой частотой вычислителя, что принципиально невозможно для позиционной ЭВМ.

Поиск путей повышения производительности и надежности ЭВМ, за счет упрощения ее структуры, привел к необходимости разработки табличных методов и алгоритмов реализации модульных операций, позволяющих повысить эффективность применения табличной арифметики.

В [1] показано, что при применении методов специального кодирования информации в СОК (целью которых является сокращение элементов таблиц ПЗУ, реализующих результаты выполнения арифметических операций) позволяет добиться того, что количество оборудования ОпУ при табличном построении может быть не больше количества оборудования необходимого при сумматорном принципе построения ОпУ ЭВМ в СОК.

Под табличной реализацией арифметических операций $c_i = f(a_i, \beta_i)$ понимается организация такой таблицы, в которой каждой комбинации входных величин a_i и β_i соответствует одно и только одно значение выходной величины c_i .

Вначале рассмотрим известный табличный метод реализации операции модульного умножения с использованием КТУ. Составим таблицу $a_i \beta_i (mod m_i)$ модульного умножения для произвольного основания m_i СОК. Эта таблица симметрична относительно левой (главной) и правой диагоналей, а также вертикали и горизонтали. Симметричность относительно левой диагонали определяется коммутативностью операции $a_i \beta_i$ умножения, а симметричность относительно правой диагонали определяется тем, что

$$(m_i - a_i)(m_i - \beta_i) \equiv a_i \beta_i (mod m_i).$$

Симметричность относительно вертикали и горизонтали определяется из условия кратности значения модуля сумме симметричных чисел таблицы умножения

$$\begin{aligned} a_i \beta_i + a_i(m_i - \beta_i) &\equiv 0(mod m_i), \\ a_i \beta_i + \beta_i(m_i - a_i) &\equiv 0(mod m_i). \end{aligned}$$

Исходя из вышеизложенного очевидно, что для табличной реализации операции модульного умножения $a_i \beta_i (mod m_i)$ достаточно иметь числовую информацию только о ее восьмой части. Отсюда возникает возможность упростить таблицу (количество схем совпадения матричного ПЗУ) модульного умножения (отметим, что уменьшение таблицы в восемь раз приводит к необходимости производить предварительный анализ величин входных операндов a_i и β_i .) Использование этого варианта увеличивает время и техническую сложность реализации данной арифметической операции.

Для наиболее эффективной реализации операции $a_i \beta_i (mod m_i)$ применяются методы специального кодирования, позволяющие в четыре раза уменьшить таблицу модульного умножения. Решение поставленной задачи возможно в результате применения специальных кодов. Рассмотрим один из вариантов выполнения операции модульного умножения посредством использования КТУ (см. табл. 1 и 2 для $m_i = 5$).

Пусть даны входные операнды a_i и β_i . Значения $a_i(\beta_i)$, лежащие в диапазоне

$[0, (m_i - 1)/2]$, могут быть закодированы произвольным способом, а значения $a_i(\beta_i)$, лежащие в диапазоне $[(m_i + 1)/2, m_i - 1]$, кодируются как $m_i - a_i(\beta_i)$. Для отличия диапазонов вводится индекс (признак) КТУ:

$$\gamma_a(\gamma_\beta) = \begin{cases} 0, & \text{если } 0 \leq a_i(\beta_i) \leq (m_i - 1)/2, \\ 1, & \text{если } (m_i + 1)/2 \leq a_i(\beta_i) \leq m_i - 1. \end{cases}$$

Алгоритм определения результата операции модульного умножения посредством КТУ следующий: если заданы два операнда в КТУ $a_i = (\gamma_a, a'_i)$, $\beta_i = (\gamma_\beta, \beta'_i)$, то для того чтобы получить произведение этих чисел по модулю m_i , достаточно найти произведение $a'_i \beta'_i \pmod{m_i}$ и инвертировать его обобщенный индекс γ_i , если γ_a отлично от γ_β , т.е.

$$a_i \beta_i \pmod{m_i} = (\gamma_i, a'_i \beta'_i \pmod{m_i}),$$

где

$$\gamma = \begin{cases} \bar{\gamma}_i, & \text{если } \gamma_a \neq \gamma_\beta, \\ \gamma_i, & \text{если } \gamma_a = \gamma_\beta; \end{cases}$$

$$a'_i = \begin{cases} a_i, & \text{если } \gamma_a = 0, \\ m_i - a_i, & \text{если } \gamma_a = 1. \end{cases}$$

При использовании данного алгоритма ПЗУ, реализующее операцию модульного умножения, конструктивно уменьшаются в четыре раза. При выполнении операции табличными методами в некоторых случаях возможно дополнительное уменьшение оборудования за счет того, что строится не единая таблица для модульных операций, а k более мелких таблиц, позволяющих дать ответы по каждому из k разрядов результата, где k – разрядность регистра, необходимая для хранения цифр остатка по рассматриваемому основанию m_i СОК.

До настоящего времени вопросы эффективной реализации арифметических операций сложения и вычитания с использованием КТУ в литературе либо не рассматривались, либо такая реализация считалась исследователями невозможной. Основная трудность заключается в том, что довольно сложно синтезировать табличные алгоритмы выполнения этих модульных операций, так как таблицы реализации модульных операций сложения и вычитания различны по своей цифровой структуре вследствие чего они не обладают теми свойствами симметрии, которыми обладают таблицы модульного умножения. Однако совершенно иные результаты можно получить, исследуя возможности реализации одной модульной операции с помощью таблиц, реализующих обратную ей операцию, и наоборот.

При исследовании цифровых свойств таблиц модульных операций сложения и вычитания доказано следующее аналитическое соотношение

$$\left[(\gamma_a, a'_i) + (\gamma_\beta, \beta'_i) \right] + \left\{ \left[m_i - (\gamma_a, a'_i) \right] - (\gamma_\beta, \beta'_i) \right\} = 0 \pmod{m_i}, \quad (1)$$

где $a_i = (\gamma_a, a'_i)$, $\beta_i = (\gamma_\beta, \beta'_i)$ – входные операнды, представленные в КТУ. Запишем выражение (1) в виде

$$(\gamma_a, a'_i) + (\gamma_\beta, \beta'_i) = m_i - \left\{ \left[m_i - (\gamma_a, a'_i) \right] - (\gamma_\beta, \beta'_i) \right\}. \quad (2)$$

Из выражения (2) следует, что для получения результата операции модульного сложения в КТУ достаточно знать результат операции модульного вычитания, т.е. возможно эффективно (с точки зрения уменьшения количества оборудования ПЗУ) использовать КТУ одновременно для трех модульных операций: умножения, сложения и вычитания.

На основе выражения (2) разработаем универсальный алгоритм (УА), посредством

которого можно будет осуществлять выполнения любой из трех арифметических операций в СОК: умножение, сложение и вычитание. Операция модульного сложения осуществляется посредством алгоритма, описанного выражением (2). Составим алгоритм выполнения операции модульного сложения с помощью таблицы, для выполнения операции модульного вычитания $(\alpha'_i - \beta'_i) \bmod m_i$. В соответствии с выражением (2) составим поэтапный алгоритм реализации операции модульного сложения.

Первый универсальный алгоритм

Первый этап. Уменьшаемое $\alpha_i = (\gamma_\alpha, \alpha'_i)$ инвертируется по модулю m_i , т.е. получим следующее выражение: $\bar{\alpha}_i = ((\gamma_\alpha + 1) \bmod 2, \alpha'_i)$. Вычитаемое (γ_β, β'_i) оставляем без изменений.

Второй этап. Посредством ПЗУ, реализующего операцию модульного вычитания, по входным операндам α'_i и β'_i определяется результат операции $(\alpha'_i - \beta'_i) \bmod m_i$. Как и для алгоритма модульного умножения индекс γ_i результата операции модульного вычитания формируется в соответствии со значениями индексов соответствующих операндов, т.е. в соответствии со значениями $(\gamma_\alpha + 1) \bmod 2$ и γ_β , где:

$$\gamma_i = \begin{cases} \bar{\gamma}, & \text{если } (\gamma_\alpha + 1) \bmod 2 \neq \gamma_\beta, \\ \gamma, & \text{если } (\gamma_\alpha + 1) \bmod 2 = \gamma_\beta. \end{cases}$$

Следовательно, результат операции модульного вычитания будет иметь следующий вид

$$(\gamma_i, (\alpha'_i - \beta'_i) \bmod m_i).$$

Третий этап. Полученный результат модульного вычитания инвертируем по модулю m_i :

$$((\gamma_i + 1) \bmod 2, (\alpha'_i - \beta'_i) \bmod m_i),$$

где индекс КТУ результата операции равен значению.

Это и будет искомый результат модульного сложения.

Последовательность выполнения полученного алгоритма схематично можно представить в упрощенном виде следующим образом

$$(\alpha_i - \beta_i) \rightarrow [(m_i - \alpha_i) - \beta_i] \rightarrow \{m_i - [(m_i - \alpha_i) - \beta_i]\} \rightarrow (\alpha_i + \beta_i).$$

Таким образом, несмотря на различие цифровой структуры таблиц модульных операций сложения, вычитания и умножения, создан новый оригинальный универсальный табличный алгоритм для реализации арифметических операций в СОК. С помощью этого универсального алгоритма можно построить конструктивно простое и высоконадежное ОпУ ЭВМ в СОК, основу которого составляют три отдельных коммутатора, каждый из которых реализует только 0,25 части соответствующей полной таблицы модульных операций умножения (табл. 2) и вычитания (табл. 4) (первый коммутатор II-квадрант таблицы умножения (табл. 5); второй и третий коммутаторы – соответственно I (табл. 7) и II (табл. 6) квадранты табл. 4 вычитания). В этом плане код табличного умножения приобрел новое качество и стал универсальным табличным кодом для выполнения трех арифметических операций в СОК.

Рассмотрим еще один вариант построения универсального алгоритма выполнения арифметических операций в СОК.

Из выражения (1) следует, что

$$(\gamma_\alpha, \alpha'_i) - (\gamma_\beta, \beta'_i) = \left\{ (\gamma_\alpha, \alpha'_i) + \left[m_i - (\gamma_\beta, \beta'_i) \right] \right\}. \quad (3)$$

Таблица 1

a_i	КТУ		a_i	КТУ	
	γ_a	a_i'		γ_a	a_i'
1	0	1	3	1	2
2	0	2	4	1	1

Таблица 2

β_i	a_i	0	1	2	3	4
0		0	0	0	0	0
1		0	1	2	3	4
2		0	2	4	1	3
3		0	3	1	4	2
4		0	4	3	2	1

Таблица 3

β_i	a_i	0	1	2	3	4
0		0	1	2	3	4
1		1	2	3	4	0
2		2	3	4	0	1
3		3	4	0	1	2
4		4	0	1	2	3

Таблица 4

β_i	a_i	0	1	2	3	4
0		0	1	2	3	4
1		4	0	1	2	3
2		3	4	0	1	2
3		2	3	4	0	1
4		1	2	3	4	0

Таблица 5

β_i		a_i	1	2
			4	3
1	4		1	2
2	3		2	4

Таблица 6

β_i		a_i	1	2
			4	3
1	4		0	1
2	3		4	0

Таблица 7

β_i		a_i	2	1
			3	4
1	4		2	3
2	3		1	2

Таблица 8

β_i		a_i	1	2
			4	3
1	4		2	3
2	3		3	4

Таблица 9

β_i		a_i	2	1
			3	4
1	4		4	0
2	3		0	1

	1	2	3	4	5
1	$\{ \{ \gamma_a, a_i' - (\gamma_\beta, \beta_i') \} \mod m_i \}$	$\gamma_a = \gamma_\beta$ ($\gamma_i = 0$)	Используется II квадрант (табл. 6) полной таблицы модуль- ного вычитания	$\gamma_a = \gamma_\beta = 0$	Результат определяется непосредственно значе- ниями узлов табл. 6 ($\gamma_0 = \gamma_i$)
2				$\gamma_a = \gamma_\beta = 1$	Результат определяется инвертированием по мо- дулю m_i значений узлов табл. 6 ($\gamma_0 = (\gamma_i + 1) \mod 2$)
3		$\gamma_a \neq \gamma_\beta$ ($\gamma_i = 1$)	Используется I квадрант (табл. 7) полной таблицы модуль- ного вычитания	$\gamma_a = 1, \gamma_\beta = 0$	Результат определяется непосредственно значе- ниями узлов табл. 7 ($\gamma_0 = \gamma_i$)
4				$\gamma_a = 0, \gamma_\beta = 1$	Результат определяется инвертированием по модулю m_i значений узлов табл. 7 ($\gamma_0 = (\gamma_i + 1) \mod 2$)
5	$\{ \{ \gamma_a, a_i' + (\gamma_\beta, \beta_i') \} \mod m_i \}$	$\gamma_a = \gamma_\beta$ ($\gamma_i = 0$)	Используется I квадрант (табл. 7) полной таблицы модуль- ного вычитания	$\gamma_a = \gamma_\beta = 0$	Результат определяется непосредственно значе- ниями узлов табл. 7 ($\gamma_0 = \gamma_i$)
6				$\gamma_a = \gamma_\beta = 1$	Результат определяется инвертированием по мо- дулю m_i значений узлов табл. 7 ($\gamma_0 = (\gamma_i + 1) \mod 2$)
7		$\gamma_a \neq \gamma_\beta$ ($\gamma_i = 1$)	Используется II квадрант (табл. 6) полной таблицы модуль- ного вычитания	$\gamma_a = 1, \gamma_\beta = 0$	Результат определяется непосредственно значе- ниями узлов табл. 6 ($\gamma_0 = \gamma_i$)
8				$\gamma_a = 0, \gamma_\beta = 1$	Результат определяется инвертированием по мо- дулю m_i значений узлов табл. 6 ($\gamma_0 = (\gamma_i + 1) \mod 2$)

Из выражения (3) следует, что результат операции модульного вычитания можно получить посредством ПЗУ, реализующего операцию модульного сложения.

В этом случае второй универсальный алгоритм будет иметь следующий вид.

Второй универсальный алгоритм

Первый этап. В соответствии с выражением (3) второе слагаемое β_i' инвертируем по модулю m_i :

$$\bar{\beta}_i = ((\gamma_\beta + 1) \mod 2, \beta_i'),$$

Второй этап. С помощью ПЗУ для модульного сложения по входным операндам a_i' и $\bar{\beta}_i$ определим значение модульной суммы в виде $(a_i' + \bar{\beta}_i) \mod m_i$.

	1	2	3	4	5
1	$[(\gamma_a, a_i' - (\gamma_\beta, \beta_i')) \bmod m_i]$	$\gamma_a = \gamma_\beta$ ($\gamma_i = 0$)	Используется I квадрант (табл. 9) полной таблицы модуль- ного сложения	$\gamma_a = \gamma_\beta = 0$	Результат определяется инвертированием по модулю m_i значений узлов табл. 9 ($\gamma_\rho = \gamma_i$)
2				$\gamma_a = \gamma_\beta = 1$	Результат определяется непосредственно значе- ниями узлов табл. 9 ($\gamma_\rho = \gamma = (\gamma_i + 1) \bmod 2$)
3		$\gamma_a \neq \gamma_\beta$ ($\gamma_i = 1$)	Используется II квадрант (табл. 8) полной таблицы модуль- ного сложения	$\gamma_a = 1, \gamma_\beta = 0$	Результат определяется инвертированием по модулю m_i значений узлов табл. 8 ($\gamma_\rho = \gamma = (\gamma_i + 1) \bmod 2$)
4				$\gamma_a = 0, \gamma_\beta = 1$	Результат определяется непосредственно значе- ниями узлов табл. 8 ($\gamma_\rho = \gamma_i$)
5	$[(\gamma_a, a_i' + (\gamma_\beta, \beta_i')) \bmod m_i]$	$\gamma_a = \gamma_\beta$ ($\gamma_i = 0$)	Используется I квадрант (табл. 8) полной таблицы модуль- ного сложения	$\gamma_a = \gamma_\beta = 0$	Результат определяется непосредственно значе- ниями узлов табл. 8 ($\gamma_\rho = \gamma_i$)
6				$\gamma_a = \gamma_\beta = 1$	Результат определяется инвертированием по мо- дулю m_i значений узлов табл. 8 ($\gamma_\rho = \gamma = (\gamma_i + 1) \bmod 2$)
7		$\gamma_a \neq \gamma_\beta$ ($\gamma_i = 1$)	Используется I квадрант (табл. 9) полной таблицы модуль- ного сложения	$\gamma_a = 1, \gamma_\beta = 0$	Результат определяется непосредственно значе- ниями узлов табл. 9 ($\gamma_\rho = \gamma = (\gamma_i + 1) \bmod 2$)
8				$\gamma_a = 0, \gamma_\beta = 1$	Результат определяется инвертированием по мо- дулю m_i значений узлов табл. 9 ($\gamma_\rho = \gamma_i$)

Окончательный результат выполнения операции модульного вычитания будет иметь следующий вид:

$$((\gamma_\beta + 1) \bmod 2, (a_i' + \beta_i') \bmod m_i),$$

где индекс КТУ результата операции равен значению

Схематично этот алгоритм может быть представлен в виде

$$(a_i + \beta_i) \rightarrow [a_i + (m_i - \beta_i)] \rightarrow (a_i - \beta_i).$$

При реализации модульных операций посредством второго УА основу ОпУ составляют

также три коммутатора, каждый из которых реализует 0,25 части соответствующей полной таблицы модульных операций умножения (табл.2) и сложения (табл. 3) (первый коммутатор - II-квадрант таблицы умножения (табл.5); второй и третий коммутаторы – соответственно I (табл.9) и II (табл.8) квадранты табл. 3 сложения).

При реализации арифметических операций умножения, сложения и вычитания второй предложенный универсальный алгоритм позволяет за меньшее время и с меньшими аппаратными затратами (по сравнению с первым универсальным алгоритмом) выполнять в СОК данные арифметические операции.

Отметим, что существует класс патентоспособных ОпУ ЭВМ в СОК, которые реализуют варианты предложенных УА.

Универсальные алгоритмы для конкретного выполнения арифметических операций, посредством КТУ, при значении модуля СОК равного $m_i = 5$ представлены в табл. 10 (первый УА) и табл. 11 (второй УА).

Выводы

В статье рассмотрены табличные универсальные алгоритмы реализации арифметических операций сложения, вычитания и умножения в СОК. Данные УА (основанные на использовании КТУ) несмотря на различие свойств цифровых данных структур таблиц, реализующих модульные операции $(a_i \otimes \beta_i) \bmod m_i$ позволяют реализовать всего 0,25 части каждой из полных таблиц что ранее предполагалось невозможным. При реализации модульных операций сложения и вычитания КТУ приобрел новые качества и стал общим для выполнения всех трех вышеперечисленных арифметических операций в СОК.

Таким образом, при выполнении посредством УА всех трех арифметических операций удалось сократить 75% оборудования коммутаторов (например, для первого УА используются табл. 5, 6 и 7; для второго УА – табл. 5, 8 и 9), посредством которых реализуются данные операции. Это в свою очередь (как показали расчеты) в зависимости от длины машинного слова (величины разрядной сетки), позволило сократить до $\approx (50-60)\%$ оборудования табличного операционного устройства ЭВМ в СОК. Отметим, что с увеличением длины разрядной сетки ЭВМ (что характерно для современной тенденции развития вычислительных средств информационно-управляющих и информационно-расчетных систем), эффективность применения предложенных универсальных алгоритмов существенно возрастает.

Результаты проведенных исследований могут быть использованы в системах и устройствах обработки в реальном времени больших массивов цифровой информации: при решении задач цифровой фильтрации, криптографических преобразованиях в полях Галуа, при разработке криптографических систем на основе использования преобразований Хартли, а также при необходимости реализации модульных преобразований в устройствах с повышенными требованиями по быстродействию выполнения сложных расчетных задач в целочисленной модульной арифметике, например, при решении задачи по обеспечению необходимого уровня стойкости криптографических средств за счет выполнения арифметических операций над целыми числами.

Список литературы: 1. Акуцкий И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М.: Сов. радио, 1968. 440 с. 2. Краснобаев В.А. Основы создания вычислителей на основе остаточных классов // Системы обработки информации. Харьков: НАНУ, ЛАНМ, ХВУ. 2001. Вып. 1(11). С. 3-7. 3. Краснобаев В.А. Методы реализации модульных операций в системах цифровой обработки информации // Радиотехника. Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С. 130-134. 4. Фурман И.А., Кошман С.А., Краснобаев В.А. Вариант синтеза процессора в системе остаточных классов // Радиоэлектроника и информатика. 2003. Вып. № 2(23). С. 94-96. 5. Жихарев В.Я., Юнес Эль Хандасси, Краснобаев В.А. Методы и алгоритмы реализации арифметических операций в классе вычетов // Открытые информационные и компьютерные интегрированные технологии. Х.: НАКУ (ХАИ). 2003. Вып. 20. С. 84-101.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 05.10.2004

ДОСВІД ПІДГОТОВКИ ТА РЕЗУЛЬТАТИ ПРОВЕДЕННЯ VI ВСЕУКРАЇНСЬКОЇ ОЛІМПІАДИ З НАПРЯМУ «ІНФОРМАЦІЙНА БЕЗПЕКА»

На базі Харківського національного університету радіоелектроніки з 26 по 28 квітня 2004 року проводився другий тур VI Всеукраїнської студентської олімпіади з напрямку "Інформаційна безпека" серед студентів III - V курсів університетів та технічних вузів України. Олімпіада провадилася за двома рівнями 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 провадилася в два тури. Перший тур був пов'язаний з розв'язком задач різної складності та реалізації однієї задачі у вигляді програмного продукту. При оцінці розробленої програми враховувалися як правильність її функціонування, так і якість реалізації (наприклад, характеристики швидкості, пам'яті, повнота проведеного тестування і т.ін.). Додатково на олімпіаді провадився другий тур, який мав вигляд презентації та захисту власних проектів та розробок за напрямком "Інформаційна безпека", у тому числі статей, патентів, програмних продуктів, апаратних засобів, тощо. Цей тур здійснювався у вигляді стендової комп'ютерної доповіді.

В олімпіаді брали участь такі вузи: Національний технічний університет України "Київський політехнічний інститут" — 3 чол.; Національний гірничий університет м. Дніпропетровськ — 3 чол.; Національний авіаційний університет, м. Київ — 2 чол.; Криворізький технічний університет — 2 чол.; Дніпропетровський національний університет — 2 чол.; Сумський державний університет — 1 чол.; Львівський національний університет ім. І. Франка — 1 чол.; Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв — 1 чол.; Запорізький національний технічний університет — 2 чол.; Харківський національний університет радіоелектроніки — 15 чол.; Харківський інститут військово-повітряних сил імені І. Кожедуба — 3 чол.; Харківський військовий університет — 2 чол.; Харківська філія Української академії банківської справи НБУ — 2 чол.

1 Конкурсні завдання

Цього року повністю оновлені конкурсні завдання, змінено підхід до їх формування.

1.1 Конкурсні завдання з напрямку "Інформаційна безпека"

Задача № 1 (12 балів)

Поясніть, в чому сутність та вразливість перекриття шифру в режимі потокового шифрування.

Вивести формулу оцінки ймовірності перекриття шифру, якщо в мережі із n станцій шифрування здійснюється на одному ключі в режимі лічильника з перезапуском синхропослідовностей.

Оцініть ймовірність перекриття шифру, якщо період повторення гами шифруючої $L = 2^{64+4k}$, число перезапусків $r = 128k$, загальна довжина зашифрованих повідомлень $z = l \cdot n \cdot r$, довжина одного повідомлення $l = 10^3 + 10^3 k$, число станцій $n = 16$, k — номер реєстрації учасника.

Задача № 2 (12 балів).

Оцініть складність, вартість та практичну можливість здійснення криптоаналітичної атаки типу „повне розкриття” на криптографічну систему, у якій цифровий підпис здійснюється з використанням ЕЦП ДСТУ 4145–2002, а направлене шифрування на основі

протоколу Діффі-Хелмана в групі точок еліптичної кривої з використанням примітивів ДСТУ 4145-2002 за наступними даними:

- порядок точки еліптичної кривої $n = 2^n$ вибрати згідно з таблицею.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
m	491	173	179	191	443	431	233	293	359	419	251	281	239	509

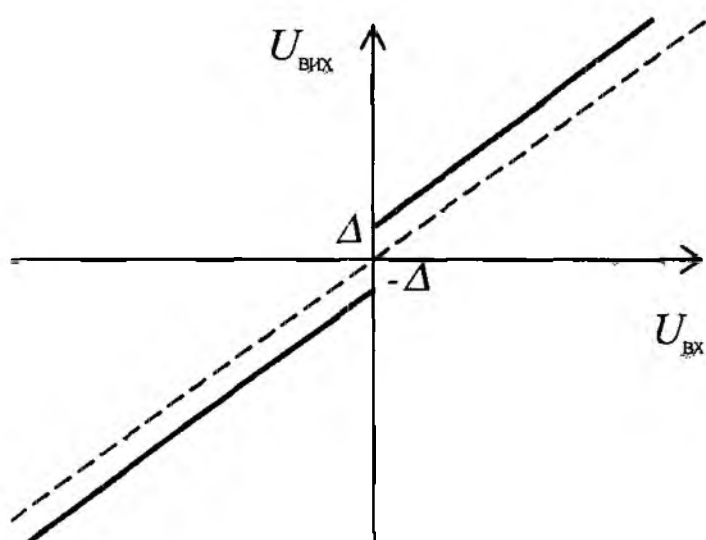
- потужність криптоаналітичної системи $\gamma = k \cdot 10^{10}$ операцій складання в групі точок еліптичної кривої за секунду;

- $i = k(\text{mod } 14)$, k – номер реєстрації учасника.

* Під атакою типу „повне розкриття” розуміється визначення особистих ключів ЕЦП та направлено розшифрування.

Задача № 3 (12 балів)

Гармонічний сигнал частоти Ω підсилюється двотактним підсилювачем до амплітуди U . Природні властивості й окреме регулювання підсилювача такі, що його прохідна характеристика має вигляд товстих ліній, зображених на рисунку: Δ – величина щаблички напруги на виході підсилювача. Характеристика ідеального підсилювача зображена пунктиром.



1) Нарисувати епюри напруг сигналів на вході й виході підсилювача.

2) Як буде залежати спектр Фур'є S_n сигналу на виході підсилювача від співвідношення $\delta = \Delta/U$? (n – номер гармоніки). Вивести аналітичну залежність і зобразити графічно. Дати фізичне пояснення ходу графіка.

3) Нарисувати спектри сигналів на вході й виході підсилювача при якомусь із значень $\delta \neq 0$.

4) До яких наслідків, з точки зору технічного захисту інформації, може призвести зміна спектра сигналу на виході підсилювача?

Довідкові матеріали

Комплексні коефіцієнти Фур'є від дійсної періодичної функції $u(t)$

$$S_n = \frac{1}{T} \int_{-T/2}^{T/2} u(t) \exp(-jn\Omega t) dt.$$

Формула Ейлера

$$\exp(-jn\Omega t) = \cos n\Omega t - j \sin n\Omega t.$$

Тригонометричні формули:

$$\sin \alpha \times \sin \beta = \frac{1}{2} [\cos(\alpha - \beta) - \cos(\alpha + \beta)];$$

$$\sin \alpha \times \cos \beta = \frac{1}{2} [\sin(\alpha - \beta) + \sin(\alpha + \beta)];$$

$$\cos \alpha \times \cos \beta = \frac{1}{2} [\cos(\alpha - \beta) + \cos(\alpha + \beta)].$$

Додаткова задача № 4 (10 балів)

Еліптична крива задана рівнянням

$$y^2 = x^3 + ax + b$$

над простим полем $GF(P)$, $p > 3$.

Довести, що, коли кубічний тричлен у правій частині рівняння кривої розкладається над $GF(P)$ на лінійні множники, то група точок ЕК не є циклічною.

Продемонструйте це на прикладі згідно з таблицею.

Член команди i^{**}	1	2	3	4	5	6
P_i	23	13	11	19	17	7

i^{**} - номер учасника команди (для всіх учасників команди номери повинні бути різними).

1.2 Конкурсні завдання з дисципліни “Захист інформації”**Задача № 1 (12 балів)**

Поясніть, в чому сутність та вразливість перекриття шифру в режимі потокового шифрування.

Вивести формули для оцінки імовірності перекриття шифру в радіолінії з n абонентами, якщо:

- період повторення гами шифруючої $L = 2^{64+4k}$;
- число перезапущів $r = k \cdot 256$;
- довжина кожного повідомлення, що шифрується, $l = 10^4 + 512k$;
- загальний об'єм інформації, що шифрується, $Z = n \cdot l \cdot r$;
- k – реєстраційний номер учасника;
- $n=2$ – число абонентів (станцій).

Задача № 2 (12 балів)

Оцініть складність та вартість здійснення атаки типу „повне розкриття”* на електронний цифровий підпис ДСТУ 4145-2002, вибравши порядок базової точки $n = 2^m$ згідно з таблицею.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
m	293	173	359	179	419	191	233	431	239	443	251	491	281	509

Потужність криптоаналітичної системи $\gamma = k \cdot 10^9$ операцій складання в групі точок еліптичної кривої за секунду, k – номер реєстрації учасника, $i = k(\bmod 14)$;

* Під атакою типу „повне розкриття” розуміється визначення особистих ключів ЕЦП та направлено розшифрування.

Задача № 3 (12 балів)

Виведіть формулу та оцініть ймовірність колізії P_k ключів, що виробляються за допомогою геш-функції. Вхідні дані, що гешуються є випадковими, довжина геш-функції $l = 160 + 4i$, число різних вхідних даних $k = 2^{64+2i}$, де i – реєстраційний номер учасника.

Додаткова задача № 4 (6 балів)

Викладіть сутність проблем або однієї проблеми захисту інформації в інформаційно-телекомунікаційних системах, які шляхи їх розв'язання? Що для цього необхідно зробити в Україні?

1.3 Конкурсні завдання з напрямку “Інформаційна безпека” з програмування

1. Реалізувати процедуру множення на скаляр крапки ЕК над $GF(2^m)$, використовуючи для цього найбільш ефективні з відомих вам алгоритмів виконання основних операцій над крапками ЕК.

У якості вихідних даних дозволено використовувати бібліотеку, що надає реалізацію основних операцій над елементами поля $GF(2^m)$ (додавання, ділення, множення, квадрат, обчислення мультиплікативної інверсії). При бажанні також дозволено використовувати будь-яку вихідну бібліотеку з аналогічною функціональністю.

Короткий опис наданого для всіх навчального варіанта відповідної бібліотеки видається у вигляді роздавального матеріалу. Приклади завдання параметрів ЕК над полем $GF(2^m)$ також наведені в роздавальному матеріалі.

2. Обґрунтовано вибрати тестові набори для перевірки працездатності реалізованих алгоритмів. Виконати тестування і продемонструвати його результати.

3. Оцінити обчислювальну складність обраних для реалізації алгоритмів. Оцінити час виконання реалізованих алгоритмів.

При оцінюванні результатів враховувались наступні фактори:

1. Ефективність обраних для реалізації методів і алгоритмів.
2. Ефективність реалізації (пріоритетним параметром вважати час виконання). Коректність оцінок обчислювальної складності.
3. Наявність (і обґрунтованість вибору) підготовлених тестових прикладів і результатів перевірки за ними.
4. Стиль програмування. Простота сприйняття програмного коду.
5. Уміння використовувати спеціалізовані бібліотеки різного типу і швидко розбиратися в правилах їх використання.
6. Час виконання завдання (необхідно укластися у фіксований проміжок часу).

Альтернативний варіант. Якщо учасника за якимись причинами не влаштовує платформа, під якою реалізована видана бібліотека, й у нього немає можливості використовувати будь-який інший варіант бібліотеки з аналогічною функціональністю (обраний за своїм бажанням), то як альтернативу можна спробувати виконати наступний варіант завдання. Самостійно реалізувати бібліотечний модуль, що провадить основні операції над елементами поля $GF(2^m)$ (додавання, ділення, множення, квадрат і обчислення мультиплікативної інверсії), не використовуючи які-небудь допоміжні бібліотеки. (Після цього, якщо дозволяє виділений час, можна спробувати виконати й основний варіант завдання.)

При оцінці результатів враховуються ті ж фактори (див. вище), а також кількість реалізованих у бібліотеці функцій і зручність їх використання.

2. Результати олімпіади та аналіз рівня підготовки конкурсантів

Всього в олімпіаді прийняла участь 39 осіб із 13 вузів України. За результатами олімпіади місця розподілилися таким чином:

Переможці з напрямку “Інформаційна безпека”(особистий залік)

- 1 місце – Михайленко Матвій Сергійович (ХНУРЕ) – 36 балів;
- 2 місце – Пометун Сергій Олександрович (НТУУ “КПІ”) – 34 бали;
- 3 місце – Фесенко Андрій В’ячеславович (НТУУ “КПІ”) – 34 бали.

Переможці з дисципліни “Захист інформації”(особистий залік)

- 1 місце – Борзяк Андрій Олександрович (ХІВПС ім. Кожедуба) – 40 балів;
- 2 місце – Литовський Віталій Дмитрович (ХНУРЕ) – 36 балів;
- 3 місце – Штанько Тарас Михайлович (ХІВПС ім. Кожедуба) – 35 балів.

Командні переможці з напрямку “Інформаційна безпека” з програмування

1 місце – команда в складі: Пометун Сергій Олександрович; Фесенко Андрій В’ячеславович; Яковлев Сергій Володимирович (НТУУ “КПІ”) – 128 балів;

2 місце – команда в складі: Бакулін Андрій Олександрович; Михайленко Матвій Сергійович; Мовчан Борис Григорович (ХНУРЕ) – 126 балів.

Командні переможці з дисципліни “Захист інформації”

1 місце – команда в складі: Артюхов Олександр Геннадійович, Михайленко Матвій Сергійович, Бакулін Андрій Олександрович (ХНУРЕ) – 132 бали;

2 місце – команда в складі: Зайнулін Костянтин Ігорович, Цимбалюк Олександр Сергійович, Макуха Андрій Володимирович (ХІВПС) – 78 балів;

3 місце – команда в складі: Підпригора Павло Олексійович, Дзигін Євгеній Валерійович, Богун Костянтин Олександрович (ДНУ) – 41 бал.

Крім того, були нагороджені:

За оригінальність при розв’язанні задач з дисципліни “Захист інформації” – Подоляник Олена Віталіївна та Циганенко Ольга Анатоліївна. Харківська філія Української академії банківської справи НБУ;

За власний проект – Кравченко Сергій Сергійович, Національний гірничий університет України, м. Дніпропетровськ.

До основних недоліків у підготовці конкурсантів слід віднести наступне.

- при вивченні тем, наприклад симетричних шифрів, студенти поверхньо засвоюють алгоритми, не володіють умінями оцінити імовірності перекриття шифрів, використовуючи для цього, наприклад, теорію колізій;

- не всі студенти можуть оцінити стійкість асиметричних алгоритмів (наприклад, в групах точок еліптичних кривих);

- не достатня, а то і слабка підготовка студентів з питань теорії еліптичних груп;

- значна кількість студентів мають слабкі знання з питань технічного захисту інформації, наприклад, з питання оцінювання спектральних складових при проходженні сигналів через нелінійні системи;

- більшість студентів не володіють методикою розв’язку наукових задач, де повинні мати місце постановка задачі, вибір критеріїв та показників, розв’язок та оцінка похибок, висновки та рекомендації.

Проведений другий тур підтвердив зрослий рівень підготовленості магістрів та студентів з проблемних питань захисту інформації з напрямку “Інформаційна безпека”, вони розуміють основні протиріччя та орієнтуються в напрямках їх розв’язку. Добре підготовлені, в тому числі математично, магістри та студенти НТУУ “КПІ” та ХНУРЕ. Зросла підготовленість студентів Національного гірничого університету. Крім того, студенти цих вузів вільно складають програми та вирішують за їх допомогою складні задачі з використанням ПЕОМ.

В номінації “Захист інформації”, в якій виступали студенти інших спеціальностей, але такі, що вивчали одну або дві загальноосвітні дисципліни з захисту інформації, студенти також продемонстрували зрослий рівень знань та умінь. Знають проблемні питання та новітні алгоритми криптографічного захисту інформації. Але математичний рівень підготовки в криптології, на наш погляд, недостатній, крім того, слід краще формувати і їх світогляд.

ПЕРСПЕКТИВНЫЕ СИММЕТРИЧНЫЕ ШИФРЫ И ИХ СВОЙСТВА

УДК 681.3.06: 519.248.681

Анализ свойств алгоритмов блочного симметричного шифрования (по результатам международного проекта NESSIE) / И.Д. Горбенко, Г.М. Гулак, Р.В. Олейников, В.И. Руженцев, М.С. Михайленко // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С. 7 – 24.

Сделан обзор блочных симметричных шифров, которые принимали участие в международном конкурсе криптографических примитивов NESSIE. Приведен выбор критериев оценки алгоритмов шифрования и выполнен сравнительный анализ алгоритмов шифрования по безусловным критериям. Сделан вывод, что безусловным критериям удовлетворяют только шифры AES, Camellia, Anubis и Shacal-2.

Табл. 4. Лп. 8. Библиогр.: 23 назв.

УДК 681.3.06: 519.248.681

Аналіз властивостей алгоритмів блокового симетричного шифрування (за результатами міжнародного проекту NESSIE) / І.Д. Горбенко, Г.М. Гулак, Р.В. Олійников, В.І. Руженцев, М.С. Михайленко // Радиотехника: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 7 – 24.

Зроблено огляд блокових симетричних шифрів, які брали участь у міжнародному конкурсі криптографічних примітивів NESSIE. Наведений вибір критеріїв оцінки алгоритмів шифрування та виконаний порівняльний аналіз алгоритмів шифрування за безумовними критеріями. Зроблений висновок про те, що безумовним критеріям задовольняють лише шифри AES, Camellia, Anubis та Shacal-2.

Табл. 4. Лп. 8. Бібліогр.: 23 назв.

UDC 681.3.06: 519.248.681

The analysis of properties of block symmetric enciphering algorithms (by results of international project NESSIE) / I.D. Gorbenko, G.M. Gulak, R.V. Oliynikov, V.I. Ruzencev, M.S. Mykhaylenko // Radiotekhnika. All-ukr. Sci. Interdep. Mag. 2005. N 141. P. 7 – 24.

An overview of block symmetric ciphers which participated in the international competition of cryptographic primitives NESSIE is performed. The criteria of choosing the enciphering algorithms estimation is presented and the enciphering algorithms comparative analysis is carried out by the unconditional criteria. It is inferred that only AES, Camellia, Anubis and Shacal-2 ciphers met the unconditional criteria.

4 tab. 8 fig. Ref.: 23 items.

УДК 681.3.06: 519.248.681

Сравнительный анализ алгоритмов блочного симметричного шифрования (по результатам выполнения проекта NESSIE) / И.Д. Горбенко, А.И. Пушкарёв, Р.В. Олейников, В.И. Руженцев, Ю.И. Горбенко, М.С. Михайленко // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С. 25 – 30.

Рассмотрены принципы использования критериев сравнения блочных симметричных шифров на основе нечётких множеств, которые принимают во внимание стойкость шифров к разным методам криптоанализа, а также юридические условия разрешения экспорта, импорта и использования алгоритма. С использованием этих критериев выполнено сравнение алгоритмов, которые стали финалистами международного конкурса криптографических примитивов NESSIE. Сделан вывод о том, что для использования в Украине целесообразно рекомендовать блочный шифр Rijndael.

Табл. 10. Библиогр.: 4 назв.

УДК 681.3.06: 519.248.681

Порівняльний аналіз алгоритмів блокового симетричного шифрування (за результатами виконання проекту NESSIE) / І.Д. Горбенко, А.І. Пушкарьов, Р.В. Олійников, В.І. Руженцев, Ю.І. Горбенко, М.С. Михайленко // Радиотехника: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 25 – 30.

Розглянуто принципи застосування критеріїв порівняння блокових симетричних шифрів на основі нечітких множин, які приймають до уваги стійкість шифрів до різних методів криптоаналізу, а також юридичні умови дозволу експорту, імпорту і використання алгоритму. З використанням цих критеріїв виконано порівняння алгоритмів, що стали фіналістами міжнародного конкурсу криптографічних примітивів NESSIE. Зроблений висновок про те, що для використання в Україні доцільно рекомендувати блочний шифр Rijndael.

Табл. 10. Бібліогр.: 4 назви.

UDC 681.3.06

The comparative analysis of algorithms of block symmetric enciphering (by results of performance of project NESSIE) / I.D. Gorbenko, A.I. Pushkaryov, R.V. Oliynikov, V.I. Ruzhentsev, U.I. Gorbenko, M.S. Mykhaylenko // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N 140. P. 25–30.

Principles of use of block symmetric ciphers comparison criteria are considered on the basis of indistinct sets which consider stability of ciphers to different methods of cryptanalysis, and also legal conditions of authorization of export, import and use of algorithm. Comparison of algorithms which became finalists of the international competition of cryptographic primitives for use in Ukraine NESSIE is executed with the use of these criteria. It is inferred that it is expedient to recommend Rijndael block code.

10 tab. Ref.: 4 items.

УДК 621.391: 519.7

Критерий примитивности группы подстановок, порожденной раундовыми шифрующими преобразованиями шифра Фейстеля / А.Н. Алексейчук, Л.В. Скрипник // Радиотехника: Всеукр. межвед. научно-техн. сб. 2005. Вип. 141. С. 31 – 39.

Получены необходимые и достаточные условия, при которых раундовые шифрующие преобразования шифра Фейстеля порождают примитивную группу подстановок, что исключает возможность проведения на шифр ряда известных алгебраических атак. Показано, что группа, порожденная раундовыми преобразованиями шифра GOST, является примитивной.

Библиогр.: 14 назв.

УДК 621.391: 519.7

Критерій примітивності групи підстановок, що породженої раундовими шифруючими перетвореннями шифру Фейстеля / А.М. Олексійчук, Л.В. Скрипник // Радіотехніка: Всеукр. міжвід. наук-техн. зб. 2005. Вип. 141. С. 31 – 39.

Отримані необхідні та достатні умови, за якими раундові шифруючі перетворення шифру Фейстеля породжують примітивну групу підстановок, що виключає можливість проведення на шифр ряду відомих алгебраїчних атак. Показано, що група, породжена раундовими перетвореннями шифру GOST, є примітивною.

Бібліогр.: 14 назв.

UDC 621.391: 519.7

The primitivity criterion of the permutation group generated by the encryption round functions of Feistel cipher / A.N. Alekseychuk, L.V. Skripnik // All-ukr. Sci. Interdep. Mag. 2005. N 141. P. 31 – 39.

Necessary and sufficient condition under which the round encryption functions of Feistel cipher generate a primitive permutation group are defined, they exclude the possibility to realize a number of known algebraic attack on the cipher.

Ref. 14 items.

УДК 681.3.06: 519.248.681

Математическая модель перекрытия шифра для поточных криптосистем / Ю.И. Горбенко // Радиотехника: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 40 – 53.

Разработана и проанализирована математическая модель перекрытия поточного шифра. Приведены методы оценки вероятности перекрытия гам шифрующих. Сделан сравнительный анализ сложности и точности оценки параметров перекрытия шифра. Выполнено практическое исследование вероятностей перекрытия шифра.

Табл. 7. Ил. 1. Библиогр.: 13 назв.

УДК 681.3.06: 519.248.681

Математична модель перекриття шифру для поточкових криптосистем / Ю.І. Горбенко // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 40 – 53.

Розроблена та проаналізована математична модель перекриття поточкового шифру. Наведені методи оцінки ймовірності перекриття гам шифруючих. Зроблений порівняльний аналіз складності та точності оцінки параметрів перекриття шифру. Виконане практичне дослідження ймовірностей перекриття шифру.

Табл. 7. Іл. 1. Бібліогр.: 13 назв.

UDC 681.3.06: 519.248.681

The mathematical model of cipher overlapping for stream-oriented cryptosystems / Y.I. Gorbenko // Radiotekhnika: All-ukr. Sci. Interdep. Mag. 2005. N 140. P. 40 – 53.

The mathematical model of cipher overlapping for stream-oriented cryptosystems is developed and analyzed. The estimation methods of cipher gammas overlapping probability are given. The comparison analysis of complexity and precision for cipher overlapping parameters is carried out. The practical research of cipher overlapping is executed.

7 tab. 1 fig. Ref.: 13 items

УДК 681.3.06: 519.248.681

Исследование статистических свойств схемы разворачивания ключей FIPS-197 / М.Ф. Бондаренко, А.Н. Лепеха // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып. 141. С. 54 – 63.

Проведено исследование статистических свойств схемы разворачивания ключей FIPS-197 с целью выявления скрытых “аномалий” в формировании цикловых подключей, которые нельзя выявить аналитическим путем. Дан углубленный анализ особенностей функционирования схемы разворачивания ключей FIPS-197.

Табл. 3. Библиогр.: 16

УДК 681.3.06: 519.248.681

Дослідження статистичних властивостей схеми розгортання ключів FIPS-197 / М.Ф. Бондаренко, О.М. Лепеха // Радиотехника: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 54 – 63.

Проведене дослідження статистичних властивостей схеми розгортання ключів FIPS-197 з метою виявлення схованих “аномалій” у формуванні циклових підключів, які не можна виявити аналітичним шляхом. Дано поглиблений аналіз особливостей функціонування схеми розгортання ключів FIPS-197.

Табл. 3. Бібліогр.: 16

UDC 681.3.06: 519.248.681

Research on FIPS-197 key schedule statistical properties. / M.F. Bondarenko, A.N. Lepekha // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N 140. P. 54 – 63.

Research on FIPS-197 key schedule statistical properties is performed in an attempt to detect latent “anomalies” in cyclic subkeys formation which cannot be revealed analytically. The detailed analysis of FIPS-197 key schedule operation is given.

3 tab. Ref.: 16 items.

УДК 681.3.06: 519.248.681

Сравнительный анализ схем разворачивания ключей блочных симметричных шифров / А.Н. Лепеха // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып. 141. С. 64 – 69.

Предложен метод классификации схем разворачивания ключей блочных симметричных шифров. Сделан сравнительный анализ тактико-технических характеристик схем разворачивания ключей ряда современных блочных шифров, представленных в проекте NESSIE.

Табл. 2. Библиогр.: 7 назв.

УДК 681.3.06: 519.248.681

Порівняльний аналіз схем розгортання ключів блочних симетричних шифрів / О.М. Лепеха // Радиотехника: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 64 – 69.

Запропоновано метод класифікації схем розгортання ключів блочних симетричних шифрів. Зроблено порівняльний аналіз тактико-технічних характеристик схем розгортання ключів ряду сучасних блочних шифрів, представлених у проекті NESSIE.

Табл. 2. Бібліогр.: 7.

UDC 681.3.06: 519.248.681

Comparative analysis of block symmetric ciphers key schedules / A.N. Lepekha // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N 141. P. 64 – 69.

A method of block symmetric ciphers key schedules classification is offered. Comparative analysis of tactical and technological characteristics of a number of modern block ciphers key schedules presented in NESSIE project is carried out.

2 tab. Ref.: 7 items.

УДК 681.3.06:519.248.681

Современные подходы к оценке линейной сложности псевдослучайных последовательностей. Усовершенствованная методика тестирования линейной сложности на основе NIST STS / С.Ю. Орлова // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып. 141. С. 70 – 81.

Рассматриваются некоторые общие понятия и определения, касающиеся оценки линейной сложности псевдослучайных двоичных последовательностей. Описан статистический подход к анализу линейной сложности, предложенный в NIST STS.

Учитывая современные тенденции при разработке поточных шифров, получено формальное выражение для оценки линейной сложности выходных последовательностей линейных рекуррентных регистров, функционирующих в расширенных полях. На основе результатов исследования современных поточных шифров предложена усовершенствованная методика статистической оценки линейной сложности шифрующих последовательностей.

Табл. 2, Ил. 3. Библиогр.: 14 назв.

УДК 681.3.06:519.248.681

Сучасні підходи до оцінки лінійної складності псевдовипадкових послідовностей. Удосконалена методика тестування лінійної складності на основі NIST STS / С.Ю. Орлова // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. С. 70 – 81.

Розглядаються деякі загальні поняття та визначення, які стосуються оцінки лінійної складності псевдовипадкових двійкових послідовностей. Описано статистичний підхід до аналізу лінійної складності, запропонований у NIST STS.

З огляду на сучасні тенденції при розробці поточкових шифрів, здобуто формальний вираз для оцінки лінійної складності вихідних послідовностей лінійних рекурентних регістрів, які функціонують у розширених полях. На основі результатів дослідження сучасних поточкових шифрів запропоновано вдосконалену методику статистичної оцінки лінійної складності шифруючих послідовностей.

Табл. 2. Іл. 3. Бібліогр.: 14 назв.

UDC 681.3.06:519.248.681

Modern approaches to evaluation of the pseudo-random sequences linear complexity. Improved methods of linear complexity testing based on NIST STS / S.Y. Orlova // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. № 141. P. 70 – 81.

Some general concepts and definitions on evaluation of pseudo-random binary sequences linear complexity are considered. The statistical approach to the linear complexity analysis offered in NIST STS is described.

Considering modern tendencies when designing the stream ciphers. The formal expression is derived to evaluate the linear complexity of output sequences of the linear shift registers which operate in extended fields. The improved methods of statistical estimation of running key linear complexity is offered as the result of stream ciphers researches.

2 tab. 3 fig. Ref.: 14 items.

КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ И ГИПЕРЭЛЛИПТИЧЕСКИХ КРИВЫХ

УДК 681.3.06:519.248.681

Алгоритмы скалярного умножения в группе точек эллиптической кривой и некоторые их модификации / А.Д. Тевяшев, С.И. Збитнев, В.Ю. Ковтун // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 82 – 96.

Проводится исследование вычислительной сложности известных алгоритмов операции скалярного умножения в группе точек эллиптической кривой, которая применяется в основных криптографических примитивах. Предлагаются авторские модификации исследованных алгоритмов с уменьшенной вычислительной сложностью. Даются рекомендации по их применению.

Табл. 3. Библиогр.: 37 назв.

УДК 681.3.06:519.248.681

Алгоритми скалярного добутку у групі точок еліптичної кривої та деякі їх модифікації / А.Д. Тевяшев, С.І. Збітнев, В.Ю. Ковтун // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 82 – 96.

Проводяться дослідження обчислювальної складності відомих алгоритмів операції скалярного добутку, яка застосовується в основних криптографічних примітивах. Пропонуються модифікації досліджених алгоритмів зі зменшеною обчислювальною складністю. Даються рекомендації щодо їх застосування.

Табл. 3. Бібліогр.: 37 назви.

UDC 681.3.06:519.248.681

Scalar multiplication algorithms in group of elliptic curve points and some their modifications / A.D. Tevashev, S.I. Zbitnev, V.Jur. Kovtun, // Radiotekhnika: all-Ukr. Sci. Interdep. Mag. 2005. N 141. C. 82 – 96.

In this paper, computation complexities are explored to scalar multiplication algorithms of elliptic curve points what are used in main cryptographic primitives. Authors modifications considered algorithms with decreased computation complexity are proposed. Recommendations for usage of considered and proposed algorithms are given too.

3 tab. Ref.: 37 items

УДК 681.3.06:519.248.681

Арифметические операции на эллиптической кривой над двончным полем в проективных координатах / С.И. Збитнев, В.Ю. Ковтун, О.Е. Илясова // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 97 – 107.

В работе проводится анализ арифметических операций на эллиптической кривой в проективных координатах. Предлагаются улучшенные выражения для вычисления результатов арифметических операций в проективных координатах, а также даются рекомендации по их применению.

Табл. 1. Библиогр.: 21 назв.

УДК 681.3.06:519.248.681

Арифметичні операції на еліптичній кривій над двійковим полем у проєктивних координатах / С.І.Збітнев, В.Ю. Ковтун, О.Є. Ільясова // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 97–107.

У роботі проводиться аналіз арифметичних операцій на еліптичній кривій у проєктивних координатах. Пропонуються покращенні вирази для обчислення результатів арифметичних операцій у проєктивних координатах. Також даються рекомендації щодо їх застосування.

Табл. 1. Бібліогр.: 21 назви.

UDC 681.3.06:519.248.681

Arithmetic operations on elliptic curve over binary field in projective coordinates / S.I. Zbitnev, V.Jur. Kovtun, O.E. Ilysova // Radiotekhnika: all-Ukr. Sci. Interdep. Mag. 2005. N 141. P. 97–107.

At this paper analyse arithmetic operations on elliptic curve in projective coordinates are considered. Efficient expressions for arithmetic computing in projective coordinates are proposed and recommendations for their usage are given too.

1 tab., Ref.: 21 items

УДК 681.3.06

Преобразования в оптимально расширенных полях и их применение в криптографии / Д.С. Балагура // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып. 141. С. 108–115.

Рассматривается математика в оптимальных расширенных полях, приводятся алгоритмы выполнения основных операций и их сложность. Рассчитываются константы инверсий для приведения по модулю в некоторых оптимальных расширенных полях. Приводятся результаты анализа практической реализации алгоритмов. Проводится сравнение полученных результатов с результатами для расширенных полей.

Табл. 4. Ил. 0. Библиогр.: 10 назв.

УДК 681.3.06

Перетворення у оптимально розширених полях та їх застосування у криптографії / Д.С. Балагура // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 108–115.

Розглядається математика в оптимальних розширених полях, наводяться алгоритми виконання базових операцій та їх складність. Розраховуються константи для приведення за модулем у деяких оптимальних розширених полях. Наводяться результати аналізу практичної реалізації алгоритмів. Проводиться порівняння одержаних результатів з результатами для розширених полів.

Табл. 4. Іл. 0. Бібліогр.: 10 назв.

UDC 681.3.06

Transformation in optimal extended fields and their use in cryptography / D.S. Balagura // Radiotekhnika: All-ukr. Sci. Interdep. Mag. 2005. N 000. P. 108–115.

Mathematic in optimal extended fields is considered. Algorithms of execution of base operations are given. Constants of inversions for module reduction in some optimal extended fields are computed. The results of analysis of practical realization of algorithms are given. The comparison of received results with results for extended fields are made.

4 tab., 0 fig., Ref.: 10 items.

УДК 681.3.06:519.248.681

Алгоритмы решения задачи дискретного логарифмирования в якобиане гиперэллиптической кривой / С.И. Збитнев, В.Ю. Ковтун, Д.В. Шевченко // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 116–132.

В работе исследуются известные алгоритмы решения задачи дискретного логарифмирования в якобиане гиперэллиптической кривой. Производится оценка и сравнение их вычислительной сложности. Приводятся перспективные направления в уменьшении вычислительной сложности рассмотренных алгоритмов.

Табл. 2. Библиогр.: 54 назв.

УДК 681.3.06:519.248.681

Алгоритми рішення задачі дискретного логарифмування в якобіані гіпереліптичної кривої / С.І.Збітнев, В.Ю. Ковтун, Д.В. Шевченко // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 116–132.

У роботі досліджуються відомі алгоритми рішення задачі дискретного логарифмування в якобіані гіпереліптичної кривої. Виконується оцінка і порівняння їх обчислювальна складність. Приводяться перспективні напрямки у зменшенні обчислювальної складності розглянутих алгоритмів.

Табл. 2. Бібліогр.: 54 назви.

UDC 681.3.06:519.248.681

Arithmetic operations on elliptic curve over binary field in projective coordinates / S.I. Zbitnev, V.Jur. Kovtun, D.V. Shevchenko // Radiotekhnika: all-Ukr. Sci. Interdep. Mag. 2005. N 141. P. 116 – 132.

In this paper, known algorithms are explored to solve discrete logarithm problem in the jacobian of hyperelliptic curve. Estimation and comparison of their computation complexity have done. Provide guidelines in decreasing computation complexity of someone give too.

2 tab. Ref.: 54 items

УДК 681.3.06

Выбор эллиптических кривых для криптографических применений / В.И. Долгов // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып. 141. С. 133 – 143.

Обсуждаются условия, которым должны удовлетворять эллиптические кривые, для того чтобы обеспечить высокие показатели стойкости при криптографических применениях. Внимание сосредотачивается на разъяснении понятий сингулярности и суперсингулярности эллиптических кривых и связанных с ними теоретических положений. Излагается сущность алгоритма сведения, предложенного А. Менезисом, Т. Окомото и С. Венстоном, выполняется теоретическое обоснование MOV условия и разъяснение связанных с ним понятий и определений. Для этого приводятся некоторые важные результаты из теории полей Галуа, в частности, следствия, которые следуют из теоремы Мордела и касаются гомоморфизмов полей, разъясняются такие понятия, как изогении, кольца эндоморфизмов, эллиптические кривые с комплексным умножением, на основе которых формируются определения суперсингулярности.

Ил. 4. Библиогр.: 11 назв.

УДК 681.3.06

Вибір еліптичних кривих до криптографічних застосувань / В.І. Долгов // Радіотехніка: Всеукр. міжв. наук.-техн. зб. 2005. Вип. 141. С. 133 – 143.

Обговорюються умови, котрим повинні задовольняти еліптичні криві, для того щоб забезпечити високі показники стійкості в криптографічних прикладеннях. Увага зосереджується на роз'ясненні понять сингулярності та суперсингулярності еліптичних кривих й пов'язаних з ними теоретичних положень. Викладається сутність алгоритму зведення, запропонованого А. Менезісом, Т. Окомото та С. Венстоном, виконується теоретичне обґрунтування MOV умови й роз'яснення пов'язаних з нею понять й означень. Для цього наводяться деякі важливі результати з теорії полів Галуа, зокрема наслідки, що випливають з теореми Мордела і стосуються гомоморфізмів полів, роз'яснюючи такі поняття, як ізогенії, кільця ендоморфізмів, еліптичні криві з комплексним множенням, на основі яких формується означення суперсингулярності.

Ил. 4. Библиогр.: 11 назв.

UDC 681.3.06

Choice of elliptic curves for cryptographic applications / V.I. Dolgov // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N141. P. 133 – 143.

Conditions are discussed, with which should satisfy elliptic curves to provide high parameters of resistance cryptographic applications. The attention is focused to an explanation of concepts singularity and supersingularity the theoretical positions elliptic curve and connected to them. The essence of algorithm of the data offered by A.Menezis, T.Okamoto and S.Venston is stated, theoretical substantiation MOV of a condition and an explanation of the concepts connected to it and definitions is carried out. For this purpose some important results from the theory of Galois fields are resulted, in particular, consequences which follow from Mordell's theorem and concern fields homomorphisms, such concepts, as isogenies, rings endomorphism, elliptic curves with complex multiplication on the basis of which definitions supersingularity are formed are explained.

4 fig. Ref.: 11 items.

МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

УДК 621.3.06

Основные положения математического аппарата субъективной логики и его практическое применение для оценки уровня зрелости систем обеспечения безопасности информации / А.В. Потий, А.В. Леншин // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып. 141. С. 144 – 160.

Излагаются основные положения и операторы субъективной логики. Определяется сущность неопределенных вероятностей, описывается модель доверия. Рассматривается геометрическая интерпретация пространства мнений. Обосновывается возможность использования данного аппарата для оценки зрелости систем обеспечения безопасности информации.

Библиогр.: 20 назв.

УДК 621.3.06

Основні положення математичного апарату суб'єктивної логіки та його застосування для оцінки рівня зрілості систем забезпечення безпеки інформації / О.В. Потій, А.В. Леншин // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 144 – 160.

Викладаються основні положення та оператори суб'єктивної логіки. Визначається сутність невизначених імовірностей, описується модель довіри. Розглядається геометрична інтерпретація простору думок. Обґрунтовано можливість використання даного апарату для оцінки зрілості систем забезпечення безпеки інформації.

Бібліогр.: 20 назв.

UDC 621.3.06

General mathematical mechanism foundations of Subjective Logic and its application in information security system maturity level evaluating / A.V. Potiy, A.V. Lenshin // Radoitekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N141. P. 144 – 160.

General foundations and operators of Subjective Logic are expounded. Uncertainly probabilities essence is defined, maturity model is described. Geometric interpretation of opinion space is considered. Using possibility of this mechanism in information security system maturity level evaluating is given.

Ref. 20 items.

УДК 681.3.06

TWO-TRACK-MAC алгоритм високої секретності / І.Д. Горбенко, О.Г. Халімов // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 161 – 167.

Задачей данной статьи является анализ основных характеристик Two-Track-MAC кодов, изложение алгоритма построения и оценка секретности. С этой целью приводятся требования к кодам аутентификации сообщений. Рассматривается алгоритм построения ТТМАС с использованием хэш-функции RIPEMD-160. Выполняется оценка секретность ТТМАС-кодов.

УДК 681.3.06

TWO-TRACK-MAC алгоритм високої таємності / І.Д. Горбенко, О.Г. Халімов // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 161 – 167.

Задачею даної статті є аналіз основних характеристик Two-Track-MAC кодів, виклад алгоритму побудови й оцінка таємності. З цією метою приводяться вимоги до кодів аутентифікації повідомлень. Розглядається алгоритм побудови ТТМАС з використанням геш-функції RIPEMD-160. Виконано оцінку таємності ТТМАС-кодів.

UDC 681.3.06

TWO-TRACK-MAC Algorithm of high privacy / I.D. Gorbenko. O.G.Halimov // Radoitekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N. 141. P. 161 – 167.

Task of given article is the analysis of basic characteristics Two-Track-MAC of codes, a statement of algorithm of construction and an estimation of privacy. With this purpose requirements are resulted in authentication codes of messages. The algorithm of construction ТТМАС with use of hash-function RIPEMD-160 is considered. The estimation privacy of ТТМАС-codes is carried out.

УДК 681.3.06: 519.248.681

Функциональность и характеристики криптографической библиотеки на языке Java / М.Ф. Бондаренко, И.Д. Горбенко, О.П. Мелецкий // Радіотехніка. Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 168 – 173.

Рассмотрена основная концепция системы безопасности Java. Рассмотрены основные принципы построения криптографической библиотеки на платформе Java. Проведена систематизация основных требований и критериев библиотек такого типа.

Табл. 3. Библиогр.: 5 назв.

УДК 681.3.06: 519.248.681

Функціональність та властивості криптографічної бібліотеки на мові Java. // М.Ф. Бондаренко, І. Д. Горбенко, О.П. Мелецький // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 168 – 173.

Розглянута основна концепція системи безпеки Java. Розглянуті основні принципи побудови криптографічної бібліотеки на платформі Java. Проведена систематизація основних вимог та критеріїв бібліотек такого типу.

Табл. 3. Бібліогр.: 5 назв.

UDC 681.3.06: 519.248.681

Functionality and behavior of cryptography library based on Java language. / M.F. Bondarenko, I.D. Gorbchenko, A.P. Melezky // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N 141. P. C. 168 – 173.

Main concept of Java security was viewed. Main creating principle's of cryptography library of the Java platform was viewed. A systematization was conducting of a main library requirements and criteria's.

3 tab. Ref.: 5 items.

УДК 621.3.06

Проектирование электронных систем со встроенным механизмом обеспечения безопасности /

В.А. Горбачев, В.В. Степаненко // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С. 174 – 180.

На основании критерия выполнения политики безопасности в КС в работе предлагается метод проектирования электронных систем со встроенным механизмом реализации политики безопасности. В соответствии с предложенным методом проектируемая система должна содержать монитор безопасности потоков для выявления и блокирования не специфицированных потоков, а также реализации траектории безопасных состояний.

На примере проектирования контроллера показано реализацию всех этапов предложенного метода, а также показано применение этого метода при проектировании комплексных СЗИ.

Табл. 2. Ил. 4. Библиогр.: 2 назв.

УДК 621.3.06

Проектування електронних систем з вбудованим механізмом реалізації безпеки / В.О. Горбачов,

В.В. Степаненко // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С. 174 – 180.

На підставі критерія виконання політики безпеки в КС у роботі пропонується метод проектування електронних систем з вбудованим механізмом реалізації політики безпеки. Відповідно до запропонованого методу, спроектована система повинна містити монітор безпеки потоків, з метою виявлення і блокування не специфікованих потоків, а також реалізації контролю траєкторії безпечних станів.

На прикладі проектування контролера показано реалізацію всіх етапів запропонованого методу, а також показано застосовність цього методу при проектуванні комплексних СЗІ.

Табл. 2. Іл. 4. Бібліогр.: 2 назв.

UDC 621.3.06

Designing electronic systems embedded mechanism of security provision / V.A Gorbachev, V.V. Stepanenko //

Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N 141. P. 174 – 180.

In this article on the basis of security policy performance criteria in computer networks the method of the electronic systems design with the built-in mechanism of security policy realization is offered. According to the offered method the system should contain the monitor of the security streams to reveal and block the not specified streams and realizations of the security conditions trajectory.

By the example of the controller design the realization of all stages of the offered method and application of this method in complex information security system design are shown.

2 tab. 4 fig. Ref.: 2 items.

КРИТЕРИИ, ПОКАЗАТЕЛИ И МЕТОДИКИ ОЦЕНКИ В КРИПТОГРАФИИ

УДК 681.3.067:681.3.016

Вероятностные характеристики схем генерации ключей с неравновероятным распределением /

Л.В. Ковальчук, С.В. Мельник, В.Т. Бездетный // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С. 181 – 187.

Дана оценка вероятности применения методов направленного перебора ключей при известном статистическом распределении символов выходной последовательности генераторов ключевых данных. Рассматривается вопрос о зависимости между количеством попыток и вероятностью успеха при известном методе направленного перебора ключей (в порядке убывания вероятности ключа).

Ил. 3. Библиогр.: 3 назв.

УДК 681.3.067:681.3.016

Ймовірнісні характеристики схем генерації ключів з нерівноймовірним розподілом / Л.В. Ковальчук,

С.В. Мельник, В.Т. Бездетный // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С. 181 – 187.

Подано оцінку ймовірності застосування методів спрямованого перебору ключів при відомому статистичному розподілі символів вихідної послідовності генераторів ключових даних. Розглядається питання щодо залежності між кількістю спроб і ймовірністю успіху при відомому методі спрямованого перебору ключів (у порядку убывання ймовірності ключа).

Іл. 3. Бібліогр.: 3 назв.

UDC 681.3.067:681.3.016

Probability characteristics of key generation circuits with non equiprobable distribution / L.V. Kovalchuk, S.V. Melnik, V.T. Bezdetny // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. № 141. P. 181 – 187.

Estimation of probability to use methods of the directed keys finding with known statistical distribution of symbols of the key data generators output sequence is given. The problem of relationship between an amount of attempts and success probability using the method of the directed keys finding (in order of decreasing probability is considered).

3 fig. Ref.: 3 items.

УДК 621.3.06

Метод формирования профилей статистического тестирования для оценки свойств случайных последовательностей / Е.В. Попович // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С. 188 – 195.

Рассматривается метод построения системы статистического контроля свойств случайных последовательностей – профиля статистического тестирования. В основу метода положен способ формализации требований к свойствам случайных последовательностей и способ оценки чувствительности статистических тестов. Предлагается порядок оценки характеристик профиля статистического тестирования.

Табл. 2. Ил. 2. Библиогр.: 15 назв.

УДК 621.3.06

Метод формування профілів статистичного тестування для оцінки властивостей випадкових послідовностей / Є.В. Попович // Радіотехніка: Всеукр. міжвід. науково.-техн. сб. 2005. Вип. 141. С. 188 – 195.

Розглядається метод побудови системи статистичного контролю властивостей випадкових послідовностей – профілю статистичного тестування. Метод ґрунтується на застосуванні способу формалізації вимог до властивостей випадкових послідовностей та способі оцінки чутливості статистичних тестів. Пропонується порядок оцінки характеристик профілю статистичного тестування.

Табл. 2. Рис. 2. Бібліогр.: 15 назв.

UDC 621.3.06

Profiles forming method of statistical testing for random sequences properties evaluation / E.V.Popovitch // Radiotekhnika: All-Ukr. Interdep. Mag. 2005. P. 188 – 195.

The construction method for system of statistical verification properties of random sequences called statistical testing profile is considered. Method is based on a method of formalization requirements to properties of random sequences and other method of sensitivity evaluation of statistical tests. The order of characteristics evaluation of statistical testing profile is proposed.

2 Tab. 2 fig. Ref.: 15 items.

УДК 621.3.06

Критерий оценки требований к генераторам случайных последовательностей в криптографических системах / Е.В. Попович, Ю.И. Горбенко // Радиотехника: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С.196 – 204.

Рассматривается способ оценки требований, выдвигаемых к генераторам случайных последовательностей. В качестве критерия предлагается использовать коэффициент эффективности криптографической атаки. Проводится анализ зависимости данного коэффициента от требований, выдвинутых к генератору.

Табл. 2. Ил. 3. Библиогр.: 14 назв.

УДК 621.3.06

Критерій оцінки вимог до генераторів випадкових послідовностей в криптографічних системах / Є.В. Попович, Ю.І. Горбенко // Радіотехніка: Всеукр. міжвед. науч.-техн. сб. 2005. Вип. 141. С.196 – 204.

Розглядається спосіб оцінки вимог, що висуваються до генераторів випадкових послідовностей. В якості критерію пропонується застосовувати коефіцієнт ефективності криптографічної атаки. Проводиться аналіз залежності даного коефіцієнту від вимог, що висуваються до генератору.

Табл. 2. Ил. 3. Бібліогр.: 14 назв.

UDC 621.3.06

Evaluation criterion of requirements to pseudorandom sequences generators in cryptographic systems / E.V.Popovitch, U.I. Gorbenco // Radiotekhnika: All-Ukr. interdep Mag. 2005.141. P. 196 – 204.

The method of requirements evaluation to pseudorandom sequences generators is considered in the article. It is proposed to use an effectiveness coefficient of cryptographic attack as a criterion. Dependence analysis of this coefficient based on requirements to generator is carried out.

2. Tab. 3. fig. Ref.: 14 items.

УДК 621.391.7

Фазовые системы передачи информации на основе численной системы остаточных классов / Ю.В. Куц, Л.М. Щербак // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып.141. С.205 – 216.

Приведены результаты исследования использования системы остаточных классов в фазовых системах передачи информации для задач кодирования информации, выявления и исправления ошибок при передаче цифровых данных, применения фазоманипулированных сигналов при передаче, в том числе и результаты имитационного моделирования.

Табл.3. Ил.8. Библиогр.: 8 назв.

УДК 621.391.7

Фазові системи передачі інформації на основі чисельної системи залишкових класів / Ю.В. Куц, Л. М. Щербак // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип.141. С. 205 – 216.

Наведені результати дослідження використання системи залишкових класів у фазових системах передачі інформації для задач кодування інформації, виявлення і виправлення помилок при передачі цифрових даних, застосування фазоманіпульованих сигналів при передачі, в тому числі і результати імітаційного моделювання.

Табл.3, Іл.8. Бібліогр.: 8 назв.

UDC 621.391.7

Phase systems of the information transfer based on numerical system of residual classes / U. V. Kuts, L.N. Scherbak // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2005. № 141. P. 205 – 216.

The results of research of system using with residual classes in phase systems for the information transfer at tasks of the information coding, revealing and mistakes correction, application phase shift keying signals by transfer, including results of imitating modeling, are given.

3 tabs. 8 figs. Ref.: 8 items.

УДК 681.04

Универсальные алгоритмы сжатия табличных цифровых данных результатов выполнения арифметических операций в системе остаточных классов / В.А.Краснобаев, Я.В. Илюшко, А.А. Замула // Радиотехника. Всеукр. межвед. науч.-техн. сб. 2005. Вып.141. С. 217 – 225.

Рассматриваются табличные универсальные алгоритмы (УА) реализации арифметических операций сложения, вычитания и умножения в непозиционной системе счисления в остаточных классах (СОК). Представленные в статье УА основываются на использовании кода табличного умножения. Приведены примеры конкретной реализации предложенных алгоритмов в СОК

Табл. 11. Библиогр.: 5 назв.

УДК 681.04

Універсальні алгоритми стиску табличних цифрових даних результатів виконання арифметичних операцій у системі залишкових класів / В.А.Краснобаев, Я.В. Ілюшко, О.А. Замула // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип.141. С. 217 – 225.

Розглядаються табличні універсальні алгоритми (УА) реалізації арифметичних операцій складання, віднімання та множення у непозиційній системі числення у залишкових класах (СЗК). Представлені у статті УА базуються на використанні кода табличного множення. Приведені приклади конкретної реалізації запропонованих алгоритмів у СЗК.

Табл. 11. Бібліогр.: 5 назв.

UDC 681.04

Universal algorithms of compression of tabulated figures of results of performance of arithmetic operations in system of residual classes / V.A.Krasnobaev, Y.V. Ilyshko, A. A Zamula // Radiotekhnika: All-Ukr. Sci Interdep. Mag. 2005. № 141.P. 217 – 225.

The tabulated universal algorithms (UA) of realization of arithmetic operations of addition, subtraction and multiplication in a not item notation in residual classes (SRC) are considered (examined). Submitted in clause UA are based on use of a code of tabulated multiplication. The examples of concrete realization of the offered algorithms in SRC are given.

Tabl. 11. Ref.: 5 items.

УДК 681.3.07

Опыт подготовки и результаты проведения VI Всеукраинской олимпиады по направлению “Информационная безопасность” / И.Д. Горбенко, О.А. Мельникова, А.И. Шумов // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2005. Вып. 141. С. 226 – 230.

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 VI Всеукраїнської олімпіади з напрямку "Інформаційна безпека" / І.Д. Горбенко, О.А. Мельникова, О.І. Шумов // Радіотехніка: Всеукр. міжвід. наук.-техн. зб. 2005. Вип. 141. С. 226 – 230.

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 VI all-Ukrainian competition in the field of information security / I.D. Gorbenko, O.A. Melnikova, A.I. Shumov // Radoitekhnika: All-Ukr. Sci. Interdep. Mag. 2005. N. 141. P. 226 – 230.

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 competitiveness level of training at the latest competitions is analyzed and typical shortages in solving the competitive tasks are pointed out.

**ЗБІРНИК НАУКОВИХ ПРАЦЬ
РАДІОТЕХНІКА**

Випуск 141

Російською та українською мовами

**СБОРНИК НАУЧНЫХ ТРУДОВ
РАДИОТЕХНИКА**

Выпуск 141

На русском и украинском языках

Коректор Л.І. Сашенко

Виконавець комп'ютерної верстки І.І. Бутримов

Підп. до друку 20.02.2005. Формат 60х90/8. Папір офсет. Гарнітура Таймс. Друк. ризограф.
Ум. друк. арк. 13,7. Обл.-вид. арк. 15,3. Тираж 300 прим. Зам. № 49. Ціна договір.

Харківський національний університет радіоелектроніки (ХНУРЕ)
Просп. Леніна, 14, Харків, 61166.

Оригінал-макет підготовлено і збірник надруковано у ПФ „Колегіум”, тел. (0572)28-53-74.