МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

РАДИОТЕХНИКА

Всеукраинский межведомственный научно-технический сборник

ТЕМАТИЧЕСКИЙ ВЫПУСК
«ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

Основан в 1965 г.

В Ы П У С К 1 8 9

Харків
Харківський національний
університет радіоелектроніки
2017

УДК 621.3

Сборник включен в список специальных изданий ВАК Украины по физико-математическим и техническим наукам.
Регистрационное свидетельство КВ № 12098-969 ПР от 14. 12. 2006.
Ответственность за содержание статей несут авторы.

Ответственные за выпуск: И.Д. Горбенко, д-р техн. наук, проф,
А.И. Лучанинов, д-р физ.-мат. наук, проф.
Технический секретарь Е.С. Полякова

Рекомендовано Ученым советом Харьковского национального университета радиоэлектроники, протокол № 57 от 29.06.2017.

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

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

СОДЕРЖАНИЕ

МЕТОДЫ ЗАЩИЩЕННОЙ ПЕРЕДАЧИ, ОБРАБОТКИ И АНАЛИЗА ДАННЫХ

МЕХАНИЗМЫ И МЕТОДЫ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

РАДИОТЕХНИЧЕСКИЕ И ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ

ФИЗИКА ПРИБОРОВ, ЭЛЕМЕНТОВ И СИСТЕМ

СИСТЕМЫ ТЕХНИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

CONTENT

УДК 681.3.06:519.248.681

И.Д. ГОРБЕНКО, д-р техн. наук, А.А. ЗАМУЛА, д-р техн. наук, В.Л. МОРОЗОВ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ПОМЕХОЗАЩИЩЕННОСТЬ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ В УСЛОВИЯХ РАЗЛИЧНЫХ ВНУТРЕННИХ И ВНЕШНИХ ВОЗДЕЙСТВИИ

Введение

К основным показателям эффективности телекоммуникационной системы относят: помехоустойчивость, надежность, живучесть, пропускную способность сети, качество обслуживания, рентабельность и стоимость, помехозащищенность, информационную безопасность и др.

Информационный обмен в ряде приложений телекоммуникационных систем (ТКС) осуществляется в условиях внутренних и внешних негативных воздействий. Примером внутренних воздействий являются помехи, создаваемые соседними станциями многопользовательских систем. Внешние воздействия связывают с преднамеренными помехами, создаваемыми станциями противодействия. При этом станция – постановщик преднамеренных помех – ставит перед собой цель лишить легальные станции возможности осуществлять надежный информационный обмен и минимизировать собственные затраты. Задача построения защищенной ТКС – это создание системы, устойчивой к воздействию множества различных угроз. В многопользовательских телекоммуникационных системах (ТКС) при передаче информации на значительные расстояния мощность преднамеренной помехи на входе приемного устройства в его полосе пропускания может значительно превышать мощность полезного сигнала, передаваемого одной из станций данной ТКС. Будем полагать, что в канале действует наиболее характерный вид помехи, описываемый гауссовским случайным процессом, спектр которого перекрывается со спектром сигнала. В этом случае вероятность ошибки зависит только от отношения мощности сигнала к общему мешающему воздействию. Необходимо подчеркнуть, что в ряде случаев возможность аппроксимации помехи гауссовским законом не так очевидна, поскольку показатели качества решения таких задач, как оценка параметров, *М*-я передача, зависят не только от отношения указанных мощностей.

Оценка помехозащищенности и информационной безопасности при различного рода воздействиях на телекоммуникационные системы.

Помехоустойчивость приема сигналов характеризует способность ТКС функционировать в условиях воздействия на систему различных помех и определяется выражением, связывающим отношение сигнал-помеха на выходе приемника (на входе согласованного фильтра или коррелятора – q^2) с отношением сигнал-помеха на входе приемника – ρ^2 [1]:

$$q^2 = 2B\rho^2, \qquad (1)$$

где $\rho^2 = \frac{P_c}{P_П}$ (P_c, $P_п$ – мощности сигнала и помехи соответственно); B =F·T – база сигнала (T – длительность сигнала).

Выражение (1) может быть представлено в виде

$$q^2 = \frac{2E}{N_П}, \qquad (2)$$

где E – энергия сигнала, $N_П = P_П / F$ – спектральная плотность мощности помехи в полосе F сигнала.

Помехоустойчивость ТКС оценивают вероятностью ошибки $P_{ош}$, которая, в свою очередь, определяется методами приема (когерентный, некогерентный) и свойствами сигналов, являющихся физическими переносчиками данных. Кроме того, вероятность ошибки в канале связи является функцией помех. Причем помеха в ряде случаев представляет собой сумму теплового шума N_0 и помехи, создаваемой станцией противодействия $N_П$. Таким образом, полная спектральная плотность мощности, вследствие наличия помех, увеличивается до значения $N_0 + N_п$ и отношение сигнал /шум можно записать в виде $E/(N_0 + N_П)$.

Как правило, мощность станции постановщика помех значительно больше мощности теплового шума. Поэтому величину отношения сигнал/шум принимают равной $\frac{E_C}{N_П}$

Известно, что энергия сигнала определяется из соотношения [1]:

$$E_C = P_C T = \frac{P}{R}, \tag{3}$$

где P – мощность полезного сигнала; T – время передачи бита; R – скорость передачи данных (бит/с).

Тогда требуемое для обеспечения заданного значения вероятности ошибки в канале отношение энергии бита данных к спектральной плотности мощности помехи может быть найдено из соотношения [2]:

$$(\frac{E_C}{N_П})_{треб.} = (\frac{P/R}{P_П/F})_{треб.} = (\frac{F/R}{P_П/P_C})_{треб.} = \frac{B}{(P_П/P_C)_{треб.}}, \tag{4}$$

где $B = F/R$ – коэффициент расширения спектра сигнала (база сигнала).

Отношение мощности помехи к мощности сигнала может быть записано в виде

$$(\frac{P_П}{P_C})_{треб.} = \frac{B}{(E_C/N_П)_{треб.}} \tag{5}$$

Выражение (5) можно интерпретировать следующим образом. В целях подавления сигналов станции постановщик помех стремится увеличить значение $(\frac{E_C}{N_П})_{треб.}$ посредством уменьшения $N_П$. Это приводит к уменьшению значения $(\frac{P_П}{P_C})_{треб.}$. Однако защищенная система в этом случае может прибегнуть к увеличению базы сигнала, усложняя задачу станции противодействия по постановке помех.

Очевидно, что станция противодействия может иметь полную информацию о режимах функционирования ТКС, частотном диапазоне работы, классах сигналов – переносчиков данных, времени сеансов связи, объеме передаваемой информации и т.д. Кроме того, станция противодействия может владеть аналогичными образцами объектов. В указанных условиях создание радиоканалов ТКС должно осуществляться таким образом, чтобы наиболее эффективной стратегией станции противодействия была стратегия нарушение функционирования системы путем постановки так называемой заградительной помехи (помеха в виде стационарного гауссова шума с нулевым средним и равномерным распределением спектральной плотности мощности, по крайней мере, в области частот, занимаемой сигналом).

Рассмотрим воздействие заградительной помехи на ТКС. Спектральная плотность мощности помехи, создаваемой станцией противодействия,

$$N_П = P_П / F, \tag{6}$$

где F – ширина полосы диапазона, в которой создаются помехи.

Вероятность ошибки на бит сообщения при некогерентной обработке сигнала [1]:

$$P_0 = Q(\frac{\sqrt{2E_C}}{N_0}), \tag{7}$$

где Q – интеграл вероятности.

Средняя вероятность ошибки на бит сообщения при когерентной обработке при наличии широкополосного шума

$$P_0 = Q(\frac{\sqrt{2E}}{N_0 + N_П}) = Q(\sqrt{\frac{2E/N_0}{1+(\frac{E}{N_0})(\frac{P_П}{P_C})/B}}) \ . \tag{8}$$

Графики зависимости P_0 от $\frac{E}{N_0}$ при фиксированном отношении $\frac{P_П}{P_C}$ приведены на рисунке [3]. Анализ кривых, приведенных на рисунке, показывает, что вероятность ошибки может быть существенно уменьшена при увеличении коэффициента расширения спектра сигнала B.

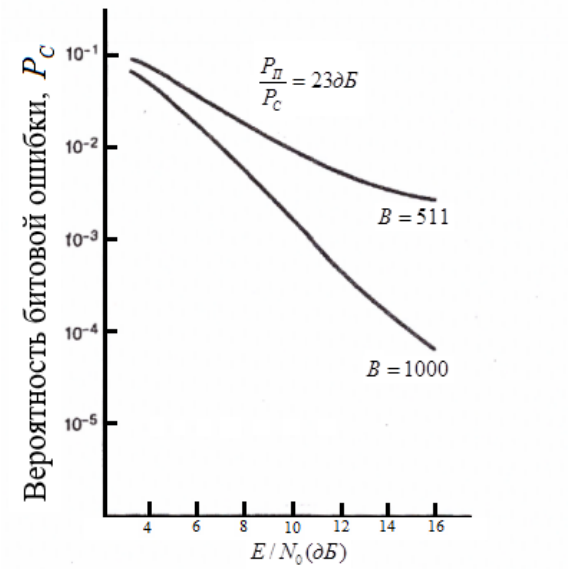


График зависимости вероятности ошибки (P_0) от $\frac{E}{N_0}$

Из соотношений (1) – (7) следует, что при ограничениях на максимальную мощность сигнала и мощность помехи, создаваемой станцией противодействия, единственной возможностью противостоять заградительной помехе является привлечение широкополосной технологии, т.е. сигналов со значительным частотно-временным ресурсом (базой сигналов).

Характерной ситуацией для практики мешающего воздействия на функционирование ТКС, является узкополосная помеха. Причем данный тип помех может быть реализован как станцией противодействия в целях нарушения работы системы, так и соседствующими станциями, создающими помехи вследствие своего обычного функционирования. Оптимальной процедурой обработки сигнала в этом случае можно считать фильтрацию, согласованную с мешающим воздействием (абелевый белый гауссовский шум и узкополосная помеха). Такая обработка эквивалентна вырезанию частотного интервала, в котором сосредоточена помеха. При этом вырезаются и частотные компоненты сигнала в пределах полосы помехи. Согласованный фильтр обеспечивает выходное отношение мощностей сигнала и шума (q_j^2) в виде [1]:

$$q_j^2 = q^2(1-\frac{F_j}{F}), \tag{9}$$

где $q^2 = \frac{2E}{N_0}$ – отношение мощностей сигнала и шума на выходе согласованного фильтра в отсутствие помехи; F_j – полоса помехи; F – полоса сигнала.

Известно [1, 2], что при воздействии на ТКС узкополосной помехи методом защиты является режекторная фильтрация части спектра, на которую воздействует помеха, кроме того, возможно реализовать передачу данных переносом его спектра в диапазон частот, свободный от воздействия помех.

Приведенные выше результаты справедливы для случая, когда помеха является нормальным случайным процессом и обладает равномерной спектральной плотностью. Станция противодействия для подавления системы может использовать мощные структурные помехи с неравномерным спектром. В таких условиях функционирования ТКС помехоустойчивость в значительной мере определяется подобием (различием) структур сигнала и помехи, т.е. тем, как подавляются отдельные элементы сигнала помехой.

Известно, что коэффициент передачи согласованного фильтра определяется выражением [1]:

$$k(\omega) = \frac{cg(\omega)}{N(\omega)}, \tag{10}$$

где c – постоянная; $g(\omega)$ – спектр сигнала.

Отношение сигнал / помеха при этом определяется выражением

$$q^2 = \frac{2}{\pi}\int_0^{\infty}\frac{|g(\omega)|^2}{N(\omega)}. \tag{11}$$

Соотношения (10) – (11) указывают на стратегию действий станции постановщика помех и защищенной системы. Помеха, создаваемая постановщиком помех, должна конструироваться таким образом, чтобы выполнялось равенство $N(\omega) = a\,|\,g(\omega)\,|$, где a – постоянная величина. Последнее равенство означает следующую стратегию: сильнее подавлять те спектральные составляющие, которые переносят большую часть энергии сигнала. Путем снижения усиления согласованного фильтра в области резких пиков в спектре помехи осуществляется исключение этой части спектра. Если имеет место «провал» в спектре помехи, то посредством увеличения усиления согласованного фильтра (согласно (11)) возможно повышение отношения сигнал/помеха. Таким образом, помехоустойчивость системы не снижается вследствие воздействия на систему помехи с неравномерным спектром.

Большинство приложений ТКС относятся к многопользовательским системам. В таких системах вследствие работы большого числа абонентов в общем частотном диапазоне возникают помехи множественного доступа, или взаимные помехи. Рассмотрим влияние взаимной помехи на помехоустойчивость приема данных в ТКС.

Пусть ширина общей полосы частот системы равна F. Предположим, что ширина спектра всех сигналов в ТКС равна ширине общей полосы частот и все активные абоненты создают на входе j-го приемника сигналы одинаковой мощности – P_C. В этом случае мощность взаимной помехи, создаваемой l мешающими абонентами, будет равна $l \cdot P_C$. Допустим, что спектральная плотность мощности взаимной помехи постоянна в пределах общей полосы частот

$$N_{П} = \frac{lP_C}{F}, \tag{12}$$

и взаимная помеха (по своим статистическим свойствам) приближается к нормальному случайному процессу. Таким образом, сделанные предположения позволяют считать

взаимную помеху нормальным случайным процессом с равномерной спектральной плотностью мощности. Нетрудно убедиться, что отношение сигнал/шум на входе решающего устройства приемника определяется из выражения

$$q^2 = \frac{B}{l} = FR/l \ , \tag{13}$$

где R – скорость передачи информации.

Из (13) следует, что при заданном числе активных абонентов l увеличение помехоустойчивости возможно только за счет увеличения базы *B* сигналов. Это объясняется тем, что с увеличением базы (с увеличением ширины спектра сигналов при постоянной скорости передачи информации *R*) уменьшается спектральная плотность мощности помехи $N_п$.

В практике работы ТКС возможны случаи, когда мощность одного или нескольких мешающих сигналов во много раз больше мощности полезного сигнала. Каким образом в этих условиях обеспечить необходимую помехозащищенность?

Пусть мощность полезного сигнала – P_C, а мощность мешающей составляющей – $P_П$. Мощность сигнальной составляющей на выходе согласованного фильтра в момент принятия решения (отсчета) пропорциональна P_C, а мощность мешающей составляющей – $P_П R_{jk}^2(\tau)$, где R_{jk}^2 – взаимнокорреляционная функция (ВКФ) полезного *k*-го сигнала и *j*-го – мешающего. Величина τ определяется смещением ВКФ относительно момента отсчета. Отношение сигнал-помеха на выходе устройства оптимального приема определяется соотношением [1]:

$$q^2(\tau) = \frac{P_C}{P_П R_{jk}^2(\tau)} \ , \tag{14}$$

Наименьшее отношение сигнал-помеха:

$$q^2(\tau) = \frac{P_C}{P_П R_{\max}^2(\tau)} \ , \tag{15}$$

где $R_{макс}$ – максимальное значение $R_{jk}(\tau)$.

Очевидно, что для повышения помехозащищенности ТКС необходимо выбирать сигналы, у которых максимальные пики ВКФ минимальны.

Если максимальные пики ВКФ уменьшены до среднеквадратического уровня: $\sigma_{j,k} = \sigma^2$, то отношение сигнал/помеха будет

$$q^2(\tau) = \frac{P_C}{P_П}\sigma^2 \ . \tag{16}$$

Например, если: $\sigma^2 = \frac{1}{2FT}$, то

$$q^2(\tau) = \frac{P_C}{P_П} FT \ . \tag{17}$$

Для дискретных фазоманипулированных сигналов $\sigma^2 = \frac{1}{2N}$ (*N* – число элементов сигнала). Для такого класса сигналов отношение сигнал-помеха определяется из выражения:

$$q^2(\tau) = \frac{P_C}{P_П} 2N \tag{18}$$

Из выражений (17), (18) следует, что увеличение базы сигнала приводит к увеличению q^2 (а значит, – к увеличению помехоустойчивости системы) и может компенсировать

уменьшение отношения $\frac{P_C}{P_П}$ в случае увеличения станцией противодействия мощности помехи $P_П$.

Помехозащищенность ТКС в условиях воздействия помех, преднамеренно создаваемых станцией противодействия, зависит от скрытности выбора и использования параметров системы. При этом под скрытностью системы в целом и скрытностью используемых в системе параметров будем понимать способность ТКС противостоять мерам радиотехнической разведки, направленным на обнаружение факта работы системы (энергетическая скрытность) и определение необходимых для радиопротиводействия параметров сигнала (структурная и информационная скрытность).

Энергетическую скрытность радиоканала определим как способность радиоканала функционировать с таким энергетическим потенциалом, которого недостаточно для того, чтобы станция противодействия осуществляла перехват и прием сигналов – физических переносчиков информации с требуемой достоверностью:

$$S_Э = P(E / N_0 < G_{треб}), \tag{19}$$

где E / N_0 – отношение энергии сигнала к спектральной плотности мощности шума на входе решающего устройства приемника станции противодействия; $G_{треб}$ – значение отношения E / N_0 для приема данных с требуемой достоверностью.

Другими словами, энергетическая скрытность радиоканала может быть определена как вероятность того, что отношение сигнал/шум на входе решающего устройства приемника станции противодействия не превысит требуемого значения, необходимого для обнаружения сигнала.

Структурная скрытность характеризует способность ТКС противостоять мерам станции противодействия, направленным на отождествление обнаруженного сигнала с одним из множества априорно известных сигналов (распознаванием формы сигнала, определяемой способами его кодирования и модуляции).

Введем понятие структурной скрытности сложного сигнала в виде соотношения

$$S = \frac{\prod_{i=1}^{K} M_i^*}{\prod_{i=1}^{K} M_i}, \tag{20}$$

где M_i^* – число координат сложного сигнала, которые необходимо знать для того, чтобы определить оставшиеся $M_i - M_i^*$ координаты.

Для случая использования в системе фазоманипулированных сигналов выражение (20) имеет вид

$$S = \frac{l}{L}, \tag{21}$$

где l – число символов, которое необходимо знать для определения правила (закона) формирования оставшихся L – l символов.

Качество услуг, которые предоставляет ТКС, оценивают уровнем обеспечения информационной безопасности [4]. При этом под информационной безопасностью будем понимать способность ТКС обеспечивать защиту от уничтожения, модификации, блокирования информации, ее несанкционированной утечки или от нарушения установленного порядка ее маршрутизации. Также под информационной безопасностью следует понимать состояние защищенности систем обработки и хранения данных, при котором обеспечивается сохранение конфиденциальности, целостности и доступности информации, аутентичности, неопровержимости и надежности и также других свойств информации.

Одной из составляющих информационной безопасности (наряду с информационной

скрытностью) является система имитозащиты (обеспечение целостности) информации. Под имитозащищенностью понимают комплекс организационно-технических мероприятий и средств, а также законодательных норм, которые направлены на обеспечение определенного уровня имитостойкости. По сути, имитостойкость является сложной услугой, которая обеспечивается предоставлением таких услуг как целостность, подлинность (аутентичность), и которая поддерживается применением различных криптографических алгоритмов и криптографических протоколов [4] .Основным методом обеспечения необходимого уровня имитостойкости является внесение в сообщение избыточности, которая может формироваться в виде контрольных сумм, избыточных символов кодов, определяющих ошибки, криптографических контрольных сумм (кодов аутентификации сообщений – имитовставок) и др. В качестве показателей оценки имитостойкости могут быть использованы сложность процедур и вероятность навязывания неправдивой (ложной, модифицированной и т.д.) информации, с учетом методов и вычислительных мощностей средств, используемых злоумышленником. К настоящему времени разработан ряд методов обеспечения имитостойкости. В основном они ориентированы на использование методов и средств криптографической защиты информации и избыточного кодирования. В то же время, как показали исследования [4], обеспечить требуемую в ТКС имитостойкость возможно на уровне источника сложных сигналов за счет: увеличения размерности пространства сигналов и размерности пространства параметров сигналов, относительно которых создается неопределенность использования сигналов со сложной структурой; изменения (через определенные промежутки времени) параметров сигналов; использования сигналов с нелинейными законами формирования, обладающих свойствами, близкими к свойствам случайных последовательностей.

На уровне источника сигналов (на физическом уровне) имитостойкость I_c зависит от размерности пространства сигналов M, числа разрешенных к использованию в интервале времени t сигналов Z, числа попыток навязывания (имитации) C и политики навязывания X:

$$I_c = F(M, Z, C, X); \tag{22}$$

или

$$I_c = l - P_{нав}, \tag{23}$$

где $P_{нав}$ – вероятность навязывания (имитации) сообщения станцией противодействия.

При равновероятном и независимом выборе сигналов, используемых в качестве физических переносчиков данных, значение имитостойкости может быть рассчитано с использованием соотношения

$$P_{нав} = C / M . \tag{24}$$

Среди основных направлений улучшения показателей эффективности функционирования ТКС, в частности помехозащищенности, скрытности, информационной безопасности, можно выделить направления, связанные с применением каналов с большой частотной избыточностью, высокой пространственной, структурной, энергетической и временной скрытностью [4]. Для обеспечения частотной избыточности в настоящее время на физическом уровне используются фазоманипулированные широкополосные сигналы (ФМ ШПС) и частотно-фазоманипулированные (ЧФМ) сигналы. При этом анализ методов информационного обмена в телекоммуникационных системах (ТКС) показывает, что для передачи данных в таких системах используют дискретные сигналы с линейными законами их формирования. Такие сигналы обладают ограниченными ансамблевыми характеристиками и, в соответствии с критерием (21), низкой кодовой устойчивостью против раскрытия законов их формирования (низкой структурной скрытностью). Применение указанных систем сигналов в ТКС не позволяет обеспечивать требуемые показатели по помехозащищенности и скрытности их функционирования [1]. Кроме того, применяемые в ТКС методы цикловой синхронизации и управления предполагают, что в течение продолжительного времени в канале синхронизации передается один и тот же широкополосный сигнал линейной формы, а в

информационном канале, т.е. на физическом уровне, соответствие: бит (m бит) сообщения – сигнал линейной формы (2^m сигналов) с течением времени остается фиксированным. Такой метод информационного обмена позволяет нарушителю на основе определения параметров используемых в системе сигналов осуществить постановку преднамеренных структурных помех с минимальными энергетическими затратами. Такие помехи (с точки зрения станции противодействия) являются оптимальными и могут быть созданы при некоторой априорной определенности нарушителя относительно пространства состояний канала передачи данных (несущие частоты, формы используемых сигналов и др.). Указанный тип помехи представляет собой либо ретранслированные, либо имитационные помехи, обработка которых совместно с полезным сигналом, приводит к энергетическому подавлению последнего. В указанных условиях, в процессе информационного противодействия, нарушитель с большой вероятностью может подавлять радиоканал, применяя станции помех с энергетическим потенциалом, соизмеримым с энергетикой радиоканала, а также осуществить навязывание ложных режимов работы системы (синхронизации, управления), ложных сообщений, что, в свою очередь, может привести к существенному ухудшению показателей функционирования ТКС (помехозащищенности, информационной безопасности, имитостойкости, живучести, вероятностно-временных показателей передачи сообщений).

Основным решением указанной проблемы является повышение помехозащищенности (в частности, энергетической, структурной и информационной скрытности) и информационной безопасности (в частности, имитостойкости) ТКС на основе усовершенствования методологических основ построения ТКС путем разработки методов информационного обмена, синтеза новых классов нелинейных дискретных сложных сигналов с необходимыми ансамблевыми, корреляционными и структурными свойствами.

Известно, что для технологии распределенного спектра свойства сигналов-переносчиков данных полностью определяются свойствами дискретных последовательностей (ДП), манипулирующих информационные биты данных пользователей системы [5]. Именно поэтому актуальным является поиск эффективных методов синтеза дискретных сигналов (последовательностей), отвечающих потенциально достижимым граничным характеристикам (минимаксным свойствам). Задача синтеза ДП оказывается еще более сложной, если выдвигаются требования к размерности (объему) системы сигналов, структурным свойствам и числу элементов ДП. При этом анализ [1 – 3, 5,6] показал, что в настоящее время отсутствуют регулярные методы синтеза дискретных последовательностей (ДП), являющихся оптимальными по минимаксному критерию и обладающих необходимыми для построения защищенных ТКС ансамблевыми, корреляционными и структурными свойствами

В работах [6 – 9] сформулирована и решена задача синтеза нелинейных дискретных последовательностей, обеспечивающих требуемые значения помехозащищенности, информационной и структурной скрытности функционирования телекоммуникационной системы. Сложные сигналы, полученные на основе таких последовательностей при использовании системы расширения спектра методом прямой последовательности, обладают, с одной стороны, структурными свойствами, аналогичными свойствам случайных (псевдослучайных) последовательностей, а с другой – требуемыми ансамблевыми и корреляционными свойствами. Кроме того, такие системы сигналов существуют и обладают указанными выше свойствами для широкого спектра значений периода последовательностей. Метод синтеза нелинейных криптографических дискретных сигналов (КС), представленный в [7], основан на использовании случайных или псевдослучайных процессов, и позволяет создавать последовательности символов (сигналов) определенного алфавита, которые удовлетворяют требованиям необратимости, неразличимости, непредсказуемости [10 – 11] и при этом обладают необходимыми (для тех или иных приложений ТКС) ансамблевыми и корреляционными свойствами [9, 12]. Практическое использование данной системы сигналов позволит повысить

(в соответствии с критерием (21)) скрытность функционирования ТКС. Так, для периода сигналов порядка 1000 элементов структурная скрытность КС превышает данный показатель для линейных классов сигналов (*M* последовательностей) более чем в 30 раз.

Характеристики корреляционных функций синтезированных КС не уступают, а в ряде случаев превосходят, соответствующие характеристикам линейных сигналов. В частности, КП обладают улучшенными по сравнению с *M*-последовательностями, взаимно-корреляционными свойствами. Так, применение синтезированных систем нелинейных криптографических сигналов (КС) позволит при использовании КС с периодом 256 элементов в качестве синхронизирующих последовательностей более чем на 3 дБ повысить помехоустойчивость приема сигналов. Значения максимальных боковых лепестков периодической функции взаимной корреляции (ПФВК) КС меньше, чем у широко применяемых в ТКС линейных классов сигналов, построенных на основе *M*-последовательностей. При этом объем системы, составленной из КС, например при периоде КС 1023 элементов, более чем на четыре порядка больше, чем объем системы, составленной из *M*-последовательностей (таблица).

Известно, что в классе линейных последовательностей, образованных на основе *M*-последовательностей, улучшенными ансамблевыми и корреляционными свойствами обладают множества Голда и Касами (так называемые последовательности с трехуровневой функцией взаимной корреляции – ПФВКТ). Так, для периода последовательностей N=1023 элемента, значения максимальных боковых лепестков ПФВК не превосходят 33 (так называемая «граница плотной упаковки» – $2\sqrt{L}$, таблица). При значении уровня боковых лепестков ПФВК $R_{max} = 3\sqrt{L}$, объем системы, составленной из КС, более чем в 15 раз больше объема множеств Голда и Касами. За счет улучшенных ансамблевых свойств КС появляется возможность улучшить, в соответствии с критерием (24), показатели информационной безопасности.

Класс сигналов	Период последовательности	Значение границы «плотной упаковки»	Число пар последовательностей, удовлетворяющих границе
M-последовательности	127	27	36
ПФВКТ	127	17	11610
КП	127	23	47 053
M-последовательности	511	63	276
ПФВКТ	511	33	147500
КП	511	63	2666671
M-последовательности	1023	100	435
ПФВКТ	1023	65	338000
КП	1023	100	5293538

Выводы

В целях улучшения показателей помехозащищенности и информационной безопасности телекоммуникационной системы в условиях внешних и внутренних воздействий необходимы новые решения научной проблемы взаимодействия удаленных информационных объектов на основе усовершенствования методологических основ построения телекоммуникационной системы путем разработки методов синтеза сложных нелинейных дискретных сигналов с необходимыми ансамблевыми, структурными и корреляционными свойствами, а также методов обработки данных в телекоммуникационной системе. Исследования и сравнительный анализ известных методов информационного

обмена показали, что одним из перспективных направлений комплексного обеспечения требуемых значений показателей помехозащищенности и информационной безопасности является реализация в радиоканалах ТКС динамического режима функционирования, когда с течением времени соответствие: m – бит – 2^m сложных сигналов изменяется по сложному закону, а в качестве сложных сигналов применяются сигналы, основанные на нелинейных принципах построения, в частности нелинейные криптографические сигналы, для синтеза которых используются случайные (псевдослучайные) процессы, и нелинейные сигналы в базисе простых и расширенных полей Галуа [6, 7, 9]. Данные классы сигналов обладают необходимыми для создания защищенных ТКС ансамблевыми, структурными и корреляционными свойствами.

Список литературы: 1. *Варакин, Л. Е.* Системы связи с шумоподобными сигналами / Л.Е Варакин. – М. : Радио и связь, 1985. – 384 с. 2. *Ipatov, Valery P.* Spread Spectrum and CDMA.Principles and Applications / Valery P. Ipatov. University of Turku, Finland and St. Petersburg Electrotechnical University ‘LETI’, Russia. – John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2005. – 385 p. 3. *Sklar, B.* Digital Communications [Текст] / B Sklar. – Prentice-Hall, Upper Saddle River, NJ, 2001. – 1082 с. 4. *Горбенко, І.Д., Горбенко, Ю.І.* Прикладна криптологія. Теорія. Практика. Застосування : монографія / І.Д. Горбенко, Ю.І Горбенко. – Харків : Форт, 2012. – 880 с. 5. *Sarvate, D.V.* Crosleration Properties of Pseudorandom and Related Sequences / D.V. Sarvate, M.V. Pursley // IEEE Trans. Commun, 1980. – Vol. Com 68 – P. 59–90. 14. *Gold, R.* Optimal binary sequences for spread spectrum multiplexing // IEEE Trans. Inform. Theory. – 1967. – Vol. 13. – P. 619–621. 6. *Замула, А.А.* Перспективы применения нелинейных дискретных сигналов в современных телекоммуникационных системах и сетях / А.А. Замула, Е.А. Семенко // Системи обробки інформації. – Харьков : ХУПС, 2015. – Вип. 5 (130). – С. 129 – 134. 7. *Gorbenko, I.D., Zamula, A.A., Semenko, Ye.A.* Ensemble and correlation properties of cryptographic signals for telecommunication system and network applications // Telecommunications and Radio Engineering. – Volume 75, 2016 Issue 2. P. 169-178. 8. *Замула, А.А.* Ансамбли дискретных сигналов с минимальными значениями боковых лепестков функций корреляции / А.А. Замула // Системи обробки інформації. – Харків : ХУПС, 2015. – Вип. 10 (135). – С. 35-39. 9. *Горбенко, И.Д., Замула, А.А.* Криптографические сигналы: требования, методы синтеза, свойства, применение в телекоммуникационных системах // Радиотехника. – 2016. – Вып. 186. – С. 7 – 23. 10. *Application* Notes and Interpretation of the Scheme (AIS) 31. Functionality classes and evaluation methodology for physical random number generators. Certification body of the BSI in context of certification scheme. BSI, 2001. 11. *NIST 800-90* b Recommendation for the Entropy Sources Used for Random Bit Generation, 2012.

Харьковский национальный университет
имени В.Н.Каразина

Поступила в редколлегию 20.04.2017

УДК 681.518: 004.658.3

В.И. ЕСИН, д-р техн. наук

КИБЕРНЕТИЧЕСКИЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ РЕИНЖИНИРИНГА БАЗ ДАННЫХ

Введение

Результаты исследований состояния информатизации в различных компаниях, организациях, учреждениях свидетельствуют о том, что в настоящее время многие из них владеют определенными информационными системами организационного управления (ИСОУ). При этом для решения возникающих новых задач, связанных, как правило, с расширением деятельности, и соответственно, рассматриваемыми предметными областями (ПрО), они хотят иметь более функциональные, с улучшенными характеристиками качества информационные системы (ИС), требующие меньших затрат по сопровождению. В этих условиях востребованными становятся проекты: по разработке новых ИСОУ и их интеграции с существующими информационными системами; разработке новых ИСОУ с целью замены существующих ИС; модернизации существующих ИСОУ. Суть данных проектов заключается в проведении процедур реинжиниринга существующих ИС и их основного функционального компонента: базы данных (БД).

Одним из важных требований, предъявляемых к процессу реинжиниринга существующих ИСОУ и их БД, является своевременность завершения соответствующих проектов в рамках запланированного бюджета с заданными характеристиками качества. Критичность этого условия наглядно демонстрируют результаты анализа IT-проектов, который провели компетентные международные организации экспертов [1, 2]. Вывод из этого анализа не оптимистичный: более 60 % проектов были провалены или завершены с опозданием, причем с гораздо большими затратами, чем планировалось.

Налицо существование нерешенной проблемы, связанной с необходимостью своевременного создания, модернизации в рамках запланированного бюджета информационных систем, обладающих требуемыми качествами, и ограниченностью возможностей существующих методов проектирования. В отношении баз данных указанная ограниченность возможностей обусловлена ориентацией традиционной методологии их проектирования, используемой при реинжиниринге БД ИСОУ, на итерационную, достаточно сложную и трудоемкую процедуру создания уникальных концептуальной модели, логической и физической схем при разработке новой БД, либо на существенное их преобразование при модернизации.

В сложившейся ситуации возникает объективная потребность в пересмотре существующих подходов, методологий и технологий реинжиниринга баз данных и оценке целесообразности применения в проектах той или иной технологии.

Чтобы оценить возможность и целесообразность применения в проектах той или иной технологии, в первую очередь традиционной технологии проектирования реляционных баз данных (РБД), получивших наибольшее распространение в ИС рассматриваемого класса и удовлетворяющих требованиям к типу источника данных создаваемой системы, использовали кибернетический подход для таких исследований, в результате чего:

– определена модель как множество взаимосвязанных характеристик, образующих базис для спецификации требований к качеству и оценивания качества подвергающейся систематической трансформации БД ИСОУ, с соответствующими метриками качества (за основу в качестве прототипов были приняты характеристики и метрики, определенные в действующем в Украине стандарте ДСТУ ISO/IEC 9126, а также в серии международных стандартов SQuaRE – ISO/IEC 250xx), которые в совокупности можно представить в виде формализованного выражения:

$$Q_{DB} = \{H_i^{DB}, S_{ij}^{DB}, M_{jk}^{DB(i)}, At_{jl}^{DB(i)}\}, \quad (1)$$

где H_i^{DB} – i-я характеристика качества БД (i=1,…,I); S_{ij}^{DB} – j-я подхарактеристика (j=1,…,J) i-й характеристики качества; $M_{jk}^{DB(i)}$ – k-я метрика (k=1,…,K) j-й подхарактеристики i-й характеристики качества; $At_{jl}^{DB(i)}$ – l-й атрибут (l=1,…,L;) j-й подхарактеристики i-й характеристики качества – переменная, которой присваивается значение в результате измерения (применения метрики), $At_{jl}^{DB(i)} \in Z$; $Z = (z_1, \ldots, z_\Theta)$, например для соответствующих H_i^{DB}, S_{ij}^{DB} атрибутами качества являются: полнота, корректность реализации объектов схемы БД, способность к взаимодействию, оперативность устранения некорректных данных в БД, адаптация объектов схемы БД, непрерывность использования данных БД, среднее время отклика на запрос, среднее время, затрачиваемое на модификацию, и другие);

– формализована задача реинжиниринга БД ИСОУ и предложен способ ее решения.

Постановка задачи

Представим задачу реинжиниринга БД ИСОУ (под которой в дальнейшем будем понимать реляционную (объектно-реляционную) БД) в виде известной из общей теории адаптивных систем задачи структурной адаптации [3].

В условиях динамичных изменений предметных областей в рамках запланированного бюджета необходимо осуществить своевременную реструктуризацию некоторой рассматриваемой реляционной БД ИСОУ с целью максимизации (минимизации) выбранных атрибутов (показателей) качества этой БД при обязательном обеспечении их требуемых значений.

Формализация задачи

Формализацию задачи структурной адаптации РБД ИСОУ начнем с представления схемы (рис. 1) системы управления объектом в условиях динамичных изменений предметных областей (схемы адаптации РБД ИСОУ к изменениям условий функционирования).

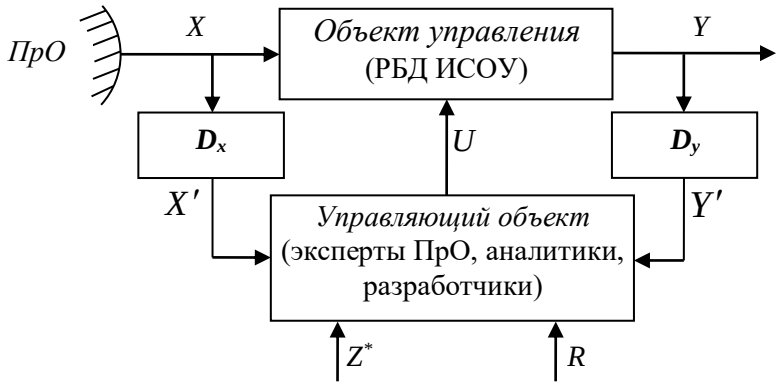


Рис. 1. Схема системы управления объектом

Каждый из приведенных на рис. 1 элементов представляет собой следующие компоненты системы управления и набор ограничительных требований:

– объект управления (ОУ) – РБД ИСОУ, которую необходимо адаптировать к изменениям условий функционирования;

– управляющий объект – задействованные в процессе реинжиниринга РБД ИСОУ различного профиля исполнители: эксперты ПрО, аналитики, разработчики, формирующие управляющие воздействия на основе анализа входной, выходной информации, используя для этого доступные модели, методы и средства;

– X – множество изменений в предметной области;

– Y – состояние сложного объекта управления – базы данных ИСОУ: $Y = F(X,U)$, где F – оператор связи входа X и выхода Y с учетом фактора управляющего воздействия U;

– U – множество доступных моделей, методов и средств, используемых при реинжиниринге РБД ИСОУ;

– X' – информация об изменениях в ПрО, получаемая управляющим объектом с помощью датчика D_x; Y' – информация о состоянии ОУ: схеме, оценках значений атрибутов качества РБД ИСОУ, получаемая управляющим объектом с помощью датчика D_y. Датчики измеряют (выделяют) только то, что используется (необходимо) в процессе управления. Поэтому получаемая из них информация не в полной мере отражает действительное состояние ОУ и реальные изменения в ПрО: $X' = D_x(X)$; $Y' = D_y(Y)$. Датчиками D_x, D_y являются соответствующие инструментальные средства (CASE-средства), обеспечивающие процедуру оценки входных и выходных данных, необходимую экспертам ПрО, аналитикам, разработчикам для принятия решений в определении эффективных управляющих воздействий;

– Z^* (цель управления) – множество требований, предъявляемых к состоянию наблюдаемого объекта управления, которые связываются с Θ-мерным пространством значимых для адаптируемой РБД ИСОУ совокупности свойств (атрибутов), обеспечивающих ее способность удовлетворять установленные или предполагаемые потребности в соответствии с назначением: $Z^* = (z_1^*,...,z_\Theta^*)$, где z_i^* – требование к значению атрибута качества – $z_i \in Z$, $Z = (z_1,...,z_\Theta)$ (под z_i^* может пониматься также требование к одновременному выполнению совокупности значений атрибутов качеств). Цели-требования z_i^* $(i = 1,...,\Theta)$ вектора Z^* могут быть представлены или при необходимости преобразованы к виду следующих форм (критериев):

а) равенств (для z_i^*): $z_i = a_i$ – это требование означает, что значение i-й целевой переменной $z_i = \varphi_i(Y')$ должно быть равно заданной величине a_i;

б) неравенств (для z_j^*): $z_j \le b_j$ $(z_j \ge b_j)$ – это требование, накладываемое на j-ю целевую переменную, – ее значение не должно быть меньше (больше) заданного порога b_j. При необходимости любое ограничение вида $z_j \le b_j$ можно привести к эквивалентному ограничению вида $-z_j \ge -b_j$, и наоборот (как и любое двустороннее ограничение можно привести к двум односторонним);

г) максимизации/минимизации (для z_ν^*): $z_\nu \to \max$ $(z_\nu \to \min)$ – это требование означает, что целевая переменная z_ν вектора Z должна быть максимальной (минимальной). В рассматриваемой задаче следует максимизировать или минимизировать (в зависимости от предназначения) определенные атрибуты качества, при обязательном выполнении, предъявляемых к ним же ограничений в виде равенств и неравенств. При этом переход от задачи максимизации к задаче минимизации осуществляется путем изменения знаков коэффициентов целевой функции на противоположный.

С учетом сказанного вектор Z^* имеет следующее представление:

$$Z^*: \begin{cases} z_i^*: z_i = \varphi_i(Y') = a_i, & (i = 1,...,l_1); \\ z_j^*: z_j = \psi_j(Y') \le b_j, & (j = 1,...,l_2); \\ z_v^*: z_v = \eta_v(Y') \to extr, & (v = 1,...,l_3), \end{cases} \quad (2)$$

где $l_1 + l_2 + l_3 = \Theta$; φ_i, ψ_j, η_v – некоторые функционалы, определяющие связь между состоянием наблюдаемого объекта управления и соответствующим параметром (атрибутом качества): z_i, z_j, z_v.

– R – временно́й, финансовый, людской ресурсы ($R = (r_t, r_f, r_p)$), выделяемые на реинжиниринг РБД ИСОУ.

Решение задачи

Решение задачи реинжиниринга РБД ИСОУ (адаптации БД к изменениям условий функционирования) заключается в нахождении на основании полученной информации об изменениях в ПрО (X'), состоянии ОУ (Y'), цели управления Z^* и ограниченных выделяемых ресурсов R такого управляющего воздействия U^*, с помощью которого стал бы возможным перевод ОУ в искомое состояние Y^*:

$$\langle X',\ Y',\ Z^*,\ R\rangle \ \to\ U^* \to\ Y^* \quad (3)$$

Для нахождения U^* необходимо определить оператор (алгоритм) A, преобразующий исходную информацию в управление:

$$U = A\,(X',\ Y',\ Z\ ,\ R). \quad (4)$$

Задачу синтеза оператора A обычно декомпозируют на две [3]:

1) синтез модели F объекта управления:

$$Y' = F(X', U); \quad (5)$$

2) синтез управления с помощью этой модели.

Синтез модели

В общем случае оператор F определяется некоторым алгоритмом (инструкцией, правилом), который указывает, как, располагая информацией об X' и U, определить выход Y'.

В рассматриваемой задаче в качестве основы такого алгоритма целесообразно взять последовательность основных этапов классической методологии проектирования реляционных баз данных (концептуального, логического и физического проектирования). Тогда оператор F можно представить в виде FEO-диаграммы (*For Exposition Only*) последовательности технологических операций с уточнением содержания их основных этапов и применяемых разработчиками соответствующих CASE-средств (рис. 2).

Действия по внесению требуемых изменений в концептуальную модель ПрО, логическую и физическую схемы БД ИСОУ осуществляются соответствующими разработчиками, экспертами ПрО с помощью доступных инструментальных средств ($U_4 \in U$) в соответствии с существующими правилами описания, структурирования данных и видами ограничений их целостности, принятыми в «расширенных» (термин, введенный

Дейтом, для обозначения моделей, используемых при семантическом моделировании [4]) – $U_1 \in U$ и реляционной – $U_2 \in U$ моделях данных, а также правилами и процедурами разработки объектов физической схемы БД ИСОУ ($U_3 \in U$).

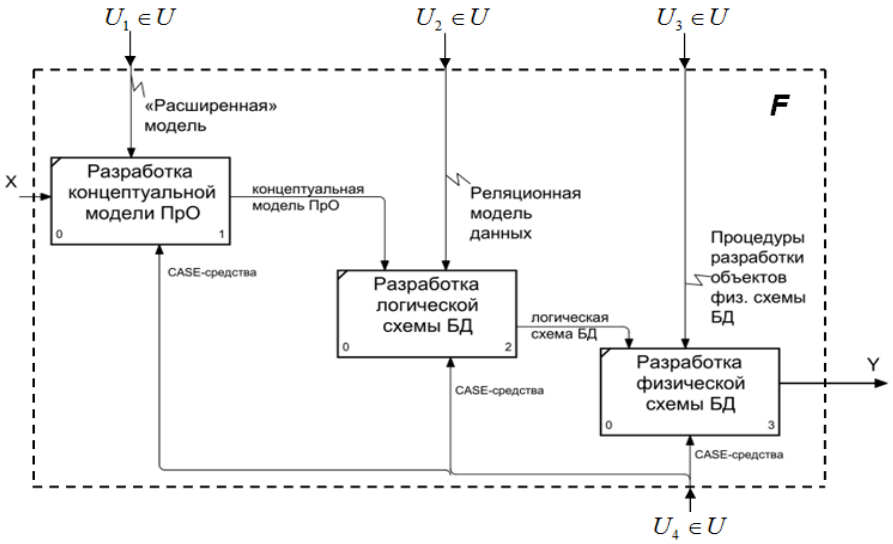


Рис. 2. Схема оператора F

В результате применения оператора F к входной информации и U ($U = U_1 \bigcup U_2 \bigcup U_3 \bigcup U_4$), как правило, изменяются структуры (структурные компоненты):

– концептуальной модели ПрО:

$$S_{sd} = \langle Ent, Rel \rangle, \tag{6}$$

где $Ent = \{Ent_1, ..., Ent_n\}$ – множество сущностей рассматриваемой ПрО, Ent_n – n-я сущность: $Ent_n = (e_{n1}, ..., e_{nm})$, e_{nm} – m-е свойство n-й сущности; $Rel = \{Rel_1, ..., Rel_i\}$ – множество связей между сущностями (и их роли), Rel_i – i-я связь: $Rel_i = rel_{kj}$, где rel_{kj} – связь между k-й и j-й сущностями, которая тоже может описываться набором свойств;

– логической схемы БД:

$$S_{ls} = \langle O, H \rangle, \tag{7}$$

где $O = \{O_1, ..., O_g\}$ – множество отношений, O_g – g-е отношение: $O_g(o_{g1}, ..., o_{g\lambda})$, $o_{g\lambda}$ – λ-й атрибут g-го отношения; $H = \{H_1, ..., H_\alpha\}$ – множество ключей, в том числе определяющих связи между отношениями (внешних ключей);

– физической схемы БД ИСОУ как собственно структуры управляемого объекта (под физической схемой БД будем понимать реализацию логической схемы в конкретной СУБД):

$$S_{ps} = \langle V, K, D \rangle, \tag{8}$$

где $V = \{V_1, ..., V_\mu\}$ – множество базовых отношений (таблиц), V_μ – μ-я таблица: $V_\mu = (v_{\mu 1}, ..., v_{\mu\psi})$, $v_{\mu\psi}$ – ψ-й столбец μ-й таблицы; $K = \{K_1, ..., K_\sigma\}$ – множество ключей, в том числе определяющих связи между таблицами (внешних ключей), $D = \{D_1, ..., D_\theta\}$ – множество иных объектов схемы, обеспечивающих эффективность выборки (индексы), целостность и безопасность данных (триггеры, функции, процедуры и т. д.).

Синтез управления

Суть этапа синтеза заключается в определении управления U^*, реализация которого в объекте дает возможность добиться заданных целей управления Z^* (выражение (2)).

Подставляя в выражение (2) полученную модель F, приходим к многокритериальной задаче оптимизации:

$$\eta_\nu(F(X^{'},U)) \to \underset{U\in\Omega}{extr},\ (\nu=1,...,l_3), \tag{9}$$

где Ω – множество допустимых управлений, определяемое следующими соотношениями:

$$\Omega:\begin{cases}\varphi_i(F(X^{'},U)) = a_i, & (i=1,...,l_1);\\ \psi_j(F(X^{'},U)) \le b_j, & (j=1,...,l_2);\\ U \in R.\end{cases} \tag{10}$$

Для решения этой задачи, прежде всего, следует осуществить свертку экстремальных целей (9). Однако ввиду того, что все атрибуты качества, входящие в выражение (9), имеют различные меры (единицы физической величины) и, как следствие, не могут непосредственно объединяться и сопоставляться, все эти атрибуты в первую очередь необходимо привести к единой шкале оценки. Один из возможных подходов приведения к такой единой безразмерной шкале оценки – от 0 до 1 приведен в работе [5]. В соответствии с данным подходом рассчитываются относительное значения этих атрибутов качества – как отношение фактического значения ($z_\nu^{факт}$, где $\nu=(1,...,l_3)$) к базовому (эталонному) значению ($z_\nu^{эталон}$):

$$S_\nu = z_\nu^{факт} / z_\nu^{эталон}. \tag{11}$$

Эталонное значение соответствует значению данного атрибута у лучшего образца БД из числа аналогов. К аналогам относятся ранее созданные БД ИСОУ, отвечающие современному уровню развития, подобные по функциональному назначению и условиям применения, что и оцениваемая БД. Кроме того, данный подход предполагает наличие сведений о важности этих атрибутов качества, обычно учитываемых с помощью специальных коэффициентов весомости, определяемых экспертным путем [5]. В результате осуществления таких преобразований был получен интегральный показатель качества создаваемой в процессе реинжиниринга БД ИСОУ, определяемый как среднее взвешенное значение от заданных относительных значений атрибутов качества с учетом коэффициентов их весомости (важности):

$$J = \sum_{\nu=1}^{l_3} k_\nu \times S_\nu, \tag{12}$$

где k_ν – весовые коэффициенты, учитывающие важность (весомость) ν-го атрибута качества; S_ν – относительное значение ν-го атрибута качества.

Ограниченность выделяемых ресурсов R заставляет разработчика искать наиболее эффективный вариант решения поставленной задачи. А это предполагает, как отмечалось в работе [6], необходимость сравнения того, что дает и во что обходится в рассматриваемом случае создаваемая в процессе реинжиниринга БД ИСОУ. То есть необходимость сравнения, не только исходя из результативности, расчет которой осуществляется в соответствии с выражением (12), но и эффективности как степени удовлетворения потребностей и достижения целей, соотнесенной с соответствующими затратами ресурсов. Об этом также указывалось в работах [7, 8]. При этом затраты ресурсов R можно привести и выразить в стоимостном

эквиваленте либо в ином виде [9], например в виде трудозатрат (финансовые затраты являются тривиальной функцией от трудоемкости [10]), значения которых далее, согласно изложенному выше подходу, привести к безразмерной величине. То есть к так называемому значению относительных трудозатрат R_n, определяемому как отношение реальных трудозатрат на обеспечение соответствующих значений атрибутов качества создаваемой в процессе реинжиниринга БД ИСОУ $P_{факт}$, к эталонному значению полученных ранее трудозатрат на создание лучшего образца БД из числа аналогов – $P_{эталон}$.

В результате соотнесения интегрального показателя качества (12) со значением относительных трудозатрат R_n был получен показатель эффективности, учитывающий рациональность использования ресурсов. Показатель эффективности процесса реинжиниринга БД ИСОУ – есть отношение среднего взвешенного значения от заданных относительных значений атрибутов качества, с учетом коэффициентов их весомости, к относительному значению трудозатрат на обеспечение этих качеств:

$$E = \frac{1}{R_n} \times \sum_{v=1}^{l_3} k_v \times S_v , \tag{13}$$

где

$$R_n = P_{факт} / P_{эталон} . \tag{14}$$

В результате проведенных преобразований получаем оптимизационную задачу:

$$E(X',U) = \frac{1}{R_n} \times \sum_{v=1}^{l_3} k_v \times S_v \to \max_{U \in \Omega} \Rightarrow U^*, \tag{15}$$

где Ω – определяется в соответствии с (10), U^* – оптимальное управление.

Решение данной задачи при незначительной сложности вносимых структурных изменений (с учетом всех схем различного уровня) и низкой частоте изменений условий функционирования (состояния предметной области, требований к функциональности, значениям атрибутов качества БД ИСОУ, расходованию ресурсов) не вызывает принципиальных затруднений. Однако при увеличении частоты изменения условий функционирования и сложности вносимых структурных изменений, обусловленных сегодня достаточно частыми обновлениями нормативно-законодательной базы, стремлением компаний к увеличению поддерживаемой функциональности систем в связи с обострившейся конкуренцией, учитывая сложность формализации правил применения моделей, методов и средств как составных элементов управляющего воздействия соответствующих технологий и характер их зависимости (разрывность, нелинейность), решение полученной оптимизационной задачи существенно затрудняется. Кроме того, эта задача для сложных объектов управления, к которым относится и БД ИСОУ, имеющих сложные законы их описания, обычно является многоэкстремальной и овражной [3]. Все это в совокупности заставляет искать и использовать для ее решения специальные методы поисковой оптимизации.

Выводы

1. Показана актуальность проблемы выбора технологии при реинжиниринге баз данных ИСОУ в условиях необходимости своевременного завершения соответствующих проектов в рамках запланированного бюджета с заданными характеристиками качества.

2. Сформулирована и формализована задача реинжиниринга БД ИСОУ в виде известной из общей теории адаптивных систем задачи структурной адаптации, которая в результате преобразований была приведена к оптимизационной задаче.

3. Отмечена сложность решения полученной оптимизационной задачи, обусловленная трудностью формализации правил применения моделей, методов и средств как составных элементов управляющего воздействия при использовании соответствующей технологии, характера их зависимости (разрывности, нелинейности), сложностью закона описания объекта управления, и приводящая к необходимости находить и использовать специальные методы поисковой оптимизации.

4. Полученное выражение для определения показателя эффективности (формула (13)) может быть практически использовано при расчете относительной (сравнительной) эффективности процесса реинжиниринга БД ИСОУ для определения конкретной количественной оценки преимущества одной информационной технологии перед другой для обоснованного выбора ее применения в проекте.

Список литературы: 1. *Chaos* Manifesto 2013: Think Big, Act Small online version. The Standish Group [Electronic resource]. – Access mode : http://www.versionone.com/assets/ img/files/ChaosManifesto 2013.pdf. 2. *Standish* Group 2015 Chaos Report – Q&A with Jennifer Lynch [Electronic resource]. – Access mode : https://www.infoq.com/articles/standish-chaos-2015. 3. *Расстригин, Л. А.* Адаптация сложных систем / Л. А. Расстригин. – Рига : Зинатне, 1981. – 375 с. 4. *Дейт, К. Дж.* Введение в системы баз данных ; 8-е изд. ; пер. с англ. / К. Дейт. – Москва : Изд. дом "Вильямс", 2005. – 1328 с. 5. *Андон, Ф. И.* Основы инженерии качества программных систем ; 2-е изд., перераб. и доп. / [Ф. И. Андон, Г. И. Коваль, Т. М. Коротун, Е. М. Лаврищева и др.]. – К. : Академпериодика, 2007. – 672 с. 6. *Захаров, В. Н.* Системы управления. Задание. Проектирование. Реализация / В. Н. Захаров, Д. А. Поспелов, В. Е. Хазацкий. – Москва : Энергия, 1977. – 423 с. 7. *Архипенков, С.* Лекции по управлению программными проектами / С. Архипенков. – Москва, 2009. – 128 с. 8. *Липаев В. В.* Качество программных средств : метод. рекомендации / В. В. Липаев ; под общей ред. проф., д.т.н. А. А. Полякова. – Москва : Яиус-К, 2002. – 400 с. 9. *Юркова, Т. И.* Термин «эффективность процесса» и его экономическая интерпретация / Т. И. Юркова // Современные проблемы науки и образования. – 2014. – № 1. 10. *Макконнелл, С.* Сколько стоит программный проект / С. Макконнелл. – Москва : Русская редакция ; СПб. : Питер, 2007. – 297 с.

Харьковский национальный университет имени В.Н.Каразина

Поступила в редколлегию 20.04.2017

УДК 004.056.53:621.39

В.І. ЗАБОЛОТНИЙ, канд. техн. наук, А.В. ЄРМОЛОВИЧ

МЕТОДИКА ОРГАНІЗАЦІЇ ЗАХОДІВ ЗАХИСТУ ВІД ТЕХНІЧНИХ ЗАСОБІВ КОНКУРЕНТНОЇ РОЗВІДКИ

Вступ

Розвиток України в економічному плані інтенсифікує процес збільшення матеріального виробництва конкурентоспроможної наукоємної продукції. Отримати суттєвий прибуток від своєї продукції власник може тільки у тому випадку, коли забезпечить раптовість її появи на відповідному ринку [1]. Суттєвим чинником для досягнення цієї мети може стати конкурентна розвідка. «Конкурентна розвідка – це реалізація системної програми збору, аналізу і розподілу інформації про діяльність конкурентів і загальні тенденції бізнесу, пов'язаних з цілями конкретної компанії» – зазначає Ю.П. Воронов [2]. У той самий час М. Логвинов зазначає, що конкурентна розвідка – це збір і обробка даних з різних джерел для вироблення управлінських рішень з метою підвищення конкурентоспроможності комерційної організації, що проводяться в рамках закону і з дотриманням етичних норм та законодавчих норм [3]. Основною проблемою в захисті від конкурентної розвідки є визначення можливих джерел витоку інформації, що становлять комерційну таємницю, несанкціоноване ознайомлення конкурентів з якої може завдати збитків. У статті 505 ЦК України визначено, що комерційною таємницею є інформація, яка є секретною в тому розумінні, що вона в цілому чи в певній формі та сукупності її складових є невідомою та не є легкодоступною для осіб, які звичайно мають справу з видом інформації, до якого вона належить, у зв'язку з цим має комерційну цінність та була предметом адекватних існуючим обставинам заходів щодо збереження її секретності, вжитих особою, яка законно контролює цю інформацію, комерційною таємницею можуть бути відомості технічного, організаційного, комерційного, виробничого та іншого характеру, за винятком тих, які відповідно до закону не можуть бути віднесені до комерційної таємниці.

Постановка задачі

Основною метою проведених досліджень є визначення змістовності основних понять при захисті від конкурентної розвідки, розмежування та постанова завдань для всіх основних етапів з розробки заходів захисту від конкурентної розвідки, розробка методики організації даних робіт.

Методи конкурентної розвідки можуть бути як законними, або такими, при використанні яких відбувається порушення закону. Існує безліч можливостей, що дозволяють отримати інформацію, що становить комерційну таємницю, не прикладаючи багатьох засиль. Метод конкурентної розвідки, при якому робота ґрунтується на вивченні інформації, отриманої з офіційних опублікованих джерел: відбувається аналіз публікацій, статей, отриманих через Інтернет і ЗМІ про конкурентів, аналіз маркетингових досліджень в даній галузі (купівля минулих маркетингових досліджень конкурентів), опитування конкурентів під виглядом маркетингового дослідження, аналіз отриманих фінансових документів конкурентів, аналіз структури компанії конкурентів, аналіз статутних документів конкурентів, аналіз структур і господарських взаємозв'язків. Існує такий метод конкурентної розвідки, як «мертві вакансії»: коли запрошують на співбесіду співробітника з фірми конкурента для роботи на більш вигідних умовах. На цій співбесіді у працівника з'ясовують подробиці його діяльності. Ніяких, природно, після цієї співбесіди пропозицій про роботу не надходить, а конкурентна розвідка володіє потрібною інформацією. Конкурентна розвідка діє: спостерігаючи (на відстані) і (або) проникаючи в організацію (коли у фірму конкурента вводиться свій (або спеціальний) співробітник).

Методами конкурентної розвідки також є:

- опитування конкурентів, постачальників, клієнтів, колишніх співробітників;
- закупівля товару у конкурента;
- відвідування конференцій, семінарів і виставок за участю конкурентів.

Для проведення конкурентної розвідки також можна використовувати ЗТР (засоби технічної розвідки):

- апаратура радіорозвідки;
- апаратури візуально оптичної розвідки;
- лазерні мікрофони;
- закладні пристрої;
- апаратуру хімічної розвідки та інше.

Конкурентна розвідка може проводитися:

- з землі;
- з БПЛА (безпілотних літальних апаратів).

Завдання конкурентної розвідки:

- виявити конкретні недоліки в роботі конкурентів, виявити товари з конкурентними пе ревагами, визначити їх цінову політику;
- виявити способи просування таких товарів на ринок;
- виявити умови співпраці з постачальниками (щоб створити для себе умови не гірші, ніж у конкурента);
- визначити постійну клієнтську базу конкурента і умови взаємодії, визначити рівень рентабельності товарів;
- виявити плани конкурентів з технічного розвитку, розширення меж ринку.

Пропозиція щодо рішення задачі

Конкурентна розвідка особливо ефективна, коли результатом буде випередження свого конкурента, а не просто копіювання його переваг. Існує безліч способів ведення конкрентної розвідки [2]. Введемо деякі поняття, що стосуються проведення конкурентної розвідки. Отже, суб'єкт конкурентної розвідки – це підприємство, організація чи особа, що з використанням певних ресурсів здійснює збір,обробку та аналіз даних з різних джерел для вироблення управлінських рішень з метою підвищення конкурентоспроможності підприємства, організації чи своєї; об'єкт конкурентної розвідки – це підприємство, організація чи особа, до діяльність якої прикута невмотивована увага з боку іншого об'єкта, що проводить збір інформації про діяльність та стан з метою нанесення шкоди чи отримання вигоди. Слід не лише приділяти увагу проведенню конкурентної розвідки. Існує можливість, що суб'єкт конкурентної розвідки, який реалізує системну програму збору, аналізу і розподілу інформації про діяльність конкурента(-ів), в той самий час може бути об'єктом конкурентної розвідки. Повністю уникнути можливих дій конкурентів з метою отримання інформації, що становить КТ, чи будь-якої іншої, розповсюдження якої завдасть шкоди, неможливо. А отже необхідно проводити заходи, що забезпечать неможливість отримання інформації такого змісту. Одним із способів запобігання впливу конкурентної розвідки на діяльність підприємства – це розробка заходів захисту відомостей з обмеженим доступом.

При розробці заходів захисту відомостей з обмеженим доступом необхідно:

1. Одержати загальні задачі щодо захисту відомостей з обмеженим доступом (ВзОД) з керівної організації вищого рівня ієрархії, замовника.
2. Створення робочої групи для розробки заходів захисту.
3. Визначити задачі захисту на різних етапах життєвого циклу продукту, його застосуванню, технології виготовлення.
4. Аналіз ВзОД для визначених етапів життєвого циклу.
5. Виявлення та аналіз основних відомостей (ОВ).

6. Виявлення небезпечних засобів технічної розвідки ЗТР для ОВ та можливих результат тів їх діяльності.
7. Розробка замислу захисту.
8. Розробка заходів захисту.
9. Розробка заходів контролю заходів захисту.

При розробці засобів захисту першочергово необхідно визначити доцільність та конкретизувати мету проведення цих заходів: чи існує на об'єкті обробка, створення, модифікація і т.д. інформації, такої, що містить відомості з обмеженим доступом. На приватних підприємствах рішення про віднесення інформації до такої, що необхідно захищати від витоку (несанкціонованого ознайомлення і т.д), приймається керівником підприємства чи його уповноваженого (групою керівників). Етап одержання загальних задач щодо захисту є дуже важливим, оскільки саме в цей момент відбувається формування загального та детального списку ВзОД, що повинен бути отриманий у вигляді офіційно оформленого розпорядження від замовника, в якому має міститися час, термін та місце проведення відповідних робіт.

У склад робочої групи доцільно ввести:
- фахівців, які володіють знаннями щодо усього об'єкту захисту або його складових частин,
- фахівця з ЗТР,
- економіста для оцінки можливих втрат від витоку ВзОД та обґрунтовування витрат на заходи захисту ВзОД.

Визначення задач захисту на різних етапах життєвого циклу продукту, його застосування, технології виготовлення:

1. Етап теоретичних досліджень з створення об'єкту захисту.
2. Етап створення та дослідження окремих фрагментів, макетів елементів об'єкту захисту.
3. Виготовлення дослідного зразка (кількох зразків).
4. Випробування дослідного зразка повномасштабні або окремі.
5. Підготовка масового виробництва, технологій масового виробництва.
6. Масове виробництво.
7. Продаж, повномасштабне застосування, включаючи експлуатацію, ремонт тощо.
8. Утилізація.

На кожному з етапів потрібно визначати заходи захисту, які необхідно впроваджувати для забезпечення захисту ВзОД. Для певного етапу життєвого циклу необхідні заходи захисту можуть відрізнятися. Це залежить від місця етапу в життєвому циклі. Також набір етапів може змінюватися в залежності від типу та характеру об'єкту захисту.

Потрібно прогнозувати можливості модернізації об'єкту ВзОД. Це може привести до проявлення зворотного зв'язку до визначення задач захисту на різних етапах життєвого циклу продукту від більш пізніших етапів життєвого циклу. Не усі етапи життєвого циклу можуть мати місце для конкретних об'єктів захисту. Також перелік та кількість етапів життєвого циклу може відрізнятися для конкретних об'єктів захисту.

Особливості та вимоги для вирішення поставленої задачі

Суть аналізу ВзОД для визначених етапів життєвого циклу полягає у декомпозиції загального переліку ВзОД та створення набору елементарних ВзОД.

1. B – відомість з обмеженим доступом (множина); $b_1,\ b_2 \dots b_n$ – множина ЕлВзОД, що визначено для певної ВзОД;

$$\exists \forall B := \{b_1,\ b_2 \dots b_n\} \tag{1}$$

Існує всяка множина B, що рівнозначна множині $\{b_1,\ b_2 \dots b_n\}$.

2. A є множиною всіх елементів, що є ВзОД.

$$A = U \tag{2}$$

3. Кожний елемент B є елементом A «і» A не дорівнює B.

$$B \subseteq A \wedge B \neq A \tag{3}$$

4. С – множина, що містить у собі елементи – ознаки відомостей(ОВ).

$$C = \{c_1, c_2 \dots c_n\} \tag{4}$$

5. P – властивість ознаки; оприлюднення завдає шкоди.

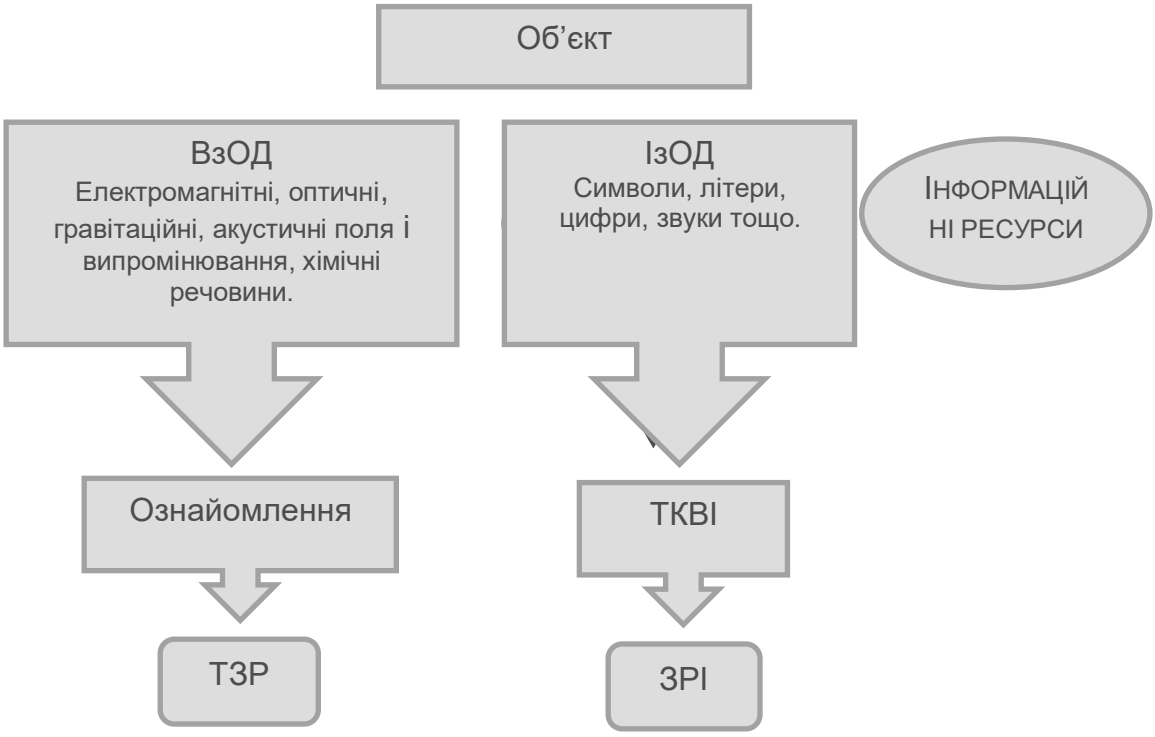
$$\exists \forall D \in \{x \in C\}: P(\{C\})\} \tag{5}$$

6. Існує множина E, що мітить відомості з інформаційних ресурсів. Серед елементів множини E не може бути елементів множини A.

$$E \cap A = \emptyset \tag{6}$$

7. Множина ЗТР – Z, елементами якої є засоби, що можна використати для отримання ІзОД по технічним каналам витоку інформації (ТКВІ), де z_n – це засоби технічної розвідки (рисунок).

$$Z \in \{z_1, z_2, z_3, \dots z_4\} \tag{7}$$



Обидва види прояву даних потребують захисту від технічних розвідок. Знакова форма представляє сукупність символів, літер, цифр, звуків, які відображають предмети та явища реального світу у віртуальному світі. Носіями інформації з обмеженим доступом (ІзОД) є

інформаційні сигнали у формі фізичних полів (електромагнітних, оптичних, акустичних), електричних сигналів, вібраційних коливань у твердих предметах. Шлях захисту від таких розвідок – технічний захист інформації (ТЗІ) [1].

Інструмент даного етапу – метод експертних оцінок, що є одинм з основних класів *методів науково-технічного прогнозування*, який ґрунтується на припущенні, що на основі думок експертів можна збудувати адекватну *модель* майбутнього розвитку об'єкта *прогнозування* та відправною інформацією при цьому є думка спеціалістів, які займаються дослідженнями й розробками в прогнозованій галузі. Робоча група проводить аналіз загальних задач захисту ВзОД для створення набору елементарних ВзОД (ЕлВзОД).

При визначенні набору ЕлВзОД необхідні:

- мета робіт та заходів, що будуть впроваджуватися;
- перелік (набір) ЕлВзОД, що повністю забезпечуватиме ВзОД;
- можливий діапазон значень ЕлВзОД;
- шкала вимірювання значень ЕлВзОД, що дозволить визначити можливі наслідки та збитки у разі розкриття конкурентами даної інформації, що становить конкретну ЕлВзОД;
- необхідна та можлива точність вимірювання значень ЕлВзОД для рішення задач конкурентної розвідки.

Етап виявлення та аналізу ОВ характерний тим, що робоча група методом експертних оцінок для кожного ЕлВзОД виявляє усі ОВ. Даний етап є найбільш відповідальним та складним. Досвід членів робочої групи є запорукою його вірної реалізації.

Коректність та вичерпність визначення ЕлВзОД може забезпечити відповідність розроблювальних та повноту заходів захисту задачам по захисту ВзОД, та з певною вірогідністю гарантувати захист ВзОД.

Вичерпність визначення ЕлВзОД забезпечується:

- розумінням призначення та мети захисту;
- коректністю та змістовністю поставлених задач по захисту;
- компетентністю та відповідно високий професійний рівеньробочої групи;
- вичерпністю переліку інформацію, захист якої необхідно забезпечити.

Виявлення небезпечних ЗТР для ОВ та можливих результатів їх діяльності, визначення відомостей, що можуть бути здобуті з використанням певного засобу технічної розвідки. Для кожної пари ОВ – ЗТР встановлюється можливість проведення розвідки – „реалізації ОВ”. Реалізація ОВ оцінюється спочатку якісно, а потім кількісно. При якісній оцінці виявляються потенційні можливості ЗТР реалізовувати ОВ. При кількісній оцінці обчислюється ймовірність реалізації ОВ. Розробка замислу захисту здійснюється методом експертних оцінок. На даному етапі методом експертних оцінок проводиться прогноз синтезу за ЗТР кожного з ЕлВзОД по набору ОВ, які реалізуються ЗТР.

Складність етапу пояснюється принциповою неоднозначністю результату синтезу по набору ОВ. Причина – набір ОВ, їх значення встановлюється імовірнісним. Тим не менш, робоча група з’ясовує яке рішення може прийняти система розвідки по одержаним розвідданим.

Якщо ОВ проявляються потужно, то краще надалі застосовувати спосіб технічної дезінформації цієї ЕлВзОД. У противному випадку – краще застосовувати спосіб приховування.

Технічна дезінформація також може стати більш дієвою у випадку, коли про існування об’єкту захисту вже відомо. У цьому випадку доцільнішим буде введення в оману конкурентів. Під «введенням в оману» слід розуміти процес надання конкуренту завідома неправдивої інформації, видаючи її за автентичну.

Не менш важливим є визначення економічної обґрунтованості розроблених заходів захисту, чи можливо після впровадження розроблених заходів захисту зберегти економічну привабливість розробки. Великі витрати на впровадження та підтримку заходів захисту

призводять до зменшення розміру чистого прибутку, через компенсування витрат на заходи захисту, що в свою чергу зменшує економічну привабливість об’єкту [4]. Повинен бути знайдений компроміс, за якого витрати на заходи захисту матимуть оптимальне поєднання гарантії безпеки від конкурентної розвідки та ціни, а витрати будуть співрозмірні із загальним бюджетом розробки та виробництва об’єкту захисту. Економічна доцільність розробленого комплексу заходів захисту повинна бути такою, що витрати на впровадження та підтримку працездатності заходів захисту не перевищуватимуть втрат від витоку інформації. Така оцінка має проводитися економістом із складу робочої групи.

Висновки

Визначено, що розробка заходів контролю заходів захисту направлена на неможливість та попередження невиконання розроблених заходів захисту для розроблюваного об’єкту. Заходи контролю необхідні для підтримки працездатності створеної системи заходів захисту.

Відображено можливість застосування апарату теорії множин для формалізації задач захисту відомостей з обмеженим доступом. Вказано на необхідність пов'язування задач захисту від конкурентної розвідки з основними етапами життєвого циклу виготовленого продукту. Методикою визначено основні цілі та завдання кожного етапу для проведення робіт. Приведена методика може використовуватися при організації заходів захисту від технічних засобів конкурентної розвідки.

Список літератури: 1.*Заболотний, В.І.*, Класифікація технічних каналів витоку інформації / В.І. Заболотний // Радіотехніка. – 2003. – Вип. 134. 2. *Воронов, Ю.П.* Конкурентна розвідка : посібник. – Новосибірськ : Вид-во Новосибір. держ. ун-ту, 2007. – С. 32. 3. *Цивільний* кодекс України, ст. 505. 2003 4. *Заболотний, В.І.* Обґрунтування вибору заходів захисту характеристик продукції від конкурентної розвідки // Прикладна радіоелектроніка. – 2013. – Т. 12. – №2. – С. 351-356.

Харківський національний
університет радіоелектроніки

Надійшла до редколегії 03.04.2017

УДК 681.142

В. А. КРАСНОБАЕВ, д-р техн. наук, С. А. КОШМАН, канд. техн. наук, А. С. ЯНКО

УСОВЕРШЕНСТВОВАННЫЙ МЕТОД ОПРЕДЕЛЕНИЯ АЛЬТЕРНАТИВНОЙ СОВОКУПНОСТИ ЧИСЕЛ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

Введение

Современный этап развития науки и техники отличается все более сложными задачами, которые требуют быстрого решения. Однако сложность решаемых задач опережает темпы нарастания мощности универсальных компьютеров. В этом аспекте основным направлением совершенствования вычислительной системы обработки данных реального времени является повышение ее производительности. Известно, что одним из возможных направлений в разработке высокопроизводительных вычислительных систем является распараллеливание решаемых задач и алгоритмов на уровне арифметических микроопераций. Одним из вариантов распараллеливания является переход к вычислениям в нетрадиционной машинной арифметике с нетрадиционным представлением операндов. Из множества нетрадиционных арифметик наибольшее практическое применение в вычислительных системах нашла непозиционная система счисления остаточных классов (СОК) [1 – 3].

Совокупность положительных свойств СОК определяет следующие классы задач, в которых она существенно эффективнее позиционной арифметики: криптографические и модульные преобразования (реализация криптопреобразований в группе точек эллиптической кривой, а также для реализации алгоритма хеширования и генератора псевдослучайных чисел [4, 5]), обработка сигналов, обработка (сжатие) изображений, целочисленная обработка данных большой (сотни бит) разрядности в реальном времени, векторная и матричная обработка больших массивов информации, нейрокомпьютерная обработка информации, реализация алгоритмов БПФ и ДПФ и оптоэлектронная табличная обработка информации.

Известно, что в СОК существует необходимость определения альтернативной совокупности (АС) $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ неправильных $\tilde{A} = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ чисел. Под понятием АС $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ неправильного (искаженного) числа $\tilde{A} = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ понимается совокупность $\left\{ m_{l_\kappa} \right\}$ $(\kappa = \overline{1, \rho})$ из ρ оснований СОК, по которым правильное (неискаженное) число (кодовое слово) $A = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ может отличаться от данной совокупности $\left\{ \tilde{A} \right\}$ возможных производных неправильных чисел. При этом предполагается, что может возникнуть только однократная (по одному из остатков m_i $(i = \overline{1, n+1})$ числа $A = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ ошибка (искажение одного из (n+1)-го остатка) в правильном числе A.

Отметим, что АС рассматривается при введении в кодовую структуру СОК минимальной информационной избыточности путем добавления к n информационным одного (k=1) дополнительного (контрольного) основания m_{n+1} СОК, при условии, что $m_i < m_{n+1}$ $(i = \overline{1, n})$. В этом случае общее количество кодовых слов в СОК $N_{OK} = \prod_{i=1}^{n+1} m_i$. Количество правильных кодовых слов $N_{ПК} = \prod_{i=1}^{n} m_i$, а количество неправильных

(искаженных) кодовых слов $N_{HK} = N_{OK} - N_{ПК} = N_{ПК} \cdot (m_{n+1} - 1)$.

Необходимость определения АС может возникать в следующих основных случаях. Во-первых, при необходимости проведения процесса контроля, диагностики и коррекции ошибок данных в СОК. Во-вторых, при организации процедур контроля, диагностики и исправлении ошибок данных в СОК в процессе решения задачи в динамике вычислительного процесса (ДВП) (в реальном времени, т.е. без останова вычислений) при введении минимальной информационной избыточности [1]. Одним из основных требований к процедуре определения АС в СОК является требование уменьшения времени определения данного набора оснований. Особенно это требование критично для второго случая – при решении вычислительных задач в ДВП [6 – 9].

Таким образом, актуальной и важной научно-технической задачей является разработка новых и совершенствование существующих методов быстрого определения АС $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ чисел в СОК.

Основная часть

Все существующие методы определения АС чисел основываются на процедуре последовательного определения искомых оснований АС чисел в СОК [1].

Первый метод. АС $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ неправильного числа $\tilde{A} = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ может быть установлена последовательной проверкой каждого из оснований m_i $(i = \overline{1, n})$ СОК следующим образом. Определяется совокупность чисел, имеющих одинаковое значение остатков по всем основаниям СОК, что и число $\tilde{A}$, кроме
одного определенного основания, и отличающихся лишь значениями возможных остатков по этому основанию. Среди этой совокупности чисел может не быть ни одного правильного числа, либо может быть только одно правильное число. В последнем случае полученное число входит в АС проверяемого неправильного числа $\tilde{A}$. Рассматриваемый метод предполагает последовательное проведение аналогичных проверок для каждого из информационных оснований СОК (контрольное основание всегда входит в состав оснований АС). Результат таких последовательных проверок полностью определяет АС $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$. Недостатки данного метода определения АС: высокая вычислительная трудоемкость и значительное время определения АС.

Второй метод определения АС основан на вычислении всех возможных проекций $\tilde{A}_i = (a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_n)$ неправильного числа $\tilde{A}$, и последующем их сравнении со значением величины информационного диапазона заданной СОК. В [1] доказано, что необходимым и достаточным условием вхождения основания СОК в АС $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ числа $\tilde{A} = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ является правильность его проекции $\tilde{A}_i$. Рассмотрим пример определения АС чисел в СОК на основе использования второго метода.

Пусть необходимо определить АС числа $A_{COK} = (0 \| 0 \| 0 \| 0 \| 5)$, заданного в СОК информационными $m_1 = 3$, $m_2 = 4$, $m_3 = 5$, $m_5 = 7$ и контрольным $m_k = m_5 = 11$

основаниями. При этом $M = \prod_{i=1}^{n} m_i = \prod_{i=1}^{4} m_i = 420$ и $M_0 = M \cdot m_{n+1} = 420 \cdot 11 = \quad = 4620$.

Ортогональные базисы B_i $(i = \overline{1, n+1})$ для данной СОК даны в табл. 1.

Таблица 1

$B_1 = (1\square 0\square 0\square 0\square 0) = 1540, \quad \bar{m}_1 = 1$
$B_2 = (0\square 1\square 0\square 0\square 0) = 3465, \quad \bar{m}_2 = 3$
$B_3 = (0\square 0\square 1\square 0\square 0) = 3696, \quad \bar{m}_3 = 4$
$B_4 = (0\square 0\square 0\square 1\square 0) = 2640, \quad \bar{m}_4 = 4$
$B_5 = (0\square 0\square 0\square 0\square 1) = 2520, \quad \bar{m}_5 = 6$

Предварительно проведем контроль данных $A_{COK} = (0, 0, 0, 0, 5)$. В соответствии с процедурой контроля [1, 4] определим значение исходного числа в позиционной десятичной системе счисления (ПСС)

$$A_{ПСС} = \left(\sum_{i=1}^{n+1} a_i \cdot B_i \right) \bmod M_0 = \left(\sum_{i=1}^{5} a_i \cdot B_i \right) \bmod M_0 =$$

$$= (a_1 \cdot B_1 + a_2 \cdot B_2 + a_3 \cdot B_3 + a_4 \cdot B_4 + a_5 \cdot B_5) \bmod M_0 = (0 \cdot 1540 + 0 \cdot 3465 + 0 \cdot 3696 + 0 \cdot 2640 +$$
$$+5 \cdot 2520) \bmod 4620 = (5 \cdot 2520) \bmod 4620 = 12600 (\bmod 4620) = 3360 > 420.$$

Таким образом, в процессе контроля определено, что $A_{ПСС} = 3360 > M = 420$. В этом случае при возможности возникновений только однократных ошибок делается вывод о том, что рассматриваемое число $\tilde{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5)$ неправильное. Далее осуществим процедуру определения АС числа $\tilde{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5)$. В соответствии со вторым методом определения АС составим возможные проекции $\tilde{A}_j$ числа $\tilde{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5)$: $\tilde{A}_1 = (0 \| 0 \| 0 \| 5)$, $\tilde{A}_2 = (0 \| 0 \| 0 \| 5)$, $\tilde{A}_3 = (0 \| 0 \| 0 \| 5)$, $\tilde{A}_4 = (0 \| 0 \| 0 \| 5)$ и $\tilde{A}_5 = (0 \| 0 \| 0 \| 0)$.

Формула для вычисления значений $\tilde{A}_{j\,ПСС}$ проекций числа в ПСС имеет вид [1]

$$\tilde{A}_{j\,ПСС} = \left(\sum_{\substack{i=\overline{1;} \\ j=\overline{1, n+1.}}}^{n} a_i \cdot B_{ij} \right) \bmod M_j = (a_1 \cdot B_{1j} + a_2 \cdot B_{2j} + \ldots + a_n \cdot B_{nj}) \bmod M_j. \qquad (1)$$

В соответствии с формулой (1) вычислим все значения $A_{j\,ПСС}$. Далее проводим $(n+1)$-о сравнение чисел $\tilde{A}_{j\,ПСС}$ и числа $M = M_0 / m_{n+1}$. Если среди проекций $\tilde{A}_{i\,ПСС}$ есть числа, не находящиеся внутри информационного $[0, M)$ числового интервала (т.е. $\tilde{A}_{k\,ПСС} \geq M$), содержащего k правильных чисел, то делается вывод о том, что эти k остатков числа $\tilde{A}_{COK}$ не искажены. Ошибочными могут быть только остатки, находящиеся среди остальных $[(n+1)-k]$ остатков числа $\tilde{A}_{COK}$. Наборы рассчитанных в [8] частных рабочих
оснований и частных B_{ij} ортогональных базисов для заданных СОК представлены соответственно в табл. 2 и 3. В этом случае имеем, что

$$\tilde{A}_{1ПСС} = \left(\sum_{i=1}^{4} a_i \cdot B_{i1} \right) \bmod M_1 = (a_1 \cdot B_{11} + a_2 \cdot B_{21} +$$

$$+a_3 \cdot B_{31} + a_4 \cdot B_{41}) \bmod M_1 = (0 \cdot 385 + 0 \cdot 616 + 0 \cdot 1100 + 5 \cdot 980) \bmod 1540 = 280 < 420 .$$

Делаем вывод, что $\bar{a}_1$ – возможно искаженный остаток;

$$\tilde{A}_{2ПСС} = \left(\sum_{i=1}^{4} a_i \cdot B_{i2} \right) \bmod M_2 = (a_1 \cdot B_{12} + a_2 \cdot B_{22} +$$

$$+a_3 \cdot B_{32} + a_4 \cdot B_{42}) \bmod M_2 = (0 \cdot 385 + 0 \cdot 231 + 0 \cdot 330 + 5 \cdot 210) \bmod 1155 = 1050 > 420 .$$

Таким образом, получим, что a_2 достоверно не искаженный остаток;

$$\tilde{A}_{3ПСС} = \left(\sum_{i=1}^{4} a_i \cdot B_{i3} \right) \bmod M_3 = (a_1 \cdot B_{13} + a_2 \cdot B_{23} +$$

$$+a_3 \cdot B_{33} + a_4 \cdot B_{43}) \bmod M_3 = (0 \cdot 616 + 0 \cdot 693 + 0 \cdot 792 + 5 \cdot 672) \bmod 924 = 588 > 420 .$$

Получим, что a_3 достоверно не искаженный остаток;

$$\tilde{A}_{4ПСС} = \left(\sum_{i=1}^{4} a_i \cdot B_{i4} \right) \bmod M_4 = (a_1 \cdot B_{14} + a_2 \cdot B_{24} +$$

$$+a_3 \cdot B_{34} + a_4 \cdot B_{44}) \bmod M_4 = (0 \cdot 220 + 0 \cdot 165 + 0 \cdot 369 + 5 \cdot 540) \bmod 660 = 60 < 420 .$$

Вывод: $\bar{a}_4$ – возможно искаженный остаток; $\tilde{A}_{5ПСС} = \left(\sum_{i=1}^{4} a_i \cdot B_{i5} \right) \bmod M_5$. Так как $M_5 = M = 420$, то остаток $\bar{a}_5$ по модулю $m_k = m_5$ всегда будет в совокупности возможных искаженных остатков числа в СОК.

Таблица 2

j \ i	m_1	m_2	m_3	m_4	M_j
1	4	5	7	11	1540
2	3	5	7	11	1155
3	3	4	7	11	924
4	3	4	5	11	660
5	3	4	5	7	420

Таблица 3

B_{ij} j \ i	1	2	3	4
1	385	616	1100	980
2	385	231	330	210
3	616	693	792	672
4	220	165	396	540
5	280	105	336	120

Таким образом, для числа $\tilde{A}_{COK} = (0, 0, 0, 0, 5)$ определились точно не искаженные остатки. Это $a_2 = 0$ и $a_3 = 0$. Ошибочными могут быть остатки по основаниям m_1, m_4 и m_5, т.е. остатки $a_1 = 0$, $a_4 = 0$ и $a_5 = 5$. В этом случае для числа $\tilde{A}_{COK} = (0, 0, 0, 0, 5)$ АС равна следующей совокупности оснований СОК: $W(\tilde{A}) = \{1, 4, 5\}$. Применение второго метода позволяет несколько ускорить процесс определения АС $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ числа $A = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ за счет возможности параллельно во времени определять значения возможных проекций $\tilde{A}_j$ неправильного числа. Данное обстоятельство

снижает временную сложность определения АС. Однако отметим, что процедура определения АС числа содержит такие основные операции: перевод числа $A=(a_1,a_2,...,a_{i-1},a_i,a_{i+1},...,a_n)$ из СОК в ПСС; перевод проекций $\tilde{A}_i$ неправильного числа $A=(a_1,a_2,...,a_{i-1},a_i,a_{i+1},...,a_n)$ из СОК в ПСС и операцию сравнения чисел. В СОК перечисленные операции относятся к непозиционным операциям, требующим больших временных и аппаратных затрат на ее реализацию. Недостатки данного метода определения АС такие же, как и при первом методе: высокая вычислительная трудоемкость и значительное время определения АС. Таким образом, остается задача совершенствования второго рассмотренного метода в плане уменьшения времени определения АС.

Совершенствование известного второго метода состоит в снижении времени определения АС. Суть предложенного в статье метода определения АС чисел в СОК заключается в предварительном формировании M таблиц соответствия (таблиц первой ступени) $A=\Phi_1(\tilde{A})$ каждого правильного $A=(a_1,a_2,...,a_{i-1},a_i,a_{i+1},...,a_n)$ числа (из числового диапазона $0\div M-1$), возможной совокупности $\{\tilde{A}\}$ неправильных чисел (из числового диапазона $M\div M_0-1$) при возникновении в числе А однократных (в одном остатке) ошибок. На основе анализа содержимого данных таблиц первой ступени составляется таблица второй ступени, в которой приведено соответствие $\tilde{A}=\Phi_2(A)$ каждого неправильного $\tilde{A}$ числа из числового диапазона $M\div M_0-1$ возможным значениям исправленных (правильных) $A=(a_1,a_2,...,a_{i-1},a_i,a_{i+1},...,a_n)$ чисел. Количество правильных A чисел соответствует количеству оснований СОК, содержащихся в АС $W(\tilde{A})=\{m_{l_1},m_{l_2},...,m_{l_\rho}\}$ числа A. Целесообразно рассмотреть использование предлагаемого метода определения АС для конкретной СОК, заданной информационными $m_1=2$, $m_2=3$ и контрольным основаниями $m_k=m_3=m_{n+1}=5$ ($M=2\cdot3=6$; $M_0=30$). Совокупность кодовых слов в позиционной (десятичной) системе счисления (ПСС) и в СОК представлена в табл. 4.

Таблица 4

A в ПСС	m_1	m_2	m_3	A в ПСС	m_1	m_2	m_3
0	0	0	0	15	1	0	0
1	1	1	1	16	0	1	1
2	0	2	2	17	1	2	2
3	1	0	3	18	0	0	3
4	0	1	4	19	1	1	4
5	1	2	0	20	0	2	0
6	0	0	1	21	1	0	1
7	1	1	2	22	0	1	2
8	0	2	3	23	1	2	3
9	1	0	4	24	0	0	4
10	0	1	0	25	1	1	0
11	1	2	1	26	0	2	1
12	0	0	2	27	1	0	2
13	1	1	3	28	0	1	3

14	0	2	4	29	1	2	4

Исходя из содержимого табл. 4 по числу правильных кодовых слов 0 – 5 составляются табл. 5 – 10 соответствий $A = \Phi_1(\tilde{A})$ первой ступени.

Таблица 5

0	0	0	0
15	1	0	0
10	0	1	0
20	0	2	0
6	0	0	1
12	0	0	2
18	0	0	3
24	0	0	4

Таблица 6

1	1	1	1
16	0	1	1
21	1	0	1
11	1	2	1
25	1	1	0
7	1	1	2
13	1	1	3
19	1	1	4

Таблица 7

2	0	2	2
17	1	2	2
12	0	0	2
22	0	1	2
20	0	2	0
26	0	2	1
8	0	2	3
14	0	2	4

Таблица 8

3	1	0	3
18	0	0	3
13	1	1	3
23	1	2	3
15	1	0	0
21	1	0	1
27	1	0	2
9	1	0	4

Таблица 9

4	0	1	4
19	1	1	4
24	0	0	4
14	0	2	4
10	0	1	0
16	0	1	1
22	0	1	2
28	0	1	3

Таблица 10

5	1	2	0
20	0	2	0
15	1	0	0
25	1	1	0
11	1	2	1
17	1	2	2
23	1	2	3
29	1	2	4

На основании этих таблиц формируется табл. 11 второй $\tilde{A} = \Phi_2(A)$ ступени. В табл. 8 приведено соответствие $\tilde{A} = \Phi_2(A)$ каждого неправильного $\tilde{A}$ числа из числового диапазона 6 – 29 возможным значениям исправленных (правильных) A чисел. В табл. 11 дан алгоритм определения АС $W(\tilde{A}) = \left\{ m_{l_1}, m_{l_2}, ..., m_{l_\rho} \right\}$ чисел в СОК.

Таблица 11

Неправильное $\tilde{A}$ число	Правильное A число	Значение АС $W(\tilde{A})$
$\tilde{A}_6 = (0\,0\,1)$	$A_0 = (0\,0\,0)$	$W(\tilde{A}_6) = \{m_3\}$
$\tilde{A}_7 = (1\,1\,2)$	$A_1 = (1\,1\,1)$	$W(\tilde{A}_7) = \{m_3\}$
$\tilde{A}_8 = (0\,2\,3)$	$A_2 = (0\,2\,3)$	$W(\tilde{A}_8) = \{m_3\}$
$\tilde{A}_9 = (1\,0\,4)$	$A_3 = (1\,0\,3)$	$W(\tilde{A}_9) = \{m_3\}$
$\tilde{A}_{10} = (0\,1\,0)$	$A_0 = (0\,0\,0)$	$W(\tilde{A}_{10}) = \{m_2, m_3\}$
	$A_4 = (0\,1\,4)$	
$\tilde{A}_{11} = (1\,2\,1)$	$A_1 = (1\,1\,1)$	$W(\tilde{A}_{11}) = \{m_2, m_3\}$
	$A_5 = (1\,2\,0)$	
$\tilde{A}_{12} = (0\,0\,2)$	$A_0 = (0\,0\,0)$	$W(\tilde{A}_{12}) = \{m_2, m_3\}$
	$A_2 = (0\,2\,2)$	
$\tilde{A}_{13} = (1\,1\,3)$	$A_1 = (1\,1\,1)$	$W(\tilde{A}_{13}) = \{m_2, m_3\}$
	$A_3 = (1\,0\,3)$	
$\tilde{A}_{14} = (0\,2\,4)$	$A_2 = (0\,2\,2)$	$W(\tilde{A}_{14}) = \{m_2, m_3\}$
	$A_4 = (0\,1\,4)$	
$\tilde{A}_{15} = (1\,0\,0)$	$A_0 = (0\,0\,0)$	$W(\tilde{A}_{15}) = \{m_1, m_2, m_3\}$
	$A_3 = (1\,0\,3)$	
	$A_5 = (1\,2\,0)$	
$\tilde{A}_{16} = (0\,1\,1)$	$A_1 = (1\,1\,1)$	$W(\tilde{A}_{16}) = \{m_1, m_3\}$
	$A_4 = (0\,1\,4)$	
$\tilde{A}_{17} = (1\,2\,2)$	$A_2 = (0\,2\,2)$	$W(\tilde{A}_{17}) = \{m_1, m_3\}$
	$A_5 = (1\,2\,0)$	
$\tilde{A}_{18} = (0\,0\,3)$	$A_0 = (0\,0\,0)$	$W(\tilde{A}_{18}) = \{m_1, m_3\}$
	$A_3 = (1\,0\,3)$	
$\tilde{A}_{19} = (1\,1\,4)$	$A_1 = (1\,1\,1)$	$W(\tilde{A}_{19}) = \{m_1, m_3\}$
	$A_4 = (0\,1\,4)$	
$\tilde{A}_{20} = (0\,2\,0)$	$A_0 = (0\,0\,0)$	$W(\tilde{A}_{20}) = \{m_1, m_2, m_3\}$
	$A_2 = (0\,2\,2)$	
	$A_5 = (1\,2\,0)$	
$\tilde{A}_{21} = (1\,0\,1)$	$A_1 = (1\,1\,1)$	$W(\tilde{A}_{21}) = \{m_2, m_3\}$
	$A_3 = (1\,0\,3)$	
$\tilde{A}_{22} = (0\,1\,2)$	$A_2 = (0\,2\,2)$	$W(\tilde{A}_{22}) = \{m_2, m_3\}$
	$A_4 = (0\,1\,4)$	
$\tilde{A}_{23} = (1\,2\,3)$	$A_3 = (1\,0\,3)$	$W(\tilde{A}_{23}) = \{m_2, m_3\}$
	$A_5 = (1\,2\,0)$	
$\tilde{A}_{24} = (0\,0\,4)$	$A_0 = (0\,0\,0)$	$W(\tilde{A}_{24}) = \{m_2, m_3\}$

	$A_4 = (0 \square 1 \square 4)$	
$\tilde{A}_{25} = (1 \square 1 \square 0)$	$A_1 = (1 \square 1 \square 1)$	$W(\tilde{A}_{25}) = \{m_2, m_3\}$
	$A_5 = (1 \square 2 \square 0)$	
$\tilde{A}_{26} = (0 \square 2 \square 1)$	$A_2 = (0 \square 2 \square 2)$	$W(\tilde{A}_{26}) = \{m_3\}$
$\tilde{A}_{27} = (1 \square 0 \square 2)$	$A_3 = (1 \square 0 \square 3)$	$W(\tilde{A}_{27}) = \{m_3\}$
$\tilde{A}_{28} = (0 \square 1 \square 3)$	$A_4 = (0 \square 1 \square 4)$	$W(\tilde{A}_{28}) = \{m_3\}$
$\tilde{A}_{29} = (1 \square 2 \square 4)$	$A_5 = (1 \square 2 \square 0)$	$W(\tilde{A}_{29}) = \{m_3\}$

Рассмотрим пример определения АС чисел в СОК предложенным в статье табличным методом. Пусть дано неправильное число $\tilde{A}_{15} = (1 \square 0 \square 0)$ (табл. 4). Необходимо определить АС этого числа. Первоначально формируются 6 таблиц (табл. 5 – 10) соответствия первой ступени каждого правильного $A = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ числа (из числового диапазона 0 – 5), возможной совокупности неправильных чисел (из числового диапазона 6 – 29) при возникновении в числе А однократных (в одном остатке) ошибок (табл. 4). На основе анализа содержимого данных таблиц первой ступени составляется таблица второй ступени, в которой приведено соответствие каждого неправильного числа из числового диапазона 6 – 29 возможным значениям исправленных (правильных) $A = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ чисел. Количество правильных А чисел соответствует количеству оснований СОК, содержащихся в АС $W(\tilde{A}) = \left\{m_{l_1}, m_{l_2}, ..., m_{l_\rho}\right\}$ числа А. При воздействии однократных ошибок неправильное число $\tilde{A}_{15} = (1 \square 0 \square 0)$ может быть образовано из следующих правильных А чисел.

Во-первых, правильное число $A_0 = (0 \square 0 \square 0)$ (табл. 5) может быть искажено в первом остатке $a_1 = 0$ $(\tilde{a}_1 = 1)$. Во-вторых, правильное число $A_3 = (1 \square 0 \square 3)$ (табл. 8) может быть искажено в третьем остатке $a_3 = 3$ $(\tilde{a}_3 = 0)$. И, наконец, в-третьих, правильное число $A_5 = (1 \square 2 \square 0)$ (табл. 10) может быть искажено во втором остатке $a_2 = 2$ $(\tilde{a}_2 = 0)$. Таким образом, АС $W(\tilde{A}) = \left\{m_{l_1}, m_{l_2}, ..., m_{l_\rho}\right\}$ неправильного числа $\tilde{A}_{15} = (1 \square 0 \square 0)$ будет равна значению $W(\tilde{A}_{15}) = \{m_1, m_2, m_3\}$ (табл. 11).

Выводы

Предложен усовершенствованный метод определения АС чисел в СОК. Совершенствование известного метода состоит в снижения времени определения АС. Суть предложенного в статье метода определения АС чисел в СОК заключается в предварительном формировании *M* таблиц соответствия (таблиц первой ступени) $A = \Phi_1(\tilde{A})$ каждого правильного $A = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ числа (из числового диапазона $0 \div M - 1$), возможной совокупности неправильных чисел (из числового диапазона $M \div M_0 - 1$) при возникновении в числе A однократных ошибок. Далее на основе анализа содержимого данных таблиц первой ступени составляется таблица второй ступени, в которой приведено соответствие $\tilde{A} = \Phi_2(A)$ каждого неправильного $\tilde{A}$ числа из числового диапазона $M \div M_0 - 1$ возможным значениям исправленных (правильных) $A = (a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ чисел. Применение данного метода, по сравнению с существующими методами, позволяет сократить время определения АС чисел. Это достигается, во-первых, за счет уменьшения количества последовательно проверяемых оснований СОК, по которым возможно искажение остатков правильного числа

$A=(a_1,a_2,...,a_{i-1},a_i,a_{i+1},...,a_n)$. И, во-вторых, за счет организации процесса быстрой (табличной) выборки предварительно рассчитанных данных значений АС $W(\tilde{A})=\left\{m_{l_1},m_{l_2},...,m_{l_\rho}\right\}$. Уменьшение времени определения АС $W(\tilde{A})=\left\{m_{l_1},m_{l_2},...,m_{l_\rho}\right\}$ чисел позволит в дальнейшем, при необходимости, повысить быстродействие процесса диагностики и коррекции ошибок данных в СОК.

Список литературы: 1. *Акушский, И. Я., Юдицкий, Д. И.* Машинная арифметика в остаточных. – Москва : Сов. радио, 1968. – 440с. 2. *Krasnobayev, V. A.* Method for Realization of Transformations in Public-Key Cryptography // Telecommunications and Radio Engineering. USA. – 2007. – Vol. 66, Issue 17. – PP. 1559 – 1572. 3. *Краснобаев, В.А., Кошман, С. А., Маврина, М. А.* Метод исправления однократных ошибок данных, представленных кодом класса вычетов // Электронное моделирование. – 2013. – Т. 35, № 5. – С. 43–56. 4. *Кузнецов, О. О., Горбенко, Ю. І., Колованова, Є. П.* Періодичні властивості шифргами у режимі Output Feedback // Прикладная радиоэлектроника. –Харьков : ХНУРЭ, 2014. – Т. 13, №3. – С. 239 – 251. 5. *Кузнецов, А. А., Швагер, А. С., Фесенко, Д. А.* Сокрытие данных в кластерных файловых системах // Радиотехника. – 2015. – Вып. 181. – С. 86–100. 6. *Мороз, С. А., Краснобаев, В.А.* Методы контроля, диагностики и коррекции ошибок данных в информационно-телекоммуникационной системе, функционирующей в классе вычетов // Інформаційно-керуючі системи на залізничному транспорті. – 2012. – № 2. – С. 60 – 78. 7. *Krasnobayev, V. A., Koshman, S. A., Mavrina, M. A.* A method for increasing the reliability of verification of data represented in a residue number system // Cybernetics and Systems Analysis. November 2014. Volume 50, Issue 6. pp 969-976.
8. *Krasnobayev, V. A., Yanko, A. S., Koshman, S. A.* A Method for arithmetic comparison of data represented in a residue number system // Cybernetics and Systems Analysis. January 2016. Volume 52, Issue 1. pp. 145-150. 9. *Материалы* Междунар. науч.-техн. конф. "50 лет модулярной арифметике" // МИЭТ. Зеленоград, Моск. обл. 23-25 ноября 2005г. – С. 101-130.

Харьковский национальный университет имени В.Н.Каразина

Поступила в редколлегию 20.04.2017

МЕХАНИЗМЫ И МЕТОДЫ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

УДК 621.391:519.7

А. Н. АЛЕКСЕЙЧУК, д-р техн. наук, А. А. МАТИЙКО

ОЦЕНКИ ВЕРОЯТНОСТИ ОБРАТИМОСТИ СЛУЧАЙНЫХ МНОГОЧЛЕНОВ, ИСПОЛЬЗУЕМЫХ В МОДИФИЦИРОВАННОЙ ВЕРСИИ КРИПТОСИСТЕМЫ NTRU

Введение

Асимметричная система шифрования NTRU предложена в 1996 г. [1] и является первым представителем широкого класса криптосистем с одноименным названием, стойкость которых основана на сложности нахождения коротких векторов в некоторых решетках (см., например, работы [2, 3] и приведенные там ссылки).

Для задания криптосистем из этого класса используют кольцо усеченных многочленов $R_{n,q} = \mathbf{Z}_q[x]/(x^n - 1)$, где n и q – взаимно простые натуральные числа, а также натуральное число p, взаимно простое с q (обычно в качестве n выбирают простое число от 100 до 2000, а q и p полагают равными 2048 и 3 соответственно [4]). Секретным ключом криптосистемы служит пара многочленов $F(x), g(x) \in R_{n,q}$ таких, что многочлен $f(x) = 1 + pF(x)$ обратим в кольце $R_{n,q}$, а открытым ключом – многочлен $h = pg(x)(f(x))^{-1} \in R_{n,q}$.

В доступных публикациях описаны различные способы формирования многочленов $F(x)$ и $g(x)$, исходя из условий практичности и стойкости криптосистем к известным атакам. Например, в [4, 5] предлагается выбирать эти многочлены случайно равновероятно из некоторых множеств многочленов с коэффициентами 0, ± 1 и малым числом ненулевых коэффициентов. В обоснованно стойком варианте NTRU [6] $F(x)$ и $g(x)$ выбираются независимо друг от друга в соответствии с дискретным гауссовым распределением, а в [7] – случайно из некоторых множеств так называемых многочленов-произведений. Отметим, что способ формирования многочленов $F(x)$ и $g(x)$ существенно влияет на стойкость и практичность соответствующей версии криптосистемы NTRU. При этом указанный способ должен гарантировать высокую вероятность обратимости многочлена $f(x) = 1 + pF(x)$.

Цель статьи – построение оценок вероятности обратимости многочлена $f(x)$ в предположении, что коэффициенты многочлена $F(x)$ являются независимыми случайными величинами, принимающими значения ± 1 и 0 с вероятностями θ и $1 - 2\theta$ соответственно, где $\theta \in (0, 1/2)$. Эта схема формирования многочленов служит приближением к традиционной схеме [4, 5], однако является более удобной при исследовании стойкости криптосистемы к некоторым атакам. В частности, эта схема используется в [5] для (эвристической) оценки стойкости NTRU относительно так называемых атак на основе ошибок расшифрования.

Дальнейшее изложение в статье построено следующим образом. В п. 1 для простого числа n и взаимно простого с ним числа q описано строение кольца $R_{n,q}$ и приведена формула для порядка группы обратимых элементов этого кольца. В п. 2 для различных простых n и q получены точные выражения вероятностей некоторых вспомогательных событий, связанных с обратимостью случайного многочлена $f(x)$, а в п. 3 – при некоторых дополнительных ограничениях на n и q – выражения и оценки вероятности обратимости этого многочлена. Представлены также результаты численных расчетов, позволяющие

судить о значениях указанной вероятности. В заключительной части статьи сформулированы краткие выводы.

1. Строение кольца усеченных многочленов

Пусть n и q – взаимно простые натуральные числа, $n,q>1$. Обозначим $\mathbf{Z}_q$ – кольцо классов вычетов по модулю q и рассмотрим кольцо $R_{n,q}=\mathbf{Z}_q[x]/(x^n-1)$.

Напомним (см., например, [8]), что кольцом Галуа порядка p^{lm} и характеристики p^l, где p – простое, l,m – натуральные числа, называется кольцо $\mathbf{Z}_{p^l}[x]/(F(x))$, где $F(x)\in\mathbf{Z}_{p^l}[x]$ – унитарный многочлен степени m, образ которого над полем $\mathbf{Z}_p$ (получаемый в результате приведения всех коэффициентов многочлена $F(x)$ по модулю p) неприводим над этим полем. Кольцо Галуа однозначно с точностью до изоморфизма определяется своими порядком и характеристикой и обозначается $\mathbf{GR}(p^{lm},p^l)$. Порядок группы обратимых элементов этого кольца $|\mathbf{GR}(p^{lm},p^l)^*|=p^{(l-1)m}(p^m-1)$.

Утверждение 1. Пусть n – нечетное простое число, $q=q_1^{\,l_1}\cdots q_s^{\,l_s}$ – каноническое разложение числа q, m_i – показатель, которому принадлежит q_i по модулю n (то есть наименьшее натуральное число, для которого $q_i^{\,m_i}\equiv 1 \bmod n$). Тогда

$$R_{n,q}\cong\bigoplus_{i=1}^{s}R_{n,q_i^{\,l_i}}\,, \tag{1}$$

и

$$R_{n,q_i^{\,l_i}}\cong\mathbf{Z}_{q_i^{\,l_i}}\oplus\underbrace{\mathbf{GR}(q_i^{\,l_i m_i},q_i^{\,l_i})\oplus\cdots\oplus\mathbf{GR}(q_i^{\,l_i m_i},q_i^{\,l_i})}_{(n-1)/m_i},\ i\in\overline{1,s}\,. \tag{2}$$

При этом вероятность события, состоящего в том, что случайный равновероятный элемент кольца обратим,

$$p_{n,q}=\prod_{i=1}^{s}\left(1-\frac{1}{q_i}\right)\left(1-\frac{1}{q_i^{\,m_i}}\right)^{\frac{n-1}{m_i}}. \tag{3}$$

Доказательство. Согласно китайской теореме об остатках [9, с. 83], $\mathbf{Z}_n\cong\bigoplus_{i=1}^{s}\mathbf{Z}_{q_i^{\,l_i}}$, откуда следует, что

$$R_{n,q}=\mathbf{Z}_q[x]/(x^n-1)\cong\bigoplus_{i=1}^{s}\mathbf{Z}_{q_i^{\,l_i}}[x]/(x^n-1)=\bigoplus_{i=1}^{s}R_{n,q_i^{\,l_i}}\,.$$

Далее, каноническое разложение многочлена x^n-1 над полем $\mathbf{Z}_{q_i}$ имеет вид $x^n-1=(x-1)f_{1,i}(x)\cdots f_{t_i,i}(x)$, где $f_{1,i}(x),\ldots,f_{t_i,i}(x)$ – различные неприводимые многочлены степени m_i, $t_i=(n-1)/m_i$ [10, теор. 2.47]. Отсюда на основании леммы Гензеля [8, с. 152], следует, что существуют попарно взаимно простые унитарные многочлены $F_{1,i}(x),\ldots,F_{t_i,i}(x)\in\mathbf{Z}_{q_i^{\,l_i}}[x]$ такие, что в кольце $\mathbf{Z}_{q_i^{\,l_i}}[x]$ выполняется равенство $x^n-1=(x-1)F_{1,i}(x)\cdots F_{t_i,i}(x)$ и для любого $j\in\overline{1,t_i}$ образ многочлена $F_{j,i}(x)$ над полем $\mathbf{Z}_{q_i}$ равен $f_{j,i}(x)$. Следовательно, на основании китайской теоремы об остатках и определения

колец Галуа,

$$R_{n,q_i^{l_i}} \cong \mathbf{Z}_{q_i^{l_i}}[x]/(x-1) \oplus \mathbf{Z}_{q_i^{l_i}}[x]/(F_{1,i}(x)) \oplus \cdots \oplus \mathbf{Z}_{q_i^{l_i}}[x]/(F_{t_i,i}(x)) \cong$$
$$\cong \mathbf{Z}_{q_i^{l_i}} \oplus \underbrace{\mathbf{GR}(q_i^{l_i m_i}, q_i^{l_i}) \oplus \cdots \oplus \mathbf{GR}(q_i^{l_i m_i}, q_i^{l_i})}_{(n-1)/m_i}.$$

Итак, справедливы соотношения (1) и (2).

Для доказательства равенства (3) достаточно заметить, что в силу (1), (2)

$$|R_{n,q}{}^*| = \prod_{i=1}^{s} |\mathbf{Z}_{q_i^{l_i}}{}^*| \cdot |\mathbf{GR}(q_i^{l_i m_i}, q_i^{l_i})^*|^{\frac{n-1}{m_i}} =$$

$$= \prod_{i=1}^{s} q_i^{l_i-1}(q_i-1)\,(q_i^{(l_i-1)m_i}(q_i^{m_i}-1))^{\frac{n-1}{m_i}},\ |R_{n,q}| = \prod_{i=1}^{s} q_i^{l_i}\,(q_i^{l_i m_i})^{\frac{n-1}{m_i}}$$

и $p_{n,q} = |R_{n,q}{}^*| \cdot |R_{n,q}|^{-1}$.

Утверждение доказано.

Следствие 1. Пусть выполняется условие утверждения 1. Тогда:

а) если q – простое число, показатель которого по модулю n равен $n-1$, то

$$R_{n,q} \cong \mathbf{GF}(q) \oplus \mathbf{GF}(q^{n-1}),\ p_{n,q} = \left(1-\frac{1}{q}\right)\left(1-\frac{1}{q^{n-1}}\right);$$

б) если $q = 2^l$ и 2 является примитивным элементом поля $\mathbf{Z}_n$, то

$$R_{n,q} \cong \mathbf{Z}_{2^l} \oplus \mathbf{Z}_{2^l}[x]/(\Phi_n(x)),$$

где многочлен $\Phi_n(x) = x^{n-1} + x^{n-2} + \cdots + 1$ неприводим над полем $\mathbf{Z}_2$, и

$$p_{n,q} = \frac{1}{2}\left(1-\frac{1}{2^{n-1}}\right).$$

2. Формулы для вероятностей вспомогательных событий

Пусть теперь n и q – различные простые числа, p – натуральное число, взаимно простое с q, $p < q-1$, $\xi_0, \xi_1, \ldots, \xi_{n-1}$ – независимые случайные величины, распределенные по закону

$$\mathbf{P}\{\xi_i = 1\} = \mathbf{P}\{\xi_i = -1\} = \theta,\ \mathbf{P}\{\xi_i = 0\} = 1-2\theta,\ i \in \overline{0,n-1}, \tag{4}$$

где $\theta \in (0, 1/2)$. Требуется оценить вероятность $\pi_{n,q}$ события, состоящего в том, что элемент кольца $R_{n.q}$, соответствующий многочлену $f(x) = 1 + pF(x)$, где $F(x) = \xi_0 + \xi_1 x + \ldots + \xi_{n-1}x^{n-1}$, необратим в этом кольце.

Отметим, что если q является степенью простого числа $\bar{q}$, то на основании изложенного $f(x) \in R_{n,q}{}^*$ тогда и только тогда, когда $f(x) \in R_{n,\bar{q}}{}^*$. Поэтому результаты, изложенные ниже для простого q, справедливы также в случае, когда q является степенью простого
числа.

Для нахождения оценок вероятности $\pi_{n,q}$ рассмотрим каноническое разложение

многочлена x^n-1 над полем $\mathbf{Z}_q$. Пусть m – показатель, которому принадлежит q по модулю n. Тогда $x^n-1=(x-1)f_1(x)\cdots f_t(x)$, где $f_1(x),\dots,f_t(x)$ – различные неприводимые многочлены степени m над полем $\mathbf{Z}_q$, $t=(n-1)/m$ [10, теор. 2.47]. Обозначим α_j произвольный
корень многочлена $f_j(x)$ в поле $\mathbf{GF}(q^m)$, $j\in\overline{1,t}$. Положим $\alpha_0=1$,

$$\pi_{n,q}(\alpha_j)=\mathbf{P}\{f(\alpha_j)=0\},\ j\in\overline{0,t}. \tag{5}$$

Ясно, что $f(x)\notin R_{n,q}{}^*$ тогда и только тогда, когда существует $j\in\overline{0,t}$ такое, что $f(\alpha_j)=0$. Отсюда вытекают следующие неравенства:

$$\max_{0\le j\le t}\pi_{n,q}(\alpha_j)\le\ \pi_{n,q}\le\pi_{n,q}(\alpha_0)+t\max_{1\le j\le t}\pi_{n,q}(\alpha_j). \tag{6}$$

Следующее утверждение устанавливает явные выражения параметров (5).

Утверждение 2. Для любого $j\in\overline{0,t}$ справедливо равенство

$$\pi_{n,q}(\alpha_j)=q^{-m}\sum_{x\in\mathbf{GF}(q^m)}\cos\left(\frac{2\pi\,\mathrm{Tr}(x)}{q}\right)\prod_{k=0}^{n-1}\left(1-2\theta\left(1-\cos\left(\frac{2\pi p\,\mathrm{Tr}(\alpha_j{}^k x)}{q}\right)\right)\right), \tag{7}$$

где $\mathrm{Tr}(z)=z+z^q+\cdots+z^{q^{m-1}}$ – абсолютный след элемента $z\in\mathbf{GF}(q^m)$.

Доказательство. Согласно определению

$$\pi_{n,q}(\alpha_j)=\mathbf{P}\{f(\alpha_j)=0\}=\mathbf{P}\{p\xi_0+p\xi_1\alpha_j+\dots+p\xi_{n-1}\alpha_j{}^{n-1}=-1\},$$

где $\xi_0,\xi_1,\dots,\xi_{n-1}$ – независимые случайные величины, распределенные по закону (4).

Обозначим $\chi(z)=\exp\left\{\frac{2\pi i\,\mathrm{Tr}(z)}{q}\right\}$, $z\in\mathbf{GF}(q^m)$ нетривиальный аддитивный характер поля $\mathbf{GF}(q^m)$ (где $i^2=-1$; см., например, [11]). Преобразование Фурье распределения случайной величины $\eta_k=p\xi_k\alpha_j{}^k$ имеет вид

$$\psi_k(x)=\sum_{a\in\mathbf{GF}(q^m)}\mathbf{P}\{\eta_k=a\}\chi(ax)=1-2\theta+\theta(\chi(\alpha_j{}^k px)+\chi(-\alpha_j{}^k px))=$$

$$=1-2\theta+\theta(\chi(\alpha_j{}^k px)+\overline{\chi(\alpha_j{}^k px)})=1-2\theta(1-\mathrm{Re}(\chi(\alpha_j{}^k px)))=$$

$$=1-2\theta\left(1-\cos\left(\frac{2\pi p\,\mathrm{Tr}(\alpha_j{}^k x)}{q}\right)\right),\ x\in\mathbf{GF}(q^m),\ k\in\overline{0,n-1}.$$

Отсюда, используя теорему о свертке и формулу обращения для преобразования Фурье (см., например, [11]), получим, что

$$\pi_{n,q}(\alpha_j)=q^{-m}\sum_{x\in\mathbf{GF}(q^m)}\overline{\chi(-x)}\,\psi_0(x)\cdots\psi_{n-1}(x)=$$

$$=q^{-m}\sum_{x\in\mathbf{GF}(q^m)}\exp\left\{\frac{2\pi i\,\mathrm{Tr}(x)}{q}\right\}\prod_{k=0}^{n-1}\left(1-2\theta\left(1-\cos\left(\frac{2\pi p\,\mathrm{Tr}(\alpha_j{}^k x)}{q}\right)\right)\right).$$

Поскольку $\pi_{n,q}(\alpha_j)$ – вещественное число, то полученное равенство равносильно формуле (7). Утверждение доказано.

Утверждение 2 позволяет установить связь между числами (5) и весовыми спектрами некоторых линейных кодов над полем $\mathbf{Z}_q$.

Для любого $j \in \overline{0,t}$ рассмотрим линейный код C_j, состоящий из всех слов вида $c = (c_0, c_1, \dots, c_{n-1})$, где $c_k = \mathrm{Tr}(\alpha_j{}^k x)$, $k \in \overline{0,n-1}$, а x пробегает все элементы поля $\mathbf{GF}(q^m)$. Заметим, что слова кода C_j – это отрезки длины n линейных рекуррент с неприводимым над полем $\mathbf{Z}_q$ характеристическим многочленом степени m, корнем которого является элемент α_j. При этом n равно порядку указанного многочлена, то есть является минимальным периодом каждой из этих рекуррент.

Для любых $c \in C_j$, $a \in \mathbf{Z}_q$ обозначим $\nu_a(c)$ число координат слова c, равных a.

Непосредственно из утверждения 2 вытекает следующий результат.

Следствие 2. Для любого $j \in \overline{0,t}$ выполняется равенство

$$\pi_{n,q}(\alpha_j) = q^{-m}\left(1 + \sum_{c \in C_j \setminus \{0\}} \cos\left(\frac{2\pi c_0}{q}\right) \prod_{a \in \mathbf{Z}_q} \left(1 - 2\theta\left(1 - \cos\left(\frac{2\pi p a}{q}\right)\right)\right)^{\nu_a(c)}\right), \tag{8}$$

где суммирование ведется по всем ненулевым словам $c = (c_0, c_1, \dots, c_{n-1})$ кода C_j.

В частности, если $q = 2$, то

$$\pi_{n,2}(\alpha_j) = 2^{-m}\left(1 + \sum_{c \in C_j \setminus \{0\}} (-1)^{c_0} (1 - 4\theta)^{wt(c)}\right), \tag{9}$$

где $wt(c)$ – вес Хэмминга слова c.

Наконец, полагая в формуле (7) $j = 0$, получаем такой результат.

Следствие 3. Справедливо равенство

$$\pi_{n,q}(1) = q^{-1} \sum_{k=0}^{q-1} \cos\left(\frac{2\pi k}{q}\right)\left(1 - 2\theta\left(1 - \cos\left(\frac{2\pi p k}{q}\right)\right)\right)^n . \tag{10}$$

В частности, если $q = 2$, то

$$\pi_{n,2}(1) = 2^{-1}(1 - (1 - 4\theta)^n). \tag{11}$$

3. Оценки вероятности обратимости случайного многочлена f

Получим оценки вероятности $\pi_{n,q}$ при некоторых дополнительных ограничениях на параметры m и θ.

Утверждение 3. Пусть $q = 2$, $\theta < 1/4$ и

$$d \stackrel{\text{def}}{=} \frac{2^{m-1} n}{2^m - 1} - 2^{m/2-1}\left(1 - \frac{n}{2^m - 1}\right) > 0. \tag{12}$$

Тогда

$$2^{-1}(1 - (1 - 4\theta)^n) \le \pi_{n,2} \le 2^{-1}(1 - (1 - 4\theta)^n) + \frac{n-1}{2^m m}\left(1 + 2^{m-1}(1 - 4\theta)^d\right). \tag{13}$$

В частности, если $n = 2^m - 1$ – простое число, то

$$2^{-1}(1 - (1 - 4\theta)^n) \le \pi_{n,2} \le 2^{-1}(1 - (1 - 4\theta)^n) + \frac{1}{\log(n+1)}\left(1 + \frac{n+1}{2}(1 - 4\theta)^{\frac{n+1}{2}}\right). \tag{14}$$

Доказательство. Покажем, что в условиях утверждения для любого $j \in \overline{0,t}$ выполняется следующее неравенство:

$$\pi_{n,2}(\alpha_j) \le 2^{-m}(1+2^{m-1}(1-4\theta)^d). \quad (15)$$

Тогда соотношения (13), (14) следуют непосредственно из формул (6), (11) и (15).

Для доказательства неравенства (15) воспользуемся формулой (9). Поскольку $\theta < 1/4$, то

$$\pi_{n,2}(\alpha_j) = 2^{-m}\left(1 + \sum_{c\in C_j\setminus\{0\}}(-1)^{c_0}(1-4\theta)^{wt(c)}\right) \le 2^{-m}\left(1 + \sum_{c\in C_j\setminus\{0\}:c_0=0}(1-4\theta)^{wt(c)}\right) \le$$

$$\le 2^{-m}(1+2^{m-1}(1-4\theta)^{d(C_j)}),$$

где $d(C_j)$ – минимальное расстояние кода C_j. Далее, согласно [10, теор. 8.84], справедливо неравенство $d(C_j) \ge d$, откуда на основании условия (12) следует формула (15).

Утверждение доказано.

Отметим, что условие (12) выполняется только для малых по сравнению с n значениях m (порядка $2\log(\varepsilon n)$, где $\varepsilon \in (0,1)$).

Следующее утверждение позволяет найти точное значение вероятности $\pi_{n,q}$ в случае, когда m принимает наибольшее возможное значение, равное $n-1$.

Утверждение 4. Пусть $m = n-1$. Тогда, если q нечетно, то

$$\pi_{n,q} = q^{-1}\sum_{k=0}^{q-1}\cos\left(\frac{2\pi k}{q}\right)\left(1-2\theta\left(1-\cos\left(\frac{2\pi pk}{q}\right)\right)\right)^n. \quad (16)$$

Если же $q = 2$, то

$$\pi_{n,2} = 2^{-1}(1-(1-4\theta)^n) + (1-2\theta)\theta^{n-1}. \quad (17)$$

Кроме того, если p' – элемент, обратный к p по модулю q, $p' \in \overline{0,q-1}$, и $n < \min\{p', q-p'\}$, то $\pi_{n,q} = 0$.

Доказательство. Если $m = n-1$, то $t = 1$, и многочлен $x^n - 1$ над полем $\mathbf{Z}_q$ раскладывается в произведение двух различных неприводимых сомножителей: $x-1$ и $\Phi_n(x) = x^{n-1} + x^{n-2} + \cdots + 1$. Поэтому элемент $f(x)$ кольца $R_{n,q}$ необратим тогда и только тогда, когда имеет место одно из двух взаимоисключающих условий: $f(1) = 0$; $f(x) = \Phi_n(x)$.

Пусть $f(x) = 1 + pF(x)$, где $F(x) = \xi_0 + \xi_1 x + \ldots + \xi_{n-1}x^{n-1}$, а $\xi_0, \xi_1, \ldots, \xi_{n-1}$ – независимые случайные величины, распределенные по закону (4). Тогда при нечетном q в силу неравенства $p < q-1$ имеем

$$\mathbf{P}\{f(x) = \Phi_n(x)\} = \mathbf{P}\{1 + p\xi_0 = 1,\ p\xi_1 = \ldots = p\xi_{n-1} = 1\} = 0,$$

$\pi_{n,q} = \pi_{n,q}(1)$, откуда на основании формулы (10) следует равенство (16). Если же $q = 2$, то

$$\mathbf{P}\{f(x) = \Phi_n(x)\} = \mathbf{P}\{\xi_0 = 0,\ \xi_1 = \ldots = \xi_{n-1} = 1\} = (1-2\theta)\theta^{n-1},$$

$\pi_{n,2} = \pi_{n,2}(1) + (1-2\theta)\theta^{n-1}$, откуда на основании формулы (11) следует равенство (17).

Покажем, наконец, то из условия $n < \min\{p', q-p'\}$ вытекает равенство $\pi_{n,q} = 0$. Заметим, что $p' > 1$, поскольку $n > 1$; следовательно, q – нечетное простое число и по доказанному

$$\pi_{n,q} = \pi_{n,q}(1) = \mathbf{P}\{\xi_0 + \xi_1 + \cdots + \xi_{n-1} \equiv (q-p') \bmod q\}.$$

Обозначим $\eta = \xi_0 + \xi_1 + \cdots + \xi_{n-1}$. Поскольку случайные величины $\xi_0, \xi_1, \ldots, \xi_{n-1}$ принимают значения 0, ± 1, то $|\eta| \le n$. Если $\eta \ge 0$, то в силу соотношений $0 \le \eta \le n < \min\{p', q-p'\} < q$ получаем, что $\eta \bmod q = \eta < q - p'$; следовательно, сравнение $\eta \equiv (q-p') \bmod q$ не выполняется. Если $\eta \le 0$, то в силу тех же соотношений получаем, что $-\eta \bmod q = -\eta < p'$, и сравнение $\eta \equiv (q-p') \bmod q$ также не выполняется. Таким образом, событие $\{\eta \equiv (q-p') \bmod q\}$ является невозможным и $\pi_{n,q} = \mathbf{P}\{\eta \equiv (q-p') \bmod q\} = 0$.

Утверждение доказано.

Следствие 4. Пусть выполняется условие утверждения 4, $p = 3$ и $q > 3n+1$. Тогда $\pi_{n,q} = 0$.

Доказательство. Достаточно заметить, что $p' = \dfrac{q+1}{3}$, если $q \equiv -1 \bmod 3$ и $p' = \dfrac{2q+1}{3}$, если $q \equiv 1 \bmod 3$.

Соотношения (16), (17) дают возможность изучить поведение вероятности $\pi_{n,q}$ как функции параметра θ в наиболее интересных с практической точки зрения случаях:

а) $q = 2^l$, n – нечетное простое число, 2 – примитивный элемент поля $\mathbf{Z}_n$;

б) q и n – различные нечетные простые числа, $p = 3 < q-1$, и показатель, которому принадлежит q по модулю n, равен $n-1$.

Как показано выше, в случае а) выполняется равенство $\pi_{n,q} = \pi_{n,2}$. При этом вероятность $\pi_{n,q}$ практически не отличается от $0{,}5$ при всех разумных, с практической точки зрения, значениях n и θ (другими словами, в среднем каждый второй случайно сгенерированный по указанному выше закону многочлен не обратим в кольце $R_{n.q}$, рис. 1, 2).

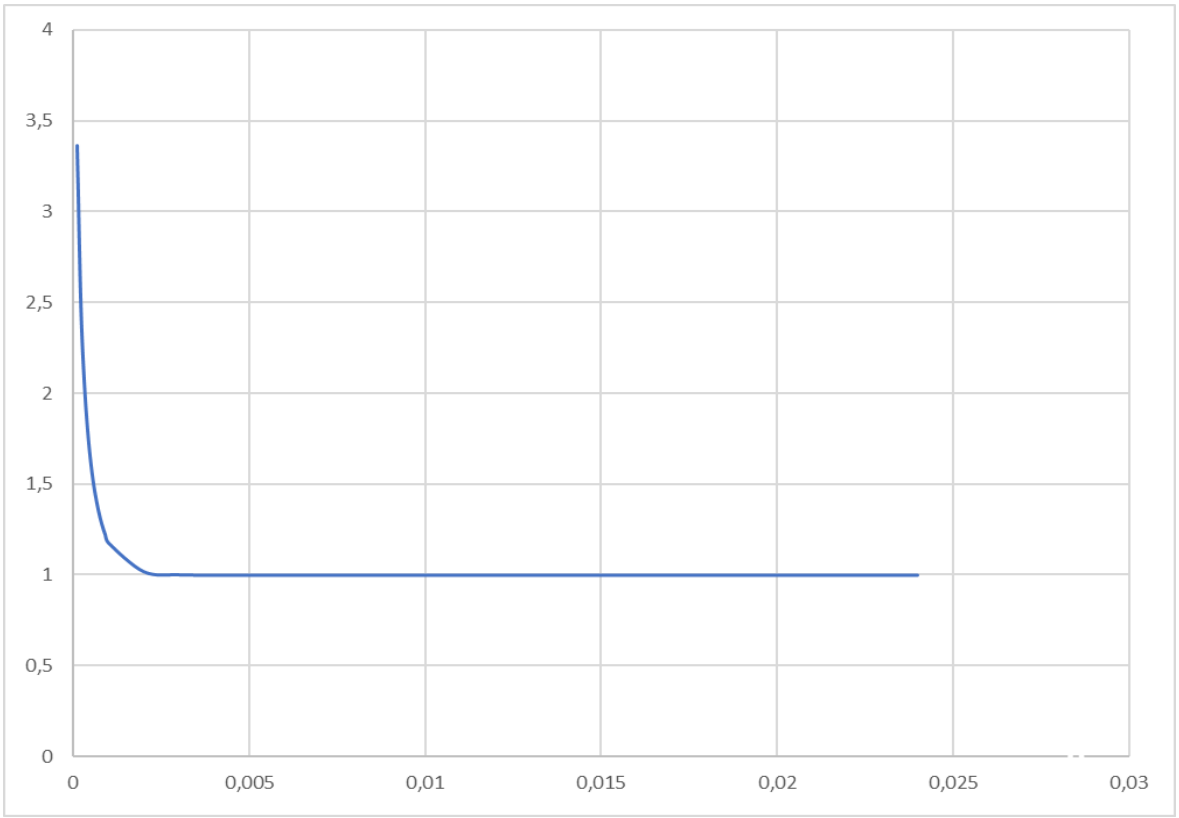


Рис. 1. Зависимость (взятого со знаком минус) двоичного логарифма

вероятности (17) от параметра θ при $n = 541$

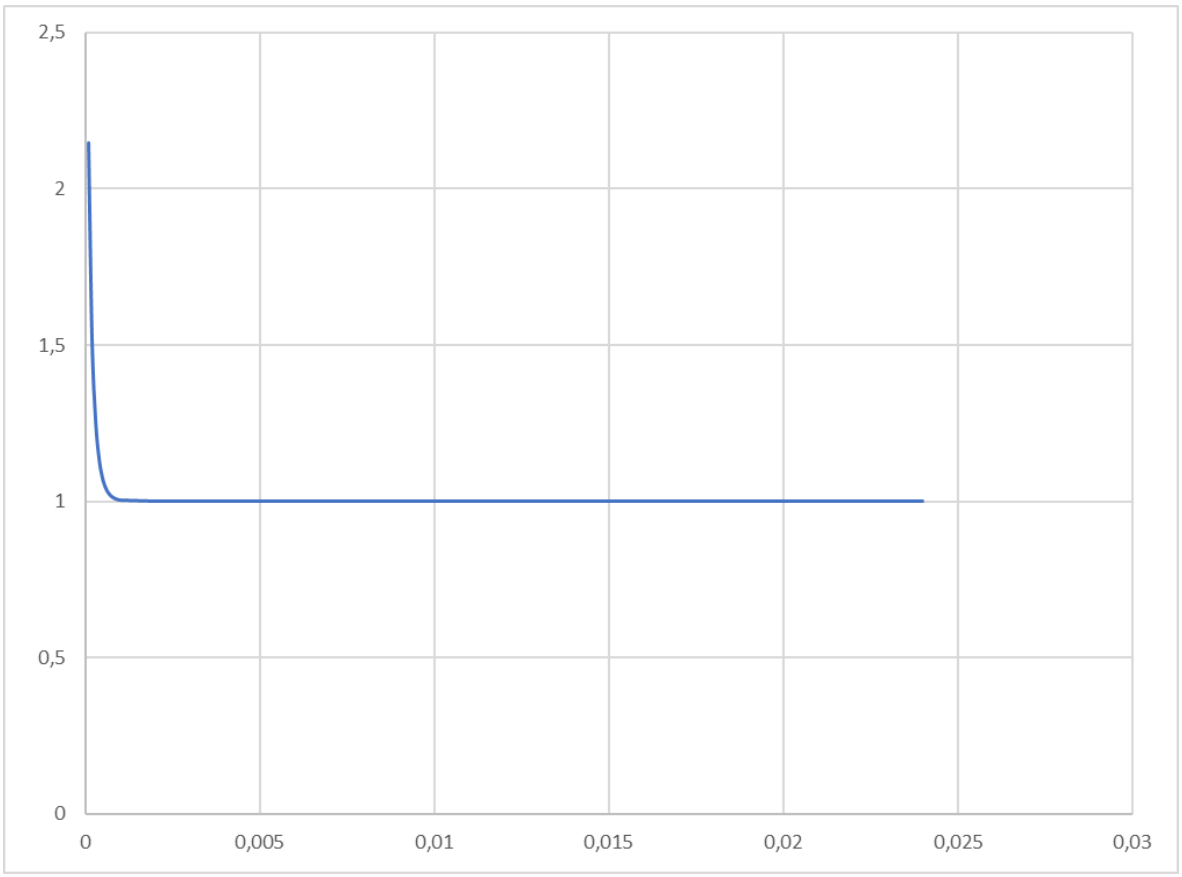


Рис. 2. Зависимость (взятого со знаком минус) двоичного логарифма вероятности (17) от параметра θ при $n = 1499$

В случае (б) вероятность $\pi_{n,q}$ быстро уменьшается с ростом q при фиксированных n и θ, обращаясь тождественно в нуль при $q > 3n+1$. Аналогичное поведение вероятности наблюдается с уменьшением параметра θ при фиксированных n и q, см. табл. 1, 2. При этом вероятность необратимости случайного многочлена в кольце $R_{n.q}$ не превосходит $1{,}5 \cdot 10^{-2}$, что существенно меньше $0{,}5$.

Таблица 1

Численные значения вероятности (16) при $n = 541$, $p = 3$

q θ	0,0001	0,001	0,01	0,1	0,2	0,3	0,4
59	$1{,}2 \cdot 10^{-44}$	$4{,}7 \cdot 10^{-25}$	$1{,}3 \cdot 10^{-8}$	$6{,}1 \cdot 10^{-3}$	$1{,}2 \cdot 10^{-2}$	$1{,}4 \cdot 10^{-2}$	$1{,}5 \cdot 10^{-2}$
73	$3{,}4 \cdot 10^{-55}$	$1{,}4 \cdot 10^{-31}$	$3{,}3 \cdot 10^{-11}$	$2{,}7 \cdot 10^{-3}$	$7{,}3 \cdot 10^{-3}$	$9{,}7 \cdot 10^{-3}$	$1{,}1 \cdot 10^{-2}$
257	$3{,}4 \cdot 10^{-243}$	$1{,}5 \cdot 10^{-157}$	$4{,}8 \cdot 10^{-75}$	$9{,}9 \cdot 10^{-17}$	$9{,}9 \cdot 10^{-10}$	$3{,}6 \cdot 10^{-6}$	$8{,}9 \cdot 10^{-6}$
331	$1{,}0 \cdot 10^{-323}$	$8{,}0 \cdot 10^{-214}$	$3{,}7 \cdot 10^{-107}$	$9{,}3 \cdot 10^{-26}$	$1{,}8 \cdot 10^{-14}$	$1{,}5 \cdot 10^{-8}$	$6{,}8 \cdot 10^{-8}$
383	0	$6{,}3 \cdot 10^{-258}$	$3{,}9 \cdot 10^{-133}$	$7{,}6 \cdot 10^{-34}$	$8{,}0 \cdot 10^{-19}$	$9{,}8 \cdot 10^{-11}$	$7{,}8 \cdot 10^{-10}$
487	0	0	$5{,}0 \cdot 10^{-186}$	$4{,}6 \cdot 10^{-52}$	$7{,}1 \cdot 10^{-29}$	$8{,}2 \cdot 10^{-16}$	$2{,}4 \cdot 10^{-14}$

Таблица 2

Численные значения вероятности (16) при $n=1499$, $p=3$

q θ	0,0001	0,001	0,01	0,1	0,2	0,3	0,4
463	0	$7{,}6\cdot10^{-250}$	$5{,}9\cdot10^{-106}$	$2{,}1\cdot10^{-19}$	$4{,}1\cdot10^{-11}$	$2{,}5\cdot10^{-8}$	$5{,}8\cdot10^{-7}$
659	0	0	$1{,}2\cdot10^{-181}$	$7{,}2\cdot10^{-37}$	$4{,}4\cdot10^{-20}$	$2{,}5\cdot10^{-14}$	$1{,}9\cdot10^{-11}$
787	0	0	$3{,}3\cdot10^{-235}$	$5{,}3\cdot10^{-51}$	$1{,}9\cdot10^{-27}$	$2{,}9\cdot10^{-19}$	$3{,}7\cdot10^{-15}$
827	0	0	$7{,}1\cdot10^{-254}$	$3{,}2\cdot10^{-56}$	$3{,}4\cdot10^{-30}$	$4{,}2\cdot10^{-21}$	$1{,}6\cdot10^{-16}$
1151	0	0	0	$9{,}9\cdot10^{-105}$	$2{,}8\cdot10^{-56}$	$1{,}2\cdot10^{-38}$	$1{,}1\cdot10^{-29}$
1289	0	0	0	$1{,}2\cdot10^{-129}$	$4{,}6\cdot10^{-70}$	$6{,}3\cdot10^{-48}$	$1{,}2\cdot10^{-36}$

Выводы

Полученные аналитические соотношения позволяют оценивать (а в ряде практически важных случаев – вычислять) значения вероятности обратимости случайных многочленов, используемых в качестве секретных ключей рассмотренной модификации криптосистемы NTRU. Эти соотношения могут быть использованы также для выбора параметров n, q и θ указанной криптосистемы.

Выбор в качестве q большого простого числа предпочтительнее (по критерию высокой вероятности обратимости многочлена) по сравнению с распространенным вариантом, в котором $q=2^l$ (см. рис. 1, 2 и табл. 1, 2). Отметим, что это условие относительно параметра q не вступает в конфликт с другими известными требованиями к криптосистеме, связанными со стойкостью и практичностью.

Список литературы: 1. *Hoffstein, J., Pipher, J., Silverman, J.H.* NTRU: a new high speed public key cryptosystem // Preprint, presented at the rump session of Crypto'96. – 1996. 2. *Steinfeld, R.* NTRU cryptosystem: resent developments and emerging mathematical problems in finite polynomial rings // http://users.monach.edu.au/~rste /NTRU_survey.pdf. – 2014. 3. *Bernstein, D.J., Chuengsatiansup Ch., Lange T., van Vredendaal Ch.* NTRU Prime // http://eprint.iacr.org/2016/461. 4. *American* National Standard X9.98-2010. Lattice-based polynomial public key encryption algorithm, Part 1: key establishment, Part 2: data encryption. – 2010. 5. *Hirschhorn, P., Hoffstein, J., Howgrave-Graham, N., Whyte, W.* Choosing NTRU parameters in light of combined lattice reduction and MITM aproaches // Applied Cryptography and Network Security, LNCS. – Vol. 5536. – 2009. – P. 437 – 455. 6. *Stehle´D., Steinfeld R.* Making NTRU as secure as worst-case problems over ideal lattices // Advances in Cryptology – EUROCRYPT 2011. – Proceedings. – Springer-Verlag. – 2011. – P.27–47. 7. *Hoffstein, J., Pipher, J., Schanck, J.M., Silverman, J.H., Whyte, W., Zhang, Z.* Choosing parameters for NTRUEncrypt // http://eprint.iacr.org/2015/708. 8. *Елизаров В.П.* Конечные кольца. – Москва : Гелиос АРВ, 2006. – 304 с. 9. *Ленг, С.* Алгебра ; пер. с англ. – Москва : Мир, 1968. – 564 с. 10. *Лидл, Р., Нидеррайтер, Г.* Конечные поля : в 2 т. ; пер. с англ. – Москва : Мир, 1988. – 818 с. 11. *Babai, L.* The Fourier transform and equations over finite abelian groups // http://people.cs.uchicago.edu/~laci /ren/fourier.pdf. – 2002.

Институт специальной связи и защиты информации
НТУ «КПИ» имени И. Сикорского

Поступила в редколлегию 12.04.2017

УДК 004.056.55

О.О. КУЗНЕЦОВ, д-р техн. наук, Ю. І. ГОРБЕНКО, канд. техн. наук, І.М. БІЛОЗЕРЦЕВ, А.В. АНДРУШКЕВИЧ, О.П. НАРІЖНИЙ, канд. техн. наук

АЛГЕБРАЇЧНИЙ ІМУНІТЕТ НЕЛІНІЙНИХ БЛОКІВ СИМЕТРИЧНИХ ШИФРІВ

1. Вступ

Криптографічне перетворення грає важливу роль в забезпеченні безпеки сучасних інформаційних систем і технологій [1, 2]. Симетричні шифри через свою простоту, ефективність і багатофункціональність застосовуються практично у всіх сучасних криптопротоколах, а також використовуються як складова частина інших криптографічних примітивів: в хешуванні, формуванні псевдовипадкових послідовностей, генерації паролів та ін. Отже, аналіз і дослідження методів синтезу симетричних криптопримітивів, розробка і теоретичне обґрунтування критеріїв і показників ефективності, в тому числі окремих вузлів сучасних шифрів є важливою і актуальною науково-технічною задачею.

Ключовим компонентом сучасних симетричних шифрів є нелінійні вузли (нелінійні підстановки, таблиці замін, S-блоки), які виконують функції приховування статистичних зв'язків відкритого тексту і шифртексту, перемішування і розсіювання даних, внесення нелінійності в процедуру шифрування для протистояння різним криптоаналітичним і статистичним атакам. Таким чином, від показників ефективності нелінійних вузлів (збалансованості, нелінійності, автокореляції, кореляційної імунності та ін.) безпосередньо залежить ефективність симетричного шифру, його стійкість до більшості відомих криптографічних атак і рівень забезпечуваної безпеки інформаційних технологій.

Окремі показники ефективності нелінійних вузлів розглянуті в [3 – 9]. Поняття алгебраїчного імунітету вперше введено в роботах [10, 11] для оцінки стійкості булевих функцій до т.з. алгебраїчного криптоаналізу, запропонованого в [12]. В роботі [13] ці положення були узагальнені для булевих відображень (S-блоків). Для обчислення алгебраїчного імунітету S-блоків використовується математичний апарат базисів Грьобнера [15 – 18].

В даній роботі розглядаються різні методи розрахунку алгебраїчного імунітету, вивчається їх взаємозв'язок, наводяться результати порівняльних досліджень алгебраїчної імунності нелінійних вузлів найбільш відомих сучасних симетричних шифрів.

2. Алгебраїчний імунітет булевих функцій

Поняття алгебраїчного імунітету вперше введено в роботах [10, 11] і докладно розглянуто в дисертації [14]. Введемо необхідні для подальшого викладення визначення та позначення, дотримуючись прийнятих в [14] формулювань.

Нехай $GF(2)$ – двійкове поле та $GF(2)^n$ – n-мірний векторний простір над $GF(2)$.

Булева функція $f(x)$ від n змінних – це відображення $f(x): GF(2)^n \to GF(2)$.

Таблиця істинності булевої функції $f(x)$ від n змінних – це двійковий вихідний вектор значень функції, який містить 2^n елементів, кожен елемент належить множині $\{0, 1\}$.

Алгебраїчна нормальна форма (*поліном Жегалкіна*) булевої функції $f(x)$ від n змінних записується у вигляді:

$$f(x) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \ldots \oplus a_n x_n \oplus a_{12} x_1 x_2 \oplus a_{13} x_1 x_3 \oplus \ldots \oplus a_{(n-1)n} x_{n-1} x_n \oplus \ldots \oplus a_{123\ldots n} x_1 x_2 x_3 \ldots x_n$$

де коефіцієнти $a_i \in \{0,1\}$ і кожна булева функція реалізується поліномом Жегалкіна єдиним чином, тобто кожне представлення $f(x)$ відповідає унікальній таблиці істинності.

Алгебраїчна ступінь Deg(f) булевої функції $f(x)$ – число змінних в найдовшому доданку алгебраїчної нормальної форми функції, що має ненульовий коефіцієнт a_i. При цьому вважаємо $Deg(0)=0$.

Позначимо V_n через множину всіх відображень $GF(2)^n \to GF(2)$, тобто це множина всіх можливих булевих функцій $f(x)$ від n змінних.

Множину V_n будемо розглядати і як кільце булевих функцій і як векторний (лінійний) простір над двійковим полем, тобто $V_n = GF(2)^{2^n}$.

Булева функція $g \in V_n$ називається *анігілятором функції* $f \in V_n$, якщо

$$f \cdot g = 0$$

або

$$(f+1) \cdot g = 0 .$$

Множина різних анігіляторів булевої функції $g(x)$ утворює лінійний простір, який позначимо

$$Ann(f) = \{g \in V_n \mid f \cdot g = 0\} .$$

Лінійний простір анігіляторів ступеня $\le d$ позначимо

$$A_d^n(f) = \{g \in V_n \mid f \cdot g = 0, Deg(g) \le d\} \subset Ann(f) .$$

Поняття анігіляторів булевих функцій тісно пов'язане з оцінкою ефективності алгебраїчного криптоаналізу поточних шифрів [10]. Зокрема, при використанні фільтруючого генератора (див. рис. 1) псевдовипадкових послідовностей (ПВП) пошук початкового стану регістра зсуву з лінійним зворотним зв'язком (РЗЛЗЗ) пов'язаний зі зниженням ступеня спільної системи поліноміальних булевих рівнянь.

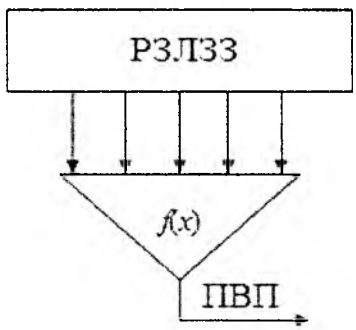


Рис. 1

Алгоритм алгебраїчного криптоаналізу, запропонований в [10], дозволяє, за певних умов, по частині перехопленої вихідної послідовності (ПВП) знаходити початковий стан РЗЛЗЗ з часовою складністю $O\left((S_n^d)^3\right)$, де

$$S_n^d = \sum_{i=0}^{d} \frac{n!}{i!(n-i)!}$$

і d – найменший ступінь ненульового анігілятора булевої функції $f(x)$, що фільтрується, або її інверсії: $f(x) + 1$.

Таким чином, завданням алгебраїчного криптоаналізу є пошук ненульових анігіляторів або, принаймні, оцінка їх мінімального ступеню. З цією метою в роботі [11] введено визначення *алгебраїчної імунності AI(f)* булевої функції $f \in V_n$:

$$AI(f) = \min\{D\,\mathrm{eg}(g) \mid g \in Ann(f) \;\; або \;\; g \in Ann(f+1)\}.$$

Величина *AI(f)* чисельно дорівнює мінімальному ступеню такої булевої функції $g \in V_n$, що $f \cdot g = 0$ або $(f+1) \cdot g = 0$.

Використовуючи введене вище поняття лінійного простору анігіляторів ступеня $\le d$, запишемо:

$$AI(f) = \min\{d \mid A_d^n(f) \neq 0 \text{ или } A_d^n(f+1) \neq 0\}, \qquad (1)$$

тобто для оцінки алгебраїчної імунності булевої функції $f \in V_n$ достатньо знайти ненульовий базис простору анігіляторів найменшого ступеню *d*.

Величина *d* дозволяє кількісно оцінити складність алгебраїчного криптоанализу і, при досить великому *d*, гарантувати стійкість поточного криптоалгоритму до алгебраїчної атаки.

Алгоритм обчислення алгебраїчної імунності булевих функцій. Один з алгоритмів розрахунку алгебраїчної імунності булевих функцій представлений в дисертаційній роботі [14]. Він заснований на побудові базису лінійного простору анігіляторів $A_d^n(f)$ заданого ступеня *d*. Ітеративно збільшуючи *d* і повторюючи побудову базису простору $A_d^n(f)$ оцінку *AI(f)* отримаємо за формулою (1), тобто через ненульовий базис анігіляторів найменшого ступеню.

Для викладу суті алгоритму необхідно ввести такі додаткові позначення.

Моном (одночлен) відносно змінних $x_1, \ldots, x_n$ запишемо у вигляді

$$x^u = \prod_{i=1}^{n} x_i^{u_i} = \begin{cases} x_i, u_i = 1, \\ 1, u_i = 0, \end{cases}$$

де вектори $x, u \in V_2^n, x = (x_1, \ldots, x_n), u = (u_1, \ldots, u_n)$.

Ступінь одночлена x^u визначається вагою Хеммінга (числом ненульових координат) $w_h(u)$ вектора $u=(u_1, \ldots, u_n)$, тобто

$$Deg(x^u) = w_h(u).$$

З урахуванням цих позначень булеву функцію $f(x)$ в алгебраїчній нормальній формі (у формі полінома Жегалкіна) запишемо у вигляді

$$f(x) = \sum_{u \in GF(2)^n} a_u x^u, \;\; a_u \in GF(2). \qquad (2)$$

Функцію (анігілятор) $g \in A_d^n(f)$ також представимо у вигляді полінома Жегалкіна

$$g(x) = \sum_{v \in GF(2)^n : w_h(v) \le d} b_v x^v, \qquad (3)$$

де $b_v \in GF(2)$ – невідомі коефіцієнти анігілятора, $w_h(v)$ – вага Хеммінга вектора $v=(v_1, \ldots, v_n)$.

Функція *g* належить простору $A_d^n(f)$ тільки в тому випадку, якщо для будь-якого $x \in GF(2)^n$ виконується рівність $f(x) \cdot g(x) = 0$.

Підставивши (2) та (3) отримаємо:

$$f(x) \cdot g(x) = \left(\sum_{u \in GF(2)^n} a_u x^u \right) \left(\sum_{v \in GF(2)^n : w_h(v) \le d} b_v x^v \right) = \sum_{u \in GF(2)^n} \left(\sum_{v \in GF(2)^n : w_h(v) \le d} a_u b_v x^{u \vee v} \right) = 0,$$

де $u \vee v = (u_1 \vee v_1, \ldots, u_n \vee v_n)$, $\vee$ – диз'юнкція (логічна операція АБО).

Після групування доданків за загальним множником, отримаємо рівність:

$$\sum_{w\in GF(2)^n}\left(\sum_{a_u,b_v:a_u\vee b_v=w} a_u b_v\right)x^w = 0, \tag{4}$$

яка виконується для будь-якого $w \in GF(2)^n$.

Отже, маємо систему лінійних однорідних рівнянь

$$\begin{cases}\sum\limits_{a_u,b_v:a_u\vee b_v=w} a_u b_v = 0, \quad \forall w \in GF(2)^n,\end{cases} \tag{5}$$

відносно невідомих коефіцієнтів b_v анігілятора $g(x)$.

Рішення системи (5), наприклад, методом Гауса, задає базис простору $A_d^n(f)$.

Приклад. Для $n = 2$ та $d = 1$ маємо:

$$f(x) = a_{00} + a_{10}x_1 + a_{01}x_2 + a_{11}x_1x_2,$$
$$g(x) = b_{00} + b_{10}x_1 + b_{01}x_2.$$

Після підстановки в $f(x)\cdot g(x) = 0$ отримаємо

$$f(x)\cdot g(x) = a_{00}b_{00} + (a_{00}b_{10} + a_{10}b_{10} + a_{10}b_{00})x_1 + (a_{00}b_{01} + a_{01}b_{01} + a_{01}b_{00})x_2 +$$
$$+ (a_{10}b_{01} + a_{01}b_{10} + a_{11}b_{00} + a_{11}b_{10} + a_{11}b_{01})x_1x_2 = 0,$$

звідки маємо систему лінійних однорідних рівнянь

$$\begin{cases} a_{00}b_{00} = 0, \\ a_{00}b_{10} + a_{10}b_{10} + a_{10}b_{00} = 0, \\ a_{00}b_{01} + a_{01}b_{01} + a_{01}b_{00} = 0, \\ a_{10}b_{01} + a_{01}b_{10} + a_{11}b_{00} + a_{11}b_{10} + a_{11}b_{01} = 0 \end{cases}$$

відносно невідомих b_{00}, b_{10}, b_{01} – коефіцієнтів функції $g(x)$.

Тоді, наприклад, для функції $f(x) = x_1 + x_2$ (тобто при $a_{00} = a_{11} = 0$ та $a_{10} = a_{01} = 1$) отримаємо систему:

$$\begin{cases} b_{10} + b_{00} = 0, \\ b_{01} + b_{00} = 0, \\ b_{01} + b_{10} = 0, \end{cases}$$

котрій задовольняє тільки два рішення:

$$b_{00} = b_{10} = b_{01} = 0, \quad тобто \quad g(x) = 0,$$
$$b_{00} = b_{10} = b_{01} = 1, \quad тобто \quad g(x) = 1 + x_1 + x_2.$$

Безпосередня перевірка показує, що $g(x) = 1 + x_1 + x_2$ дійсно є анігілятором функції $f(x) = x_1 + x_2$:

$$f(x)\cdot g(x) = (x_1 + x_2)(1 + x_1 + x_2) = x_1 + x_2 + x_1 + x_1x_2 + x_1x_2 + x_2 = 0.$$

Узагальнюючи вищевикладене, визначимо основні кроки алгоритму пошуку базису простору анігіляторів [14].

Вхід: $n \in N$, $d \in \{1,\ldots,n\}$, функція $f(x)$ (задана списком одночленів x^u з ненульовими коефіцієнтами a_u в (2)).

Вихід: Лінійний простір $A_d^n(f)$, заданий у вигляді параметричного сімейства багаточленів Жегалкіна від n булевих змінних ступеня $\le d$.

Крок 1. Представляємо функції $f(x)$ і $g(x)$ у вигляді сум (2) і (3), відповідно.

Крок 2. Відкриваємо дужки в $f(x)\cdot g(x)$ і, групуючи доданки $a_u b_v x^w$ шляхом сортування по $a_u \vee b_v = w$, отримуємо рівняння (4).

Крок 3. Складаємо систему лінійних однорідних рівнянь (5).

Крок 4. Знаходимо загальне рішення системи (5) в параметричному вигляді і подаємо на вихід алгоритму.

У дисертації [14] наводиться оцінка $O\left(m\cdot\left(S_n^d\right)^3\right)$ бітової складності розглянутого алгоритму, де m – кількість ненульових коефіцієнтів a_u в (2).

Використовуючи наведений алгоритм пошуку базису простору анігіляторів можемо обчислити алгебраїчну імунність булевої функції $f(x)$ послідовно перебираючи всі значення $d > 0$ до тих пір, поки не отримаємо нульовий простір анігіляторів $A_d^n(f)$ або $A_d^n(f+1)$. Мінімальне значення $d > 0$, для якого $A_d^n(f) \neq 0$ та/або $A_d^n(f+1) \neq 0$ відповідає значенню алгебраїчної імунності булевої функції $f(x)$.

Алгоритм обчислення алгебраїчної імунності *AI(f)*.

Вхід: $n \in N$, функція $f(x)$ (задана списком одночленів x^u з ненульовими коефіцієнтами a_u в (2)).

Вихід: Значення алгебраїчної імунності *AI(f)*.

Крок 1. Присвоюємо $d = 1$.

Крок 2. Обчислюємо простір анігіляторів $A_d^n(f)$ і $A_d^n(f+1)$.

Крок 3. Якщо $A_d^n(f) = 0$ і $A_d^n(f+1) = 0$ присвоюємо $d = d+1$ і переходимо до кроку 2.

Крок 4. Якщо $A_d^n(f) \neq 0$ та/або $A_d^n(f+1) \neq 0$ присвоюємо $AI(f) = d$ і подаємо на вихід алгоритму.

3. Алгебраїчний імунітет булевих відображень (S-блоків)

Поняття алгебраїчної імунності булевих функцій в [13] узагальнено на випадок булевих відображень $F : GF(2)^n \to GF(2)^m$ (векторних булевих функцій), які реалізуються вузлами замін (таблицями підстановок, S-блоками) блокових симетричних шифрів. Для визначення алгебраїчної імунності *AI(F)* скористаємося термінами та визначеннями з [15].

Зафіксуємо натуральні числа n, m і деяке поле K. Розглянемо кінцеву систему S з m алгебраїчних рівнянь

$$\begin{cases} P_1(x_1, x_2, ..., x_n) = 0, \\ P_2(x_1, x_2, ..., x_n) = 0, \\ ... \\ P_m(x_1, x_2, ..., x_n) = 0 \end{cases} \qquad (6)$$

від змінних $x_1, x_2 \ldots, x_n$ з коефіцієнтами над полем K.

Нехай $K[x_1, x_2 \ldots, x_n]$ – множина всіх багаточленів від змінних $x_1, x_2 \ldots, x_n$ з коефіцієнтами над полем K. На цій множині визначені операції додавання і множення, а саму множину називають *кільцем багаточленів*. Це кільце комутативне (для будь-яких елементів $a, b \in K[x_1, x_2, ..., x_n]$ виконується рівність $a \cdot b = b \cdot a$), з одиницею (для всіх $a \in K[x_1, x_2, ..., x_n]$ виконується рівність $a \cdot e = a$, де $e = 1$).

Непуста підмножина I комутативного кільця з одиницею R називається *ідеалом* в R (позначається як $I \lhd R$), якщо виконуються наступні дві умови:

– для будь-яких елементів $a,b \in I$ елемент $a-b \in I$;

– для будь-яких $a \in I$ і $c \in R$ елемент $a \cdot c \in R$.

Елементи $a_1, a_2 \ldots, a_k$ складають *базис ідеалу*

$$I = (a_1, a_2, ..., a_k) = \{a_1 \cdot r_1 + a_2 \cdot r_2 + ... + a_k \cdot r_k ; r_1, r_2, ..., r_k \in R\} \subseteq R$$

Кажуть, що ідеал $I \lhd R$ *допускає кінцевий базис*, якщо в ньому знайдуться такі елементи $a_1, a_2 \ldots, a_k$, що $I = (a_1, a_2, ..., a_k)$.

Фундаментальна ***теорема Гілберта про базис*** стверджує, що кожен ідеал $I \lhd K[x_1, x_2, ..., x_n]$ допускає кінцевий базис, тобто знайдуться такі $f_1(x_1, x_2, ..., x_n)$, $f_2(x_1, x_2, ..., x_n)$, …, $f_k(x_1, x_2, ..., x_n) \in I$, що

$$I = (f_1, f_2, ..., f_k) = \{f_1 \cdot r_1 + f_2 \cdot r_2 + ... + f_k \cdot r_k ; r_1, r_2, ..., r_k \in K[x_1, x_2, ..., x_n]\}$$

З системою S (6) зв'яжемо ідеал I, породжений $P_1(x_1, x_2, ..., x_n)$, $P_2(x_1, x_2, ..., x_n)$, …, $P_m(x_1, x_2, ..., x_n)$, що відповідає рівнянням системи:

$$I(S) = (P_1, P_2, ..., P_m) = \{P_1 \cdot r_1 + P_2 \cdot r_2 + ... + P_m \cdot r_m ; r_1, r_2, ..., r_m \in K[x_1, x_2, ..., x_n]\}$$

Якщо $F \in I(S)$, тоді для кожного рішення (X_1, X_2 …, X_n) системи (6) буде виконуватися рівність

$$\begin{aligned}
&F(X_1, X_2, ..., X_n) = \\
&= P_1(X_1, X_2, ..., X_n) \cdot r_1(X_1, X_2, ..., X_n) + P_2(X_1, X_2, ..., X_n) \cdot r_2(X_1, X_2, ..., X_n) + ... + \\
&+ P_m(X_1, X_2, ..., X_n) \cdot r_m(X_1, X_2, ..., X_n) = \\
&= 0 \cdot r_1(X_1, X_2, ..., X_n) + 0 \cdot r_2(X_1, X_2, ..., X_n) + ... + 0 \cdot r_m(X_1, X_2, ..., X_n) = 0.
\end{aligned}$$

Якщо $\{P_1, P_2, ..., P_m\}$ і $\{\overline{P}_1, \overline{P}_2, ..., \overline{P}_k\}$ – два базиси одного ідеалу I, тоді системи алгебраїчних рівнянь

$$\begin{cases} P_1(x_1, x_2, ..., x_n) = 0, \\ P_2(x_1, x_2, ..., x_n) = 0, \\ ... \\ P_m(x_1, x_2, ..., x_n) = 0, \end{cases} \qquad \begin{cases} \overline{P}_1(x_1, x_2, ..., x_n) = 0, \\ \overline{P}_2(x_1, x_2, ..., x_n) = 0, \\ ... \\ \overline{P}_k(x_1, x_2, ..., x_n) = 0 \end{cases}$$

еквівалентні, тобто множини їх рішень збігаються.

Отже, множина рішень системи алгебраїчних рівнянь однозначно визначається ідеалом системи, а різні базиси одного ідеалу відповідають еквівалентним системам [15].

Припустимо, що є деякий багаточлен $h(x_1, x_2, ..., x_n) \in K[x_1, x_2, ..., x_n]$ і потрібно за кінцеве число кроків з'ясувати, чи належить він ідеалу $I \lhd K[x_1, x_2, ..., x_n]$, заданому своїм базисом $I=(f_1, f_2 \ldots, f_m)$. Іншими словами, потрібно вирішити т.зв. *задачу входження*: з'ясувати, чи існують такі багаточлени $r_1(x_1, x_2, ..., x_n)$, $r_2(x_1, x_2, ..., x_n)$, …, $r_m(x_1, x_2, ..., x_n)$, що $h = f_1 \cdot r_1 + f_2 \cdot r_2 + ... + f_m \cdot r_m$ і $h \in I = (f_1, f_2, ..., f_m)$.

Задачу входження вирішують за допомогою спрощення виразу $h(x_1, x_2 \ldots, x_n)$, використовуючи т.зв. *редукцію багаточлена*. Запишемо поліном $h(x_1, x_2 \ldots, x_n)$ у вигляді суми $h=h_C+h_M$, де h_C – старший одночлен (моном), а h_M – сума решти одночленним в h. Припус-

тимо також, що h_C ділиться на старший член f_{iC} одного з багаточленів f_i, тобто $h_C = f_{iC} \cdot Q$ і $h = f_{iC} \cdot Q + h_M$ для деякого одночлена Q. Тоді *операція редукції* задається виразом

$$h_1 = h - f_i \cdot Q = f_{iC} \cdot Q + h_M - f_{iC} \cdot Q - f_{iM} \cdot Q = h_M + (-f_{iM}) \cdot Q, \qquad (7)$$

де f_{iM} – сума решти одночленним в $f_i = f_{iC} + f_{iM}$. При цьому старший член багаточлена h_1 менше старшого члена багаточлена h. Якщо багаточлен h належить ідеалу $I = (f_1, f_2, ..., f_m)$, тоді і редукований багаточлен h_1 також буде належати до цього ідеалу. Дійсно, якщо $h \in (f_1, f_2, ..., f_m)$, тоді $h - h_1 = f_i Q \in (f_1, f_2, ..., f_m)$. Отже, задачу входження тепер можна вирішувати вже не для багаточлена h, а для редукованого багаточлена h_1. Якщо за кінцеве число редукцій (7) багаточлен h зведеться (редукується) до нуля (нуль належить будь-якому ідеалу), тоді $h \in (f_1, f_2, ..., f_m)$.

Базис $f_1, f_2 \ldots, f_m$ ідеалу називається ***базисом Грьобнера*** цього ідеалу, якщо всякий багаточлен $h \in I$ редукується до нуля за допомогою $f_1, f_2 \ldots, f_m$. Інакше: набір багаточленів $f_1, f_2 \ldots, f_m$ є базисом Грьобнера в ідеалі $I = (f_1, f_2, ..., f_m)$, якщо для будь-якого $h \in I$ одночлен h_C ділиться на один з одночленів $f_{1C}, f_{2C}, ..., f_{mC}$ [15].

Для операції редукції багаточленів використовується поняття старшого одночлена (монома). Іншими словами, передбачається, що на множині всіх одночленів кільця $K[x_1, x_2 \ldots, x_n]$ заданий *лінійний порядок* (мономіальне упорядкування $\prec$), який задовольняє наступним властивостям [16]:

– з $x^u \prec x^v$ випливає, що $x^w \cdot x^u \prec x^w \cdot x^v$ для будь-яких одночленів x^u, x^v, x^w (одночлени визначені як в (2), тобто $x, u, v, w \in V_2^n, x = (x_1, ..., x_n), u = (u_1, ..., u_n), v = (v_1, ..., v_n), w = (w_1, ..., w_n)$;

– $1 \preceq x^v$ для будь-якого одночлена x^v.

Як приклади мономіального упорядкування наведемо:

– *словниковий* (*лексикографічний*) *порядок* (*lex*): $x^u \prec_{\text{lex}} x^v$, якщо існує таке i, що $u_i < v_i$ і $u_j = v_j$ для $j < i$ (спочатку упорядковуємо змінні в одночленах в необхідному алфавітному порядку, а потім дивимося до першої відмінності в одночленах);

– *ступінево-словниковий порядок* (*deglex*): $x^u \prec_{\text{deglex}} x^v$, якщо $w_h(u) < w_h(v)$ або $w_h(u) = w_h(v)$, але при цьому $x^u \prec x^v$ в словниковому порядку (упорядковуємо за сумою ступенів, в разі рівності сум порівнюємо за словниковим порядком);

– *ступінево-зворотній словниковий порядок* (*degrevlex*): $x^u \prec_{\text{degrevlex}} x^v$, якщо $w_h(u) < w_h(v)$ або $w_h(u) = w_h(v)$, але при цьому $x^u \succ_{\text{lex}} x^v$ в словниковому порядку (упорядковуємо за сумою ступенів, в разі рівності сум порівнюємо за зворотним словниковим порядком).

Рішення задачі входження, тобто визначення приналежності багаточлена h ідеалу $I = (f_1, f_2, ..., f_m)$, полягає в побудові всіх можливих редукцій h за допомогою елементів базису Грьобнера ідеалу I. Багаточлен h належить ідеалу $I = (f_1, f_2, ..., f_m)$ тоді і тільки тоді, коли в результаті редукції одержано нуль [15].

Для кожного ідеалу $I \lhd K[x_1, x_2, ..., x_n]$ існує базис Грьобнера, а сама побудова базису Грьобнера ґрунтується на вирішенні *зачеплень* [15]. Багаточлени f_i і f_j мають зачеплення, якщо їх старші члени діляться одночасно на деякий одночлен ω, відмінний від константи. Нехай $f_{iC} = \omega \cdot q_1$, $f_{jC} = \omega \cdot q_2$, де ω – найбільший спільний дільник старших одночленів f_{iC} і f_{jC}. Розглянемо багаточлен $F_{i,j} = f_i \cdot q_2 - f_j \cdot q_1 \in I$ і редукуємо його за допомогою базису

$f_1, f_2, ..., f_m$ до тих пір, поки це можливо. Якщо отриманий в результаті багаточлен $F'_{i,j} \equiv 0$, тоді кажуть, що *зачеплення вирішується*. Інакше, додамо до базису $f_1, f_2, ..., f_m$ ідеалу I отриманий багаточлен $f_{m+1} = F'_{i,j}$, після чого процедуру пошуку і редукування зачеплення продовжимо. Після редукування кінцевого числа зачеплення отримаємо набір $f_1, f_2, ..., f_m, f_{m+1}, ..., f_M$, в якому кожне зачеплення вирішується.

Відповідно до *діамантової лемми* базис $f_1, f_2 \ldots, f_m$ ідеалу $I \lhd K[x_1, x_2, ..., x_n]$ є базисом Грьобнера тільки тоді, коли в ньому немає *нерозв'язних зачеплень* [15].

Розв'язання зачеплень дозволяє визначити ефективний алгоритм побудови базису Грьобнера ідеалу $I = (f_1, f_2, ..., f_m)$ (*алгоритм Бухбергера*).

Крок 1. Перевіряємо наявність зачеплень в наборі $f_1, f_2 \ldots, f_m$. Якщо зачеплень немає, тоді набір $f_1, f_2 \ldots, f_m$ є базисом Грьобнера ідеалу $I = (f_1, f_2, ..., f_m)$. Якщо зачеплення є, тоді переходимо до кроку 2.

Крок 2. По знайденому зачепленню багаточленів f_i і f_j складаємо багаточлен $F_{i,j} = f_i \cdot q_2 - f_j \cdot q_1$ і редукуємо його за допомогою набору $f_1, f_2 \ldots, f_m$ поки це можливо. Якщо багаточлен $F_{i,j}$ редукувався до ненульового багаточлена переходимо до кроку 3, інакше – до кроку 4.

Крок 3. Додаємо многочлен f_{m+1} до набору $f_1, f_2 \ldots, f_m$ і переходимо до кроку 4.

Крок 4. Шукаємо раніше нерозглянуте зачеплення і переходимо до кроку 2. Якщо всі зачеплення розглянуті, тоді виводимо отриманий набір $f_1, f_2, ..., f_m, f_{m+1}, ..., f_M$, в якому всі зачеплення можна розв'язати. Це і є базис Грьобнера ідеалу $I = (f_1, f_2, ..., f_m)$.

На сьогоднішній день відомі й інші алгоритми побудови базису Грьобнера, наприклад алгоритми F4, F5 [17, 18].

Базис Грьобнера можна спростити наступними способами [15].

1. *Мінімізація базису Грьобнера*. Якщо f_i і f_j два елементи базису Грьобнера, причому їх старші члени f_{iC} і f_{jC} діляться один на одного, наприклад $f_{jC} \mid f_{iC}$, тоді багаточлен f_i можна видалити з набору $f_1, f_2 \ldots, f_m$. Базис Грьобнера називають *мінімальним*, якщо f_{iC} не ділиться на f_{jC} для всіх $i \neq j$.

2. *Редукування базису Грьобнера*. Якщо деякий член q багаточлена f_i ділиться на старший член багаточлена f_j, тоді редукуємо q за допомогою f_j і результат редукції запишемо замість члена q в багаточлен f_i. При цьому базис Грьобнера залишиться базисом Грьобнера, число елементів базису не зміниться, проте ступеня багаточленів $f_1, f_2 \ldots, f_m$ понижуються. Базис Грьобнера називають *редукованим*, якщо жоден член багаточлена f_i не ділиться на старший член многочлена f_j для всіх $i \neq j$.

Мінімальний редукований базис Грьобнера ідеалу $I \lhd K[x_1, x_2, ..., x_n]$ визначено однозначно (з одиничними коефіцієнтами при старших ступенях елементів базису), тобто не залежить від вибору вихідного базису ідеалу $I = (f_1, f_2, ..., f_m)$ і від послідовності операцій, що проводяться (але залежить від упорядкування змінних $x_1, x_2 \ldots, x_n$) [15].

Поняття мінімального редукованого базису Грьобнера використано в роботі Жан-Шарля Фожера (Jean-Charles Faugère) [13] для визначення алгебраїчної імунності S-блоків (нелінійних вузлів ускладнення) блокових симетричних шифрів.

Розглянемо нелінійний вузол (S-блок) блочного симетричного шифру (див. рис. 2), який реалізує булеве відображення $S : GF(2)^n \to GF(2)^m$ [1-9].

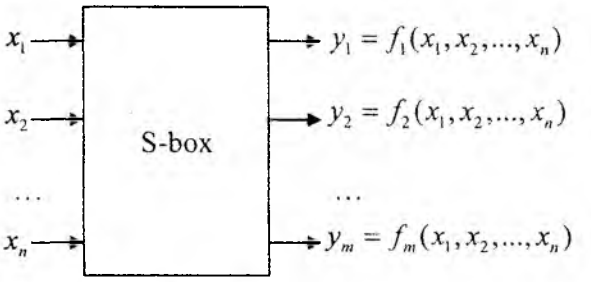


Рис. 2

S-блок задається системою алгебраїчних рівнянь над двійковим полем:

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = y_1, \\ f_2(x_1, x_2, ..., x_n) = y_2, \\ ... \\ f_m(x_1, x_2, ..., x_n) = y_m, \end{cases} \tag{8}$$

тобто сукупністю булевих багаточленів

$$\begin{array}{l} y_1 - f_1(x_1, x_2, ..., x_n), \\ y_2 - f_2(x_1, x_2, ..., x_n), \\ ....., \\ y_m - f_m(x_1, x_2, ..., x_n) \end{array}, \tag{9}$$

в кільці $K[x_1, x_2, ..., x_n, y_1, y_2, ..., y_m]$ від змінних $x_1, x_2, ..., x_n, y_1, y_2, ..., y_m$ з коефіцієнтами над полем $K = GF(2)$.

З системою рівнянь (8), що алгебраїчно задають структуру S-блоку, зв'яжемо ідеал I, породжений багаточленами (9):

$$I(S) = (y_1 - f_1(x_1, x_2, ..., x_n), y_2 - f_1(x_1, x_2, ..., x_n), ..., y_m - f_m(x_1, x_2, ..., x_n)) =$$
$$= \{(y_1 - f_1) \cdot r_1 + (y_2 - f_1) \cdot r_2 + ... + (y_m - f_m) \cdot r_m; r_1, r_2, ..., r_m \in GF(2)[x_1, x_2, ..., x_n, y_1, y_2, ..., y_m]\}.$$

Алгебраїчна імунність нелінійного вузла блокового симетричного шифру визначається як мінімальна ступінь багаточлена P з ідеалу $I(S)$ [13]:

$$AI(S) = \min\{\deg(P), P \in I(S) \lhd GF(2)[x_1, x_2, ..., x_n, y_1, y_2, ..., y_m]\}, \tag{10}$$

причому мінімальний редукований базис Грьобнера ідеалу $I(S)$ при ступенево-зворотньому словниковому впорядкуванні (degrevlex) містить лінійний базис поліномів P з $I(S)$, таких, що $AI(S) = \deg(P)$. Іншими словами, для обчислення алгебраїчної імунності досить побудувати мінімальний редукований базис Грьобнера ідеалу $I(S)$, заданого рівняннями (9) і знайти многочлен мінімального ступеня серед елементів цього базису. Значення мінімального ступеня і є значенням алгебраїчної імунності вузла замін блокового симетричного шифру.

Зв'язок алгебраїчної імунності S-блоку (10) і булевої функції (1) показаний в [19, с. 337]. Розглянемо булеву функцію $f_S(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m) : GF(2)^{2n} \to GF(2)$, значення якої визначимо наступним чином:

$$f_S(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m) = \begin{cases} 1, \forall i, j : f_i(x_1, x_2, ..., x_n) = y_j, \\ 0, \forall i, j : f_i(x_1, x_2, ..., x_n) \neq y_j. \end{cases}$$

Множина рішень рівняння

$$f_S(x_1, x_2, ..., x_n, y_1, y_2, ..., y_m) - 1 = 0.$$

Співпадає з множиною рішень системи (8). Отже, маємо різні базиси $(f_S - 1)$ та $(y_1 - f_1, y_2 - f_1, ..., y_m - f_m)$ одного ідеалу еквівалентних систем. Тобто

$$I(f_S - 1) = I(y_1 - f_1, y_2 - f_1, ..., y_m - f_m).$$

Ідеал простору анігіляторів $Ann(fs)$ в кільці $GF(2)[x_1, x_2, ..., x_n, y_1, y_2, ..., y_m]$ збігається з ідеалом $I(f_S - 1)$, отже, алгебраїчна імунність (10) булевого відображення $S: GF(2)^n \to GF(2)^m$ збігається з мінімальним ступенем ненульових поліномів, що належать анігілятору функції fs:

$$AI(S) = \min\{Deg(g) \mid g \in Ann(f_S)\}.$$

Таким чином, будь-який S-блок можна однозначно описати булевою функцією [19], алгебраїчну імунність цієї функції можна обчислити, наприклад, за допомогою алгоритму з п. 2.

4. Значення алгебраїчної імунності нелінійних вузлів сучасних шифрів

В даній роботі проведено порівняльні дослідження алгебраїчної імунності нелінійних вузлів сучасних симетричних шифрів. Як об'єкти дослідження обрані широко відомі і стандартизовані на національному та/або міжнародному рівні блокові симетричні криптоперетворення:

– криптоалгоритм AES, стандартизований в США як федеральний стандарт обробки даних FIPS-197 [20], а також на міжнародному рівні в ISO/IEC 18033-3 [21];
– криптоалгоритм Camellia, стандартизований в ISO/IEC 18033-3 [21];
– криптоалгоритм CAST, стандартизований в ISO/IEC 18033-3 [21];
– криптоалгоритм SEED, стандартизований в ISO/IEC 18033-3 [21];
– криптоалгоритм «Калина», національний стандарт України ДСТУ 7624: 2014 [22];
– криптоалгоритм «Кузнечик», стандартизований в Росії як ГОСТ 34.12-2015 [23];
– алгоритм «BelT» симетричного шифрування і контролю цілісності Республіки Білорусь, стандартизований в СТБ 34.101.31-2011 [24];
– криптографічна хеш-функція Whirlpool, заснована на використанні блокових симетричних криптоперетворень, стандартизована в ISO/IEC 10118-3: 2004 [25].

Для обчислення алгебраїчного імунітету використовувався вираз (10). Для безпосередніх обчислень використано пакет прикладного програмного забезпечення Magma [26], який реалізує широкий спектр функцій, пов'язаних з алгеброю, теорією груп, кілець і полів, теорією чисел і багатьма іншими розділами математики.

Досліджувані вузли замін, окрім S-блоку хеш-функції Whirlpool, були детально розглянуті в нашій роботі [9], в таблиці наведено деякі результати досліджень.

Криптоалгоритм	B	N	A	AD	PC	CI	AI
AES	+	112	32	7	0	0	2
SEED	-	110	40	7	0	0	2
CAST-128	-	120	0	4	8	0	2
Camellia	+	112	32	7	0	0	2
«Калина»	+	104	72	7	0	0	3
«Кузнечик»	+	102	72	7	0	0	3
«BelT»	+	104	72	7	0	0	3
Whirlpool	+	95	80	7	0	0	3

У таблиці використані наступні позначення [9]:

- B – збалансованість;
- N – нелінійність;
- A – автокореляция;
- AD – алгебраїчна ступінь;
- PC – критерій поширення;
- CI – кореляційний імунітет.

В останній колонці «AI» таблиці наведено значення алгебраїчної імунності нелінійних вузлів заміни сучасних шифрів. Ці дані отримано за формулою (10) за допомогою побудови базисів Грьобнера ідеалів $I(S)$, заданих сукупностями багаточленів (9) з рівнянь (8) відповідних S-блоків.

Отримані результати дозволяють судити про недостатню алгебраїчну імунність нелінійних вузлів блокових шифрів, які були розроблені в кінці 90-х – початку 2000-х років. Розглянуті алгоритми (AES, SEED, CAST-128, Camellia) представлені в сучасному міжнародному стандарті ISO/IEC 18033-3, мають порівняно низьку алгебраїчну імунність і потенційно можуть розглядатися в якості реальних кандидатів на побудову ефективних алгебраїчних атак.

Блокові симетричні криптоалгоритми «Калина», «Кузнечик», «BelT», а також криптографічна функція хешування Whirlpool були розроблені з урахуванням можливого застосування алгебраїчних атак. Нелінійні вузли замін цих алгоритмів мають високу алгебраїчну імунність і, вочевидь, залишаться стійкими до нових методів алгебраїчного криптоаналізу.

5. Висновки

Методи алгебраїчного криптоаналізу, з моменту перших публікацій [27, 28], перетворилися з абстрактних і маловживаних математичних ідей в розвинений і широко обговорюваний в науковому співтоваристві розділ сучасної криптології. На сьогоднішній день в цій галузі знань проводиться величезна кількість дослідницьких проектів і, очевидно, слід очікувати в найближчі роки появи ефективних обчислювальних алгоритмів алгебраїчного криптоаналізу сучасних симетричних шифрів.

У даній роботі були розглянуті окремі аспекти алгебраїчного криптоаналізу, зокрема, досліджені методи обчислення алгебраїчної імунності нелінійних вузлів симетричних шифрів. Це поняття, вперше введене для потокових криптоалгоритмів в роботах [10, 11], було узагальнено в [13] на випадок булевих відображень, тобто для нелінійних вузлів з довільною розмірністю входів-виходів. Алгебраїчна імунність в деякому розумінні характеризує складність вирішення системи рівнянь, що описують нелінійний вузол і дозволяє, таким чином, отримати уявлення про стійкість симетричного шифру до алгебраїчного криптоаналізу. Зокрема, в роботі [10] запропонований алгоритм алгебраїчного криптоаналізу потокових шифрів, побудованих за схемою фільтр-генератора, складність реалізації цього алгоритму є функцією від значення алгебраїчної імунності нелінійного вузла ускладнення.

Обчислення алгебраїчної імунності в загальному випадку пов'язане з побудовою базису Грьобнера ідеалу кільця многочленів, заданого многочленами з рівнянь вузла ускладнення. Ця задача вирішується обчислювально ефективними алгоритмами Бухбергера, F4, F5 та ін. [15 – 18]. Крім того, розглянуті математичні методи можуть також використовуватися і для пошуку ефективних алгебраїчних атак [29], що підтверджує перспективність і актуальність проведених робіт в даній області.

У даній роботі наведені значення алгебраїчного імунітету для вузлів замін деяких сучасних шифрів. Зокрема, встановлено, що криптоалгоритми, розроблені на рубежі 90-х – початку 2000-х років, не володіють граничними значеннями алгебраїчної імунності, тобто потенційно можуть розглядатися як кандидати на побудову ефективних алгебраїчних атак. Блокові шифри останнього покоління («Калина», «Кузнечик», «BelT») розроблялися з урахуванням можливого застосування алгебраїчного криптоаналізу і володіють граничними значеннями алгебраїчного імунітету.

Перспективним напрямком є подальші дослідження методів алгебраїчного криптоаналізу, зокрема, застосування технологій квантових обчислень для вирішення систем алгебраїчних рівнянь, що описують симетричний шифр. На думку авторів даної роботи, саме в цьому напрямку досліджень очікуються найбільш значущі і цікаві наукові результати.

Список літератури: 1. *Alfred, J. Menezes, Paul C. van Oorschot, Scott, A.* Vanstone. Handbook of Applied Cryptography – CRC Press, 1997. – 794 p. 2. *Горбенко, І.Д., Горбенко, Ю.І.* Прикладна криптологія. Теорія. Практика. Застосування: Підручник для вищих навч. закладів. – Харків : Форт, 2013. – 880 с. 3. *Bart Preneel.* Analysis and Design of Cryptographic Hash Functions. [Электронный ресурс] – Режим доступа:homes.esat.kuleuven.be/ ~preneel/phd_preneel_feb1993.pdf 4. *Carlet, C.* Vectorial Boolean functions for // Cambridge Univ. Press, Cambridge. – 95 p. [Электронный ресурс] – Режим доступа: www.math.univ-paris13.fr/~carlet/chap-vectorial-fcts-corr.pdf 5. *Carlet, C.* Boolean functions for cryptography and error correcting codes // Cambridge Univ. Press, Cambridge. – 2007. – 148 p. [Электронный ресурс] – Режим доступа: www1.spms.ntu.edu.sg/~kkhoongm/chap-fcts-Bool.pdf 6. *Zhuo Zepeng, Zhang Weiguo* On correlation properties of Boolean functions // Chinese Journal of Electronics. Jan, Vol.20, 2011, №1, 143-146 pp. 7. *O'Connor, L.* An analysis of a class of algorithms for S-box construction // J. Cryptology. -1994. – p. 133-151. 8. *Clark J.A., Jacob J.L., Stepney S.* The Design of S-Boxes by Simulated Annealing // New Generation Computing. – 2005. – 23(3). – p.219–231. 9. *Кузнецов, А.А., Белозерцев, И.Н., Андрушкевич, А.В.* Анализ и сравнительные исследования нелинейных узлов замены современных блочных симметричных шифров // Прикладная радиоэлектроника. – Харьков : ХНУРЭ, 2015. – Т. 14. №4. – С.343 – 350. 10. *Courtois, N., Meier, W.* Algebraic Attacks on Stream Ciphers with Linear Feedback, Eurocrypt 2003, LNCS 2656, Springer, 2003. – pp. 345-359. 11. *Meier, W., Pasalic, E., Carle,t C.* Algebraic Attacks and Decomposition of Boolean Functions, Eurocrypt 2004, LNCS 3027, Springer, 2004. – pp. 474-491. 12. *Nicolas Courtois; Josef Pieprzyk* (2002). Cryptanalysis of Block Ciphers with Overdefined Systems of Equations // LNCS. 2501: 267–287. 13. *Gw'enol'e Ars, Jean-Charles Faug`ere.* Algebraic Immunities of functions over finite fields. [Research Report] RR-5532, INRIA. 2005, pp.17. 14. *Баев, В. В.* Эффективные алгоритмы получения оценок алгебраической иммунности булевых функцій : дис. ... канд. физ.-мат. наук : 01.01.09 / Баев Владимир Валерьевич; [Место защиты: Моск. гос. ун-т им. М.В. Ломоносова. Фак. вычислит. математики и кибернетики]. – Москва, 2008. – 101 с. 15. *Аржанцев, И.В.* Базисы Грёбнера и системы алгебраических уравнений. Летняя школа. Современная математика. Дубна, июль 2002. – Москва : МЦНМО, 2003. – 68 с. 16. *Злобин, А.И., Соколова, О.В.* Компьютерная алгебра в системе Sage. Учебное пособие. – Москва : МГТУ им. Баумана, 2011. – 55 с. 17. *Faugère, J.-C.* (June 1999). A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra. Elsevier Science. 139 (1): 61–88. 18. *Faugère, J.-C.* (July 2002). A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). Proceedings of the 2002 international symposium on Symbolic and algebraic computation (ISSAC). ACM Press: 75–83. 19. *Massimiliano Sala, Teo Mora, Ludovic Perret, Shojiro Sakata, Carlo Traverso Gröbner Bases,* Coding, and Cryptography. Springer-Verlag Berlin Heidelberg. – 426 p. 20. *FIPS 197.* National Institute of Standards and Technology. [Electronic resource]: Advanced Encryption Standard. – 2001. – Available at: http://www.nist.gov/aes. 21. *ISO/IEC 18033-3.* Information technology – Security techniques – Encryption algorithms, Part 3: Block ciphers, 80 p. 22. *ДСТУ 7624:2014.* Інформаційні технології. Криптографічний захист інфомації. Алгоритм симетричного блокового перетворення. – Київ : Мінекономрозвитку України, 2015. – 238 с. 23. *ГОСТ Р 34.12-2015.* Информационная технология. Криптографическая защита информации. Блочные шифры. – Москва : Стандартинформ, 2015. – 25с. 24. *СТБ 34.101.31-2011.* Информационные технологии и безопасность. Криптографические алгоритмы шифрования и контроля целостности. – Минск : Госстандарт, 2011. – 32 с. 25. *ISO/IEC 10118-3:2004.* Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions, 94 p. 26. *Magma* Computational Algebra System. Available at: http://magma.maths.usyd.edu.au/magma/ 27. *Nicolas Courtois, Alexander Klimov, Jacques Patarin, Adi Shamir.* Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. Proceeding EUROCRYPT'00 Proceedings of the 19th international conference on Theory and application of cryptographic techniques. P. 392-407. 28. *Nicolas Courtois, Josef Pieprzyk.* Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Advances in cryptology – ASIACRYPT 2002. P.267-287. 29. *Andrey Pyshkin.* Algebraic Cryptanalysis of Block Ciphers Using Grobner Bases. Dissertation zur Erlangung des Grades Doktor rerum naturalium. Technischen Universit"at Darmstadt. – Darmstadt, 2008, 118 p.

Харківський національний університет імені В.Н.Каразіна

Надійшла до редколегії 06.04.2017

УДК 004.056.55

А.А. КУЗНЕЦОВ, д-р техн. наук, А.И. ПУШКАРЕВ, А.С. КИЯН

АЛГОРИТМЫ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ НА ОСНОВЕ АЛГЕБРАИЧЕСКОГО КОДИРОВАНИЯ

Введение

Электронная цифровая подпись (ЭЦП) является цифровым эквивалентом подписи (печати, штампа и пр.), наличие которого в сообщении позволяет с высокой точностью определить источник сообщения (документа) и юридически доказать, что, с определенной вероятностью, только он мог создать и подписать этот документ [1 – 3].

Для формирования ЭЦП на сегодняшний день используются криптографические механизмы и протоколы [4 – 10], в которых задача поиска секретного ключа по известному открытому ключу связана с решением известной и чрезвычайно сложной математической задачи (например, факторизации, дискретного логарифмирования, дискретного логарифмирования в группе точек эллиптической кривой) [1 – 3]. Однако квантовые вычисления позволяют существенно ускорить решение многих математических задач [11 – 20], в том числе лежащих в основе современных алгоритмов ЭЦП [20]. Например, алгоритм Шора (Shor) позволяет найти за приемлемое время все простые множители в системе RSA, т.е. найти секретный ключ и/или подделать ЭЦП без знания секретного ключа [16].

Появление квантовых компьютеров анонсировано на ближайшие 10 – 15 лет [21, 22]. По этой причине Национальный Институт Стандартов и Технологий (NIST) США в конце 2016 г. объявил открытый конкурс (http://csrc.nist.gov/groups/ST/post-quantum-crypto/) с целью принятия в течение пяти-семи лет новых постквантовых криптостандартов [22]. В этом смысле разработка, исследование и обоснование рекомендаций по практическому использованию новых криптосистем, устойчивых к квантовому криптоанализу, т.е. на постквантовый период (Post-Quantum Cryptography), имеет особую актуальность и востребованность [20 – 22].

Одно из направлений в развитии постквантовой криптографии основывается на использовании алгебраических блоковых кодов (Code-based Cryptography) [20, 21, 23 – 31]. При этом обеспечивается высокая скорость криптографического преобразования, стойкость к классическому и квантовому криптоанализу, а также возможность дополнительного контроля возникающих ошибок [28 – 30].

В основе построения кодовых криптосистем лежит использование специальных маскирующих матриц, которые выступают в качестве секретного ключа и, по предположению, надежно скрывают алгебраическую структуру кода [26]. Злоумышленник, не зная секретного ключа, не может воспользоваться алгебраическим алгоритмом декодирования (полиномиальной сложности) и вынужден декодировать кодовое слово, решая NP-сложную задачу [32].

В данной статье рассматриваются наиболее известные кодовые криптосистемы Мак-Элиса и Нидеррайтера [23, 24], а также схема CFS (Courtois, Finiasz, Sendrier) [31] для формирования и проверки ЭЦП. Предлагается новая схема ЭЦП с использованием алгебраических кодов, приводятся сравнительные оценки эффективности по различным показателям (стойкость, сложность формирования и проверки ЭЦП, объемы ключевых данных, длина подписи и пр.).

Кодовые криптосистемы Мак-Элиса и Нидеррайтера

В 1978 г. Мак-Элисом (McEliece) была предложена первая криптосистема на алгебраических блоковых кодах [23]. В ее основе лежит маскирование быстрого правила декодирования посредством матричного умножения порождающей матрицы алгебраического блокового кода на случайные невырожденные матрицы (секретный ключ). Полученный результат (открытый ключ) представляет собой порождающую матрицу, имеющую вид случайно выбран-

ных линейно независимых векторов. Злоумышленник, имеющий только открытый ключ, вынужден использовать сложный алгоритм неалгебраического декодирования (NP-полная задача). Уполномоченный пользователь, знающий секретный ключ, снимает действие маскирования и применяет быстрый алгебраический алгоритм декодирования (полиномиально разрешимая задача).

Введем необходимые обозначения и определения.

Зафиксируем конечное поле $GF(q)$. Пусть G – порождающая матрица алгебраического (n,k,d) кода над $GF(q)$ (в оригинальной статье предлагалось использовать двоичные сепарабельные коды Гоппы), X – невырожденная $k\times k$ матрица с элементами из $GF(q)$, P и D – перестановочная и диагональная $n\times n$ матрицы, соответственно (для двоичных кодов используется только матрица P). Матрица

$$G_X = X\cdot G\cdot P\cdot D$$

является открытым ключом, маскирующие матрицы X, P и D являются секретным ключом. Криптограммой является искаженное ошибкой e кодовое слово

$$c_X^* = I\cdot G_X + e, \tag{1}$$

причем вес Хемминга вектора ошибок $w_h(e)$ удовлетворяет ограничению

$$w_h(e)\le t=\left\lfloor\frac{d-1}{2}\right\rfloor.$$

Вектор $c_X = I\cdot G_X$ является кодовым словом замаскированного кода, т.е. c_X принадлежит (n,k,d) коду с порождающей матрицей G_X, I – k-разрядный информационный вектор над $GF(q)$. Не зная матрицы X, P и D злоумышленник не может восстановить матрицу G и воспользоваться алгоритмом декодирования полиномиальной сложности. Из этих соображений величину $w_h(e)$ следует максимизировать. Например, при $w_h(e)=t$ обеспечивается наивысший уровень стойкости кодовой криптосистемы для заданных параметров n,k,q. Уполномоченный пользователь, получив вектор c_X^*, строит вектор $\bar{c}^* = c_X^*\cdot D^{-1}\cdot P^{-1}$. Используя алгоритм полиномиальной сложности, он декодирует вектор $\bar{c}^* = I'\cdot G + e'$, т.е. находит I'. Затем он вычисляет k-разрядный информационный вектор $I = I'X^{-1}$.

Другим важным примером кодовых криптосистем является схема Нидеррайтера (Niederreiter) [24]. Открытым ключом в этой криптосистеме есть матрица

$$H_X = X\cdot H\cdot P\cdot D,$$

где H – проверочная матрица алгебраического (n,k,d) кода над $GF(q)$ (в оригинальной статье предлагалось использовать обобщенные коды Рида – Соломона), X – невырожденная $(n-k)\times(n-k)$ матрица с элементами из $GF(q)$, P и D – перестановочная и диагональная $n\times n$ матрицы (для двоичных кодов используется только матрица P). Матрицы X, P и D (как и для криптосистемы Мак-Элиса) являются секретным ключом, который маскирует используемый алгебраический блоковый код под случайный код (код общего положения), т.е. открытый ключ H_X представляется злоумышленнику как случайно сформированная проверочная матрица некоторого линейного кода, для которого неизвестен алгоритм быстрого декодирования. Напротив, уполномоченный пользователь, знающий секретный ключ (матрицы X, P и D), может снять действие маскирующих матриц и воспользоваться быстрым алгоритмом декодирования алгебраического кода с проверочной матрицей H.

Криптограмма s_X представляет собой вектор длины $(n-k)$ и вычисляется по правилу

$$s_X = e\cdot H_X^T, \tag{2}$$

где вектор e – вектор длины n и веса $w_h(e) \le t$, который несет конфиденциальную информацию (информационное сообщение, подлежащее зашифрованию). Наибольшая стойкость обеспечивается при $w_h(e) = t$.

Для расшифрования криптограммы s_X выполняются следующие действия [19]. Уполномоченный пользователь (имеющий секретный ключ) находит одно из q^k решений выражения $s_X = c_X^* \cdot H_X^T$. Найденное решение – суть кодовое слово с ошибками $c_X^* = I \cdot G_X + e$.

Далее, как и в схеме Мак-Элиса, уполномоченный пользователь строит вектор $\bar{c}^* = c_X^* \cdot D^{-1} \cdot P^{-1}$ и декодирует полученное слово. Однако вместо восстановления информационного слова I', он вычисляет кодовое слово $c' = I' \cdot G$, а затем и вектор ошибок $e' = \bar{c}^* - c'$. На последнем шаге производится вычисление вектора $e = e' \cdot P \cdot D$, который несет конфиденциальную информацию.

Расшифрование s_X может быть выполнено и по следующей схеме [20]. Сперва заметим, что выражение (2) можно переписать в виде

$$s_X{}^T = H_X \cdot e^T . \qquad (3)$$

В этом случае, уполномоченный пользователь (имеющий матрицы X, P и D) для расшифрования криптограммы вычисляет вектор

$$s_X^{*T} = X^{-1} \cdot s_X{}^T = X^{-1} \cdot H_X \cdot e^T = H \cdot P \cdot D \cdot e^T = H \cdot \bar{e}^T ,$$

значение которого зависит от вектора

$$\bar{e}^T = P \cdot D \cdot e^T .$$

Вектор $s_X^{*T} = H \cdot \bar{e}^T$ представляет собой синдром, вычисленный по проверочной матрице H алгебраического (n, k, d) кода, т.е. алгоритм быстрого (алгебраического) декодирования позволяет найти вектор $\bar{e}^T$, после чего уполномоченный пользователь снимает действие матриц маскирования P, D и находит вектор

$$e^T = D^{-1} \cdot P^{-1} \cdot \bar{e}^T = D^{-1} \cdot P^{-1} \cdot P \cdot D \cdot e^T .$$

В работах [33 – 38] исследованы свойства кодовых криптосистем Мак-Элиса и Нидеррайтера, приведены оценки стойкости, в том числе к квантовому криптоанализу, показано, что относительная скорость передачи данных для наиболее важных с прикладной точки зрения случаев существенно ниже единицы. Предложена новая криптосистема, которая позволяет существенно (на 30 – 40 %) повысить относительную скорость передачи информации.

ЭЦП на основе алгебраических кодов

Первый известный алгоритм формирования и проверки ЭЦП с использованием алгебраических кодов основан на криптосистеме Нидеррайтера и был представлен Courtois, Finiasz и Sendrier в работе [29]. Оценка стойкости этой схемы (названной по инициалам ее изобретателей – CFS) против подделки подписи может быть сведена к оценке сложности решения задачи синдромного декодирования. Знание секретного ключа позволяет декодеру решить эту задачу для некоторой доли случайных кодовых слов.

В **схеме CFS** для формирования ЭЦП реализуется многократное хеширование информационного сообщения, рандомизированного счетчиком битовой длины r, до тех пор, пока не будет получен правильно выделенный (допускающий декодирование) синдром. Уполномоченный пользователь использует секретный ключ для определения соответствующего вектора ошибок. Вместе с текущим значением счетчика этот вектор ошибок используется в качестве подписи.

Реализация схемы CFS для формирования и проверки ЭЦП осуществляется в соответствии со следующими алгоритмами [31].

1. Генерация общесистемных параметров: выбираются положительные целые числа m, t.

2. Генерация ключа: формируются пары ключей как в криптосистеме Нидеррайтера на основе алгебраического кода из класса ($n = 2^m$, $k = n - mt$, $2t + 1$) двоичных кодов Гоппы. Для этого формируются:

– матрица H: $(n-k)\times n$ – проверочная матрица алгебраического кода с исправляющей способностью t ошибок,

– матрица X: $(n-k)\times(n-k)$ – случайная обратимая матрица,

– матрица P: $n\times n$ – случайная матрица перестановок;

Открытым ключом является матрица $H_X = X \cdot H \cdot P$ и число t (исправляющая способность кода).

Секретным ключом являются матрицы H, X и P, а также связанный с матрицей H быстрый (полиномиальной сложности) алгоритм декодирования алгебраического кода.

Алгоритм декодирования позволяет по введенной синдромной последовательности $s = (s_0, s_1, ..., s_{n-k-1})$ в случае успеха декодирования найти вектор ошибок $e = (e_0, e_1, ..., e_{n-1})$ и кодовое слово $c = (c_0, c_1, ..., c_{n-1})$. В противном случае (если декодирование не удалось) алгоритм выдает отказ в обработке синдрома $s = (s_0, s_1, ..., s_{n-k-1})$, т.е. по такой последовательности алгоритм не может найти вектор ошибок $e = (e_0, e_1, ..., e_{n-1})$ и кодовое слово $c = (c_0, c_1, ..., c_{n-1})$.

3. Формирование подписи.

Вход:

1. h – функция хеширования, которая применяется к входным данным x (аргументу функции) произвольной длины. Результатом хеширования является хеш-код $h(x)$ длины $n-k$ бит;

2. Быстрый (полиномиальной сложности) алгоритм декодирования алгебраического кода, который применяется к синдромной последовательности $s = (s_0, s_1, ..., s_{n-k-1})$. Предполагается, что в результате выполнения алгоритма декодирования возможны две ситуации:

– если декодирование успешно – выводится найденный вектор ошибок $e = (e_0, e_1, ..., e_{n-1})$, который соответствует вектору $s = (s_0, s_1, ..., s_{n-k-1})$;

– если декодирование не успешно – выдается сообщение о невозможности найти вектор ошибок $e = (e_0, e_1, ..., e_{n-1})$ для введенного вектора $s = (s_0, s_1, ..., s_{n-k-1})$;

3. Открытый текст M, для которого необходимо сформировать ЭЦП по схеме CFS.

Выход:

ЭЦП по схеме CFS Y для открытого текста M.

Алгоритм формирования ЭЦП по схеме CFS [20, 31]

Шаг 1. Хеширование открытого текста M, т.е. вычисление хеш-кода $h(M)$. Присваивание переменной i значения $i = 1$;

Шаг 2. Вычисление хеш-кода $h(h(M)\|i)$, где $h(M)\|i$ – конкатенация (объединение) значений $h(M)$ и i, представленных в виде битовых последовательностей;

Шаг 3. Значение $h(h(M)\|i)$ интерпретируется как синдромная последовательность $s_X = (s_0, s_1, ..., s_{n-k-1})$, вычисленная для некоторого (произвольного) кодового слова и вектора ошибок $e = (e_0, e_1, ..., e_{n-1})$, т.е. предполагается выполнение равенства (3) для соответствующего открытого ключа $H_X = X \cdot H \cdot P$;

Шаг 4. Вычисление значение вектора

$$s_X^{*T} = X^{-1} \cdot s_X{}^T ,$$

который (как предполагается) представляет собой синдром, вычисленный по проверочной матрице H алгебраического (n, k, d) кода, т.е. предполагается, что

$$s_X^{*T} = X^{-1} \cdot s_X{}^T = X^{-1} \cdot H_X \cdot e^T = H \cdot P \cdot e^T = H \cdot \overline{e}^T$$

и алгоритм быстрого декодирования позволит найти вектор $\overline{e}^T = P \cdot e^T$;

Шаг 5. Для синдромной последовательности s_X^* реализуется выполнение быстрого алгоритма декодирования:

– если декодирование успешно – выводится найденный вектор ошибок $\overline{e}^T = P \cdot e^T$, который соответствует вектору s_X^*;

– если декодирование не успешно – выдается сообщение о невозможности найти вектор ошибок $\overline{e}^T = P \cdot e^T$ для введенного вектора s_X^*. В этом случае переменной i присваивается значение $i = i + 1$ и осуществляется переход на Шаг 2;

Шаг 6. Вычисление вектора

$$e^T = P^{-1} \cdot \overline{e}^T = P^{-1} \cdot P \cdot e^T;$$

Шаг 7. Формирование ЭЦП по схеме CFS $Y = (e, i)$ для открытого текста M.

Таким образом, в результате выполнения рассмотренного алгоритма формирования ЭЦП по схеме CFS вычисляется такое наименьшее положительное целое число i, для которого значение $h(h(M)\|i)$, интерпретируемое как синдромная последовательность $s_X = (s_0, s_1, ..., s_{n-k-1})$, соответствует вектору ошибок $e = (e_0, e_1, ..., e_{n-1})$, т.е. формально запишем:

$$Y = (e, i) : H_X \cdot e^T = h(h(M)\|i)^T. \quad (4)$$

Задача вычисления вектора $e = (e_0, e_1, ..., e_{n-1})$ по известному вектору $h(h(M)\|i)$ сопряжена с решением задачи декодирования (n, k, d) кода:

– для уполномоченного пользователя (знающего секретный ключ) это вычислительно простая задача (полиномиальной сложности);

– для злоумышленника (знающего только открытый ключ) это вычислительно сложная задача декодирования случайного кода (относящаяся к классу сложности NP-полных задач).

Для верификации (проверки правильности ЭЦП $Y = (e, i)$ сообщения M) необходимо убедиться в том, является ли результат хеширования $h(h(M)\|i)$ синдромной последовательностью, вычисленной по вектору $e = (e_0, e_1, ..., e_{n-1})$ (который интерпретируется как вектор ошибок).

4. Верификация

Ввод:

1. Открытый ключ (матрица $H_X = X \cdot H \cdot P$ и число *t*);
2. Функция хеширования *h*;
3. ЭЦП $Y = (e, i)$;
4. Открытый текст M.

Выход:

решение о *правильности* или *неправильности* ЭЦП.

Алгоритм верификации (алгоритм проверки ЭЦП по схеме CFS) [20, 31]

Шаг 1. Вычисление вектора

$$(s'_X)^T = H_X \cdot e^T;$$

Шаг 2. Вычисление вектора

$$(s''_X)^T = h(h(M)\|i);$$

Шаг 3. Принятие решение о *правильности* или *неправильности* ЭЦП:

– если $s'_X = s''_X$, тогда принимается решение о *правильности* ЭЦП;

– если $s'_X \neq s''_X$, тогда принимается решение о *неправильности* ЭЦП.

В работах [34, 39 – 41] приведены оценки конструктивных характеристик схемы CFS, исследована эффективность криптосистемы по показателям стойкости, сложности формирования и проверки ЭЦП, объемам ключевых данных, длине подписи и пр. Выявлен существенный недостаток схемы CFS, не отмеченный ранее в других работах, который состоит в возможности быстрой подделки подписи $Y = (e, i)$ используя кодовые слова применяемого (n, k, d) кода. Обоснованы рекомендации для защиты от такой подделки. В частности, шаг 3 необходимо переписать в виде:

Шаг 3. Принятие решение о *правильности* или *неправильности* ЭЦП:

- если $s'_X = s''_X$ и $w(e) \le t$ тогда принимается решение о *правильности* ЭЦП;

- если $s'_X \neq s''_X$ и (или) $w(e) > t$ тогда принимается решение о *неправильности* ЭЦП.

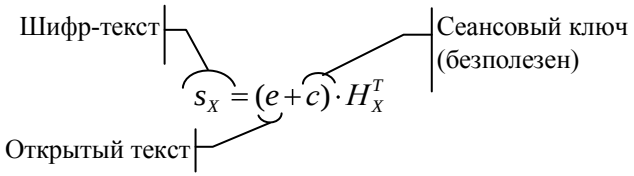
Дополнительная проверка веса Хемминга вектора e позволяет защититься от подделки подписи на основе добавления произвольного кодового слова.

Рассмотренная схема ЭЦП CFS основана на использовании кодовой криптосистемы Нидеррайтера. С момента публикации авторской статьи [31] (т.е. уже более 15 лет) предполагается, что это единственный возможный вариант ЭЦП на алгебраических кодах [20]. В данной работе предлагается новая схема ЭЦП, которая построена на использовании кодовой криптосистемы Мак-Элиса. По своим основным параметрам (стойкости, длине ключа и длине подписи) она сопоставима со схемой ЭЦП CFS.

Предлагаемая схема формирования и проверки ЭЦП. Для наглядной демонстрации принципов построения схемы CFS и предлагаемой схемы ЭЦП на рис. 1 приведены основные аналитические соотношения, связывающие открытый и шифр-текст в кодовых криптосистемах Мак-Элиса и Нидеррайтера, а также их интерпретации, используемые для формирования и проверки подписи.

В криптосистеме Нидеррайтера [24] информационное сообщение в виде равновесной последовательности e умножается на открытый ключ – матрицу H_X, что в теории кодирования эквивалентно нахождению синдрома s_X, который однозначно определяется вектором e. Уполномоченный пользователь, знающий секретные матрицы X и P в $H_X = X \cdot H \cdot P$, сможет применить быстрый (полиномиальной сложности) алгоритм декодирования для нахождения вектора e. Без знания матриц X и P неуполномоченный пользователь вынужден использовать сложный алгоритм декодирования случайного кода, для которого достаточно использовать матрицу H_X. Таким образом, в схеме Нидеррайтера для реализации асимметричного шифрования используется т.н. «односторонняя функция», кода вычислить s_X по известным H_X и e вычислительно легко (полиномиальная сложность), а найти e по известным H_X и s_X чрезвычайно сложно (NP-полная задача).

Криптосистема Нидеррайтера:

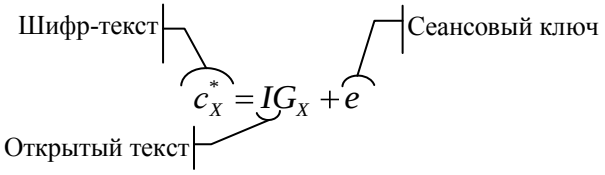


Односторонняя функция:

– для вычисления s_X (синдромной последовательности) по известному e требуется алгоритм полиномиальной сложности;

– для вычисления e по известному s_X (без знания алгебраической структуры кода) требуется алгоритм декодирования случайного кода (NP-сложная задача);

Криптосистема Мак-Элиса:



Односторонняя функция:

– для вычисления c_X^* (кодового слова со случайно сформированной ошибкой) по известным I и e требуется алгоритм полиномиальной сложности;

– для вычисления I и (или) e по известному c_X^* (без знания алгебраической структуры кода) требуется алгоритм декодирования случайного кода (NP-сложная задача);

CFS-схема ЭЦП:

Хеш-образ $s_X = h(h(M)\|i)$, вычисленный по подписываемому сообщению M и значению счетчика i, интерпретируется как синдромная последовательность;

Зная секретный ключ (алгебраическую структуру кода), уполномоченный пользователь использует алгоритм полиномиальной сложности для быстрого нахождения вектора e. Если декодирование не успешно, тогда выбирается другое значение счетчика i и процедура повторяется;

Найденные значения e и i являются частями ЭЦП $Y = (e, i)$ сообщения M. При этом обязательно выполняется равенство:

$$Y = (e, i) : H_X \cdot e^T = h(h(M)\|i)^T,$$

которое лежит в основе процедуры проверки (верификации) ЭЦП.

Предлагаемая схема ЭЦП:

Хеш-образ $c_X^* = h(h(M)\|i)$, вычисленный по подписываемому сообщению M и значению счетчика i, интерпретируется как кодовое слово со случайно сформированной ошибкой;

Зная секретный ключ (алгебраическую структуру кода) уполномоченный пользователь использует алгоритм полиномиальной сложности для быстрого нахождения векторов I и e. Если декодирование не успешно, тогда выбирается другое значение счетчика i и процедура повторяется;

Найденные значения I, e и i являются частями ЭЦП $Y = (I, e, i)$ сообщения M. При этом обязательно выполняется равенство:

$$Y = (I, e, i) : IG_X + e = h(h(M)\|i),$$

которое лежит в основе процедуры проверки (верификации) ЭЦП.

Рис. 1. Кодовые криптосистемы и схемы ЭЦП на их основе

В схеме ЭЦП CFS [31] используется односторонняя функция из схемы Нидеррайтера, но только для формирования и проверки ЭЦП. Для этого подписываемое информационное сообщение M (его сжатый образ) непосредственно связывается со значением синдрома s_X и только уполномоченный пользователь, знающий секретные матрицы X и P в $H_X = X \cdot H \cdot P$, сможет применить быстрый (полиномиальной сложности) алгоритм декодирования для нахождения вектора e. Этот найденный вектор совместно со вспомогательной информацией (значение счетчика i) и составляют подпись $Y = (e, i)$ сообщения M. Для проверки (верификации) подписи достаточно владеть открытым ключом (матрицей H_X) – для этого достаточно вычислить синдром s_X и сравнить его с сжатым образом информационного сообщения. Таким образом, для формирования и проверки ЭЦП используется односторонняя функция из схемы Нидеррайтера: вычислить s_X по известным H_X и e (проверить ЭЦП) вычислительно легко (полиномиальная сложность), а найти e по известным H_X и s_X (сформировать ЭЦП) чрезвычайно сложно (NP-полная задача).

В криптосистеме Мак-Элиса [23] информационное сообщение M рассматривается как информационный вектор I, подлежащий избыточному кодированию. Кодовое слово вычисляется через произведение $c_X = I \cdot G_X$, где порождающая матрица $G_X = X \cdot G \cdot P$ представляет собой открытый ключ, а матрицы X и P – секретный (закрытый) ключ. Шифрограмма $c_X^* = I \cdot G_X + e$ представляет собой кодовое слово c_X с добавленным к нему случайным вектором ошибки e. Уполномоченный пользователь, знающий секретные матрицы X и P в $G_X = X \cdot G \cdot P$, сможет применить быстрый (полиномиальной сложности) алгоритм декодирования для нахождения векторов I и e. Без знания матриц X и P неуполномоченный пользователь вынужден использовать сложный алгоритм декодирования случайного кода, для которого достаточно использовать лишь матрицу G_X. Таким образом, в схеме Мак-Элиса для реализации асимметричного шифрования используется следующая одностороння функция: вычислить c_X^* по известным G_X, I и e вычислительно легко (полиномиальная сложность), а найти I и e по известным G_X и c_X^* чрезвычайно сложно (NP-полная задача).

Предлагаемая схема формирования и проверки ЭЦП использует одностороннюю функцию из схемы Мак-Элиса. Для этого подписываемое информационное сообщение M (его сжатый образ) непосредственно связывается со значением c_X^* и только уполномоченный пользователь, знающий секретные матрицы X и P в $G_X = X \cdot G \cdot P$, сможет применить быстрый (полиномиальной сложности) алгоритм декодирования для нахождения векторов I и e. Эти векторы совместно со вспомогательной информацией (значение счетчика i) составляют подпись $Y = (I, e, i)$ сообщения M. Для проверки (верификации) подписи достаточно владеть открытым ключом (матрицей G_X) – для этого достаточно вычислить c_X^* и сравнить его с сжатым образом информационного сообщения. Таким образом, вычислить c_X^* по известным G_X, I и e (проверить ЭЦП) вычислительно легко (полиномиальная сложность), а найти I и e по известным G_X и c_X^* (сформировать ЭЦП) чрезвычайно сложно (NP-полная задача).

Реализация предлагаемой схемы формирования и проверки ЭЦП осуществляется в соответствии со следующими алгоритмами.

1. Генерация общесистемных параметров: выбираются положительные целые числа m, t.

2. Генерация ключа: формируются пары ключей как в криптосистеме Нидеррайтера на основе алгебраического кода из класса $(n, k, d = 2t + 1)$ кодов. В частном случае может использоваться код из класса $(n = 2^m, k = n - mt, 2t + 1)$ двоичных кодов Гоппы.

Для этого формируются:

– матрица $G: k \times n$ – порождающая матрица алгебраического кода с исправляющей способностью t ошибок,

– матрица $X: k \times k$ – случайная обратимая матрица,

– матрица $P: n \times n$ – случайная матрица перестановок.

В случае применения недвоичных кодов используется также матрица $D: n \times n$ – случайная диагональная матрица. Если код двоичный, тогда под D в дальнейшем будем понимать единичную матрицу.

Открытым ключом является матрица $G_X = X \cdot G \cdot P \cdot D$ и число t (исправляющая способность кода).

Секретным ключом являются матрицы G, X, P и D, а также связанный с матрицей G быстрый (полиномиальной сложности) алгоритм декодирования алгебраического кода.

Алгоритм декодирования позволяет по введенному кодовому слову с ошибками $c^*_X = (c^*_0, c^*_1, ..., c^*_{n-1})$ в случае успеха декодирования найти вектор ошибок $e = (e_0, e_1, ..., e_{n-1})$ и вектор $I = (I_0, I_1, ..., I_{k-1})$, причем $c^*_X = I \cdot G_X + e$. В противном случае (если декодирование не удалось) алгоритм выдает отказ в декодировании вектора $c^*_X = (c^*_0, c^*_1, ..., c^*_{n-1})$, т.е. по такой последовательности алгоритм не может найти вектор ошибок $e = (e_0, e_1, ..., e_{n-1})$ и вектор $I = (I_0, I_1, ..., I_{k-1})$.

3. Формирование подписи

Вход:

1. h – функция хеширования, которая применяется к входным данным x (аргументу функции) произвольной длины. Результатом хеширования является хеш-код $h(x)$ длины n кодовых символов (для двоичных кодов – n бит);

2. Быстрый (полиномиальной сложности) алгоритм декодирования алгебраического кода, который применяется к кодовому слову с ошибками $c^*_X = (c^*_0, c^*_1, ..., c^*_{n-1})$. Предполагается, что в результате выполнения алгоритма декодирования возможны две ситуации:

– если декодирование успешно – выводятся векторы $e = (e_0, e_1, ..., e_{n-1})$ и $I = (I_0, I_1, ..., I_{k-1})$, которые соответствует вектору $c^*_X = (c^*_0, c^*_1, ..., c^*_{n-1})$;

– если декодирование не успешно – выдается сообщение о невозможности найти векторы $e = (e_0, e_1, ..., e_{n-1})$ и $I = (I_0, I_1, ..., I_{k-1})$ для введенного вектора $c^*_X = (c^*_0, c^*_1, ..., c^*_{n-1})$;

3. Открытый текст M, для которого необходимо сформировать ЭЦП.

Выход:

ЭЦП Y для открытого текста M.

Предлагаемый алгоритм формирования ЭЦП

Шаг 1. Хеширование открытого текста M, т.е. вычисление хеш-кода $h(M)$. Присваивание переменной i значения $i = 1$;

Шаг 2. Вычисление хеш-кода $h(h(M) \| i)$, где $h(M) \| i$ – конкатенация (объединение) значений $h(M)$ и i, представленных в виде двух последовательностей;

Шаг 3. Значение $h(h(M)\|i)$ интерпретируется как кодовое слово с ошибками $c_X^* = (c*_0, c*_1, ..., c*_{n-1})$, вычисленное для некоторых $I = (I_0, I_1, ..., I_{k-1})$ и $e = (e_0, e_1, ..., e_{n-1})$, причем $c = IG_X$, $c_X^* = c + e$, т.е. предполагается выполнение равенства (19) для соответствующего открытого ключа $G_X = X \cdot H \cdot P \cdot D$;

Шаг 4. Вычисление значения вектора

$$\bar{c}^* = c_X^* \cdot D^{-1} \cdot P^{-1},$$

который (как предполагается) представляет собой искаженное не более чем в t разрядах кодовое слово алгебраического (n, k, d) кода с порождающей матрицей G и его можно декодировать быстрым алгоритмом полиномиальной сложности, т.е. предполагается, что

$$\bar{c}^* = c_X^* \cdot D^{-1} \cdot P^{-1} = (I \cdot G_X + e) \cdot D^{-1} \cdot P^{-1} = (I \cdot X \cdot H \cdot P \cdot D + e) \cdot D^{-1} \cdot P^{-1} = I \cdot X \cdot H + e \cdot D^{-1} \cdot P^{-1}$$

и алгоритм быстрого декодирования позволит найти вектор $I' = I \cdot X$ посредством декодирования слова $\bar{c}^* = I' \cdot G + e'$, $e' = e \cdot D^{-1} \cdot P^{-1}$;

Шаг 5. Для слова $\bar{c}^* = I' \cdot G + e'$ реализуется выполнение быстрого алгоритма декодирования:

– если декодирование успешно – выводятся найденные векторы $I' = I \cdot X$ и $e' = e \cdot D^{-1} \cdot P^{-1}$, которые соответствует вектору $\bar{c}^* = I' \cdot G + e'$;

– если декодирование не успешно – выдается сообщение о невозможности найти векторы $I' = I \cdot X$ и $e' = e \cdot D^{-1} \cdot P^{-1}$ для введенного вектора $\bar{c}^*$. Присваивание переменной i значения $i = i + 1$ и переход на Шаг 2;

Шаг 6. Вычисление векторов

$$I = I' X^{-1} \text{ и } e = e' \cdot D \cdot P;$$

Шаг 7. Формирование ЭЦП $Y = (I, e, i)$ для открытого текста M.

Таким образом, в результате выполнения рассмотренного алгоритма формирования ЭЦП, вычисляется такое наименьшее положительное целое число i, для которого значение $h(h(M)\|i)$, интерпретируемое как кодовое слово с ошибками c_X^*, соответствует кодовому слову $c = IG_X$ и вектору ошибок e, т.е. формально запишем:

$$Y = (I, e, i) : IG_X + e = h(h(M)\|i) \ . \qquad (5)$$

Задача вычисления векторов I и e по известному вектору $h(h(M)\|i)$ сопряжена с решением задачи декодирования (n, k, d) кода:

– для уполномоченного пользователя (знающего секретный ключ) это вычислительно простая задача (полиномиальной сложности);

– для злоумышленника (знающего только открытый ключ) это вычислительно сложная задача декодирования случайного кода (относящаяся к классу сложности NP-полных задач).

Для верификации (проверки правильности ЭЦП $Y = (I, e, i)$ сообщения M) необходимо убедиться в том, является ли результат хеширования $h(h(M)\|i)$ кодовым словом с ошибками c_X^*, вычисленным по векторам I и e.

4. Верификация

Ввод:

1. Открытый ключ (матрица $G_X = X \cdot H \cdot P \cdot D$ и число t);

2. Функция хеширования h;

3. ЭЦП $Y = (I, e, i)$;

4. Открытый текст M.
Выход:
решение о *правильности* или *неправильности* ЭЦП.

Предлагаемый алгоритм верификации (алгоритм проверки ЭПЦ)

Шаг 1. Вычисление вектора

$$c_X^{*}{}' = IG_X + e;$$

Шаг 2. Вычисление вектора

$$c_X^{*}{}'' = h(h(M)\|i);$$

Шаг 3. Принятие решение о *правильности* или *неправильности* ЭЦП:

– если $c_X^{*}{}' = c_X^{*}{}''$ и $w(e) \le t$ тогда принимается решение о *правильности* ЭЦП;

– если $c_X^{*}{}' \ne c_X^{*}{}''$ и (или) $w(e) > t$ тогда принимается решение о *неправильности* ЭЦП.

Отметим, что предлагаемая процедура верификации защищена от быстрой подделки подписи $Y = (I, e, i)$ на основе добавления произвольного кодового слова применяемого (n, k, d) кода (предложена и подробно рассмотрена в работе [34]). Так, если выбрать произвольное кодовое слово $\hat{c}$ используемого (n, k, d) кода с порождающей матрицей G_X, тогда можно попытаться подделать подпись $Y = (I, e + \hat{c}, i)$. Однако равенство (5) не будет выполняться:

$$Y = (I, e + \hat{c}, i) : IG_X + e + \hat{c} \ne h(h(M)\|i) \ .$$

Это очевидное преимущество предлагаемой схемы ЭЦП дополнительно усилено введенной проверкой веса вектора e, которая предназначена для защиты от других гипотетических атак (например, одновременной подделки и вектора I и вектора e).

Оценим конструктивные характеристики предлагаемой схемы формирования и проверки ЭЦП с использованием алгебраических кодов. При оценке будем предполагать, что используются двоичные сепарабельные коды Гоппы с параметрами [42, 43]:

$$n = 2^m, k = n - mt, t = deg\, G(x), d \ge 2t + 1, \qquad (6)$$

где $G(x)$ – многочлен Гоппы, $deg\, G(x)$ – степень многочлена $G(x)$.

Сложность формирования и проверки ЭЦП. Наиболее затратной частью алгоритма формирования ЭЦП является Шаг 5, на котором выполняется многократная попытка декодирования до достижения успеха. Оценим вероятность успеха декодирования, покажем, что сложность формирования ЭЦП предлагаемым способом сопоставима со схемой CFS [31].

При использовании (n, k, d) кода над $GF(q)$ на вход декодера поступает кодовое слово с ошибками

$$\bar{c}^{*} = I' \cdot G + e'$$

длины n символов из $GF(q)$, причем k-символьный вектор I' может принимать одно из q^k значений, а вектор e' – одно из

$$N = \sum_{i=0}^{t} (q-1)^i C_n^i, \ C_n^i = \frac{n!}{i!(n-i)!}$$

значений (т.к. $w(e') \le t$). Следовательно, число возможных значений вектора $\bar{c}^{*}$ равно $q^k \cdot N$, а общее число возможных n-символьных векторов с элементами из $GF(q)$ определяется как q^n. Тогда вероятность успеха декодирования для единичной попытки на шаге 5:

$$P_{у.д.} = \frac{q^k \cdot \sum_{i=0}^{t} (q-1)^i C_n^i}{q^n} = \frac{\sum_{i=0}^{t} (q-1)^i C_n^i}{q^{n-k}}, \quad (7)$$

что для двоичного случая

$$P_{у.д.} = \frac{\sum_{i=0}^{t} C_n^i}{2^{n-k}} \quad (8)$$

совпадает с аналогичным выражением из [31, с. 163]. При этом, по мере улучшения кодовых соотношений сложность формирования ЭЦП снижается. Так, для совершенного линейного (n,k,d) кода над $GF(q)$, удовлетворяющего верхней кодовой границе Хемминга [44]

$$q^k \le \frac{q^n}{\sum_{i=0}^{t} (q-1)^i C_n^i}, \quad (9)$$

вероятность успеха декодирования будет равна единице.

С учетом кодовых соотношений (6) для двоичных кодов Гоппы и аппроксимации [31]

$$\sum_{i=0}^{t} C_n^i = \frac{n^t}{t!}$$

оценка (8) примет вид

$$P_{у.д.} = \frac{\sum_{i=0}^{t} C_n^i}{2^{n-k}} = \frac{\frac{n^t}{t!}}{n^t} = \frac{1}{t!}, \quad (10)$$

что совпадает с аналогичным выражением из [31, с. 163] для схемы ЭЦП CFS.

Таким образом, успех в декодировании (на шаге 5 алгоритма формирования ЭЦП) будет достигнут при реализации в среднем после $t!$ попыток. Для двоичных кодов Гоппы каждая попытка требует $t^2 \cdot m^3$ двоичных операций [20, 31], т.е. среднее число битовых операций, которые необходимо затратить для формирования ЭЦП как по схеме CFS, так и по предлагаемой схеме, определяется как

$$N_{ф.п.} = t^2 \cdot m^3 \cdot t!. \quad (11)$$

В (11) не учтены затраты на формирование хеш-кодов (шаги 2 и 3 алгоритма формирования ЭЦП), а также операции с матрицами маскирования X и P (на шагах 4 и 6 алгоритма).

Для проверки (верификации) ЭЦП $Y = (I,e,i)$ необходимо вычислить хеш-код $h(h(M)\|i)$ и сравнить его с результатом вычисления $IG_X + e$. Если не учитывать сложность хеширования, тогда сложность проверки ЭЦП будет определяться выражением

$$N_{п.п.} = k \cdot n = m \cdot t \cdot 2^m. \quad (12)$$

Стойкость ЭЦП. В работе [26] показано, что стойкость несимметричных криптосистем Мак-Элиса и Нидеррайтера эквивалентна. Если принять предположение о тождественности оценок стойкости соответствующих ЭЦП (по схеме CFS и по предлагаемой схеме), тогда следует использовать формулы (17) и (18) из работы [34].

Длина ЭЦП. Цифровая подпись $Y = (I,e,i)$ состоит из трех частей: вектора I длиной k бит, вектора e длиной n бит и целого числа i, которое может принимать значение в диапазоне $0,1,..,q^{n-k}-1$ (после q^{n-k} попыток будет гарантирован успех в декодировании на Шаге 5 алгоритма формирования ЭЦП). Таким образом, при использовании двоичных кодов битовая длина ЭЦП (записанной в виде последовательности (I,e,i)) будет определяться выражением

$$l_{ЭЦП} = 2 \cdot n = 2^{m+1}. \quad (13)$$

При этом вектор e может быть преобразован в безызбыточную последовательность e^* длины $\lceil \log_2(N_{w(e)\le t}) \rceil$ бит. С учетом безызбыточного кодирования вектора e^* выражение (13) для длины ЭЦП $Y = (I, e^*, i)$ перепишем в виде

$$l_{ЭЦП}^* = \left\lceil \log_2 \left(\sum_{i=0}^{t} C_n^i \right) \right\rceil + n = \left\lceil \log_2 \left(\sum_{i=0}^{t} C_{2^m}^i \right) \right\rceil + 2^m .$$

Используя выражение (9) для верхней границы Хемминга получим:

$$l_{ЭЦП}^* \le \left\lceil \log_2 \; 2^{n-k} \right\rceil + n = m \cdot t + 2^m . \qquad (14)$$

m	t	Криптосистема Нидеррайтера (ЭЦП по схеме CFS)			Криптосистема Мак-Элиса (ЭЦП по предложенной схеме)		
		$l_{ЭЦП}^*$	$l_{о.к.}$	$l_{з.к.}$	$l_{ЭЦП}^*$	$l_{о.к.}$	$l_{з.к.}$
10	10	200	102400	20240	1124	946176	864016
	20	400	204800	50240	1224	843776	689216
	30	600	307200	100240	1324	741376	534416
	40	800	409600	170240	1424	638976	399616
	50	1000	512000	260240	1524	536576	284816
	60	1200	614400	370240	1624	434176	190016
	70	1400	716800	500240	1724	331776	115216
	80	1600	819200	650240	1824	229376	60416
	90	1800	921600	820240	1924	126976	25616
	100	2000	1024000	1010240	2024	24576	10816
12	20	480	983040	106752	4336	15794176	14917888
	40	960	1966080	279552	4576	14811136	13124608
	60	1440	2949120	567552	4816	13828096	11446528
	80	1920	3932160	970752	5056	12845056	9883648
	100	2400	4915200	1489152	5296	11862016	8435968
	120	2880	5898240	2122752	5536	10878976	7103488
	140	3360	6881280	2871552	5776	9895936	5886208
	160	3840	7864320	3735552	6016	8912896	4784128
	180	4320	8847360	4714752	6256	7929856	3797248
	200	4800	9830400	5809152	6496	6946816	2925568
14	50	1400	11468800	719376	17084	256966656	246217232
	100	2800	22937600	2189376	17784	245497856	224749632
	150	4200	34406400	4639376	18484	234029056	204262032
	200	5600	45875200	8069376	19184	222560256	184754432
	250	7000	57344000	12479376	19884	211091456	166226832
	300	8400	68812800	17869376	20584	199622656	148679232
	350	9800	80281600	24239376	21284	188153856	132111632
	400	11200	91750400	31589376	21984	176685056	116524032
	450	12600	103219200	39919376	22684	165216256	101916432
	500	14000	114688000	49229376	23384	153747456	88288832
	550	15400	126156800	59519376	24084	142278656	75641232
	600	16800	137625600	70789376	24784	130809856	63973632
	650	18200	149094400	83039376	25484	119341056	53286032
	700	19600	160563200	96269376	26184	107872256	43578432

Объем ключевых данных в предлагаемой схеме определяется объемом ключевых данных несимметричной криптосистемы Мак-Элиса. Для двоичных кодов имеем:

- битовая длина открытого ключа (число двоичных ячеек матрицы $G_X = X \cdot G \cdot P$)

$$l_{о.к.} = k \cdot n = (2^m - m \cdot t) \cdot 2^m; \quad (15)$$

- битовая длина закрытого ключа (число двоичных ячеек матрицы X плюс битовая длина n целых чисел в диапазоне $0,1,..,n-1$ для указания правила заполнения матрицы P)

$$l_{з.к.} = k^2 + n \cdot \lceil \log_2 n \rceil = (2^m - m \cdot t)^2 + 2^m \cdot m. \quad (16)$$

Таким образом, основные конструктивные характеристики предлагаемой схемы формирования и проверки ЭЦП сопоставимы с характеристиками ЭЦП по схеме CFS [34]. При этом для высокоскоростных кодов (с $R = \frac{k}{n} > \frac{1}{2}$) объем ключевых данных схемы Нидеррайтера меньше, чем у схемы Мак-Элиса, а для низкоскоростных (с $R = \frac{k}{n} < \frac{1}{2}$) – наоборот, меньший объем ключей имеет схема Мак-Элиса. Схемы ЭЦП (предлагаемая и CFS) наследуют это свойство. Ввиду добавления в ЭЦП $Y = (I, e, i)$ вектора I битовая длина подписи больше, по сравнению со схемой CFS, на $2^m - m \cdot t$ бит. В остальном при оценке параметров предлагаемой схемы ЭЦП следует ориентироваться на данные [34, табл. 3].

В таблице приведены оценки длин ключей и длин подписей для схемы CFS (с использованием криптосистемы Нидеррайтера) и по предлагаемой схеме (с использованием криптосистемы Мак-Элиса). Для оценки параметров использовались конструктивные кодовые соотношения (6) для двоичных сепарабельных кодов Гоппы.

Выводы

Кодовые криптосистемы рассматриваются на сегодняшний день как реальная альтернатива в построении надежных постквантовых алгоритмов криптографической защиты информации [21, 22]. Исследование стойкости, быстродействия и возможности эффективной программно-аппаратной реализации таких криптосистем представляет важную и актуальную научно-техническую проблему, непосредственно связанную с обеспечением услуг информационной безопасности в условиях использования квантовых вычислительных систем и алгоритмов.

В статье рассмотрены кодовые криптосистемы Мак-Элиса и Нидеррайтера, а также алгоритмы формирования и проверки ЭЦП на их основе. В частности, с использованием криптопреобразований по схеме Мак-Элиса была предложена новая схема ЭЦП, которая по своим основным параметрам (стойкости, длине ключей и длине подписей) сопоставима с уже известной схемой CFS. Основное отличие предложенной схемы ЭЦП состоит в способе формирования подписи: информационная последовательность (ее сжатый образ) интерпретируется не как синдром кодового слова (как в схеме CFS), а как искаженное ошибками кодовое слово. Уполномоченный пользователь формирует ЭЦП в результате быстрого (полиномиальной сложности) декодирования. Неуполномоченный пользователь (не знающий правило маскирования алгебраического кода) для подделки подписи вынужден декодировать случайный код, решая NP-полную задачу. Проверка подписи осуществляется посредством матричного умножения элементов подписи с проверкой полученного результата. Предложенная схема ЭЦП защищена от быстрой подделки подписи на основе добавления произвольного кодового слова применяемого кода (эта атака предложена и подробно рассмотрена в работе [34]). Указанное преимущество дополнительно усилено введенной проверкой веса Хемминга, которая предназначена для защиты от других гипотетических атак (например, одновременной подделки нескольких элементов подписи).

Проблемным вопросом практического применения ЭЦП на алгебраических кодах остается чрезвычайно высокая сложность формирования подписи. Ввиду того, что реальные кодовые характеристики при большой длине кода значительно уступают верхним кодовым границам, сложность формирования ЭЦП растет как факториал от исправляющей способности кода. Фактически, это означает, что с увеличением исправляющей способности практическое использование таких ЭЦП вычислительно недостижимо. Однако для совершенных кодов (удовлетворяющих верхней кодовой границе Хемминга) сложность формирования ЭЦП минимальна, она определяется сложностью быстрого декодирования используемого алгебраического кода. В этом смысле поиск кодов, удовлетворяющих верхним кодовым границам, приобретает особую актуальность.

Другим возможным направлением снижения сложности формирования ЭЦП является совершенствование самой схемы формирования подписи. Это направление представляется особенно актуальным при условии использования уже известных кодовых конструкций, например двоичных сепарабельных кодов Гоппы.

Список литературы: 1. *Alfred, J. Menezes, Paul C. van Oorschot, Scott, A.* Vanstone. Handbook of Applied Cryptography – CRC Press, 1997. – 794 p. 2. *Горбенко, І.Д., Горбенко, Ю.І.* Прикладна криптологія. Теорія. Практика. Застосування : підручник для вищих навч. закладів. – Харків : Форт, 2013. – 880 с. 3. *Nigel Smart*. Cryptography: An Introduction (3rd Edition). – 432 pp. https://www.cs.umd.edu/~waa/414-F11/IntroToCrypto.pdf 4. *ISO/IEC 9796-2:2010* "Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms". 5. *ISO/IEC 9796-3:2006* "Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms". 6. *ISO/IEC 14888-1:2008* "Information technology – Security techniques – Digital signatures with appendix – Part 1: General". 7. *ISO/IEC 14888-2:2008* "Information technology – Security techniques – Digital signatures with appendix – Part 2: Integer factorization based mechanisms". 8. *ISO/IEC 14888-3:2006* "Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms". 9. *ISO/IEC 15946-1:2008* "Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General". 10. *ISO/IEC 15946-5:2009* "Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 5: Elliptic curve generation". 11. *David Deutsch and Richard Jozsa*. Rapid solutions of problems by quantum computation // Proceedings of The Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 439, no. 1907. – 1992. – P. 553-558. 12. *Cleve, R., Ekert,A., Macchiavello, C., Mosca, M*. Quantum algorithms revisited // Proceedings of The Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 454, no. 1969. – 1998. – P. 339-354. 13. *Simon, D. R.* On the power of quantum computation // Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium. – P. 116-123. 14. *Grover, L.* A fast quantum mechanical algorithm for database search. // Proceedings of the 28th annual ACM symposium on the theory of computing (STOC, 96). ACM Press, New York. – 1996. – P. 212–219. 15. *Grover, L.* A framework for fast quantum mechanical algorithms // Proceedings of the 13th annual ACM symposium on theory of computing (STOC' 98). ACM Press, New York. – 1998. – P. 53–62. 16. *Shor, P. W*. Algorithms for quantum computation: discrete logarithms and factoring // Foundations of Computer Science : Conference Publications. – 1994. – P. 124-134. 17. *Shor, P. W*. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer // Foundations of Computer Science: Conference Publications. – 1997. – P. 1484-1509. 18. *Neal Koblitz* and *Alfred J. Menezes*. A Riddle Wrapped in an Enigma. https://eprint.iacr.org/2015/1018.pdf 19. *Committee* on National Security Systems, Use of public standards for the secure sharing of information among national security systems, Advisory Memorandum 02-15, July 2015. https://cryptome.org/2015/08/CNSS_Advisory_Memo_02-15.pdf. 20. *Bernstein, Daniel J., Buchmann, Johannes, and Dahmen, Erik.* Post-Quantum Cryptography. – 2009, Springer-Verlag, Berlin-Heidleberg. – 245 p. 21. *Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone*. NISTIR 8105. Report on Post-Quantum Cryptography. National Institute of Standards and Technology. http://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf. 22. *Dustin Moody*. Post Quantum Cryptography: NIST's Plan for the Future. National Institute of Standards and Technology. http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/pqcrypto-2016-presentation.pdf. 23. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory. DSN Progress Report 42-44, Jet Propulsion Lab., Pasadena, CA, January-February, 1978. P. 114-116. 24. *Niederreiter, H.* Knapsack-type cryptosystems and algebraic coding theory // Problem Control and Inform Theory, 1986, v. 15. P. 19-34.

25. *Stasev, Yu. V., Kuznetsov, A. A.* Asymmetric Code-Theoretical Schemes Constructed with the Use of Algebraic Geometric Codes // Cybernetics and Systems Analysis, Volume 41, Issue 3, May 2005, Pages 354 – 363. 26. *Сидельников, В.М.* Криптография и теория кодирования // Материалы конф. «Московский университет и развитие криптографии в России». – Москва : МГУ, 2002. – 22 с. 27. *Сидельников ,В.М., Шестаков, С.О.* О системе шифрования, построенной на основе обобщенных кодов Рида – Соломона. // Дискретная математика. – 1992. – Т.4.№3. – С.57-63. 28. Кузнецов, А.А. Алгебраическая теория блоковых кодов и ее приложения в криптографии // Перша міжнар. наук. конф. 25–27 травня 2005р. „Теорія та методи обробки сигналів". Тези доповідей. – К. : НАУ. – 2005. – С. 6 – 8. 29. *Кузнецов, А.А.* Исследование эффективности криптосистем на алгебраических блоковых кодах // Системи обробки інформації. – Харків : ХУПС, 2005 – Вип. 4. – С. 202 –206. 30. *Кузнецов, А.А.* Исследование помехоустойчивости и криптостойкости теоретико-кодовых схем // Моделювання та інформаційні технології. – Київ : НАНУ, 2005. – №33. – С. 81-84. 31. *Courtois, N., Finiasz, M., and N.Sendrier* How to achieve a McEliece-based digital signature scheme // Advances in Cryptology – ASIACRYPT 2001, volume 2248, pages 157–174. 32. *E. Berlekamp, R. McEliece, H. van Tilborg.* On the Inherent Intractability of Certain Coding Problems // IEEE Transactions on Information Theory, vol. IT-24, No. 3, May 1978. – P. 384-386. 33. *Кузнецов, А., Пушкарев, А., Кавун, С., Калашников, В.* Несимметричные криптосистемы на основе алгебраического кодирования: современное состояние, существующие противоречия и перспективы практического использования на постквантовый период // Computer science and cybersecurity. – Kharkiv : V.N. Karazin Kharkiv National University, 2016. – Issue 3(3) 2016. – P. 36-60. 34. *Кузнецов, А.А., Пушкарев, А.И., Сватовский, И.И., Шевцов, А.В.* Несимметричные криптосистемы на алгебраических кодах для пост-квантового периода // Радиотехника. – 2016. – Вып. 186. – С. 70-90. 35. *Кузнецов, О., Пушкарьов, А., Шевцов, О., Кузнецова, Т.* Несиметричне криптографічне перетворення з використанням алгебраїчних блокових кодів // Захист інформації. – Київ : Нац. авіаційний ун-т, 2016. – Т. 18, №4, жовтень-грудень 2016. – С. 269-275. 36. *Кузнецов, О.О., Пушкарьов, А.І., Шевцов, О.В, Кузнецова, Т.Ю.* Несиметричне криптоперетворення з використанням алгебраїчних блокових кодів // Актуальні задачі та досягнення у галузі кібербезпеки: матеріали Всеукр. наук.-практ. конф., 23–25 листопада 2016 року. – Кропивницький : КНТУ, 2016. – С. 124-127. 37. *Кузнецов О.О., Пушкарьов А.І., Шевцов О.В., Кузнецова Т.Ю.* Несиметричні крипто-кодові системи захисту інформації // 71-а наук.-техн. конф. професорсько-викладацького складу, науковців, аспірантів та студентів : матеріали конф. (6-8 грудня 2016 р.) – Одеса : ОНАЗ, 2016. – 84-87 с. 38. *Кузнецов, О.О., Пушкарьов, А.І., Сватовський, І.І., Шевцов, А.В., Кузнецова, Т.Ю.* Несиметричні криптосистеми на алгебраїчних блокових кодах // Актуальні питання забезпечення кібербезпеки та захисту інформації: тези доповідей учасників III Міжнар. наук.-практ. конф., 22-25 лютого 2017 р. – К. : Вид-во Європейського ун-ту, 2017. – С. 108-112. 39. *Кузнецов, А.А., Сватовский, И.И., Шевцов, А.В.* Схемы электронной цифровой подписи на основе помехоустойчивых кодов // Труды науч.-техн. конф. с международным участием «Компьютерное моделирование в наукоемких технологиях», 26-31 мая 2016 г. – Харьков : ХНУ имени В.Н. Каразина, 2016. – С. 191-193. 40. *Кузнецов, А., Сватовский, И., Шевцов, А.* Электронная цифровая подпись на алгебраических кодах для постквантового периода // Матеріали V-ї міжнар. наук.-техн. конф. "Захист інформації і безпека інформаційних систем", 2-3 червня 2016 р. – Львів : НУ "Львівська політехніка", 2016. – С. 122-123. 41. *Кузнецов, А.А., Сватовский, И.И., Шевцов, А.В.* Алгоритми цифрового підпису для постквантового періоду із застосування алгебраїчних кодів: сучасний стан, існуючі протиріччя та перспективи практичного застосування // Безпека інформації в інформаційно-телекомунікаційних системах : Матеріали міжнар. наук.-практ. конф. Вип. 18, 25-26 травня 2016 р. – К. : Державна служба спеціального зв'язку та захисту інформації України, 2016 – С. 17. 42. *Гоппа, В. Д.* Новый класс линейных корректирующих кодов // Проблемы передачи информации. – 1970. – Т. 6, вып. 3. – С. 24–30. 43. *Гоппа, В. Д.* На неприводимых кодах достигается пропускная способность ДСК // Проблемы передачи информации. – 1974. – Т. 10, вып. 1. – С. 111–112. 44. *MacWilliams, F. J. and Sloane,N. J.* A. The theory of error-correcting codes. – North-Holland, Amsterdam, New York, Oxford, 1977. – 762 pp.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 11.04.2017

УДК 004.056.55

Е.Г. КАЧКО, канд. техн. наук, Д.К. ТЕЛЕВНЫЙ

ИССЛЕДОВАНИЕ ПРИМЕНИМОСТИ SMT/SAT ДОКАЗАТЕЛЬСТВ В КРИПТОАНАЛИЗЕ ХЕШ-ФУНКЦИЙ СЕМЕЙСТВА КЕССАК

Введение

Большой областью применения средств доказательства SMT (*SMT solvers*) является генерация тестовых случаев, статический анализ программ, ограниченная проверка моделей (*BMC*) и *k*-индукция, проверка на эквивалентность.

SMT (*Satisfiability modulo theory)* это обобщение Boolean SAT (*Satisfiability theory*) [1], где набор переменных замещается предикатами из теорий. Одними из таких теорий являются теории структур данных, применяемых при верификации программ. Особую роль в криптологии имеют теории массивов и битовых векторов, применяющиеся при моделировании функций.

В настоящее время существует множество SMT решателей: CVC, CVC3, Z3, многие выпущены под свободными лицензиями (BSD, MIT).

Основная идея исследования

Проблема выполнимости – задача определения – имеет ли формула модель. В булевых выражениях такой моделью является присваивание истинности булевым переменным. В логике предикатов первого порядка модель присваивает значения из домена переменным и доменных интерпретаций функциям и предикатным символам. Для теорий, таких как арифметические или теории структур данных, модель принимает определенное множество интерпретаций символов теории.

В компьютерных науках и прикладной математике SMT проблема – проблема нахождения решения для логических формул на основе теорий, описанных логикой первого порядка.

В криптологии применение SMT сводится к валидации путем доказательств выполнимости или невыполнимости формул F в теориях. Теории первого порядка являют собой множество дедуктивно-полных высказываний.

Применительно к криптоанализу SMT можно использовать для нахождения выполнимых или невыполнимых формул на основе теорий, для групп криптографических преобразований. Таким образом, при моделировании криптопреобразований с применением SMT возможно не только тестирование, но и нахождение возможных слабых мест.

Криптографические модели строятся на алгебраических и логических преобразованиях.

Рассмотрим основные определения для доказательств, основанных на теориях. Пусть *DC(G)* дедуктивное замыкание множества высказываний *G*. Тогда для каждой теории $DC(\tau) = \tau$. Теория называется совместимой если $\text{False} \notin \tau$.

Формула $\phi(x)$ выполнима в теории τ, если существует модель в $DC(\tau \cup \exists x.\phi(x))$. Таким образом, существует модель M для τ, в которой формула приобретает значение истины. Это также называется τ-выполнимостью. Запись имеет вид $M \models_\tau \phi(x)$. Формула ϕ называется валидной в теории τ, если $\forall \vec{x}.\phi(x) \in \tau$ формула истинна в любой модели M теории τ.

SAT/SMT средства предполагают, что формула задается в виде конъюнктивной нормальной формы (КНФ), элементами которой являются дизъюнкции атомарных элементов или их отрицаний. Процесс решения модели – поиск конечных значений булевых выражений и анализа конфликтов. Большая часть средств построена на алгоритме *DPLL* [1]. Для КНФ, построенных на предикатах определенной теории, существует комплексный алгоритм *DPLL(T)* [1] с решателем теорий *T*. В этом виде *DPLL* рассматривает модели, в которых

входящие без отрицания литералы могут быть атомарными высказываниями теории первого порядка.

Существует несколько применений SMT/SAT в области криптологии: верификация моделей преобразований, статистический анализ Side-channel атак, генерация тестовых случаев.

Стандарт SHA3 – конструкция на основе функций семейства Keccak-p-преобразований (permutation) [2], которые могут быть использованы в моделях различных шифрах. Некоторые инструкции раунда Keccak-p являются операциями над полиномами в $GF(2)^n$. Модель преобразования может быть построена на основе массивов слов или битовых векторов. Таким образом, для верификации могут быть применимы предикаты из теории битовых векторов (англ. *bit vector theory*) или массивов (англ. a*rray theory*) .

Задача исследования применимости SMT в моделировании и криптоанализе хеш-функций являет собой в первую очередь поиск доказательств существования возможных мест атак времени выполнения. Отдельно можно выделить применимость SMT/SAT для доказательства эквивалентности программных моделей и поиск возможных ошибок.

Применение SMT доказательства для верификации защиты преобразования Keccak-p от side-channel attack

Факультет компьютерной инженерии (англ. *ECE)* Технического Университета Вирджинии (англ. *Virginia Tech*) описывает применимость SMT на примере side-channel атак реализации криптопреобразований и метода противодействия атакам, где с секретным битом и множеством случайных битовых переменных происходит набор логических преобразований,
которые исключают возможность статистического анализа зависимости результата от секрета [3].

SMT верификация методов противодействия сводится к проверке существования зависимости критических данных от промежуточных вычислений и поиск уязвимых инструкций.

Рассмотрим математическую модель применимую для верификации метода «masking secret input» (1).

$$\begin{gathered}(x,k)\\ F(x,k)\\ I_1(x,k,r),\ldots,I_n(x,k,r)\\ \sum_{r\bullet 0,1s} I(x,k,r)\\ \sum_{r\bullet 0,1s} I(x,k',r)\\ D_{x,k},D_{x,k'}\end{gathered} \tag{1}$$

где *(x, k)* – кортеж из сообщения и ключа; *F(x, k)* – функция-преобразование; *r* – случайное значение; *I(x, k, r)* – формулы промежуточных вычислений; *s* – количество бит в r; *Σ(I(…))* – количество решений для формулы *I(x, k r)*, $D_{x,\ k}$ – вероятность *I(p, k, r)* быть логической единицей.

Тогда условие можно записать в виде выражения логики предикатов первого порядка или его отрицания:

$$\begin{gathered}\bullet\ x.\bullet\ k,k'.\left(\sum I(x,r,k)\neq\sum I(x,r',k)\right)\\ \forall\ x.\forall\ k,k'.\left(\sum I(x,r,k)=\sum I(x,r',k)\right)\end{gathered} \tag{2}$$

Выражение (1) является формулой выполнимости (англ. s*atisfiability*), когда ее отрицание – валидности (англ. *validity*). Если валидность из (2) истинно для всех значений, значит что операция *I* предельно скрыта (англ. *masked*) [3].

На практике такая запись обладает вычислительной экспоненциальной сложностью и не может быть применима.

Задача выполнимости булевых формул важна как с теоретической, так и с практической точек зрения. В теории сложности это первая задача, для которой было доказана принадлежность классу NP-полных задач. Теорема Кука утверждает, что задача о выполнимости булевой функции является NP-полной [1].

Решением на практике является использование статического анализа кода и верификации. Верификация SMT проводится для небольших размером регионов кода с применением таких подходов: для каждой инструкции производится классификация дополнительных
переменных и удалением ненужных случайных, декомпозицией и выделением атомарных операций в потоке управления, статическим анализом регионов, где безопасность гарантирована с последующим удалением их из верификации, формальной верификацией остальных частей из потока [1].

Пусть Φ определяет SMT формулу, созданную для проверки промежуточных результатов *I(x, k, r)*. Метод верификации определяет, что Φ – выполнима тогда и только тогда, когда хотя бы одна I(…) неполностью сокрыта. Определим Φ (3) как дизъюнкцию выражений исходя из (1):

$$\Phi := \left(\psi_k^{ri}\right) \wedge \left(\psi_{k'}^{ri}\right) \wedge \psi_{bti} \wedge \psi_s \wedge \psi_{diff},\ ri = \overline{0,..,2^s - 1}\ , \tag{3}$$

где составные части определены как:

$\psi_{k(')}^{r}$ – программная логика, каждая из формул кодирует копию функциональности I,

r – случайное значение взятое на промежутке ri, а ключ – k, k'. Все копии принимают одинаковое значение текста;

ψ_{bit} – Boolean-to-int кодирует преобразование булевого результата I(…) в целое число;

ψ_s – кодирует 2 суммирования логических 1 из множества результатов 2^s копий I(…);

ψ_{diff} – высказывание определяет, то что сумы должны иметь разные результаты.

Формула (3) является SMT кодированием анализа для статистической проверки зависимости результата от секретного ключа.

Для того чтобы скрытый код мог противостоять *DPA* – атакам первого порядка (англ. *First-order DPA*) [4], все подпрограммы *I(x, r, k)* должны быть скрыты. Тем не менее это свойство эффективно при *DPA* атаках высших порядков (англ. *High-order DPA*) [4], где наблюдатель может считывать значения сразу с нескольких промежуточных результатов.

Таким образом можно записать условие выполнимости защиты против *high-order DPA:*

$$\bullet\ x\ .\bullet\ k,k'\ .\left(\sum_{r\bullet\ 0,1}\sum_{i=0}^{d} I_i(x,k,r) \neq \sum_{r\bullet\ 0,1}\sum_{i=0}^{d} I_i(x,k,r)\right) \tag{4}$$

Для эксперимента были выбрана функции *Keccak-p*, реализация которой взята из официального репозитория разработчиков. Таблица хранит условия и результат работы алгоритма Sleuth [3] для различных вариантов:

P1 – Mac-keccak 512 masked;

P2 – Masked Chi step from Keccak-p;

P3 – Mac-keccak 512 not-perfect mask in Chi.

Таблица содержит усредненные результаты проведенных испытаний нахождения регионов кода, которые не полностью защищены. Реализация взята из NIST Keccak Reference [2].

P	Nodes	Plains	Keys	Rands	Masked	Nodes failed	Nodes checked	Time
P1	128k	288	288	805	T	0	128k	92m

P	Nodes	Plains	Keys	Rands	Masked	Nodes failed	Nodes checked	Time
P2	19	3	0	4	F	1	19	0.15s
P3	128	288	288	805	F	512	128k	113m

Поиск минимальных диференциальных характеристик с помощью CryptoSMT

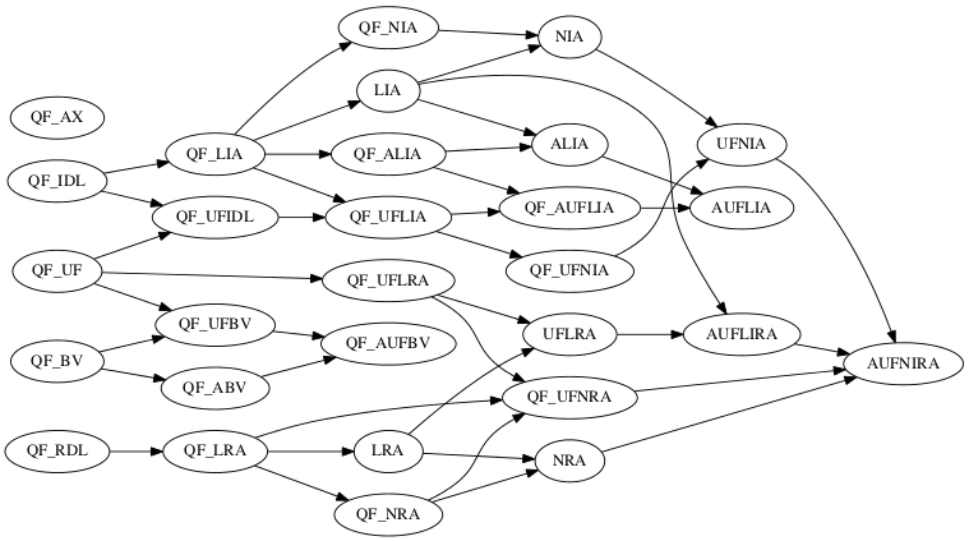
Криптоанализ хеш-преобразований обычно включает себя дифференциальный криптоанализ и поиск коллизий, обратных отражений функций.

Семейство преобразований Keccak-p основано на операции над массивами бит, таким образом КНФ должна быть основана на высказываниях теории битовых векторов.

CryptoSMT – инструмент, построенный на базе SMT доказательств. Для работы утилиты необходимы STP, CryptoMiniSat, Boolector решатели. Работа этих инструментов основана на стандарте SMT-lib 2.0, который содержит требования к решателю теорий [5].

Стандарт SMT-lib 2.0 опредеяет теорию как множество высказываний и операций элементов теории.

На рисунке изображена иерархия теорий, представленная в SMT-lib 2.0 [6].



Применение SAT доказательств для дифференциального анализа преобразований подразумевает: реализацию КНФ модели преобразования, формальной верификации преобразований, КНФ поиска минимального веса дифференциальной характеристики.

Рассмотрим в качестве примера формирование теоремы модели линейного преобразования в раунде, основанной на атомах теории битовых векторов из SMT-LIB 2.0. [6]

```
  for i in range(5):
  command += "ASSERT({} = BVXOR({}, BVXOR({}, BVXOR({}, BVXOR({},
{})))));\n".format(
      c[i + 5*rnd], s[i + 5*0 + 25*rnd], s[i + 5*1 + 25*rnd],
      s[i + 5*2 + 25*rnd], s[i + 5*3 + 25*rnd], s[i + 5*4 + 25*rnd])
```

Формирование теоремы вычислений веса дифференциальной характеристики можно привести как:

```
    command += ("ASSERT({0} = (IF {1} = 0b{4} THEN BVSUB({5},0b{4},0b{6}1)"
"ELSE BVXOR({2}, {3}) ENDIF));\n".format(
    mp[z + wordsize*y + 5*wordsize*rnd],
    xin[z + wordsize*y + 5*wordsize*rnd],
    varibits, doublebits, "1"*5,
    5, "0"*4))
```

Среднее время поиска характеристик на основе сконструированной теоремы преобразований (Intel Core i7 2.2Hz, 4Gb RAM) – 113min.

Проверка эквивалентности функций с помщью Cryptol

Язык Cryptol является функциональным DSL *(Domain Specific Language*) языком, с возможностью верификации описанной модели посредством теорем. В Cryptol 2.0 основным решателем является Z3 от Microsoft Research.

Следующая запись объявляет теорему для проверки эквивалентности двух аргументов. Тип возвращаемого значения – всегда логическое значение в виде бита.

```
    T: [8] → [8]→ [Bit]
    theorem T: {x, y}. x == y.
```

Cryptol предоставляет несколько инструментов для работы с теоремами. Quickcheck – технология, появившаяся для языка Haskell, идеей которой является генерация случайных тестов, основанных на определении теоремы с мономорфными, конечными типами аргументов. Так команда *«:check»* возвращает не только результат, но и покрытие кода тестами.

В случае когда результат теоремы – FALSE, интерпретатор возвращает набор значений, в которых не выполняется условие. Команда *«:exhaust»* проводит тестирование для всех возможных наборов аргументов теоремы. [7]

Формальные доказательства теорем (*«:prove»*) реализованы с помощью проверки на эквивалентность функций, где вторая функция – логическая ИСТИНА.

Модульное тестирование Keccak-p можно определить как множество теорем, проверяющих эквивалентность преобразования битового сообщения с валидным результатом из технической документации. Следующий код содержит пример использования доказательств теорем для тестирования.

```
    theorem testsPass: tests == ~zero;
    tests = [t00 t01 t02 t03 ...];
    t01 = SHA_3_224 (r `{n=0} 0x00) == md
0xF71837502BA8E10837BDD8D365ADB85591895602FC552B48B7390ABD;
    t02 = SHA_3_224 (r `{n=1} 0x00) == md
0x860E3EC314C5CBF19C1A4314E9EA8CB85CECD18BD850B42F5C6F2A07;
    ...
    t40 = Keccak_r544c256 (r `{n=0} 0x00) == md
0xA3CEA55CFD9F4432AD3F9AE33673AE12665F66D150A11AF54E007C7F26F7C9A6E69862E14A2BAD
40048D439E26FB67B40807412BAE2EB42B6896B1D4D602755B23EC19E4;
    t41 = Keccak_r544c256 (r `{n=70} 0xF8CB65B7FE6995F200) == md
0x730AD05FCC3DD4E250D00D5426A42BDAEB8615772F1D1BC0F9AE2639F8920A1BF36CE60F893EAF
E782095A903848C7B150E79B3AAB32C1C3EDE8AB4E64D4015E5E47021B;
```

Выводы

Изучены варианты применения SMT/SAT в криптологии, где верификация осуществляется с помощью логических выражений и является формальным методом. Был проведен эксперимент доказательства достаточной защиты шагов Keccak от side-channel атак.

CryptoSMT является ярким представителем инструментов, использующих SMT (SAT) для криптоанализа преобразований, с применением решателей теорий первого порядка.

Поиск обратных отражений раунда Keccak-p можно произвести с помощью теории битовых векторов.

Язык Cryptol использует Z3 prover для проверки эквивалентности функций и формальной верификации на основе теорем. Теоремы могут быть использованы для генерации и тестирования частей модели в случае мономорфности и конечности аргументов. Верификация модели основана на технологии эквивалентности.

Список литературы: 1. *Nieuwenhuis, R., Oliveras, A., Tinelli, C.* Solving SAT and SAT Modulo Theories: From an Abstract Davis-Putnam-Logemann-Loveland Procedure to DPLL(T) // Journal of the ACM. – 2006. – Т. 53, № 6. – P. 937 – 977. 2. *Guido Bertoni, Joan Daemen, Michaël Peeters Gilles Van Assche.* The Keccak sponge function family [Электронный ресурс]. – Режим доступа: http://keccak.noekeon.org/ – 21.04.2017. – Загл. с экрана. 3. *Hassan Eldib, Chao Wang, and Patrick Schaumont* SMT-Based Verification of Software Countermeasures against Side-Channel Attacks [Электронный ресурс]. – Режим доступа: http://www.faculty.ece.vt.edu/chaowang/pubDOC/Eldib14maskVerif.pdf – 21.02.2011. – Загл. с экрана. 4. *Брюс Шнайер.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке С. – Москва : Триумф, 2002. 5. *Clark Barrett, Aaron Stump, Cesare Tinelli.* The SMT-LIB Standard Version 2.0 [Электронный ресурс]. – Режим доступа: http://smtlib.cs.uiowa.edu/papers/smt-lib-reference-v2.0-r10.12.21.pdf – 21.11.2011. – Загл. с экрана. 6. *SMT-LIB* The satisfiability modulo theories library [Электронный ресурс]. – Режим доступа: http://smtlib.cs.uiowa.edu/logics.shtml – 21.03.2017. – Загл. с экрана. 7. *Levent Erk* ok Theorem declarations in Cryptol [Электронный ресурс]. – Режим доступа: http://community.galois.com/pipermail/cryptol-users/attachments/20110316/f66e99b2/attachment-0003.pdf – 02.10.2008. – Загл. с экрана.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 04.04.2017

УДК 621.3.06

К.Е. ЛИСИЦКИЙ

ЗАКОН РАСПРЕДЕЛЕНИЯ ВЕРОЯТНОСТЕЙ СМЕЩЕНИЙ ТАБЛИЦ ЛИНЕЙНЫХ АППРОКСИМАЦИЙ СЛУЧАЙНЫХ ПОДСТАНОВОК

Введение

В соответствии с развиваемой в последнее время новой методологией оценки стойкости блочных симметричных шифров к атакам дифференциального и линейного криптоанализа [1] обострился интерес к математическому описанию случайных подстановок. Он вызван тем, что оказалось, что все блочные симметричные шифры асимптотически приобретают свойства случайных подстановок. В этом направлении был получен ряд важных результатов [2 – 6].

Здесь возвратимся к работе [3], посвященной построению закона распределения вероятностей переходов таблиц линейных аппроксимаций случайных подстановок. В этой работе была представлена оригинальная версия доказательства теоремы, сформулированной еще Люком О'Конором [7]. Сегодня стало понятно, что представленное доказательство теоремы в ней выполнено не совсем аккуратно. В работе не хватает обоснований некоторых важных моментов, и явно не выдерживает критики Утверждение 2, точнее, его доказательство нельзя считать правильным (убедительным). Учитывая теоретическую важность результата, которому посвящена работа, здесь предлагается уточненная версия доказательства теоремы и дополнительные подходы к интерпретации ее результатов. Для уменьшения неоднозначности трактовок и повышения прозрачности доказательств излагаемого материала здесь вводятся и доказываются дополнительные утверждения, устраняющие имеющиеся в работе [3] недоработки и пробелы.

Уточненное доказательство теоремы

Следуя [3, 7], напомним обозначения, использованные в этих работах и саму теорему.

Пусть $\pi : Z_2^n \to Z_2^n$ – биективное n-битное отображение и пусть S_2^n будет множеством всех таких отображений. Для n-битного вектора $X \in Z_2^n$ пусть X_i обозначает i-й бит вектора X. Линейная аппроксимационная таблица для подстановки π обозначается LAT_π и является таблицей размера $2^n \times 2^n$ с элементами $LAT_\pi(\alpha, \beta)$, определяемыми соотношением

$$LAT_\pi(\alpha,\beta) \overset{def}{=} \overset{def}{=} \#\left\{ X / X \in Z_2^n,\ \bigoplus_{i=1}^{n} X[i]\cdot\alpha[i] = \bigoplus_{i=1}^{n} \pi\big(X[i]\big)\cdot\beta[i] \right\}, \quad (1)$$

где $\alpha, \beta \in Z_2^n$ и $'\cdot'$ обозначает операцию скалярного произведения.

В соответствии с приведенным определением $LAT_\pi(\alpha, \beta)$ представляет собой число равенств четности между линейной комбинацией входных битов (определяемых маской α по входу в LAT_π подстановки по строкам) и линейной комбинацией выходных битов (определяемых маской β по входу в таблицу LAT_π подстановки по столбцам).

Теорема, о которой идет речь, сформулирована в [3, 7] следующим образом.

Теорема 1: *Пусть $\lambda(\alpha, \beta)$ будет случайным числом, соответствующим значению линейной аппроксимационной таблицы подстановки $LAT_\pi(\alpha, \beta)$, когда подстановка π выбрана равновероятно из множества S_2^n и маски α, β ненулевые. Тогда $\lambda(\alpha, \beta)$ для целых значений k, $0 \le k \le 2^{n-1}$ принимает только четные значения и вероятность, что*

$\lambda(\alpha,\beta)=2k$
определяется выражением

$$\Pr\left(\lambda(\alpha,\beta)=2k\right)=\frac{\left(2^{n-1}!\right)^2}{2^n!}\cdot\begin{pmatrix}2^{n-1}\\ k\end{pmatrix}^2. \qquad (2)$$

Далее излагается новый (уточненный) вариант ее доказательства.

Д о к а з а т е л ь с т в о. Нас интересует число подстановок из общего их числа $2^n!$, ячейки таблиц $LAT_\pi(\alpha,\beta)$ которых для заданного значения входа в таблицы по строкам α и заданного значения входа в таблицы по столбцам β (заданного сочетания пары входов в LAT_π) имеют заполнением (значением) число $2k$.

По определению, если подстановка π имеет значение ячейки таблицы $LAT_\pi(\alpha,\beta)$ равное $\lambda(\alpha,\beta)=2k$, то это означает, что число входов в подстановку X из общего их числа 2^n, прошедших при построении таблицы маску α с признаками чет и нечет (имеющих результатом скалярного произведения $\alpha\cdot X$ нуль или единицу), совпадающих с числом соответствующих выходов подстановки $\pi(X)$, прошедших маску β с признаками чет и нечет (имеющих результатом скалярного произведения $\beta\cdot\pi(X)$ нуль или единицу), равно $2k$, т.е. число входов и выходов, удовлетворяющих равенству четности $\alpha\cdot X=\beta\cdot\pi(X)$ равно $2k$.

Итак, интересующее нас событие связано с совпадением признаков чет или нечет для входных текстов подстановки и ее выходных текстов, прошедших соответствующие маски.

В дальнейшем нам потребуется несколько дополнительных утверждений.

Утверждение 1. *Для любого сочетания масок входа α и масок выхода β ровно половина* (2^{n-1}) *из общего числа скалярных произведений для всего множества входов в подстановку (как и для всего множества выходов подстановки) принимают значения "чет"* (0), *а остальные* 2^{n-1} *скалярных произведений принимают значения "нечет"* (1).

Предварительно докажем еще одно утверждение

Утверждение 2

Сумма по модулю 2 *двоичных символов полного набора из* 2^l *l-битных векторов на всем наборе этих векторов в половине случае равна нулю, а в другой половине случаев равна* 1.

Д о к а з а т е л ь с т в о. Число единиц в векторах полного набора из 2^l двоичных векторов меняется от 0 до l, причем число векторов без единиц $C_l^0=1$, число векторов с одной единицей $C_l^1=l$, с двумя единицами $C_l^2=\frac{l(l-1)}{2}$, …, число векторов с k единицами C_l^k, . . ., число векторов с l единицами будет $C_l^l=1$. Но векторы с нечетным числом единиц при сложении битов по модулю 2 (при вычислении скалярных произведений) будут давать результатом единицу, а векторы с четным числом единиц при сложении битов по модулю 2 будут давать результатом нуль.

Тогда на полном множестве из 2^l векторов число результатов "чет" (0) и "нечет" (1) для $l=2s$ (четного) будет определяться выражением $\sum_{k=0}^{s}C_l^{2k}$ для векторов с четным числом единиц, либо соответственно $\sum_{k=0}^{s}C_l^{2k-1}$ для векторов с нечетным числом единиц, а для $l=2s+1$` (нечетного) будет соответственно определяться выражением $\sum_{k=0}^{s}C_l^{2k}$ для векторов

с четным числом единиц и $\sum_{k=0}^{s} C_l^{2k+1}$ для векторов с нечетным числом единиц. Нетрудно убедиться (непосредственной проверкой), что $\sum_{k=0}^{s} C_l^{2k} = \sum_{k=0}^{s} C_l^{2k-1} = 2^{l-1}$ для $l = 2s$ (четного), как и то, что $\sum_{k=0}^{s} C_l^{2k} = \sum_{k=0}^{s} C_l^{2k+1} = 2^{l-1}$ для $l = 2s+1$` (нечетного). Тем самым доказана справедливость утверждения 2.

Вернемся к Утверждению 1.

Д о к а з а т е л ь с т в о. Будем интерпретировать маски входа в подстановку и маски выхода подстановки как *l*-битные и *m*-битные единичные вектора, наложенные на *n*-битные блоки входов и выходов в подстановку. Тогда скалярные произведения $\alpha \cdot X$ для масок, содержащих *l* единиц, (также как и скалярные произведения $\beta \cdot \pi(X)$ для масок, содержащих *m* единиц), наложенных на полный набор *n*-битных блоков входов и выходов) независимо от расположения единичных символов маски будут включать полные наборы из *l*-битных (*m*-битных) двоичных последовательностей, которые повторяются на полном наборе *n*-битных векторов $2^n / 2^l = 2^{n-l}$ ($2^n / 2^m = 2^{n-m}$) раз.

Учитывая, что в соответствии с Утверждением 2 сумма по модулю 2 двоичных символов для полного набора из 2^l *l*-битных векторов на всем наборе этих векторов в половине случаев равна нулю, а в другой половине случаев равна 1, т.е. имеем 2^{l-1} нулей и столько же единиц, приходим к выводу, что для любой *l*-битной маски входа на всем множестве *n*-битных векторов $2^{l-1} \times 2^{n-l} = 2^{n-1}$ скалярных произведений будут давать результатом нуль и столько же 2^{n-1} скалярных произведений будут давать результатом единицу. Аналогичный результат следует и для скалярных произведений с *m*-битными масками выхода. Тем самым доказано утверждение 1.

Таким образом, в соответствии с доказанным выше положением ровно половина (2^{n-1}) из общего числа скалярных произведений для всего множества входов в подстановку (как и для всего множества выходов из подстановки) принимают значения "чет" (нуль), а остальные 2^{n-1} скалярных произведений принимают значения "нечет" (единица). В результате различные подстановки при вычислении $LAT_\pi(\alpha, \beta)$ практически будут отличаться только распределением четных (нулевых) и нечетных (единичных) значений множеств скалярных произведений $\alpha \cdot X$ и $\beta \cdot \pi(X)$ из одного и того же набора, включающего 2^{n-1} четных и 2^{n-1} нечетных значений этих произведений.

Результирующее число "проходов" (выполнений равенства $\alpha \cdot X = \beta \cdot \pi(X)$) для любой пары входов в таблицу α и β будет определяться числом совпадений признаков чет или нечет в "списках" соответствующих наборов скалярных произведений для входов и выходов подстановки, причем одно и то же значение числа "проходов" будут иметь подстановки, которые отличаются переходами (перестановками), сохраняющими четность (нечетность) компонент, формирующих значение $\lambda(\alpha, \beta)$. Напомним, что параметр $\lambda(\alpha, \beta)$ фиксирует число случаев выполнения равенства $\alpha \cdot X = \beta \cdot \pi(X)$ для каждой пары α, β.

Нам потребуется еще два утверждения.

Утверждение 3. *Две последовательности, составленные из* 2^n *двоичных элементов, содержащие одинаковое число* 2^{n-1} *символов каждого типа, имеют только четное число совпадений* (*несовпадений*).

Д о к а з а т е л ь с т в о. Пусть $\xi = \{0,1\}^{2^n}$ и $\zeta = \{0,1\}^{2^n}$ – две случайно взятые 2^n-битные последовательности с одинаковым числом нулей и единиц. Доказательство будем вести

«от противного». Предположим, что последовательности ξ и ζ имеют нечетное число совпадений. Пусть для конкретности совпадающие символы имеют четное число нулей и нечетное число единиц. Тогда оставшиеся (несовпадающие) символы последовательностей для одной из них будут иметь нечетное число нулей и четное число единиц, в то время как вторая последовательность тогда должна иметь четное число нулей и нечетное число единиц (ведь они противоположные). В результате получается, что в одной последовательности должно быть четное число нулей и четное число единиц, в то время как во второй должно быть нечетное число нулей и нечетное число единиц, а это противоречит исходному предположению, что обе последовательности состоят из одинакового четного числа нулей и четного числа единиц. Следовательно, наше предположение о том, что последовательности ξ и ζ могут иметь нечетное число совпадений не верно.

Из доказанного следует справедливость утверждения первой части теоремы о том, что параметр $\lambda(\alpha,\beta)$ линейных аппроксимационных таблиц подстановок принимает только четные значения, так как наборы признаков чет и нечет в скалярных произведениях можно интерпретировать как соответствующие двоичные последовательности.

Справедливо также и такое утверждение.

Утверждение 4. *Для двух последовательностей, составленных из* 2^n *числа двоичных элементов и содержащих одинаковое число* 2^{n-1} *символов каждого типа, совпадающие* (*несовпадающие*) *последовательности символов содержат для каждой из последовательностей одинаковое число единиц и нулей.*

Д о к а з а т е л ь с т в о. Пусть $\xi = \{0,1\}^{2^n}$ и $\zeta = \{0,1\}^{2^n}$ – две случайно взятые 2^n-битные последовательности с одинаковым числом нулей и единиц и пусть $2k = s+t$, $s \neq t$ будет числом совпадающих символов. Пусть далее для конкретности совпадающие символы содержат *s* единиц и *t* нулей (в обеих частях равенства четности содержится по *s* единиц и *t* нулей). Но каждая последовательность состоит из одинакового числа 2^{n-1} символов каждого типа. Опять доказательство от противного. Пусть теперь, скажем, для первой последовательности ξ в числе несовпадающих символов оказывается $2^{n-1} - s$ единиц и $2^{n-1} - t$ нулей. Но это несовпадающие символы и, значит, вторая последовательность должна содержать $2^{n-1} - s$ нулей и $2^{n-1} - t$ единиц (противоположные символы).

В результате получается, что первая последовательность содержит, как и положено, $2^{n-1} - s + s = 2^{n-1}$ единиц и $2^{n-1} - t + t = 2^{n-1}$ нулей, а вторая – $2^{n-1} - s + t$ нулей и $2^{n-1} - t + s$ единиц, что при $s \neq t$ противоречит исходному положению, что каждая из последовательностей состоит из одинакового числа нулей и единиц, т.е. мы должны считать, что $s = t$.

Таким образом, среди $2k$ пар совпадений признаков ″чет″ ″нечет″ в равенствах $\lambda(\alpha,\beta)$ для каждой подстановки половина совпадений ″четы″ и еще половина ″нечеты″.

Перейдем теперь к определению значений интересующего нас числа $\lambda(\alpha,\beta)$ для подстановки степени 2^n.

Заметим сразу, что подстановки с одним и тем же значением параметра $\lambda(\alpha,\beta)$ отличаются друг от друга распределением в левой и правой части равенств $\alpha \cdot X = \beta \cdot \pi(X)$ четных и нечетных компонент.

Ранее уже отмечалось, что из 2^n скалярных произведений в правых частях равенств (как и в левых) половина произведений имеют признак ″чет″, а другая половина признак ″нечет″ (их 2^{n-1} каждого типа). Причем равенство сохраняется, если меняются местами между собой переходы (выходы) подстановки, которые дают результатами скалярные произведения с одинаковым признаком четности.

Для 2^{n-1} входов в подстановку с одинаковым признаком четности соответствующие 2^{n-1} выходов образуют подстановку степени 2^{n-1}, т.е. саму подстановку степени 2^n можно

рассматривать как две подстановки степени 2^{n-1} (ведь переходы случайной подстановки формируются независимо друг от друга). Это значит, что для каждого значения маски выходов β существует $2^{n-1}!$ различных подстановок, отличающихся между собой закреплением (расстановкой) выходов, формирующих признаки ″чет″ и столько же, т.е. $2^{n-1}!$ различных подстановок, отличающихся между собой закреплением выходов, формирующих признаки ″нечет″. Действительно, выходы подстановки, соответствующие входам с одинаковым признаком четности для фиксированной маски α, сохраняются по числу четов и нечетов для выходной маски β при их перестановке между собой.

В силу независимости распределения выходов подстановок по входам всего получается, что существует $(2^{n-1}!)^2$ вариантов различных подстановок, имеющих одно и то же распределение признаков ″чет″ и ″нечет″ (2^{n-1} скалярных произведений $\beta \cdot C$ каждого типа четности), отличающихся расстановкой выходов подстановки по своим входам.

Теперь каждая из таких подстановок реализует интересующее нас значение параметра $\lambda(\alpha, \beta) = 2k$ привязкой (совпадениями признаков ″чет″ и ″нечет″) входящих в $2k$ равенств $\alpha \cdot X = \beta \cdot \pi(X)$.

В соответствии с утверждением 4 таких совпадений будет в $2k$ переходах $\lambda(\alpha, \beta)$ по k каждого типа четности. Эти два набора по из одинакового числа переходов каждого типа (k равенств $\alpha \cdot X = \beta \cdot \pi(X)$ каждого из типов) могут быть осуществлены для каждого уникального набора из 2^{n-1} скалярных произведений $\beta \cdot C$ одного типа четности $C_{2^{n-1}}^{k}$ вариантами расстановки выходов подстановки по ее входам, а всего получается, что равенства обоих типов четности, образующих $2k$ интересующих нас переходов $\lambda(\alpha, \beta)$, могут быть реализованы $\left(C_{2^{n-1}}^{k}\right)^2$ различными способами ($C_{2^{n-1}}^{k} = \binom{2^{n-1}}{k}$ – биномиальный коэффициент).

В результате приходим к результату, который и утверждается в теореме.

Далее уже цитируются результаты работы [3].

В линейном криптоанализе интересуются входами (значениями) в линейную аппроксимационную таблицу подстановки порядка 2^n, которые после вычитания нормировочного значения 2^{n-1} являются отличием (смещением) действительного значения на число 2^{n-1} и, как отмечается в [3, 7, 8], представляют собой корреляцию линейных комбинаций входов и выходов подстановки. В результате приходят к так называемым линеаризованным таблицам подстановок, которые в [3, 7] обозначены $LAT_{\pi}^{*}(\alpha, \beta)$. Они определяются выражением

$$LAT_{\pi}^{*}(\alpha, \beta) = \left| LAT_{\pi}(\alpha, \beta) - 2^{n-1} \right|. \qquad (3)$$

В этом случае модуль в правой части записанного соотношения приводит к тому, что значение $\lambda(\alpha, \beta) = 2k$ для $2k' = 2k - 2^{n-1}$, $0 \le k \le 2^{n-1}$ может быть получено как при положительном смещении $k' = k - 2^{n-2}$ ($2^{n-2} \le k \le 2^{n-1}$), так и при отрицательном смещении $k' = k - 2^{n-2}$ ($0 \le k \le 2^{n-2}$), причем возможны и нулевые значения смещений ($\lambda^*(\alpha, \beta) = 0$), когда $k = 2^{n-2}$. Возвращаясь к старому обозначению переменной k (теперь уже для смещения), теорему 1 теперь можем переписать в виде теоремы 2.

Теорема 2: *Пусть $\lambda^*(\alpha, \beta)$ будет случайным числом, соответствующим значению линейной аппроксимационной таблицы подстановки $LAT_{\pi}^{*}(\alpha, \beta) = \left| LAT_{\pi}(\alpha, \beta) - 2^{n-1} \right|$, когда подстановка π выбрана равновероятно из множества S_2^n и маски α, β ненулевые. Тогда*

$\lambda^*(\alpha,\beta)$ для целых значений k, $|k| \le 2^{n-2}$ принимает только четные значения и вероятность, что $\lambda^*(\alpha,\beta) = 2k$ определяется выражением

$$\Pr\left(\lambda^*(\alpha,\beta) = 2k\right) = \frac{\left(2^{n-1}!\right)^2}{2^n!} \cdot \begin{pmatrix} 2^{n-1} \\ 2^{n-2} + |k| \end{pmatrix}^2. \qquad (4)$$

Определение наибольшего значения входа *в* LAT_π^*

Наша цель - определить наибольшее значение таблицы LAT_π^*.

Пусть теперь, как и в [3,7], $\lambda(\pi)$ будет наибольшим значением таблицы LAT_π^* взятым над всеми невырожденными α и β:

$$\lambda(\pi) \overset{def}{=} \max_{\alpha,\beta \ne 0} LAT_\pi^*(\alpha,\beta).$$

Здесь пропустим ряд соображений, представленных в работе [1], в частности проверку условий нормировок для полученного закона распределения вероятностей.

Обозначим $\mathrm{E}[\lambda(\pi,2k)]$ ожидаемое число ячеек таблицы LAT_π^*, имеющих значение $2k$.

В этом месте, повторяя рассуждения работы [3], сделаем переход от свойств ансамбля подстановок к свойствам отдельной подстановки. Считая, что полученный закон распределения вероятностей (4) справедлив для каждой отдельно взятой подстановки, рассмотрим его теперь применительно к множеству $(2^n-1)^2$ ячеек таблицы LAT_π^*, соответствующих ненулевым ее входам и выходам.

В результате можем получить выражение для вычисления $\mathrm{E}[\lambda^*(\pi,2k)]$ как простое умножение формулы (4) на общее число ячеек таблицы подстановки, исключая первую строку и первый столбец

$$\mathrm{E}[\lambda^*(\pi,2k)] = \frac{(2^n-1)^2 \cdot \left(2^{n-1}!\right)^2}{2^n!} \cdot \begin{pmatrix} 2^{n-1} \\ 2^{n-2} + |k| \end{pmatrix}^2, \qquad (5)$$

(для положительных и отрицательных значений смещения k результат будет один и тот же).

Выражение (5) имеет тенденцию быстро стремиться к нулю с ростом k. Среднему значению максимума таблицы LAT_π^* подстановки, как следует из сопоставления результатов вычислений с экспериментальными данными, будет соответствовать значение k^*, при котором получается наименьшее значение $\mathrm{E}[\lambda^*(\pi,2k)]$, превышающее или равное единице, т.е. для определения k^* необходимо найти округленное в сторону увеличения до ближайшего целого решение уравнения

$$\frac{(2^n-1)^2 \cdot \left(2^{n-1}!\right)^2}{2^n!} \cdot \begin{pmatrix} 2^{n-1} \\ 2^{n-2} - |k^*| \end{pmatrix}^2 = 1. \qquad (6)$$

Сравнение расчетных и экспериментальных результатов

Варианты решения уравнения (6) (переборным методом, который существенно упрощается при использовании результатов экспериментов), вместе с данными экспериментов заимствованные из [3] иллюстрирует табл. 1.

Как следует из результатов, представленных в табл. 1, найденные значения максимумов смещений таблиц линейных аппроксимаций случайных подстановок хорошо согласуются с данными, полученными экспериментальным путем.

Заметим, что в правой колонке таблицы представлены и результаты расчетов по предлагаемой в [9] упрощенной формуле

$$\mathrm{E}\left[\lambda(\pi,2k)\right]=\left(\frac{3}{2}\right)^{n}. \qquad (7)$$

Таблица 1

Сравнение теоретических
и экспериментальных результатов

n	$2k$*	$\mathrm{E}\left[\lambda(\pi,2k)\right]$	Эксперимент
4	4 6 8	3,89 1,118 0,017	5,498 $\left(\frac{3}{2}\right)^{8}=5{,}06$
6	12 14 16	9,013 1,7 0,239	14,48 $\left(\frac{3}{2}\right)^{6}=11{,}39$
8	32 34	2,12 0,7457	34,68 $\left(\frac{3}{2}\right)^{8}=25{,}62$
10	74 76	1,16 0,64	78,8 $\left(\frac{3}{2}\right)^{10}=57{,}66$
12	162 164	1,129 0,82	116,24 $\left(\frac{3}{2}\right)^{12}=129{,}74$
14	350 352	1,069 0,900	314 $\left(\frac{3}{2}\right)^{14}=291$
16	748 750	1,027 0,93	720 $\left(\frac{3}{2}\right)^{16}=657$

Если идти далее, то видно, что выражение (4) можно рассматривать как закон распределения вероятностей значений $\lambda^*(\alpha,\beta)$ таблицы $LAT_{\pi}^{*}(\alpha,\beta)$ отдельной взятой случайной подстановки π. В работе [3] показано, что для него выполняется условие нормировки ($\Pr\left(\left|\lambda(\pi)\right|\le 2^{n-1}\right)=1$).

Далее приводятся дополнительные материалы из работы [9].

Как показывает анализ, полученное расчетное соотношение хорошо работает для значений $n\le 32$. Для получения практических результатов при больших значениях n в работе [9] предложено воспользоваться теоремой 9 из работы [10], в которой обосновывается допустимость аппроксимациb соотношения (4) нормальным законом распределения вероятностей. Эта теорема здесь приводится под номером 3.

Теорема. 3: *Для случайной n-битовой подстановки, с* $n \ge 5$ *дисбаланс* $\mathrm{Imb}(v,u)$ *аппроксимации является случайным значением с распределением, которое может быть аппроксимировано в виде*

$$\Pr(\mathrm{Imb}(v, u) = z) \approx 2Z\left(\frac{z}{2^{(n-2)/2}}\right) \tag{8}$$

для z четного и нуль для z нечетного.

Если в (8) подставить $z = 2x$, то его можно переписать в виде

$$\Pr(\mathrm{Imb}(v, u) = 2x) \approx Z\left(\frac{x}{2^{(n-4)/2}}\right).$$

При выводе этого соотношения авторы пользуются леммой, повторяющей наш результат (2):

$$\Pr(\mathrm{Imb}(v, u) = 2x) = \frac{\binom{2^{n-1}}{2^{n-2}+x}^2}{\binom{2^n}{2^{n-1}}}, \tag{9}$$

т.е. в наших обозначениях дисбаланс $\mathrm{Imb}(v, u) = z$ при $z = 2k$ как раз соответствует $\lambda(\pi) = 2k$.

В табл. 2, заимствованной из [9], приводятся для сравнения результаты оценки максимальных значений смещений линейной таблицы случайной подстановки (половинных значений), полученных при использовании аппроксимирующего выражения (7), аппроксимации, использованной в выражении (8), и точного расчета по формуле (4).

Таблица 2

Сравнение результатов, полученных различными путями

n	Значение $x = k_L^*$ для нормального закона	Значение x для нашей аппроксимации	Расчетное значение k_L^*
8	16	12,81	16-17
16	334	328,42	374
20	1466	1662,62	1670
24	6342	8417	7302
28	27142	42611,346	31504
32	115080	215719	135649
128	62316975567822669939 $\approx 2^{67} \rightarrow \approx 2^{-120}$	17324119207854702237560 $\approx 2^{75} \rightarrow \approx 2^{-104}$	-

Как следует из представленных результатов расчетов, итоговая аппроксимация в виде нормального закона (8) оказывается достаточно хорошей. Видно также, что если аппроксимация нормальным законом дает оценки заниженные, то предлагаемая в [9] аппроксимация приводит к завышенным оценкам.

Для значений $n > 32$ остается ориентироваться на аппроксимирующие соотношения. Если полагать, что соотношение граничных (оценочных) значений, следующих из соответствующих аппроксимирующих выражений, с истинными значениями сохраняется и для значений $n > 32$, то в соответствии с идеологией, развиваемой в работе [9], можно перейти к расчетам ожидаемых значений максимумов линейных вероятностей для полномасштабных шифров.

Например, использование представленных аппроксимирующих соотношений для 128-битного шифра позволяет получить граничные значения для максимальной линейной вероятности (слева и справа) вида

$$2^{-120} \le LP_{\max}^{f} \le 2^{-104}.$$

Заметим, однако, что аппроксимация нормальным законом для значений $n \ge 32$ получается существенно точнее предлагаемой аппроксимации (судя по приведенным данным, отношение расчетного и аппроксимирующего выражений для аппроксимации в виде нормального закона принимает значения: при $n = 20 \rightarrow \frac{1670}{1466} = 1{,}139$ при $n = 24 \rightarrow \frac{7302}{6384} = 1{,}144$, при $n = 32 \rightarrow \frac{135649}{115080} = 1{,}178$). Если считать, что близкое к этому соотношение (отношение меньше двойки) сохранится и для больших значений n, то можно прийти к ожидаемому значению вероятности, более близкому к левой из двух приведенных границ, т.е. в качестве достаточно точной оценки линейной вероятности для интересующего нас битового размера входа в шифр $n = 128$ следует рассматривать значение $LP_{\max}^{f} \approx 2^{-119}$. Предложенная аппроксимация в этом случае дает ошибку в 2^{15} раза. Она оказывается хорошей для малых версий шифров. Работа по уточнению результатов продолжается.

Заключение

Приведенный закон распределения вероятностей для числа смещений таблиц линейных аппроксимаций случайных подстановок позволяет, как показано в работах [1, 9], ввести более строгие критерии отбора случайных подстановок, вплотную приближающих их свойства к свойствам шифрующих преобразований блочных симметричных шифров.

Выведенные расчетные соотношения для средних значений максимумов таблиц линейных аппроксимаций позволяют более обоснованно подойти к оценке показателей стойкости блочных симметричных шифров к атакам линейного криптоанализа.

Список литературы: 1. *Лисицкая, И. В.* Методология оценки стойкости блочных симметричных криптопреобразований на основе уменьшенных моделей : дис. … д-ра техн. наук : 05.13.05 / Лисицкая Ирина Викторовна. – 2012. – 293 с. 2. *Олейников, Р.В.* Дифференциальные свойства подстановок / Р.В. Олейников, О.И. Олешко, К.Е. Лисицкий, А.Д. Тевяшев // Прикладная радиоэлектроника. – 2010. – Т.9. – № 3. – С. 326-333. 3. *Долгов, В.И.* Свойства таблиц линейных аппроксимаций случайных подстановок / В.И. Долгов, И.В. Лисицкая, О.И. Олешко // Прикладная радиоэлектроника. – Харьков : ХНУРЭ, 2010. – Т. 9, №3. – С. 334-340. 4. *Долгов, В.И.* Случайные подстановки в криптографии / В.И. Долгов, И.В. Лисицкая, К.Е. Лисицкий // Радіоелектронні та комп'ютерні системи. – 2010. – № 5 (46). – С. 79-85. 5. *Лисицкая, И.В.* Оценка числа случайных подстановок с заданным распределением парных разностей XOR таблиц и смещений таблиц линейных аппроксимаций / И.В. Лисицкая, А.В. Широков, Е.Д. Мельничук, К.Е. Лисицкий // Прикладная радиоэлектроника. – Харьков : ХНУРЭ. – 2010. – Т. 9, № 3. – С. 341-345. 6. *Олейников, Р. В., Лисицкий, К. Е.* Исследование дифференциальных свойств подстановок различных цикловых классов // Двенадцатая Междунар. науч.-практ. конф. "Безопасность информации в информационно-телекоммуникационных системах ", 19-22 мая 2009 г., Тезисы докладов. – К. : ЧП "ЕКМО", НИЦ "ТЕЗИС" НТУУ "КПИ", 2009. – С. 24-25. 7. *Luke O'Connor*. Properties of Linear Approximation Tables. Email: oconnor@dsts. Edu. au, 1995. 8. *Luke O'Connor*. On Linear Approximation Tables and Ciphers secure against Linear Cryptanalysis. Email: oconnor@dsts. Edu. au, 1995. 9. *Долгов, В. И.* Методология оценки стойкости блочных симметричных шифров к атакам дифференциального и линейного криптоанализа : монография / В.И. Долгов, И.В. Лисицкая. – Харьков : Форт, 2013. – 420 с. 10. *Joan Daemen, Vincent Rijmen* Probability distributions of Correlation and Differentials in Block Ciphers / Joan Daemen, Vincent Rijmen // April 13, 2006, pp. 1–38.

Харьковский национальный университет имени В.Н. Каразина

Поступила в редколлегию 07.04.2017

УДК 004.056.5

П.И. СТЕЦЕНКО, Г.З. ХАЛИМОВ, д-р техн. наук

ПРОБЛЕМА СОВЕРШЕНИЯ УСЛОВНЫХ ТРАНЗАКЦИЙ В ЗАКРЫТЫХ BLOCKCHAIN-СИСТЕМАХ

Достижение консенсуса в открытых Blockchain-системах (например, Bitcoin и Ethereum), где отсутствует доверие между пользователями и имеются злоумышленники, является актуальным [1, 2]. Достижение консенсуса в распределенной системе, включающей злоумышленника, будет невозможным при условии, что сообщения не могут быть доставлены в течение ограниченного времени [3]. Базовые механизмы достижения консенсуса никогда не достигают консенсуса в случае умышленных задержек распространения сообщений [4].

Использование технологии Blockchain в закрытой, контролируемой среде в последнее время становится востребованным среди многих компаний. В 2015 г. одиннадцать банков успешно сотрудничали в развертывании закрытого регистра Blockchain проекта Ethereum для обработки транзакций [5].

В данной работе рассмотрена проблема, которая лишает пользователя возможности совершать транзакции на основе текущего состояния регистра Blockchain, что может привести к потере цифровых активов или атакам двойной траты. Показано, что некоторые смарт-контракты также могут иметь данную проблему. Представленные в работе результаты описывают риск использования технологии Blockchain в закрытой, контролируемой среде без учета ее сложных конструктивных особенностей.

1. Анализ связанных работ

Сравнительный анализ механизмов достижения консенсуса на основе доказательства проделанной работы (PoW) и протоколов с византийской отказоустойчивостью (BFT) проводился в работах [6 – 8]. BFT обеспечивает более низкую задержку и более высокую пропускную способность при обслуживании транзакций, чем PoW-механизмы. Недостатками BFT-протокола достижения консенсуса являются ограничения масштабируемости и вследствие этого необходимость использования сегментирования.

Вероятностные гарантии PoW-механизма достижения консенсуса и гарантии детерминирования BFT-протоколов исследуются в работе [8]. Достижение консенсуса на основе PoW сопоставляется с BFT-протоколами по двум ключевым моментам – масштабируемости и производительности. В результате исследования установлено, что PoW-протоколы считаются масштабируемыми, но неэффективными; а BFT-протоколы, напротив, считаются эффективными, но не масштабируемыми.

Одним из проектов, позволяющих избежать рассматриваемую в данной работе проблему, является PeerCensus [9]. Этот алгоритм состоит из двух компонентов: 1) выполнение BFT-протокола поверх Bitcoin с простой системой голосования; 2) сведение к минимуму последствий атак Сибиллы во время таких голосований. Последний компонент мешает злоумышленнику создавать множественные идентификаторы, чтобы превзойти число голосов своими собственными голосами.

Определение 1. Атака Сибиллы – разновидность атаки двойной траты, в которой злоумышленник наполняет сеть подконтрольными ему узлами, вследствие чего остальные пользователи смогут подключиться только к блокам, созданным злоумышленником.

Используя такой подход, PeerCensus усиливает гарантии Bitcoin и немедленно устраняет разветвления, что позволяет избежать проблемы совершения условных транзакций.

Определение 2. Условная транзакция – это транзакция, которая должна выполняться только в текущем наблюдаемом фиксированном состоянии регистра Blockchain или в более позднем его состоянии.

Аномалия протокола Paxos, схожая с рассматриваемой в данной работе проблемой, анализировалась в работах [10]. Следует отметить, что аномалия протокола Paxos отличается от

проблемы совершения условных транзакций, так как может произойти по двум транзакциям, выпущенным одним и тем же пользователем. В Blockchain-системах для упорядочения транзакций используется метка времени. Другим отличием является то, что, если консенсус достигнут, индекс решения не может измениться, а проблема совершения условных транзакций строго обусловлена тем, что индекс определенной транзакции или порядок блока данной транзакции в регистре Blockchain может измениться. Однако для приложений, которым не нужны параллельные зависимые запросы, достаточно протокола Paxos [10].

2. Основные сведения

Большинство Blockchain-систем отслеживают транзакцию путем включения ее в блок. Это осуществляется с помощью процесса майнинга перед добавлением к цепочке существующих блоков, которая и является регистром Blockchain. Механизм достижения консенсуса гарантирует общий порядок этих блоков, так что цепочка блоков в конечном итоге не становится деревом. В процессе функционирования системы возможен вариант, когда несколько новых блоков будут одновременно присоединены к последнему блоку регистра – такой временный процесс известен как разветвление. Как только обнаруживается разветвление, это означает, что участники знают о двух альтернативных версиях регистра Blockchain (ветвях), и самая длинная ветвь принимается как действительная. Blockchain-системы обычно предполагают, что разветвления могут вырасти до некоторой ограниченной глубины, так как рост ветви требует решения сложной вычислительной проблемы, которая сводится к трате большого количества времени, в течение которого кто-то получит информацию о самой длинной ветви регистра Blockchain. Криптовалюта Bitcoin предполагает присоединение еще шести блоков к регистру после выпущенной транзакции, чтобы данная транзакция рассматривалась как принятая системой. Аналогично проект Ethereum подразумевает присоединение от пяти до одиннадцати блоков после выпущенной транзакции для принятия ее системой [2].

Однако консенсус не может быть достигнут, если нет верхнего предела времени для доставки сообщения и если какой-то участник может потерпеть неудачу [3]. Консенсус обычно выражается тремя свойствами:

1) соглашение – если два корректных участника принимают решение, то они принимают его на одном и том же блоке;

2) корректность – блок, на котором принималось решение, должен быть одним из блоков, которые были предложены участниками;

3) завершение – в конечном итоге правильный участник принимает решение.

Процесс принятия общего решения в протоколах Paxos и Raft разработан таким образом, чтобы гарантировать свойства соглашения и корректности при получении сообщений с задержкой [11, 12]. Это достигается за счет того, что алгоритм жертвует свойством завершения, чтобы гарантировать, что только корректные ответы, удовлетворяющие свойствам корректности и соглашения, могут быть возвращены. Эти алгоритмы консенсуса являются привлекательными, потому что, если через некоторое время сеть стабилизируется и сообщения будут доставляться в ограниченное время, то консенсус будет достигнут [4].

В качестве базовой системы для проведения анализа выбран проект Ethereum, так как это основная Blockchain-система, позволяющая развертывание закрытых регистров.

2.1. Blockchain системы

Регистр Blockchain можно рассматривать как реплицируемую машину состояний, в которой обратная связь между блоками является указателем от текущего состояния к его предыдущему состоянию (рис. 1) [13].

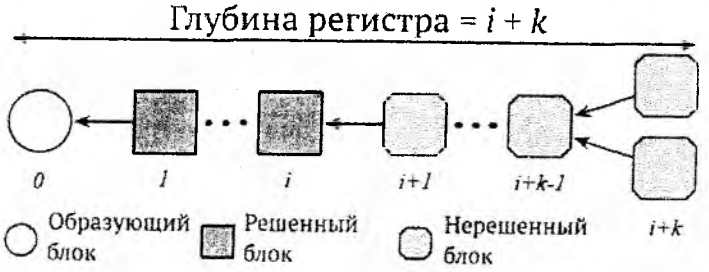


Рис. 1. Структура регистра Blockchain

Структура регистра Blockchain начинается с образующего блока с индексом 0 и связывает последовательно блоки в порядке, обратном их индексу; новый решенный блок имеет индекс $i > 0$, когда глубина регистра достигает $i + k$. Регистр, имеющий глубину 0, – это образующий блок.

Консенсус необходим для того, чтобы полностью упорядочить блоки, сохранив структуру цепочки блоков. В традиционных Blockchain-системах применяется PoW-механизм достижения консенсуса, требующий затрат на решение сложной вычислительной задачи [14]. Данный метод используется для того, чтобы достичь консенсуса, несмотря на произвольные неудачи, в том числе злонамеренное поведение пользователя. Специализированные одноранговые узлы (майнеры) должны доказуемо решить криптографическую HashCash задачу, прежде чем новый блок может быть добавлен к регистру Blockchain [15].

PoW-механизм достижения консенсуса находится в основе децентрализованной криптовалюты Bitcoin [1]. Примечательно, что PoW-консенсус не гарантирует достижения консенсуса детерминировано. Вместо этого он гарантирует, что консенсус будет достигнут с некоторой вероятностью, близкой к 1. Сложность решения криптографических головоломок, используемых в криптовалюте Bitcoin, приводит к присоединению нового блока к регистру Blockchain каждые 10 минут. Преимуществом этого длительного периода является то, что разветвления на регистре Blockchain из-за одновременного присоединения новых блоков возникают достаточно редко и Bitcoin решает эти разветвления, выбирая самую длинную ветвь и отбрасывая другую (другие). Проект Ethereum – это современная криптовалютная платформа с открытым исходным кодом, которая также опирается на PoW-консенсус [16]. В противоположность протоколу достижения консенсуса Bitcoin Ethereum генерирует один блок каждые 12-15 секунд. В то время как это увеличивает пропускную способность (количество транзакций в секунду), это также способствует увеличению количества переходных разветвлений регистра, так как вероятность одновременного присоединения майнерами новых блоков к регистру существенно возрастает. Чтобы избежать частой напрасной траты ресурсов на осуществление процесса майнинга, для разрешения разветвлений Ethereum использует протокол GHOST (Greedy Heaviest Observed Subtree), который не обязательно отбрасывает все блоки, не принадлежащие основной ветви. Проект Ethereum предлагает Тьюринг-полный язык программирования, который может быть использован для написания смарт-контрактов, определяющих новые правила владения активами [16].

2.2. Свойство завершения механизма достижения консенсуса

Путем ослабления свойства соглашения механизма достижения консенсуса Blockchain-системы могут гарантировать завершение достижения консенсуса детерминированно. В контексте Blockchain завершение механизма достижения консенсуса указывает на то, что блок был решен для следующего доступного индекса.

Пусть все транзакции решенного блока совершены. Здесь используется термин "совершенный", а не "подтвержденный", так как, в Blockchain терминологии транзакция предполагается "подтвержденной" иногда, когда только над её блоком осуществлен процесс майнинга, а иногда, когда над $k + 1$ блоками осуществлен процесс майнинга (собственно блок транзакции и k последующих блоков).

Такое решение на основе включения блока в цепочку необходимо для обменных платформ криптовалют. Например, чтобы определить, что монеты того или иного вида, которые были недавно созданы (не добавлены посредством майнинга) в пределах этого блока, могут быть конвертированы в монеты криптовалют различного типа или в декретные валюты (например, EUR, USD). В частности, обзор того, что над блоком был осуществлен процесс майнинга и этот блок добавлен к цепочке блоков, не является достаточным условием для гарантирования решения: этот блок может быть частью одной переходной ветви без достигнутого (пока что) консенсуса ни по одной из этих переходных ветвей.

На рис. 1 представлено прекращение механизма достижения консенсуса на индексе i регистра Blockchain. Стрелка, направленная слева направо, указывает на то, что блок содержит хэш своего блока-предшественника, который расположен сразу с левой стороны от него. Блоки, над которыми только что осуществлен процесс майнинга, добавляются в правый конец регистра Blockchain, что может привести к кратковременному разветвлению, в случае если над несколькими блоками, относящимися к одному и тому же предшественнику, процесс майнинга совершен одновременно. Разветвления являются лишь переходными и их решение зависит от используемой Blockchain-системы. Механизм достижения консенсуса для индекса i заканчивается, когда участники принимают решение о присвоении новому блоку индекса i. Решение на блоке с индексом i имеет место для всех $i > 0$, когда глубина регистра Blockchain достигает $i + k$, где $k \geq 0$ является константой, зависящей от используемой Blockchain-системы.

Различные Blockchain-системы могут принимать различные значения k, чтобы определять завершение механизма достижения консенсуса. В Bitcoin (btc), $k_{btc} = 5$, что означает, что блок с индексом i решен (т.е., достижение консенсуса для индекса i завершается), когда над $k_{btc} + 1 = 6$ блоками с индексами i, ..., $i + 5$ был успешно проведен процесс майнинга. Как уже упоминалось ранее, в криптовалюте Bitcoin новый блок решается каждые 10 минут. Следовательно, время совершения транзакции в Bitcoin $(k_{btc} + 1) * 10мин = 1час$. В проекте Ethereum (eth), начиная с версии 1.3.5 Homestead, $k_{eth} = 11$, что означает, что блок с индексом i решен (т.е., достижение консенсуса в отношении индекса i завершается), когда глубина регистра Blockchain достигает $i + 11$. Поэтому время для совершения транзакций в Ethereum равняется $(k_{btc} + 1) * 15сек = 180сек$. Следует обратить внимание, что некоторые обменные платформы криптовалют принимают разные значения k для регулирования вероятности соглашения. Например, обменная платформа QuadrigaCX ждет, пока над $k'_{btc} + 1 = 4$ блоками будет осуществлен процесс майнинга в регистре Blockchain криптовалюты Bitcoin, в то время как для регистра Ethereum данная платформа ждет осуществление процесса майнинга над $k'_{eth} + 1 = 12$ блоками [17].

3. Проблема совершения условных транзакций в закрытых Blockchain-системах

В данной работе рассмотрена проблема совершения условных транзакций, которая затрагивает ведущие Blockchain-системы, чей протокол достижения консенсуса не является гарантированно детерминированным.

Причины возникновения проблемы совершения условных транзакций. Данная проблема связана с асинхронностью сети, в которой задержки сообщения не могут быть ограничены, и со свойством завершения механизма достижения консенсуса.

Пусть два майнера осуществляют процесс майнинга на общей цепи блоков, начиная с общего образующего блока. Достаточно длительная задержка в сообщениях между ними может привести к тому, что майнеры будут, казалось бы, иметь согласие по состоянию регистра отдельно на разных ветвях цепи, содержащих более k блоков каждая, для любого k.

Данное обстоятельство является серьезной проблемой, так как может привести к простым атакам внутри любой закрытой, частной сети, в которой пользователи имеют стимул для максимизации своих прибылей – в виде монет, фондовых опционов или произвольной

собственности. Кроме того, этот сценарий является реалистичным в контексте частной сети, где сотрудники компании имеют прямой доступ к некоторым сетевым ресурсам. Когда сообщения в итоге доставляются, результаты несогласованности создают противоречия на регистре Blockchain.

Неподтвержденные транзакции. Проблема совершения условных транзакций, в которой транзакция t_i становится подтвержденной в рамках слота i с точки зрения некоторых узлов, представлена на рис. 2.

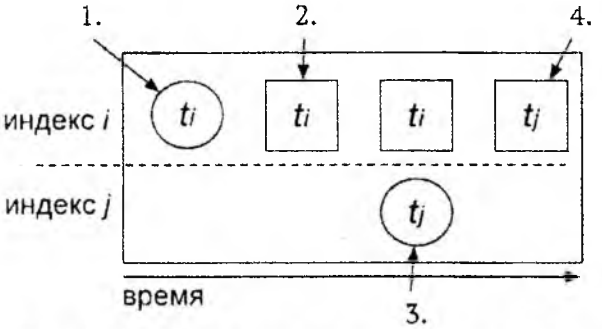


Рис. 2. Проблема совершения условных транзакций

На рис. 2:

1. Транзакция t_i предложена.
2. Транзакция t_i кажется подтвержденной.
3. Транзакция t_j предложена другим узлом.
4. Транзакция t_j подтверждена первой.

Основываясь на данном наблюдении, один пользователь предлагает новую транзакцию t_j, зная, что t_i успешно подтверждена. Можно представить простой сценарий, в котором "Боб передает некоторую сумму средств Алексу" (t_j) только тогда, когда ранее "Боб успешно получил деньги от Алисы" (t_i). Однако, как только эти узлы будут уведомлены о другой ветви подтвержденных транзакций, они примут решение реорганизовать ветвь для решения разветвления. Реорганизация удаляет подтвержденную транзакцию t_i из слота i. Позже транзакция t_j будет успешно подтверждена в слоте i.

Первый пользователь выпускает транзакцию t_i, над которой успешно проводится процесс майнинга и которая подтверждается. Далее второй пользователь выпускает транзакцию t_j, которая обуславливается подтверждением t_i. Следует отметить, что должно выполняться $j \geq i + k$ для того, чтобы t_i была подтверждена, перед тем как t_j будет выпущена. Однако транзакция t_j в итоге реорганизовывается и успешно подтверждается перед t_i, тем самым нарушая зависимость между транзакциями t_i и t_j.

Проблема, рассматриваемая в данной работе, связана с нарушением зависимости между транзакциями t_j и t_i: Транзакция t_j произошла и это значит, что Боб передал денежную сумму Алексу, однако транзакция t_i не произошла, а это значит, что Боб не получал деньги от Алисы. Следует обратить внимание, что в криптовалюте Bitcoin транзакция t_i будет отброшена, тогда как в Ethereum транзакции t_i в некоторых случаях могут быть подтверждены в слоте j.

Упрощение реализации атаки двойной траты. Одним из очевидных и серьезных последствий проблемы совершения условных транзакций является возможность злоумышленника провести атаку двойной траты: например, дважды конвертируя все свои монеты в товары. Сценарий состоит из выпуска злоумышленником первой транзакции t_1, которая конвертирует все его монеты на товары в блоке i, и старта майнинга блоков после того, как блок $i-1$ находится в изоляции сети. Как часть этого процесса майнинга злоумышленник осуществляет процесс майнинга над другой транзакцией t_2, при помощи которой он также конвертирует все свои монеты в товары. Затем злоумышленник ждет, пока глубина регистра Blockchain не достигнет $i+k$, после чего может забрать свои товары как результат транзакции t_1. Далее злоумышленник публикует свою более длинную цепочку, не содержащую транзакции t_1, таким образом, чтобы эта цепочка была принята остальной сетью. Транзакция t_2 под-

твердается в блоке j, и, после того как глубина цепочки блоков достигнет $j+k$, злоумышленник может забрать свои товары во второй раз. Следует обратить внимание, что даже если кто-то пытается позже вновь подтвердить транзакцию t_1, то сделка будет считаться недействительной, так как средств на балансе окажется недостаточно.

Отслеживание проблемы совершения условных транзакций. Другим серьезным аспектом рассматриваемой проблемы является незаметное выполнение. Точнее, данная проблема полагается на ошибочно подтвержденное состояние регистра Blockchain. После того, как ошибочно подтвержденное состояние становится неподтвержденным, нет никакого способа рассмотреть это проблемное состояние цепочки блоков и заметить проблему совершения условных транзакций. Несмотря на то, что можно наблюдать, как одноранговый узел осуществляет процесс майнинга над несколькими блоками подряд, отсутствует какой-либо способ отследить бенефициаров данной проблемы. Таким образом, это подталкивает пользователей закрытых регистров эффективно использовать рассматриваемую проблему для атаки на регистр Blockchain.

4. Смарт-контракты как способ решения проблемы совершения условных транзакций

Смарт-контракты являются основополагающим аспектом системы Ethereum. Смарт-контракты могут быть запрограммированы обеспечить выполнение кода при определенных условиях. Установлено, что предотвращение возникновения проблемы совершения условных транзакций тесно связано с корректным программированием смарт-контракта. Это означает, что если смарт-контракт был запрограммирован так, что соответствующим образом не проверено условие появления первой транзакции, то будет выполняться, действуя как нормальная, транзакция.

Вычисление на регистре и вне регистра Blockchain. Смарт-контракт conditionalPayment, написанный на языке Solidity, при котором не возникает проблемы совершения условных транзакций, представлен ниже.

```
1 contract conditionalPayment {
2
3 uint32 paid; // для отслеживания за выплаченной Алисой суммой при принятии решения о переводе средств Боба
4 mapping (address => uint256) public balances; // сопоставление адресов с соответствующим им балансом
5 address A = 0x57ec7927841e2d25aad5f335e3b701369b177392; // адрес счета Алисы
6 address B = 0x5ae58375c89896b09045de349289af9034902905; // адрес счета Боба
7
8 modifier onlyFrom(address _address) { // позволяет выполнение функций в зависимости от вызывающего
9 if (msg.sender != _address) throw;
10 _
11 }
12
13 function sendTo(address B, uint32 _amount) onlyFrom(A) { // Алиса отправляет деньги Бобу
14 if (balances[A] >= _amount) { // проверка достаточно ли доступных средств на счете
15 balances[A] -= _amount;
16 balances[B] += _amount;
17 paid = _amount; // сортировка выплачиваемой суммы
18 }
19 }
20
21 function sendIfReceived(address C, uint32 _amount) onlyFrom(B) { // Боб отправляет деньги Алексу
22 if (paid > _amount) { // только если предыдущий платеж был успешным
23 balances[B] -= _amount;
24 balances[C] += _amount;
25 } else {
```

```
26 throw; // отмена выполнения контракта
27 }
28 }
29 }
```

Ключевой особенностью является то, что функция sendIfReceived группирует два шага: проверяет, была ли сумма оплачена на строке 22, и платеж, полученный в результате этой успешной проверки на строках 23 и 24. Поскольку эти два шага выполняются на цепочке блоков, то для того, чтобы второе условие произошло, первое должно быть обязательно истинным.

Однако, если эти два шага были частью двух отдельных функций контракта, одна из них, проверяющая, что сумма была уплачена, и другая, которая будет производить платеж и будет использовать возвращенную стоимость первой, в данном случае может возникнуть проблема совершения условных транзакций.

Рассмотрим функцию checkPayment в смарт-контракте problematicConditionalPayment, представленном ниже.

```
1 contract problematicConditionalPayment {
2 ...
3 function checkPayment(address B, uint32 _amount) onlyFrom(B) constant returns
(bool result) {
4 if (paid > _amount) { // проверка, что Алиса заплатила
5 return true;
6 } else throw;
7 }
8
9 function sendIfReceived(address C, uint32 _amount) onlyFrom(B) { // Боб от-
правляет деньги Алексу
10 balances[B] -= _amount;
11 balances[C] += _amount;
12 }
13 }
```

Данная функция проверяет, что платеж от Алисы прошел корректно (строки 3 – 7), а другая функция sendIfReceived была изменена для выполнения платежа без каких-либо условий (строки 9 – 13).

Даже если Боб вызывает функцию checkPayment и замечает, что функция выполняется успешно до вызова sendIfReceived, может возникнуть рассматриваемая в работе проблема. Причина в том, что проверка сделана вне регистра Blockchain, и ничто не гарантирует, что платеж от Алисы не был реорганизован, в то время как Боб проверял результат в автономном режиме.

Смарт-контракт conditionalPayment имеет более высокие шансы не быть подверженным проблеме совершения условных транзакций, поскольку выполняет на регистре Blockchain как проверку, так и перевод средств после выполнения условия. Однако это не гарантирует, что такой смарт-контракт застрахован от рассматриваемой проблемы. Для формального обоснования данного утверждения необходимо дальнейшее исследование. Кроме того, как и транзакции, смарт-контракты должны быть включены в блок, над которым осуществляется процесс майнинга и который добавляется к цепочке блоков. Включение его в регистр Blockchain, даже с k последующими блоками, может также оставаться подверженным переупорядочению и приводить к другим типам проблем совершения условных транзакций.

Мультиподпись и вариант криптовалюты Bitcoin. Даже без Тьюринг-полного языка программирования в криптовалюте Bitcoin могут быть способы обойти проблему совершения условных транзакций. Идея заключается в том, чтобы изменить условную транзакцию на совместный платеж, который включает в себя как условную транзакцию, так и действие, обеспечивающее это условие. Идея включения транзакции и действия аналогична идее группировки в функции SendIfReceived смарт-контракта conditionalPayment, проверки и перевода средств, которые описаны ранее.

Совместный платеж будет представлять собой совершение платежа Алексу как Алисой, так и Бобом. Таким образом, платеж будет принимать два входа, принадлежащие разным

людям и дающие один выход. Поскольку монеты этих двух входов поступают с разных адресов, то для совместного платежа необходимы две разные подписи. Совместный платеж может быть достигнут при помощи транзакции с использованием мультиподписи в криптовалюте Bitcoin. Такая транзакция для выполнения требует две подписи от Алисы, Боба и стороннего арбитра. Если и Алиса, и Боб подписывают транзакцию, то она выполняется, и Алекс получает деньги. Однако, если Алиса или Боб отказываются подписывать транзакцию, то арбитр может помочь в разрешении транзакции путем своей подписи. Важно отметить, что семантика совместного платежа отличается от условной транзакции тем, что Боб не может ждать, пока получит деньги от Алисы, чтобы выбрать, что делать вне зависимости от платежа Алексу.

5. Обсуждение

Следует отметить, что в PoW-механизме достижения консенсуса имеется одна важная деталь. Если система достаточно велика и вычислительная мощность майнинга достаточно рассеяна среди достаточного количества пулов майнинга, то вероятность наличия пула майнинга, осуществляющего процесс майнинга быстрее других, может быть сделана сколь угодно малой. По этой причине рассматриваемая в работе проблема совершения условных транзакций имеет очень низкую вероятность появления в крупномасштабных свободно доступных Blockchain-системах, таких как Bitcoin. Для майнеров существуют стимулы не раскрывать блок, над которым они успешно осуществили процесс майнинга, чтобы другие тратили усилия на майнинг. Это, в свою очередь, позволяет им присоединить более длинную цепочку блоков, чем другим, а это может потенциально привести к проблеме совершения условных транзакций [18].

Атака 51 процента может привести к проблеме совершения условных транзакций в открытой Blockchain-системе.

Определение 3. Атака 51 процента – атака, в которой злоумышленник, контролирующий более половины вычислительной мощности майнинга в общедоступной сети, может быстрее осуществлять процесс майнинга над блоками по сравнению с другими майнерами.

Злоумышленник может выпустить транзакцию для конвертации некоторого количества монет Bitcoin, чтобы снять некоторую сумму денег. Как только над транзакцией осуществлен процесс майнинга в блоке с индексом i, злоумышленник может создать разветвление на цепочке блоков с индекса $i-1$, следовательно, изымая свою транзакцию новой серией блоков, которая в конечном итоге будет длиннее главной цепочки блоков. По мере того, как наиболее длинная ветвь становится принятой, транзакция злоумышленника не появляется в цепочке, так что, в конце концов, злоумышленник снимает некоторое количество денег, сохраняя свои монеты. Аналогичным методом, можно было бы легко переопределить блок, содержащий транзакцию от Алисы к Бобу. Возможность такой атаки возросла в контексте открытого регистра Bitcoin, поскольку было отмечено, что вычислительная мощность майнинга недостаточно рассеяна [18].

Предположение, что проблема совершения условных транзакций характерна исключительно для Pow-механизмов достижения консенсуса, поскольку существуют Blockchain-системы с другими механизмами достижения консенсуса, которые не подвержены подобной проблеме, является верным лишь частично. Это имеет место в случае некоторых Blockchain-систем, использующих механизм достижения консенсуса на основе подтверждения доли (proof-of-stake), таких как Tendermint, которые гарантируют согласие и достоверность консенсуса детерминировано. Однако проблема совершения условных транзакций применима и для Blockchain-систем, использующих PoS-механизм достижения консенсуса. Например, Casper представляет собой основанную на подтверждении доли альтернативу протоколу реорганизации GHOST, который используется в системе Ehtereum. Подтверждение доли не обязательно решает рассматриваемую проблему, поскольку даже Casper ставит в приоритет доступность по отношению к согласованности [19].

Другой проблемой является то, что реорганизация регистра может повлиять на исходный порядок транзакций. Это имеет значение при выполнении, в котором две транзакции нацелены на перевод $100 с одной учетной записи, первоначальный баланс которой составляет всего $100, поскольку может быть выполнена только транзакция, подтвержденная первой [20]. Однако проблема совершения условных транзакций является более общей по отношению к проблеме реорганизации регистра Blockchain. В частности, проблема совершения условных транзакций позволяет успешно выполнять конфликтующие транзакции и подтверждать их в двух разных состояниях цепочки блоков. Таким образом, решение проблемы, рассматриваемой в данной работе, позволит решить и проблему реорганизации регистра Blockchain.

Как известно, в асинхронной системе невозможно достичь консенсуса при наличии сбоев. Считается, что протокол гарантирует свойства или завершения, или соглашения детерминировано, но не оба эти свойства одновременно. В данной работе рассмотрено, что Blockchain консенсус завершается детерминировано на основе рекомендуемых 6-12 последующих добавленных блоков в Bitcoin [1] и Ethereum [16] соответственно, но иногда не достигает соглашения. Следует обратить внимание, что другие формализации также считают, что завершение достижения консенсуса в Bitcoin является детерминированным и что только его свойство безопасности является вероятностным [21]. Однако можно утверждать, что завершение не гарантируется детерминировано, а скорее вероятностно, и что можно увеличить вероятность достижения соглашения по консенсусу путем простого затягивания завершения. Характеристика консенсуса в Bitcoin-NG принимает это определение [22]. На практике, однако, Blockchain-приложения предполагают завершение достижения консенсуса, как описано в подразд. 2.2. Например, в Ethereum ожидание осуществления процесса майнинга над 12 последующими блоками объясняется тем, что, вероятно, такое ожидание будет достаточным для того, чтобы первый блок стал необратимым. Это может быть справедливо в крупномасштабной системе с открытым доступом, в которой вычислительная мощность майнинга достаточно рассеяна среди пулов майнинга, но его можно легко вернуть в контексте закрытой цепочки блоков.

Выводы

1. Проблема совершения условных транзакций предотвращает совершение пользователями наиболее популярных Blockchain-систем условных транзакций в случае разворачивания закрытого, частного регистра Blockchain. Данная проблема может привести к потере цифровых активов или атакам двойной траты. Возможным способом избегания проблемы совершения условных транзакций может быть создание смарт-контрактов вместо использования обычных транзакций, но это добавляет сложности и применимо только для систем с Тьюринг-полным языком программирования (например, Ethereum).

2. Если система достаточно велика и вычислительная мощность майнинга достаточно рассеяна среди достаточного количества пулов майнинга, то вероятность наличия пула майнинга, осуществляющего процесс майнинга быстрее других, может быть сделана сколь угодно малой. По этой причине рассматриваемая в работе проблема совершения условных транзакций имеет очень низкую вероятность появления в крупномасштабных свободно доступных Blockchain-системах.

3. В работе рассмотрены связи проблемы совершения условных транзакций с другими проблемами и показано, что она не ограничивается технологией Blockchain проекта Ethereum, но и потенциально может распространяться на частные, закрытые регистры Blockchain, использующие механизм достижения консенсуса proof-of-stake.

4. Blockchain-приложения требуют достижения консенсуса, чтобы быстро завершаться, в то время как базовые протоколы Blockchain гарантируют достижение консенсуса вероятностно. Это приводит к негативным результатам применительно к закрытым, частным распределенным регистрам.

5. В качестве направлений для последующих исследований можно выделить следующие: анализ проблемы совершения условных транзакций в Blockchain-системах, использующих PoS-механизм достижения консенсуса; изучение альтернативных Blockchain-систем, предоставляющих исключительно детерминированные гарантии для частной сети.

Список литературы: 1. *S. Nakamoto.* Bitcoin: a peer-to-peer electronic cash system, 2008. http://www.bitcoin.org. 2. *G. Wood.* Ethereum: A secure decentralised generalized transaction ledger final draft – under review, 2014. ttps://github.com/ethereum/wiki/wiki/White-Paper. 3. *M. J. Fischer, N. A. Lynch, and M. S. Paterson.* Impossibility of distributed consensus with one faulty process. J. ACM, 32(2):374–382, Apr. 1985. 4. *C. Dwork, N. Lynch, and L. Stockmeyer.* Consensus in the presence of partial synchrony. J. ACM, 35(2), Apr. 1988. 5. International Business Time Journal, 2015. http://www.ibtimes.co.uk/r3-connects-11-banksdistributed-ledger-using-ethereum-microsoft-azure-1539044. 6. *M. Castro and B. Liskov.* Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst., 20(4):398–461, Nov. 2002. 7. *K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer.* On scaling decentralized blockchains. In 3rd Workshop on Bitcoin Research (BITCOIN), Barbados, February 2016. 8. *M. Vukol'ıc.* The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In Proceedings of the IFIP WG 11.4Workshop on Open Research Problems in Network Security (iNetSec 2015), LNCS, 2016. 9. *C. Decker, J. Seidel, and R. Wattenhofer.* Bitcoin meets strong consistency. In Proceedings of the 17th International Conference on Distributed Computing and Networking (ICDCN), page 13, 2016. 10. *V. Gramoli, L. Bass, A. Fekete, and D. Sun.* Rollup: Nondisruptive rolling upgrade with fast consensus-based dynamic reconfigurations. IEEE Trans. on Parallel and Distributed Systems, 2015. 11. *L. Lamport.* The Part-Time parliament. ACM Transactions on Computer Systems, 16(2):133–169, May 1998. 12. *D. Ongaro and J. Ousterhout.* In search of an understandable consensus algorithm. In 2014 USENIX Annual Technical Conference (USENIX ATC 14), pages 305–319, Philadelphia, PA, 2014. USENIX Association. 13. *X. Xu, C. Pautasso, L. Zhu, V. Gramoli, S. Chen, A. Ponomarev, and A. B. Tran.* The blockchain as a software connector. In Proceedings of the 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), April 2016. 14. *C. Dwork and M. Naor.* Pricing via processing or combating junk mail. In Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '92, pages 139–147, 1993. 15. *A. Black.* Hashcash – a denial of service countermeasure. Technical report, Cypherspace, 2002. 16. *G.Wood.* Ethereum: A secure decentralized generalized transaction ledger, 2015. Yellow paper. 17. QuadrigaCX Bitcoin Trading Platform. https://www.quadrigacx.com/. 18. *I. Eyal and E. G. Sirer.* Majority is not enough: Bitcoin mining is vulnerable. In Financial Cryptography and Data Security – 18th International Conference, FC 2014, hrist hurch, Barbados, March 3-7, 2014, Revised Selected Papers, pages 436–454, 2014. 19. *Ethereum* Stack Exchange. https://ethereum. stackexchange.com/questions/332/what-is-the-difference-between-casper-and-tendermint /536 20. *Wood G.* Chain Reorganization Depth Expectations, 2015. https://blog.ethereum.org/2015/08/08/ chain-reorganisation-depth-expectations/. 21. *J. A. Garay, A. Kiayias, and N. Leonardos.* The Bitcoin backbone protocol: Analysis and applications. In Advances in Cryptology – EUROCRYPT 2015 – 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II, pages 281–310, 2015. 22. *I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse.* Bitcoin-NG: A scalable blockchain protocol. In 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2016.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 29.03.2017

УДК 621.3.06

М.Ю. РОДІНКО, Р.В. ОЛІЙНИКОВ, д-р техн. наук

ПОСТКВАНТОВИЙ МАЛОРЕСУРСНИЙ СИМЕТРИЧНИЙ БЛОКОВИЙ ШИФР«КИПАРИС»

Вступ

Поширення таких технологій як Інтернет речей (англ. Internet of Things), смарт-лічильники, системи безпеки для автомобілів та ін. висуває нові вимоги до криптографічних примітивів. Існуючі симетричні блокові шифри з високим рівнем криптографічної стійкості (такі, як AES [1] або «Калина» [2]) можуть не задовольняти вимогам компактної реалізації для застосування у пристроях з обмеженою кількістю споживання енергії. Для таких цілей існує цілий ряд малоресурсних блокових шифрів (PRESENT [3], XTEA [4], SPECK [5] та ін.). Однак більшість з них не підтримують довжину ключа, достатню для забезпечення високого рівня безпеки, тим більше, у постквантовий період. В Україні також не має власного постквантового малоресурсного блокового шифру. Таким чином, актуальною є задача створення шифру, який водночас має високий рівень криптографічної стійкості та високу швидкодію перетворень на різних платформах.

Серед підходів до побудування малоресурсних примітивів наряду з уже відомими SPN-структурою та мережею Фейстеля поширення набуває так зване ARX (Add-Rotate-XOR)-перетворення (SPECK, ChaCha [6]). Це обумовлено тим, що додавання за модулем та циклічний зсув є швидкими та простими у реалізації операціями.

У роботі пропонується постквантовий малоресурсний блоковий шифр «Кипарис», який заснований на мережі Фейстеля, а в якості циклової функції використовує ARX перетворення. Запропонований шифр має оптимізацію як під 32-бітові, так і під 64-бітові платформи та забезпечує високу швидкодію перетворень (у декілька разів вищу за AES).

1. Вимоги до перспективного малоресурсного блокового шифру

Виходячи з поставленої задачі, до перспективного малоресурсного симетричного блокового шифру були сформульовані наступні основні вимоги.

– Високий рівень криптографічної стійкості проти перебірних атак, що обумовлює необхідність підтримки ключів довжиною 256 та 512 біт.

– Застосування «сильної» схеми розгортання ключів (СРК) з метою захисту від криптоаналітичних атак на СРК.

– Висока швидкодія на різних програмно-апаратних платформах (в тому числі, мобільних).

– Компактна програмна реалізація як для стаціонарних систем (робочі станції, сервери), так і для мобільних платформ (смартфони, планшети).

– Наявність варіантів алгоритму, оптимізованих як під 64-бітові, так і під 32-бітові системи.

– Мінімальний об’єм пам’яті для швидкодіючої реалізації (компактний код та відсутність таблиць передобчислень).

– Постійний час шифрування блока на сучасних процесорах незалежно від параметрів, що обробляються, для захисту від атак по побічних каналах.

2. Опис алгоритму шифрування

2.1. Загальні параметри

Алгоритм шифрування «Кипарис» виконує перетворення блоків даних розміром l біт, із використанням ключа шифрування довжиною k біт, $l,k \in \{256,512\}$, $l = k$. Операції виконуються над s-бітними словами, $s \in \{32,64\}$. Основні загальні параметри шифру наведені в табл. 1.

Таблиця 1

Загальні параметри шифру «Кипарис»

	Кипарис-256	Кипарис-512
Розмір блока (l), біт	256	512
Довжина ключа (k), біт	256	512
Довжина слова (s), біт	32	64
Кількість ітерацій перетворення (t)	10	14

Варто відмітити, що «Кипарис-256» орієнтований на використання на 32-бітних платформах, «Кипарис-512» – на 64-бітних платформах, в т.ч. із вимогами до компактної реалізації та обмеженого енергоспоживання.

2.2. Процедура зашифрування/розшифрування

Загальна схема процедури зашифрування представлена на рис. 1.

На вхід процедури зашифрування подається блок відкритого тексту $P = (P_0, P_1, ..., P_7)$ та циклові ключі $RK^{(0)}, RK^{(1)}, \ldots, RK^{(t-1)}$. Кожний ключ $RK^{(i)} = (RK_0^{(i)}, RK_1^{(i)}, RK_2^{(i)}, RK_3^{(i)})$ складається з чотирьох s-бітних слів. Циклові ключі формуються за допомогою СРК на основі ключа шифрування $K = (K_0, K_1, ..., K_7)$.

Блок відкритого тексту P ділиться на два підблока: $L_0 = (P_0, P_1, P_2, P_3)$, $R_0 = (P_4, P_5, P_6, P_7)$. Вихід i-ї ітерації перетворення обчислюється як

$$L_i = R_{i-1} \oplus F\left(L_{i-1}, RK^{(i-1)}\right),$$

$$R_i = L_{i-1}.$$

Циклова функція F представляє собою додавання підблока L_{i-1} з ключем $RK^{(i-1)}$ за модулем 2 та двократне повторення функції $h(P_0', P_1', P_2', P_3')$, на вхід якої подається чотири s-бітних слова. Вихідне значення функції h обчислюється як

$$P_0' = ADD(P_0', P_1'),\ P_3' = XOR(P_3', P_0'),\ P_3' = ROTL(P_3', r1),$$
$$P_2' = ADD(P_2', P_3'),\ P_1' = XOR(P_1', P_2'),\ P_1' = ROTL(P_1', r2),$$
$$P_0' = ADD(P_0', P_1'),\ P_3' = XOR(P_3', P_0'),\ P_3' = ROTL(P_3', r3),$$
$$P_2' = ADD(P_2', P_3'),\ P_1' = XOR(P_1', P_2'),\ P_1' = ROTL(P_1', r4),$$

де $ADD(x, y)$ – додавання за модулем s двох s-бітних слів;

$XOR(x, y)$ – XOR двох s-бітних слів;

$ROTL(x, r)$ – циклічний зсув s-бітного слова вліво на r біт.

Значення циклічних зсувів (r_0, r_1, r_2, r_3) залежать від довжини блока і дорівнюють:

- для шифру «Кипарис-256» $(r_0, r_1, r_2, r_3) = (16, 12, 8, 7)$.
- для шифру «Кипарис-512» $(r_0, r_1, r_2, r_3) = (32, 24, 16, 15)$.

Процедура розшифрування є ідентичною до процедури зашифрування, лише циклові ключі подаються у зворотному порядку.

2.3. Схема розгортання ключів шифру «Кипарис»

Циклові ключі формуються за допомогою неін'єктивної СРК, в основі якої лежить структура СРК шифру «Калина». У [7] показано, що складність перебірних атак на неін'єктивні СРК у порівнянні з ін'єктивними схемами не знижується, при цьому

неін’єктивні схеми забезпечують додатковий захист від атак на реалізацію та інших криптоаналітичних атак.

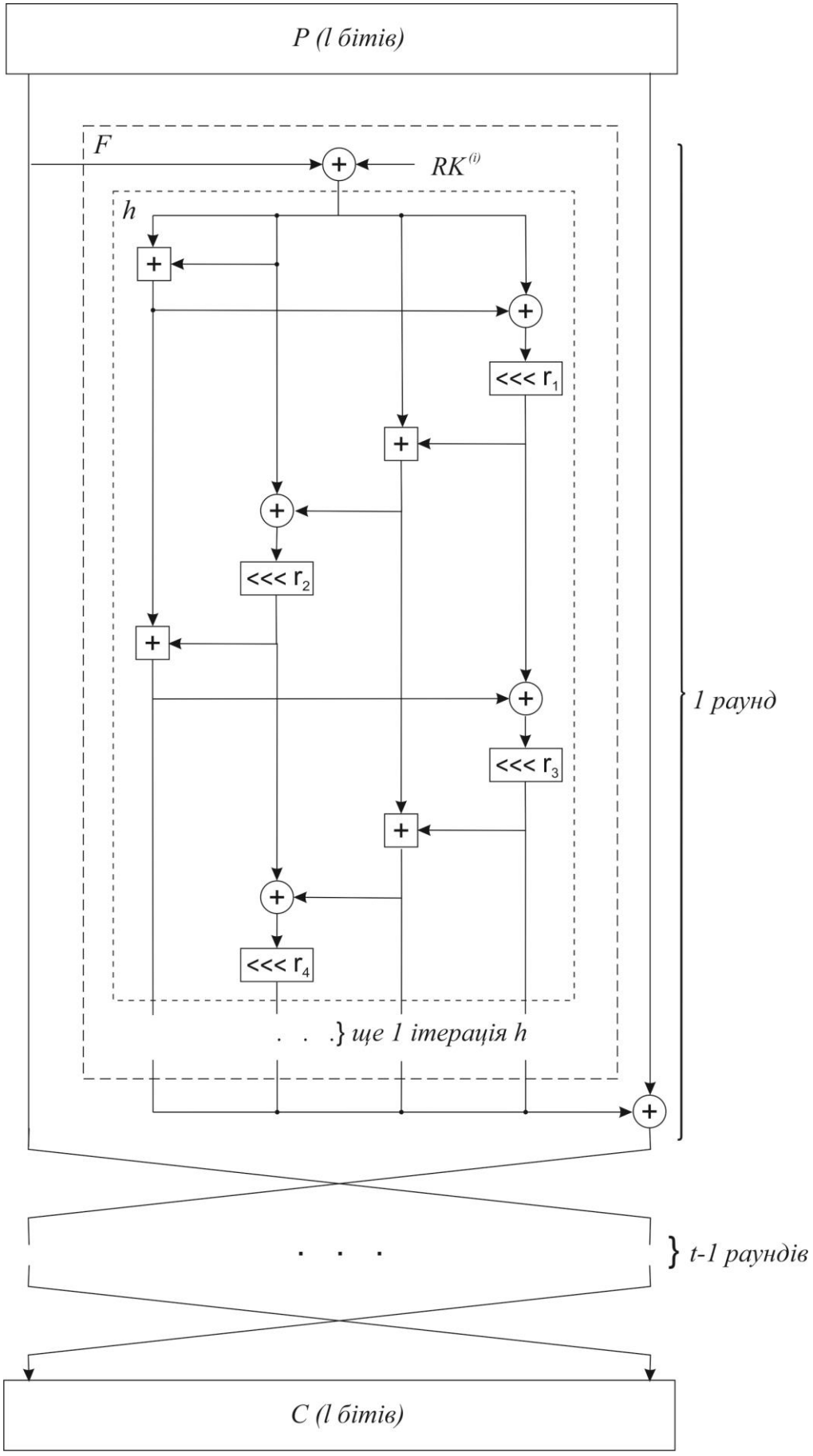


Рис. 1. Процедура зашифрування шифру «Кипарис»

При шифруванні використовується t циклових ключів довжиною $4\times s$ біт.

Спочатку формується проміжний ключ K_σ довжиною $4\times s$ біт. Формування ключа здійснюється із використанням ключа шифрування $K=(K_0,K_1,...,K_7)$ наступним чином:

$$K_l=(K_0,K_1,K_2,K_3), K_r=(K_4,K_5,K_6,K_7),$$
$$st=XOR(0x1,K_l), st=h(h(st)),$$
$$st=ADD(st,K_r), st=h(h(st)),$$
$$st=XOR(st,K_l), K_\sigma=st.$$

Циклові ключі формуються на основі ключа шифрування K, проміжного ключа K_σ та константи tmv = (0x000F000F, 0x000F000F, 0x000F000F, 0x000F000F) наступним чином:

$$st=(K_0,K_1,K_2,K_3), K_t=ADD(K_\sigma,tmv),$$
$$st=ADD(st,K_t), st=h(h(st)),$$
$$st=XOR(st,K_t), st=h(h(st)),$$
$$st=ADD(st,K_t), RK_{2i}=st,$$
$$tmv=ShiftLeft(tmv,0x1),$$
$$st=(K_4,K_5,K_6,K_7), K_t=ADD(K_\sigma,tmv),$$
$$st=ADD(st,K_t), st=h(h(st)),$$
$$st=XOR(st,K_t), st=h(h(st)),$$
$$st=ADD(st,K_t), RK_{2i+1}=st,$$

де функція *ShiftLeft* передбачає зсув кожного s-бітного слова вліво на один біт.

Після формування кожної пари парний/непарний ключ значення tmv та K модифікуються наступним чином:

$$tmv=ShiftLeft(tmv,1),$$
$$K=ROTLKey(K,1),$$

де $ROTLKey(K,1)$ – циклічний зсув масиву s-бітних слів ключа вліво.

4. Дослідження швидкодії шифру «Кипарис»

У ході досліджень на різних програмно-апаратних платформах була оцінена швидкодія алгоритмів «Кипарис-256» та «Кипарис-512» та порівняна зі швидкодією шифру AES.

Вимірювання швидкодії блокових шифрів здійснювалося на наступних платформах:

– Intel Core i3 / Windows 7 x32 з компілятором Visual C++ 2010;
– Intel Core i3 / Windows 7 x64 з компілятором Visual C++ 2010;
– Intel Core i5 / Linux (64 bit) з компілятором g++ версії 4.8;
– ARM Cortex-A7 / Android 4.2.2 Jelly Bean (32 bits).

Результати вимірювання швидкодії шифрів на різних платформах наведені в таблиці 2.

Таблиця 2

Швидкодія шифрів «Кипарис» та AES, Мбіт/с

Платформа	Кипарис-256	Кипарис-512	AES-256
Intel Core i3 / Windows 7 x32	1796,86	786,24	711,13
Intel Core i3 / Windows 7 x64	1878,5	2617,74	858,77
Intel Core i5 / Linux (64 bit)	3954,55	5395,81	1969,65

ARM Cortex-A7 / Android 4.2.2 (32 bit)	122	136	43

Як видно з табл. 2, блоковий шифр «Кипарис» за швидкістю перевершує алгоритм AES на всіх обраних платформах. На платформі x86 з 32-бітовою архітектурою шифр «Кипарис-256» у 2,5 рази швидший, ніж AES-256. На платформі x86 з 64-бітовою архітектурою шифр «Кипарис-512» приблизно у 3 рази швидший, ніж AES-256. На платформі ARM Cortex-A7 «Кипарис-256» та «Кипарис-512» приблизно у 3 рази швидші, ніж AES-256.

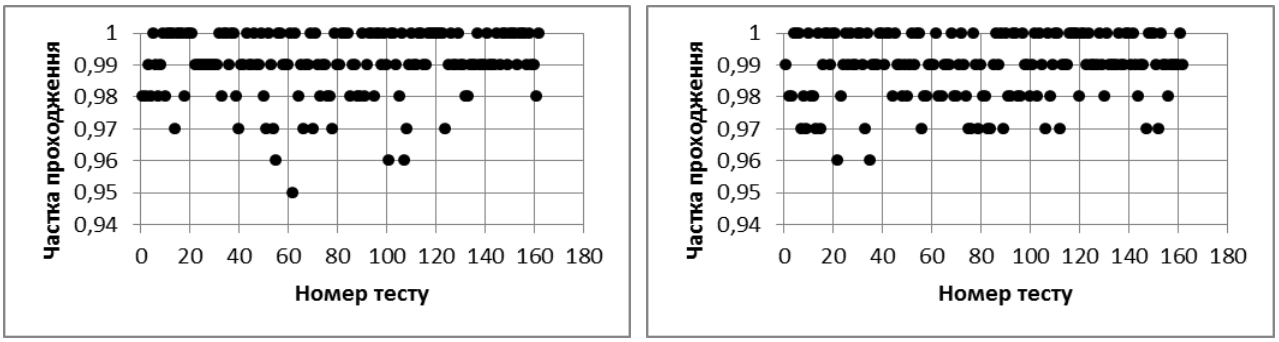
5. Основні властивості шифру

5.1. Статистичні властивості шифру «Кипарис»

Статистичні властивості шифруючого перетворення та СРК оцінювалися згідно з методикою NIST STS [8]. Для оцінки статистичних властивостей були обрані вхідні послідовності, що мають максимальну збитковість.

Для тестування шифруючого перетворення була обрана послідовність відкритих текстів $m_0 = 0$, $m_1 = 1$, …, $m_i = i$, $m_n = 100000$ (з максимальною збитковістю). Вихідна послідовність шифртекстів (отримана в режимі електронної кодової книги) тестувалася згідно з NIST STS.

Статистичні профілі вихідних послідовностей для розміру блока 256 та 512 біт наведені на рис. 2, *а* та *б* відповідно.

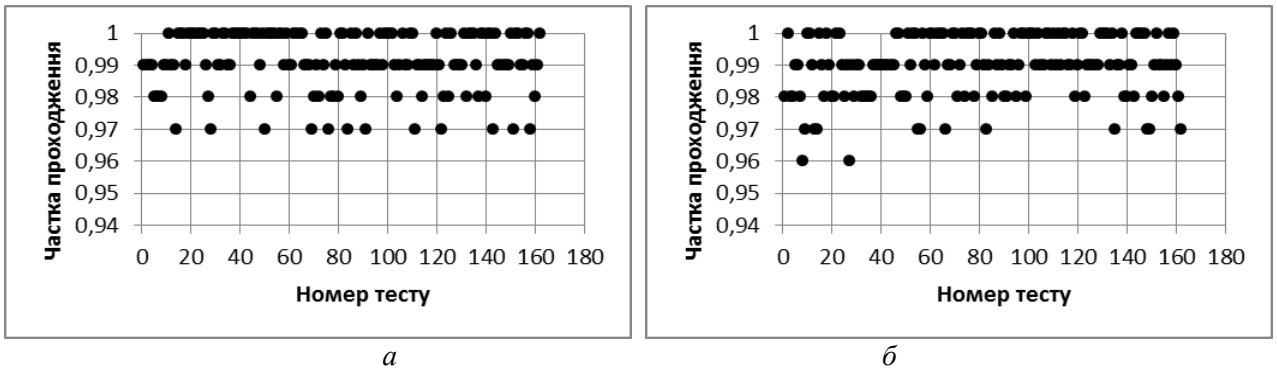


а *б*

Рис. 2. Статистичні профілі вихідних послідовностей для розміру блока 256 та 512 біт

Для тестування схеми розгортання ключів була обрана послідовність ключів шифрування $K_0 = 0$, $K_1 = 1$, …, $K_i = i$, $K_n = 100000$ (з максимальною збитковістю). Отримані циклові ключі формували вихідну послідовність, що тестувалася згідно з NIST STS.

Статистичні профілі вихідних послідовностей циклових ключів для довжини ключа 256 та 512 біт наведені на рис. 3, *а* та *б* відповідно.



а *б*

Рис. 3. Статистичні профілі вихідних послідовностей для розміру блока 256 та 512 біт

6.2. Лавинні показники шифру «Кипарис»

Лавинний ефект [9] є властивістю шифру, яка означає, що зміна малої кількості бітів у відкритому тексті призведе до «лавинної» зміни значень бітів шифртексту. Якщо блоковий шифр не володіє достатнім лавинним ефектом, криптоаналітик може зробити припущення щодо вхідної інформації, спираючись на вихідну інформацію, тому досягнення лавинного ефекту є важливою метою при розробці блокового шифру.

Вважається, що алгоритм задовольняє лавинному критерію, якщо зміна одного біта відкритого тексту призводить до зміни не менше половини бітів шифртексту.

Для оцінки лавинного ефекту шифру «Кипарис» були обчислені наступні показники:

1) мінімум математичного сподівання (МС) кількості вихідних бітів, що змінилися при зміні одного вхідного біта для N блоків даних;
2) максимум математичного сподівання кількості вихідних бітів, що змінилися при зміні одного вхідного біта для N блоків даних;
3) мінімум середньоквадратичного відхилення (СКВ) кількості вихідних бітів, що змінилися при зміні одного вхідного біта для N блоків даних;
4) максимум середньоквадратичного відхилення кількості вихідних бітів, що змінилися при зміні одного вхідного біта для N блоків даних.

У табл. 3, 4 наведені результати обчислення лавинних показників для шифрів «Кипарис-256» та «Кипарис-512» відповідно.

Таблиця 3

Лавинні показники шифру «Кипарис-256»

Кількість циклів шифрування	Показник	Значення	Кількість циклів шифрування	Показник	Значення
1	Мінімум МС	1	6	Мінімум МС	127,941
	Максимум МС	65,0254		Максимум МС	128,078
	Мінімум СКВ	0		Мінімум СКВ	63,2596
	Максимум СКВ	49,8347		Максимум СКВ	64,7305
2	Мінімум МС	62,3417	7	Мінімум МС	127,94
	Максимум МС	128,016		Максимум МС	128,066
	Мінімум СКВ	32,1742		Мінімум СКВ	63,1499
	Максимум СКВ	81,6093		Максимум СКВ	64,7323
3	Мінімум МС	125,375	8	Мінімум МС	127,927
	Максимум МС	128,06		Максимум МС	128,064
	Мінімум СКВ	63,3573		Мінімум СКВ	63,2817
	Максимум СКВ	82,1095		Максимум СКВ	64,7861
4	Мінімум МС	127,929	9	Мінімум МС	127,924
	Максимум МС	128,079		Максимум МС	128,072
	Мінімум СКВ	63,2875		Мінімум СКВ	63,2531
	Максимум СКВ	64,6699		Максимум СКВ	64,7662
5	Мінімум МС	127,926	10	Мінімум МС	127,941
	Максимум МС	128,09		Максимум МС	128,075
	Мінімум СКВ	63,2186		Мінімум СКВ	63,2797
	Максимум СКВ	64,8311		Максимум СКВ	64,778

Таблиця 4

Лавинні показники шифру «Кипарис-512»

Кількість циклів шифрування	Показник	Значення	Кількість циклів шифрування	Показник	Значення
1	Мінімум МС	1	8	Мінімум МС	255,889
	Максимум МС	161,034		Максимум МС	256,096
	Мінімум СКВ	0		Мінімум СКВ	126,625
	Максимум СКВ	417,279		Максимум СКВ	129,672
2	Мінімум МС	98,0896	9	Мінімум МС	255,9
	Максимум МС	256,052		Максимум МС	256,078
	Мінімум СКВ	63,6531		Мінімум СКВ	126,646

	Максимум СКВ	481,193		Максимум СКВ	129,51
3	Мінімум МС	225,032	10	Мінімум МС	255,911

Продовження табл. 4

Кількість циклів шифрування	Показник	Значення	Кількість циклів шифрування	Показник	Значення
3	Максимум МС	256,081	10	Максимум МС	256,109
	Мінімум СКВ	126,813		Мінімум СКВ	126,321
	Максимум СКВ	482,083		Максимум СКВ	129,6
4	Мінімум МС	255,911	11	Мінімум МС	255,912
	Максимум МС	256,084		Максимум МС	256,1
	Мінімум СКВ	126,4		Мінімум СКВ	126,691
	Максимум СКВ	129,707		Максимум СКВ	129,297
5	Мінімум МС	255,915	12	Мінімум МС	255,909
	Максимум МС	256,1		Максимум МС	256,138
	Мінімум СКВ	125,956		Мінімум СКВ	126,219
	Максимум СКВ	129,338		Максимум СКВ	129,697
6	Мінімум МС	255,862	13	Мінімум МС	255,895
	Максимум МС	256,096		Максимум МС	256,103
	Мінімум СКВ	126,708		Мінімум СКВ	126,53
	Максимум СКВ	129,394		Максимум СКВ	129,524
7	Мінімум МС	255,915	14	Мінімум МС	255,89
	Максимум МС	256,102		Максимум МС	256,108
	Мінімум СКВ	126,838		Мінімум СКВ	126,607
	Максимум СКВ	129,639		Максимум СКВ	129,597

Як видно з табл. 3, 4, «Кипарис-256» та «Кипарис-512» задовольняють лавинному критерію вже після четвертого циклу.

Висновки

З урахуванням сучасних вимог до симетричних примітивів був розроблений постквантовий малоресурсний блоковий шифр «Кипарис», що заснований на мережі Фейстеля. Циклова функція шифру представляє собою ARX-перетворення, схема розгортання циклових ключів неін'єктивна та використовує принципи побудови СРК шифру «Калина». Алгоритм підтримує довжину блока (ключа) 256 та 512 бітів, що дозволяє забезпечити необхідний рівень криптографічної стійкості. «Кипарис-256» орієнтований на використання на 32-бітних платформах, «Кипарис-512» – на 64-бітних платформах.

Дослідження статистичних властивостей показали, що шифр «Кипарис» та його СРК задовольняють вимогам зі статистичного тестування випадкових послідовностей NIST STS.

Дослідження лавинних показників показали, що шифр «Кипарис-256» (число циклів дорівнює 10) та шифр «Кипарис-512» (число циклів дорівнює 14) відповідають вимогам щодо лавинного ефекту починаючи з чотирьох циклів шифрування.

Вимірювання швидкодії виконувалося на платформі x86 (x86_64) під управлінням ОС Windows та Linux і платформі ARM-v7 під управлінням ОС Android. На різних програмно-апаратних платформах шифр продемонстрував наступні показники швидкодії:

- на платформі x86 з 32-бітною архітектурою «Кипарис-256» у 2,5 рази швидший, ніж AES-256;
- на платформі x86 з 64-бітною архітектурою «Кипарис-512» приблизно у 3 рази швидший, ніж AES-256;
- на платформі ARM-v7 з 32-бітною архітектурою «Кипарис-256» та «Кипарис-512» приблизно у 3 рази швидші за AES-256.

Таким чином, у порівнянні з шифрами «Калина» та AES, блоковий шифр «Кипарис» має наступні переваги:

- висока швидкодія перетворень незалежно від використовуваної платформи;

– наявність варіантів алгоритму, оптимізованих як під 64-бітові, так і під 32-бітові системи;
– компактна реалізація як для стаціонарних систем, так і для мобільних платформ;
– швидкодіюча реалізація не потребує використання таблиць передобчислень, що забезпечує використання мінімального об'єму пам'яті;
– можливість організації ефективних захищених високошвидкісних каналів зв'язку між мобільними системами та серверами, у тому числі тими, що використовують апаратні прискорювачі;
– постійний час шифрування блока на сучасних процесорах незалежно від параметрів, що обробляються.

Список літератури: 1. *Standard, Advanced Encryption*. Federal Information Processing Standards Publication 197 / FIPS PUB, 46-3. – 2001. – 51 p. 2. *ДСТУ* 7624:2014. *Інформаційні* технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення. – Введ. 01–07–2015. – К. : Мінекономрозвитку України, 2015. – 119 с. 3. *Bogdanov, A.* PRESENT: An Ultra-Lightweight Block Cipher / A. Bogdanov, L.R. Knudsen, G. Leander et al.: Springer Berlin Heidelberg, 2007. – pp. 450-466. 4. *Needham, Roger M.* Tea extensions] / Roger M. Needham, D. J. Wheeler // Report, Cambridge University, Cambridge, UK. – 1997 – 4 p. 5. *Beaulieu, Ray, et al.* The SIMON and SPECK lightweight block ciphers.Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE. IEEE, 2015. 6. *Bernstein, Daniel J.* ChaCha, a variant of Salsa20. Workshop Record of SASC. Vol. 8. 2008. 7. *Родінко, М.Ю.* Математична модель оцінки властивостей неін'єктивних схем розгортання ключів симетричних блокових шифрів / М.Ю. Родінко, Р.В. Олійников // Прикладна радіоелектроніка. – 2016. – Т. 15. – № 3. – С. 179-183. 8. *Rukhin, Andrew, et al.* A statistical test suite for random and pseudorandom number generators for cryptographic applications. Booz-Allen and Hamilton Inc Mclean Va, 2001. 9. *Feistel, Horst.* Cryptography and Computer Privacy // Scientific American, Vol. 228, No. 5, 1973.

Харківський національний університет
імені В.Н. Каразіна

Надійшла до редколегії 14.04.2017

УДК 004.056.55

Н.А. ПОЛУЯНЕНКО, А.В. ПОТИЙ, д-р техн. наук

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ГРАФИЧЕСКИХ ПРОЦЕССОРАХ ДЛЯ ГЕНЕРАТОРОВ ПОТОКОВОГО ШИФРОВАНИЯ

Введение

Практически все программно-аппаратные методы защиты информации неразрывно связаны с криптографией [1]. Криптография существует уже много веков, однако в том виде, в котором мы ее знаем и используем сейчас, она насчитывает несколько десятилетий. Можно сказать, что основным (хотя и не единственным) направлением развития современной криптографии является создание стойких алгоритмов шифрования, которое неразрывно связано с вычислительными мощностями.

В последние несколько лет перспективным направлением увеличения вычислительной мощности компьютерных систем стал перенос вычислений с центрального процессорного устройства (CPU – от англ. central processing unit) на графические процессоры (GPU – от англ. Graphics processing units). Эту концепцию поддержали основные производители графических ускорителей, в частности компания NVIDIA.

В настоящий момент процессоры, используемые в компьютерах общего назначения, подошли к пределу тактовой частоты – около 3 ГГц. Поэтому дальнейший рост производительности осуществляется путем распараллеливания [2]. Средства параллельного выполнения процессоров от Intel и AMD включают в себя наборы команд для выполнения векторных операций, а также реализацию на одном кристалле нескольких процессорных ядер – реальных или виртуальных (с использованием HyperThreading). Однако возможности по распараллеливанию традиционных процессоров ограничены. Этому мешают их изначально (начиная с 8086) последовательная архитектура и достаточно сложный набор команд, требующий, для быстрого выполнения реализации, достаточно сложных элементов процессора – конвейера, системы прогнозирования переходов, кешей и т. п.

Архитектура видеокарт NVIDIA, позволяющая использовать их для неграфических вычислений, получила название CUDA (от англ. Common Unified Device Architecture – общая унифицированная архитектура устройства), а разработанная для нее спецификация [3] облегчила процесс создания алгоритмов, работающих на GPU. Технология CUDA была впервые представлена компанией NVIDIA в 2007 г. [4]. CUDA может использоваться на GPU производства NVIDIA, таких как GeForce, Quadro, Tesla. Последняя линейка устройств (Tesla) вообще не имеют видеовыхода и предназначены исключительно для высокопроизводительных вычислений.

NVIDIA распространяет ряд примеров, демонстрирующих решение некоторых задач из различных областей математики, физики и информатики. Прирост производительности в этих задачах при переходе от CPU к GPU составляет от 10 до 100 раз.

Поскольку графическая карта – это отдельный вычислительный модуль с собственным вычислительным узлом и оперативной памятью, то в программе для CUDA код разделяется на две части: host – выполняется на CPU; device – выполняется на GPU.

Параллелизм в CUDA обеспечивается следующим образом: при запуске кода на выполнение создается не один, а группа потоков – рис. 1. Вся группа потоков, созданных при запуске кода, называется сеткой (grid). Элементами сетки являются блоки (block). Они формируют одномерную или двухмерную структуру. Каждый блок, в свою очередь, состоит из группы нитей (thread), объединенных в одномерный, двумерный или трехмерный массив. Число блоков в сетке и нитей в блоке задается при старте кода.

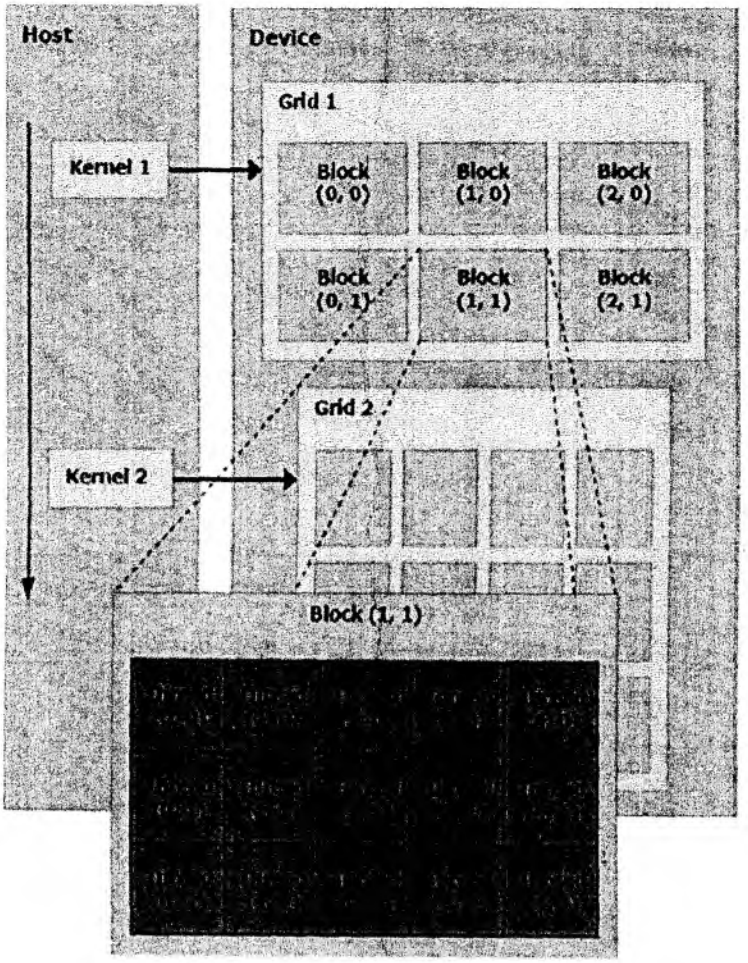


Рис. 1. Структура организации потоков на GPU

Параллельная программа в модели передачи сообщений представляет собой набор обычных последовательных программ, которые обрабатываются одновременно. Обычно каждая из этих последовательных программ выполняется на своем процессоре и имеет доступ к своей, локальной памяти [5]. Явным достоинством при такой организации вычислений является возможность написания и отладки программы на однопроцессорной системе.

Для достижения высокой эффективности при массивно-параллельных вычислениях, нужно учитывать множество факторов: архитектурные особенности GPU, быстродействие и порядок доступа к памяти, механизмы синхронизации между вычислительными потоками. Важную роль играет также приспособленность самого алгоритма к исполнению в параллельном режиме. Перенос простых арифметических операций на GPU дает выигрыш в производительности по сравнению с CPU при использовании даже самых простых моделей графических адаптеров [6].

Архитектура CPU позволяет использовать практически любые шаблоны параллельных вычислений, в то время как архитектура CUDA в каждый определенный момент времени позволяет выполнять на всех своих ядрах только одну инструкцию [7]. Поэтому архитектура CUDA может эффективно применяться только при вычислениях с большим параллелизмом и интенсивной арифметикой. Все функции, выполнимые на GPU, не поддерживают рекурсии и имеют некоторые другие ограничения, которых нет в архитектуре CPU.

Современные GPU предоставляют выгодное соотношение цены, производительности и энергопотребления. Многие суперкомпьютерные кластеры имеют высокий рейтинг в ТОП500, благодаря использованию GPU [8]. Однако, будучи специализированными устройствами, рассчитанными на потоковую обработку однотипных данных, GPU на многих алгоритмах не показывают значимого преимущества перед процессорами традиционной архитек-

туры [9]. Это обусловлено тем, что современные GPU используют SIMD-архитектуру [10] и специально рассчитанные на работу с ней контроллеры памяти [11]. GPU являются «массивно-параллельными» процессорами, т.е. вычислительными устройствами, на которых одновременно выполняется большое (по сравнению с CPU) количество вычислительных потоков.

В [12] проводится сравнение эффективности CPU и GPU реализаций некоторых комбинаторных алгоритмов, используемых в криптоанализе. Показывается, что применение специальных техник трансформации потока управления позволяет существенно компенсировать потери производительности, возникающие из-за неэффективного исполнения условных переходов на SIMD-устройстве. Однако ограничения, которые накладывают механизмы работы с памятью, применяемые в современных GPU, для рассматриваемых алгоритмов оказываются непреодолимыми.

Постановка задачи

Цель данной работы – изучение возможности применения GPU для систем потокового шифрования, в частности исследование возможностей переноса части вычислений по генерации псевдослучайной последовательности с CPU на GPU и сравнение производительности полученных решений. Под производительностью будем понимать количество генерируемых бит в единицу времени.

Для оценки возможностей GPU использовались регистры сдвига с нелинейной обратной связью генерирующие последовательность максимального периода (M-РСНОС) взятые из алгоритма, представленного в 2008 году на международном конкурсе eStream Achterbahn-128/80 version 1.1 [13].

Следует учитывать, что потоковый шифр Achterbahn в первую очередь разрабатывался для аппаратной реализации. Некоторые его возможности, такие как применение многократной реализации функции обратной связи с использованием параллельной реализации РСНОС, позволяющее увеличить количество генерируемых бит за такт, нами не рассматривались.

Функции обратных связей для всех 13 M-РСНОС (A_0, …, A_{12}), которые используются в потоковом шифре Achterbahn-128/80 были выполнены в соответствии с оптимизированной реализацией, приведенной в [14]. Функции обратных связей для каждых из 13 регистров имеют одинаковую логическую глубину [12], что обеспечивает примерно равную скорость генерации каждым регистром.

Кроме M-РСНОС, взятых из конструкции потокового шифра Achterbahn-128/80, был также исследован M-РСНОС найденный нами, и имеющий нелинейность второго порядка. Функция обратной связи, используемая в нашем M-РСНОС второго порядка, задается следующим соотношением:

$$A_{30} = x_{30} + x_{28} + x_{27} + x_{26} + x_{25} + x_{22} + x_{21} + x_{17} + \\ + x_{15} + x_{12} + x_{11} + x_4 + x_{27} \cdot x_{28} + x_{27} \cdot x_{29},$$

где x_i – значение x_i ячейки в регистре. Отсчет идет от ячейки, в которой помещается новое значение (определяемое функцией обратной связи A_{30}) и начинается с номера 1, и заканчивается последней (30 ячейкой), значение которой является выходным битом в следующей итерации.

Для компиляции программного кода использовалось программное обеспечение Microsoft Visual Studio 2015 и программно-аппаратная платформа для организации параллельных вычислений на графических процессорах NVIDIA CUDA 8.0. Вычисления проводились на персональном компьютере под управлением 64-разрядной Windows 7 SP 1, с процессором Intel Core i5-3210M CPU 2,5GHz и видеопроцессором NVIDIA GeForce GT 630M.

Полученные результаты

Вначале приведем результаты измерения скорости генерации М-РСНОС на CPU и GPU. В табл. 1 приведено время (t_{GB}), затраченное на генерацию 1 Гбайта (8 589 934 592 итераций) данных каждым из М-РСНОС A_0, …, A_{12} как по отдельности, так и для всех вместе.

Таблица 1

Временные затраты на генерацию 1 Гбайта на CPU (t_{GB}^C) и GPU (t_{GB}^G)

М-РСНОС	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_0...A_{12}
t_{GB}^C (с.)	107,4	112,5	118,1	115,6	117,7	114,2	115,3	118,3	128,4	130,5	140,9	106,8	133,8	133,8
t_{GB}^G (с.)	2408	2457	2310	2719	2621	2736	2506	2834	2556	2589	2687	2556	2294	26033

Необходимо оговорить тот факт, что здесь и далее под генерацией мы понимаем холостой прогон всех регистров необходимое число раз без полной передачи или записи полученных значений. При реальной генерации необходимо учитывать передачу генерируемых значений, на что также будет затрачено определенное время.

Как видим, благодаря оптимизированной реализации, время выполнения каждого из регистров как на CPU так и на GPU примерно равно (разброс не более 14 % для CPU и 11 % для GPU). Таким образом, если мы будем выполнять на GPU каждый из РСНОС параллельно в отдельном потоке, мы будем получать выходной бит без существенного простоя какого-либо из регистров.

В табл. 2 приведены результаты испытаний для одновременного расчета итераций с A_0 по A_{12} регистр. Так как в A_{12} используется регистр, состоящий из 33 ячеек, то для универсальности во всех регистрах применялись 64 битные переменные, с которыми GPU работает медленнее чем с 32 битными. Вычисления проводились в один поток. Каждый регистр вычислялся в своей нити, т.е. всего для расчета одного набора из 13 регистров запускалось 13 нитей. Общее число используемых нитей обозначим символом n_{trd}.

Таблица 2

Временные затраты на генерацию 1 Гбайта на GPU при одновременной генерации нескольких М-РСНОС

n_{trd}		13	26	39	52	65	78	91	104	117	130	143	156	169	182
t_{GB}^G (с.)	один блок	26033	26476	26509	26509	26689	29474	29540	30113	30572	30867	34258	34373	34603	34865
	два блока	26476	26492	26492	26492	26705	29720	29605	30244	30523	30900	34717	34504	34799	34881

Как видим из табл. 2, с ростом числа одновременно вычисляемых регистров время, затраченное на проведения расчетов для каждой из итерации, растет по нелинейному закону. В связи с чем, более целесообразно рассматривать производительность генерации τ, которую определим как количество значений, генерируемых одновременно регистрами во всех потоках в единицу времени:

$$\tau = \frac{n_{trd}}{t_{GB}} \left[\frac{\text{Гбайт}}{\text{сек}} \right].$$

Более наглядно результаты табл. 2 отображены на графике рис. 2. Как видим, с ростом числа одновременно просчитываемых регистров производительность достигает своего предельного значения и дальнейшее увеличения числа одновременно рассчитываемых регистров не дает прироста производительности.

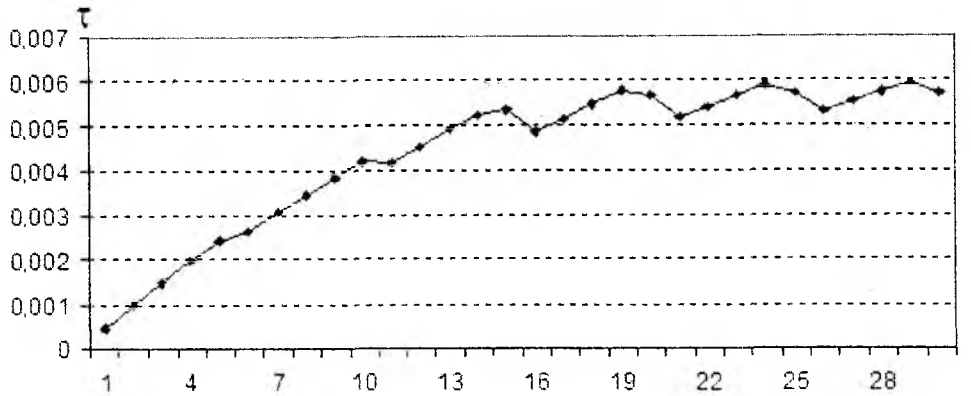


Рис. 2. Зависимость производительности τ генерации на GPU от числа одновременно рассчитываемых регистров с A_0 по A_{12}

На рис. 3 и 4 отображены результаты производительности при различном количестве задействованных блоков и нитей в каждом блоке для 32-битных и 64-битных переменных, используемых в качестве регистров.

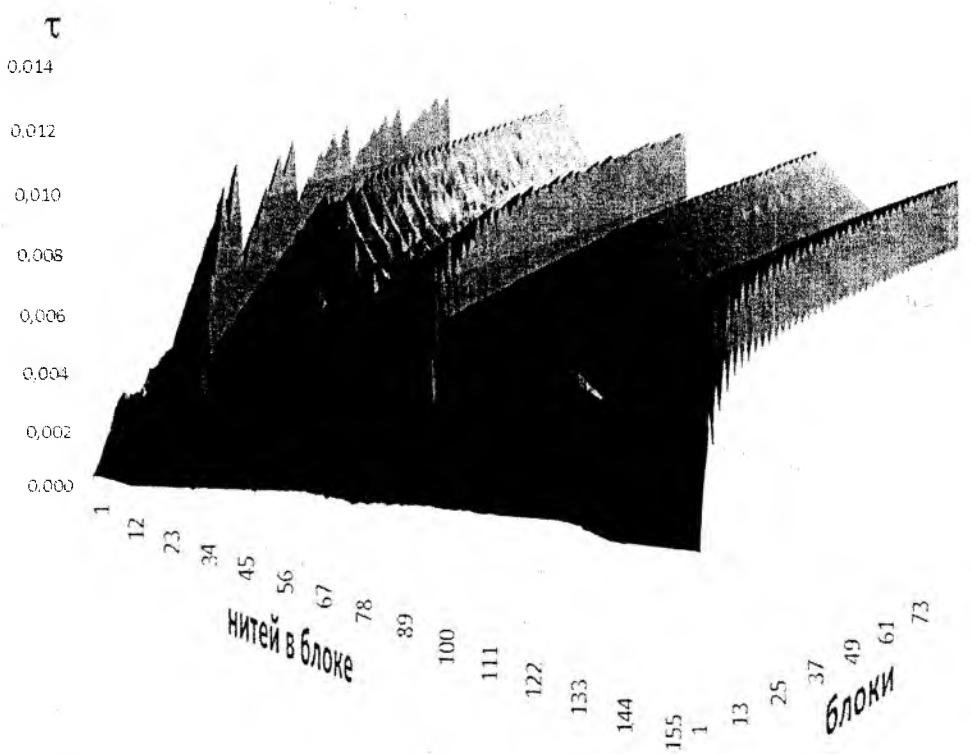


Рис. 3. Зависимость производительности τ генерации на GPU от количества блоков и нитей в блоке при размерности регистров в 64 бита

Как видим, на GPU генерация при размере используемых переменных в 64 бита значительно медленнее чем в 32 бита.

УДК 003.026:004.056

Ю.І. ГОРБЕНКО, канд. техн. наук, Т.В. МЕЛЬНИК, І.Д. ГОРБЕНКО, д-р техн. наук

АНАЛІЗ ПОТЕНЦІЙНИХ ПОСТКВАНТОВИХ МЕХАНІЗМІВ ЕЛЕКТРОННИХ ПІДПИСІВ НА ОСНОВІ ГЕШ-ФУНКЦІЙ

Вступ

В 2015 – 2017 рр. відбувся ряд значущих подій та прийняті на світовому рівні рішення, які визначили необхідність розробки та стандартизації постквантової асиметричної криптографії. До них необхідно віднести Інтернет – статтю [1], VII міжнародної конференцію з постквантової криптографії та її рекомендації, звіт «Report on Post – Quatum Cryptography. NISTIR 8105 (DRAFT) [2]», а також оголошення NIST США конкурсу на постквантові стандарти електронного підпису(ЕП) та асиметричного шифрування (АШ) [3] тощо. Серед можливих кандидатів на стандарт ЕП певні переваги мають методи, що грунтуються на використанні функцій гешування (НВ криптографія) [4 – 8]. Основні переваги НВ криптографії у використанні існуючих криптографічно стійких функцій гешування, значною швидкодією, прозорістю математичних перетворень та випробуванням функцій гешування часом. В той же час висунуті в [3] вимоги до постквантових ЕП вимагають проведення значних досліджень та порівняння потенційних кандидатів, що будуються на основі НВ криптографії та і інших математичних основ[4, 9] .

В [3] наведено основні вимоги до постквантових криптографічних примітивів – з безпеки застосування, техніко-економічні та техніко-експлуатаційні вимоги. Там же запропоновано критерії та показники відбору серед можливих кандидатів.

Мета цієї статі – відбір, розгляд сутностей та порівняльний аналіз, а також розробка пропозицій з удосконалення та застосування ЕП в обмеженому класі тільки НВ перетворень. В процесі досліджень використовуються визнані чи рекомендовані критерії та показники оцінки та відповідні методики.

Таким чином, нині вирішується проблема створення та стандартизації постквантових криптографічних примітивів типу ЕП. Вона надзвичайно актуальна і вимагає свого розв’язання в найближчі декілька років у відповідності з вимогами NIST США [2 – 4] та ETSI ЄС [10]. Отримані результати досліджень та пропозиції планується подати у вигляді серії статей.

1. Сутність та властивості постквантових ЕП Лампорта

На сьогодні існує декілька механізмів (схем) та складових ЕП, заснованих на геш-функціях. Серед них необхідно відмітити механізми Лампорта [5], Лампорта – Діффі [7], Вінтерніц [6], MSS (Merkle), XMSS, SPHINC, HORS та їх певні модифікації – XMSS-T, SPHINCS, SPHINCS-256, HORST [4, 8, 9]. Особливістю вказаних підписів є те, що їх криптографічна стійкість грунтується на колізійній стійкості функцій гешування та /чи стійкості до знаходження прообразу, так як нині уже розроблено та прийнято у якості геш-функцій ряд по суті стандартизованих постквантових геш-функцій. Запропоновані механізми, що названі вище, на наш погляд, є перспективними [9, 11]. Але залишається ряд проблемних питань, що повинні певною мірою бути вирішеними до початку етапу стандартизації: доведення криптографічної стійкості; мінімізація розмірів загальних параметрів та ключів; мінімізація довжини підпису та підвищення швидкодії ; вироблення та перевіряння ЕП тощо згідно вимог [2 – 7, 8 – 13].

Розглянемо у загальному вигляді сутності та властивості (переваги і недоліки), а також умови, відповідність вимогам, можливості удосконалення та застосування механізмів ЕП на основі НВ криптоперетворень в історичному ракурсі, але розглядаючи цю статтю як першу в досягненні поставленої мети.

1.1. Сутність та особливості механізму ЕП Лампорта

В [4, 5, 9] запропоновані криптографічні перетворення типу електронний підпис (ЕП) для постквантового періоду. Усі вони грунтуються на механізмах ЕП з використанням геш-функцій. Серед перших із них є механізм Лампорт ЕП, який уже раніше досліджувався, але потребує, на наш погляд, додаткового дослідження у зв'язку з новими вимогами та можливими застосуваннями в критичних додатках. Розглянемо та проведемо його аналіз у такій послідовності: узгодження геш-функції та загальних параметрів; генерування ключів; підпис та перевірка підпису; стійкість да потенційних атак та складність реалізації.

Загальні параметри. Підписувач А та перевірник Б узгоджують стандартизовану геш-функцію та її параметри, довжину геш-значення та генератори випадкових чи псевдовипадкових послідовностей (наявність та відповідність вимогам).

Генерування ключів. З використанням генератора чи генераторів випадкових чи псевдовипадкових послідовностей підписувач А генерує *n* секретних ключових пар (X, Y):

$$\begin{aligned} X &= (x_1,\ldots, x_i, \ldots, x_n) \\ Y &= (y_1,\ldots, yi, \ldots, y_n) \end{aligned} \quad (1)$$

з довжиною кожного із секретних ключів l_h, де l_h – довжина геш-значення вибраної геш-функції. При цьому кожна пара (x_i, y_i) є *i*-ю частиною секретного (особистого) ключа.

Відкритий ключ обчислюється засобом гешування секретних ключів (1), як наслідок отримуємо *n* пар відкритих ключів:

$$\begin{aligned} H(X) &= (H(x_1), \ldots, H(x_i), \ldots, H(x_n)) \\ H(Y) &= (H(y_1), \ldots, H(y_i), \ldots, H(y_n)), \end{aligned} \quad (2)$$

з довжиною геш-значення l_h .

Секретні ключі (1) загалом є обов'язково конфіденційними (особистими) та повинні бути доступними тільки підписувачу *A*. Відкриті ключі (2) опубліковуються чи є доступними усім користувачам Б іншим чином, що можуть отримувати від *A* підписані повідомлення.

Підпис повідомлення. При підписі повідомлення *M* підписувач *A* гешує повідомлення *M* з використанням узгодженої (як правило криптографічної) геш-функції з параметрами *Pr* та отримує геш-значення

$$h_M = H(M, Pr)) \quad (3)$$

Далі значення h_M, по суті, зашифровується засобом заміни бітів геш-значення h_M секретними одноразовими ключами із (1), причому кожен h_{Mi} біт, що приймає значення «0», замінюється послідовно секретним ключем із множини (1) X, а h_{Mi} біт, що приймає значення «1» заміняється послідовно секретним ключем із множини (1) Y. Процес такого зашифрування продовжується для усіх бітів геш-значення h_{Mi} .

Таким чином, l_h бітів геш-значення h_{Mi} заміняються (зашифровуються) по суті безумовно стійким шифром, оскільки послідовність бітів h_{Mi} заміняється одноразовими секретними ключами (x_i, y_i). Вказана послідовність l_h секретних ключів і є S ЕП повідомлення M, він разом з вибраними із x_i чи y_i стає відкритим та доступним як користувачам (перевірникам) відповідного домену так і порушнику (криптоаналітику). В подальшому такий ЕП у відповідному форматі передається та зберігається разом з повідомленням і є його ЕП.

У загальному випадку підписане повідомлення можна подати у такому вигляді

$$\{M; Z = (\{x_1 \mid y_1\}), \{x_2 \mid y_2\}, ..., \{x_i \mid y_i\}, ..., \{x_n \mid y_n\}) = \{M, Z = (z_1, z_2, ..., z_i, ..., z_n)\} \quad (4)$$

В (4) символ «|» означає, що при зашифрування в ЕП появляється один із використаних секретних сигналів – x_i чи y_i, що визначається *i*-м бітом геш-значення h_{Mi}. Таким чином, випадкові послідовності $(z_1, z_2, ..., z_i, ..., z_n)$ із секретного ключа стають ЕП повідомлення М.

Зрозуміло, що після зашифрування використані, що були секретними, ключі x_i чи y_i стають відкритими. Але в подальшому будемо враховувати, що відкритими із множини (1) стають лише *n* ключів, а *n* залишились секретними, вони в механізмі Лампорта після вироблення ЕП повинні бути знищеними та більше не використовуватись.

Перевірка ЕП повідомлення. При перевірянні ЕП підписане повідомлення $M*$ має такий вигляд

$$\{M^*; Z^* = (\{x_1 \mid y_1\}), \{x_2 \mid y_2\}, ..., \{x_i \mid y_i\}, ..., \{x_n \mid y_n\}) =$$
$$Z* = (z^*_1, z^*_2, ..., z^*_i, ..., z^*_n), \quad (5)$$

де символ «*» означає, що як повідомлення М, і підпис Z можуть бути викривленими чи підробленими тощо.

Нехай користувач Б хоче перевірити ЕП повідомлення (5). Це він може здійснити у такій послідовності.

1. Здійснює гешування повідомлення M^*, в результаті отримує геш-значення

$$h_{Mi*} = H(M^*, \Pr) \quad (6)$$

2. У відповідності зі значеннями h_{Mi*} із ЕП Z^* у відповідності з усіма значеннями бітів h_{Mi*} із відкритого ключа (2) вибираються геш-значення $H(x_i)$ чи $H(y_i)$.

Причому, якщо обчислене значення h_{Mi*} приймає значення «0», то із відкритого ключа ЕП підписувача А (2) вибирається відповідне значення $H(x_i)$, а якщо «1», то із відкритого ключа ЕП (2) підписувача А вибирається відповідне значення $H(y_i)$. Вказане виконується для усіх значень блоків бітів геш-значення (6) і користувач Б отримує

$$Z' = (\{H(x_1) \mid H(y_1)\}), \{H(x_2) \mid H(y_2)\}, ..., \{H(x_i) \mid H(y_i)\}, ..., \{H(x_n) \mid H(y_n)\}) =$$
$$= (z'_1, z'_2, ..., s'_i, ..., s'_n) \quad (7)$$

3. Користувач Б послідовно гешує усі ключі ЕП (4) та порівнює отримані значення зі значеннями (7) $(z'_1, z'_2, ..., z'_i, ..., z'_n)$. Якщо усі *n* значень при порівнянні співпали, то ЕП вважається справжнім, в іншому випадку ЕП вважається викривленим. Формально перевіряється, що для кожного *i* виконується вимога

$$z'_i = H(z_i). \quad (8)$$

Із наведеного механізму Лампорта слідує, що після зашифрування (4) половина секретних ключів із множин (Х, У) залишились секретними, так як вони знищуються після здійснення ЕП (зашифрування).

Нижче наводяться результати попереднього аналізу криптографічної стійкості ЕП Лампорта. Тут відмітимо основний недолік чи, скоріше всього, проблему реалізації ЕП Лампорта, яка зводиться до великих довжин ключів – як секретного, так і відкритого. Вважаємо, що для застосування ЕП Лампорта основні зусилля повинні бути направленими на її практичне вирішення.

1.2. Особливості одноразового механізму ЕП Лампорта – Діффі

Одною із перших модифікацій механізму ЕП Лампорта є механізм Лампорта – Діффі (LD-OTS) [7 – 9]. Особливістю модифікації механізму є уточнення вимог до геш-функцій. Так, геш-функція, що використовується для обчислення відкритих ключів (2) підписувачем та перевірником при перевірці підпису, є однонаправленою, а геш-функція, що використовується для гешування повідомлення *M* при виробленні ЕП, повинна бути криптографічною.

Загальні параметри. Нехай *l* – додатне ціле число, довжина геш-значення, яке є параметром безпеки механізму. В механізмі використовується однонаправлена геш-функція

$$f :\{0, 1\}^l \to \{0, 1\}^l \quad (9)$$

та криптографічна геш-функція

$$g :\{0, 1\}^* \to \{0, 1\}^l \quad (10)$$

Генерація ключової пари. Секретний ключ ЕП *X* механізму LD-OTS, як і у механізму Лампорта, складається з *2n* випадкових бітових строчок довжини l_h, причому не виключається, що $l_h = n$, тому

$$X= (x_{n-1}[0], x_{n-1}[1], \ldots, x_1[0], x_1[1], x_0[0], x_0[1]) \in R\{0, 1\}^{(n, 2n)} \quad (11)$$

Відкритим ключем підпису *Y* є послідовність строчок

$$Y= (y_{n-1}[0], y_{n-1}[1], \ldots, y_1[0], y_1[1], y_0[0], y_0[1]) \in R\{0, 1\}^{(n, 2n)}, \quad (12)$$

що отримана засобом гешування строчок секретного ключа (11), тобто як

$$y_i[j]= f(x_i[j]),\ 0 \le i \le n-1,\ j =0,1.$$

Таким чином, секретні ключі та відкриті ключі перевірки ЕП Лампорта – Діффі кожен складаються з *2n* строчок довжини *l*.

Підпис повідомлення. Повідомлення M = $\{0, 1\}^*$ підписується з використанням секретного ключа X (11). Спочатку за допомогою криптографічної геш-функції *g* обчислюється геш-значення повідомлення *M*

$$g(M) = h= (h_{l-1}, \ldots, h_0) \quad (13)$$

Безпосередньо ЕП повідомлення *M* буде, аналогічно (4), *n* строчок:

$$\sigma = (x_{n-1}[h_{n-1}], \ldots, x_1[h_1], x_0[h_0]) \in R\{0, 1\}^{(l, n)} \quad (14)$$

Таким чином, ЕП з використанням механізму LD-OTS є послідовність з *n* бітових рядків, кожен з яких довжиною *l*. Тобто ЕП механізму LD-OTS здійснюється аналогічно як і в механізмі Лампорта згідно з (4), *i*-й біт підпису є $x_i[0]$, якщо *i*-й біт геш-значення *h* дорівнює 0, інакше $x_i[1]$, якщо *i*-й біт геш-значення *h* дорівнює 1. *Всього довжина ЕП* складає $l \times n$ бітів, а при *l*=*n* буде $l^2 = n^2$.

Перевірка підпису. Перевірка ЕП механізму LD-OTS здійснюється аналогічно як і в механізмі Лампорта (6) та (8). Спочатку обчислюється геш-значення від повідомлення $h(M^*)$, підпис щодо якого перевіряється. Потім у відповідності зі значеннями h_{Mi*} із ЕП Z^* у відповідності з усіма значеннями бітів h_{Mi*}(0 чи 1) із відкритого ключа (12) вибираються геш-значення *yi[0]* чи *yi[1]*. *Наприкінці за допомогою однонаправленої геш-функції обчислюються геш-значення ЕП* (14), *які порівнюються з отриманими вище з* (12).

Необхідно підкреслити, що особливістю механізму ЕП Лампорта – Діффі у порівнянні з механізмом Лампорта є визначення функцій гешування (9) та (10), а також інший формат запису та використання секретного та відкритого ключів – в зворотному порядку. Тобто, в ньому зроблені уточнення.

1.3. Одноразовий ЕП Вінтерніц

Аналіз показує, що генерація ключа і підпису для механізму LD-OTS є ефективною, але розмір ЕП досить великий. Зменшення розміру ЕП досягається в механізмі одноразового ЕП, що запропонована в [6, 7], яка отримала назву механізму Вінтерніц (W-OTC). Ідея механізму Вінтерніц полягає в тому, щоб підписувати декілька бітів геш-значення, використовуючи один рядок одноразового ключа. Така пропозиція була запропонована вперше Merkle [7] як узагальнення його одноразового механізму. Однак, як слідує із [7], вона була описана в ній вперше.

Потрібно відмітити, що така ідея значний час застосовується в системах зв'язку при основному кодуванні, коли декілька бітів *w* при модуляції кодуються відповідним числом сигналі – переносників [12].

Загальні положення. Як і в механізмі Лампорта – Діффі – (LD-OTS) в механізмі Вінтерніц (W-OTS) використовуються одностороння геш-функція (6) та криптографічна геш-функція (10). Параметр Вінтерніц ЕП w ≥ 2 обирається як кількість бітів, що повинні бути підписані одночасно з використанням одноразового ключа.

У загальному випадку в механізмі Вінтерніц введено параметри t_1, t_2, t [6, 8], причому

$$t_1 = \lceil l / w \rceil, t_2 = \lceil \log_2 t_1 / w \rceil, t = t_1 + t_2, \tag{15}$$

де *l* довжина геш-значення, $w \geq 2$ – параметр, що визначає кількість бітів, які підписуються одним рядком одноразового ключа (LD-OTS). Параметр t_2 визначає необхідне число нулів, які потрібно добавити на початку геш-значення, щоб отримана в результаті нова довжина *t* була кратна w.

Для спрощення та з урахування практичного застосування параметр *w* можна визначити з обмеженням у вигляді (але не обов'язково)

$$w = 2^{\partial}, \ \partial = 1, 2, 3, 4, 5, 6, 7, 8... , \tag{16}$$

Генерування ключів. По аналогії з (11) у загальному випадку секретним ключем ЕП є $t_1 2^w$ випадкових бітових строчок довжини *l*

$$X = (x_{t1-1}, \ldots xi, \ldots, x_1, x_0) \in R\{0, 1\}^{(l,\ t_1 2^w)} \tag{17}$$

Таким чином, секретним ключем (17) є $t_1 2^w$ секретних ключів довжини *l*.

Відкритим ключем ЕП *Y* є послідовність строчок, що обчислюється шляхом застосування геш-функції *f*, по аналогії з (12), до кожного бітового рядку (17), внаслідок маємо

$$Y = (y_{t1-1}, \ldots, y_i, \ldots, y_1, y_0) \in \{0, 1\}^{(l,\ t_1 2^w)}, \tag{18}$$

де

$$y_i = f(x_i), 0 \leq i \leq t_1 2^w - 1. \tag{19}$$

Тобто, при генеруванні відкритого ключа необхідно виконати $t_1 2^w$ викликів однонаправленої геш-функції *f*. Необхідно підкреслити, що число секретних та відкритих ключів, які потрібні для виконання ЕП, залежить від величини *w*. Тому, змінюючи обгрунтувано w,
можна змінювати довжини секретного та відкритого ключів. Вказане досліджується нижче для ЕП, що розглядаються.

Вироблення ЕП W-OTC. Нехай необхідно підписати повідомлення *M* з геш-значенням g (*M*) = *d* = (d_{l-1}, ..., d_0). Для того щоб довжина *d* ділилась на *w*, спочатку необхідно встановити

мінімальне число нулів таким чином, щоб довжина d геш-значення ділилась на w. За цієї умови розширений рядок d розділяється на t_1 бітових блоків довжини w, тобто

$$d = b_{t1-1} \,||\, .bi\, .\, .\, ||\, b_0, \tag{20}$$

де знак || означає конкатенацію.

Засобом заміни кожного bi блоку геш-значення повідомлення g (M) зашифровується з використанням секретного ключа (17). Як результат ЕП повідомлення, тобто зашифроване геш-значення g (M), має такий вигляд

$$g(M) = S^* = \left(f^{b_{t1-1}}(x_{t1-1}), \ldots f^{bi}(x_i), \ldots, f^{b_1}(x_1), f^{b_0}(x_0)\right) = \\ = (s^*_{t1-1}, \ldots, s^*_i, \ldots, s^*_1, s^*_0) \tag{21}$$

Необхідно відмітити, що в (21) $s^*_{t1-1}, \ldots, s^*_i, \ldots, s^*_1, s^*_0$ – це ключі, які до вироблення ЕП були секретними. Але після вироблення ЕП вони стають відкритими, як і підписане повідомлення M , разом з ЕП.

Перевірка ЕП W-OTS. Для перевірки ЕП отриманого повідомлення M^*, що має вигляд

$$S^* = (s^*_{t1-1}, \ldots, s^*_i, \ldots, s^*_1, s^*_0) \tag{22}$$

спочатку аналогічно (20) обчислюється геш-значення $g(M^*)$, що має такий вигляд

$$d^* = b^*_{t1-1}, \ldots, b^*_i, \ldots, b^*_0. \tag{23}$$

Наявність символу (*) у M^* та у всіх b^*_i елементах означає, що вони могли бути викривленими штучно чи в результаті помилок при обробленні, передаванні та прийманні.

Далі, для всіх b^*_i (23) із відкритого ключа (18) у відповідності з їх значеннями вибираються певним чином геш-значення, що є складовими відкритого ключа. В результаті отримуємо

$$Y = (y_{t1-1}, \ldots, y_i, \ldots, y_1, y_0) \in \{0, 1\}^{(l,\, t_1)} \tag{24}$$

Потім здійснюється гешування значень $s^*_{t1-1}, \ldots, s^*_i, \ldots, s^*_1, s^*_0$, тобто ключів безпосередньо ЕП (22), в результаті отримуємо

$$Y^* = (y^*_{t1-1}, \ldots, y^*_i, \ldots, y^*_1, y^*_0) \in \{0, 1\}^{(l,\, t_1)} \tag{25}$$

Наостанок порівнюємо значення (24) y_i та (25) y^*_i для j = 0, 1, ...,i,…, t1 – 1.

Якщо $y_i = y^*_i$ для усіх j = 0, 1, ...,i,…, t1 – 1, то ЕП правильний, що підтверджує цілісність та справжність ЕП та підписане за його допомогою повідомлення M, а також дозволяє встановити авторство повідомлення M.

На наш погляд, викладений вище механізм одноразового ЕП *W-OTS* має суттєвий недолік – число секретних та відкритих ключів згідно (17) та (18) зі збільшенням параметра w зростає експоненційно. Крім того, довжина ЕП у залежності від значення w також э суттєвою. Вказане практично виключає можливість застосування механізму Вінтерніц на практиці, навіть у особливих технологіях з малим числом користувачів – підписувачів. Нижче наводиться удосконалений механізм Вінтерніц – Лампорта

1.4. Сутність та реалізація удосконаленого ЕП Вінтерніц – Лампорта

Аналіз підтверджує, що ключі і підписи Лампорта та Лампорта – Діффі (LD-OTS) є безумовно стійкими, але розмір ЕП досить великий, а також в Вінтерніц OTS (W-OTS) механізмі ЕП, що розглянуто вище, існує можливість виробляти коротші ЕП, але число секретних та відкритих ключів зі збільшенням параметра w зростає експоненційно. Особливість механізму Вінтерніц W-OTS в тому, що існує можливість використати секретний ключ одночасно для підпису декількох та значно більше бітів геш-значення [8, 9,

13]. Використовуючи цю ідею, розглянемо удосконалений механізм W-OTS з одноразовими ключам, основними перевагами якого є можливість суттєвого зменшення довжин секретних та відкритих ключів, а також довжини ЕП.

Як і в LD-OTS в W-OTS будемо використовувати односторонню геш-функцію

$$f : \{0, 1\}^{l} \rightarrow \{0, 1\}^{l}$$

та криптографічну геш-функцію

$$g : \{0, 1\}^{*} \rightarrow \{0, 1\}^{l}.$$

Генерація ключів для механізму W-OTS. Будемо вважати, що параметр $w \geq 2$ визначає кількість бітів геш-значення, що повинна бути підписано одночасно, тобто замінена одним секретним ключем. Цей механізм схожий на багатослівне кодування, що застосовується у системах зв'язку [12].

Для здійснення ЕП спочатку уточнимо параметри підпису – t_1 та t. Якщо довжина ЕП l кратна w, то $t1$ визначає кількість блоків бітів геш-значення, що будуть підписуватись (зашифровуватись) одним секретним ключем, причому

$$t = t1 = n / w \tag{26}$$

Якщо n не кратне w, то в останньому блоці буде менше чим w бітів, тому число блоків які потрібно підписати, необхідно збільшити на $t_{2=1,}$ записавши в перший блок додатково необхідне число нулів. У загальному випадку

$$t = t_1 + t_2 \tag{27}$$

Секретним ключем ЕП по аналогії з (1) є (X, Y) послідовність t множин 2^w секретних ключів (x_i, y_i) як і у механізмі Лампорта (1) та (2) , тобто

$$X = (x_{t-1}, \ldots, x_i, \ \ldots, x_0), \ Y = (y_{t-1}, \ldots, yi, \ \ldots, y_0) \tag{28}$$

з довжиною кожного із секретних ключів l_h, де l_h – довжина геш-значення для однонаправленої геш-функції f. При цьому кожна із множин 2^w секретних ключів (x_i, y_i) є i-ю частиною секретного (особистого) ключа. Згідно з (27) в (28) число секретних множин ключів менше ніж для механізмів Лампорта та Лампорта – Діффі (1). Точні оцінки наводяться нижче.

Відкритий ключ перевірки ЕП обчислюється засобом гешування секретних ключів (28) з застосуванням однонаправленої геш-функції f, внаслідок отримуємо t множин 2^w відкритих ключів:

$$H(X) = H(x_{t-1}), \ldots, \mathrm{H}(x_i), \ \ldots, \mathrm{H}(x_0),$$
$$H(Y) = H(y_{t-1}), \ldots, \mathrm{H}(y_i), \ \ldots, \mathrm{H}(y_0) \tag{29}$$

також з довжиною l_h кожного відкритого ключа пари $(H(x_i), \mathrm{H}(y_i))$. Але у випадку (29) параметр t в w разів менше за l_h.

Вироблення ЕП для механізму W-OT. Нехай по аналогії з (13) повідомлення M має геш-значення

$$g(M) = h = (h_{t-1}, ..., h_i, ..., h_0), \tag{30}$$

яке потрібно підписати з використанням криптографічної функції g.

У загальному випадку, якщо l_h не кратно *w*, добавимо до *h* деяке число нулів, так, щоб довжина l_h була кратна *w*. Рядок *h*, бітів розділяється на *t* блоків $b_{t-1},...,b_i,...,b_0$ довжини *w* бітів кожний.

В подальшому для ЕП та перевірки ЕП застосовуємо інше правило зашифрування: якщо значення b_i блоку знаходиться в інтервалі

$$0 \le b_i \le (2^w / 2) - 1 \tag{31}$$

то b_i блок зашифровується (заміняється) послідовно секретним ключем із множини (28) X, інакше заміняється послідовно секретним ключем із множини (28) Y. Тоді підписане повідомлення має такий вигляд

$$\{M; Z = (\{x_{t-1} \mid y_{t-1}\}), \{x_{t-2} \mid y_{t-2}\},...,\{x_i \mid y_i\},...,\{x_0 \mid y_0\}) = \{M, Z = (z_t, z_{t-1},..., z_i,..., z_0)\} \tag{32}$$

В (32) символ « | » означає, що при зашифруванні в ЕП з'являється один із використаних секретних сигналів – x_i чи y_i, що визначається *i*-м блоком бітів довжини w.

Правило (31) будемо застосовувати принципово по-новому. На відміну від механізму Вінтерніц не зашифровуються усі можливі стани 2^w, коли потрібно 2^w секретних ключів для кожного блоку b_i. В нашому випадку для зашифрування b_i потрібно всього $r = 2$ секретних та відкритих ключів. Крім того, в якості довжин блоків можна розглядати значення

$$l_h, l_h / 2, l_h / 4, l_h / 8... \tag{33}$$

В цьому випадку необхідне число секретних та відкритих ключів, а також довжина ЕП суттєво скорочується. Детально покажемо це нижче.

Перевірка ЕП для механізму W-OT. Перевірка ЕП здійснюється аналогічно (5) – (8), тобто у такій послідовності.

1. Із використанням криптографічної геш-функції *g* здійснюється гешування повідомлення M^*, для якого робиться перевірка ЕП, в результаті отримується геш-значення

$$h_{Mi^*} = H(M^*, \Pr).$$

Якщо довжина h_{Mi^*} не кратна w, то до рядка бітів h_{Mi^*} у відповідності з домовленістю добавляється деяке число нулів, так, щоб довжина h_{Mi^*} була кратна w. Рядок h_{Mi^*} бітів розділяється на *t* блоків $b_{t-1},...,b_i,...,b^0$ довжини w бітів кожен.

2. У відповідності зі значеннями b_i блоків h_{Mi^*} згідно правила (31) із відкритого ключа перевірки ЕП (29) згідно з критерієм (31) вибираються геш-значення $H(x_i)$ чи $H(y_i)$, внаслідок отримуємо

$$Z^* = (\{H(x_1) \mid H(y_1)\}), \{H(x_2) \mid H(y_2)\},...,\{H(x_i) \mid H(y_i)\},...,\{H(x_n) \mid H(y_n)\}) =$$

$$= (z^*_{t-1}, z^*_{t-2},..., z^*_i,..., z^*_0) \tag{34}$$

3. Користувач Б послідовно гешує усі ключі ЕП (32) та отримує значення

$$(H(z_t), H(z_{t-1}),..., H(z_i),..., H(z_0) \tag{35}$$

та порівнює отримані значення зі значеннями (34), тобто $(z^*_{t-1}, z^*_{t-2},..., z^*_i,..., z^*_0)$. Якщо усі *t* значень при порівнянні співпали, то ЕП вважається справжнім, в іншому випадку ЕП вважається викривленим.

Нижче наводяться результати порівняльного аналізу криптографічної стійкості та складності розглянутих вище постквантових механізмів ЕП на основі геш-функцій.

2. Аналіз властивостей ЕП з одноразовими ключами на основі геш-функцій

2.1. Загальні положення щодо умов здійснення атак

Згідно з вимогами [3, 4, 8, 9] постквантові ЕП повинні бути стійкими проти усіх відомих постквантових та класичних атак. Попередній аналіз показав, що основними загрозами щодо ЕП Лампорта, Лампорта – Діффі, Вінтерніц та удосконаленого з разовими ключами є підробка ЕП та створення хибного ЕП. Такі загрози повинні бути здійсненими в умовах використанні при ЕП одноразового ключа (1), (11), (17) та (26). Вказані загрози можуть бути потенційно реалізованими засобом здійснення силових атак типу «брутальна сила» та аналітичного типу [11].

Відповідно до загальновизнаних підходів основною задачею вказаних атак будемо вважати визначення секретного (особистого) ключа користувача (підписувача), тобто наприклад відносно ЕП Лампорта – визначення секретного ключа (1), для механізму Лампорта – Діффі секретного ключа (11), удосконаленого механізму секретного ключа (28). Це пов'язане з тим, що знаючи секретний ключ ЕП, криптоаналітик може здійснювати як модифікацію підписаного, так і створювати хибне повідомлення з дійсним підписом.

Для проведення криптоаналізу введемо модель порушника та модель загроз. Особливістю розгляду в тому, що за основу моделі порушника беруться модель квантового та класичного комп'ютерів та їх можливості, а за основу моделі загроз – методи та алгоритми квантового та класичного криптоаналізу щодо постквантових ЕП на основі геш-функцій.

В якості основних вихідних даних візьмемо такі [2 – 4, 5 – 9].

1. В якості геш-функцій можуть застосовуватись : ДСТУ ISOIEC 10118 (SHA -2); FIPS 202 (SHA –3); ДСТУ 7564-2014.
2. Функції гешування є стійкими проти класичного та квантового криптоаналізу, в тому числі: до знаходження прообразу; до знаходження другого прообразу; до виникнення чи створення колізій.
3. Порушник має повний доступ до математичної та програмних моделей геш-функцій та може отримувати з високою ймовірністю підписані повідомлення;
4. В якості секретних (особистих) ключів ЕП використовуються випадкові чи псевдовипадкові послідовності, що задовольняють вимогам нормативно-правових документів, наприклад NIST SP 800 –22: 2009 тощо.
5. Усі n секретних послідовностей секретного ключа, що використані при виробленні ЕП, є відкритими та доступні криптоаналітику, а ті, що не використані, зменшились секретними та знищені після ЕП.
6. Відкритим ключем ЕП є геш-значення усіх секретних послідовностей секретного ключа.
7. ЕП, що задовольняють вимогам 1 – 6, наведеним вище, згідно з [3, 11] будемо називати повністю автентичними з мінімальною ймовірністю обману щодо усіх відомих атак.
8. Основними атаками при цих дослідженнях будемо вважати атаки типу модифікація чи підробка (створення хибного) підпису довільного повідомлення.

Для підтвердження «повної автентичності» стійкості до модифікації та підробки підписаних повідомлень наведемо наступне.

2.2. Метод оцінки складності атаки типу модифікація ЕП

Нехай за умов 1 – 8 підрозд. 2.1 криптоаналітик змінює хоча б один біт в підписаному повідомленні *M*. Але при цьому він хоче або змінити ЕП, так щоб перевірка ЕП дала позитивний результат, або щоб значення ЕП не змінилося. Нехай при перевірці, наприклад на приймальній стороні, отримується викривлене значення повідомлення *M*', тобто криптоаналітик реалізує атаку зі зміною змісту повідомлення *M*. При перевірці ЕП спочатку обчислюється геш-значення

$$H(M') = (b'_{t-1}, ..., b'_i, ..., b'_0) \tag{36}$$

Далі, якщо геш-функція, що застосовується, відповідає вимогам до криптографічних геш-функцій, то після модифікації одного біта в повідомленні *M* в середньому зміниться половина бітів геш-значення H(m'), тобто приблизно $l_h / 2$ бітів. Також криптоаналітик знає тільки ті *n* значень секретного ключа, що були в невикривленому ЕП. Тому для модифікації ЕП йому необхідно знати *n* інших випадкових послідовностей секретного ключа. Це необхідно для того, щоб мати можливість використати їх для підстановки у відповідні місця підпису S, для тих бітів геш-значення, що змінилися.

Відповідно до прийнятої моделі будемо вважати, що криптоаналітик знає все про геш-функцію та може обчислити геш-значення модифікованого повідомлення M'. Це означає , що криптоаналітик може визначити біти геш-значення, що змінились після модифікації повідомлення. Тоді сутність атаки згідно з (1), (11), (17), (26) зводиться до того, щоб визначити та підставити в ЕП в середньому *n* секретних послідовностей, що йому недоступні.

Ймовірність знаходження однієї послідовності визначимо, вважаючи, що атака типу модифікація здійснюється методом створення колізії (наприклад, методом Гровера) [11]. Дійсно, із [11] слідує, що одним із ефективних алгоритмів криптоаналізу симетричних криптоперетворень, в тому числі стосовно геш-функцій, є алгоритм Гровера. При його використанні секретний ключ симетричного криптоперетворення можна знайти засобом виконання (за час) $\sqrt{N}$ групових операцій, де *N* – число можливих ключів.

Показано, що в цьому випадку при застосуванні методу ρ- Полларда ймовірність виникнення колізії *P(N,k)* в *k* спробах, а значить знаходження ключа, при довжині геш-значення l_h та $N = 2^{l_h}$ можна визначити як

$$p_\rho(N,k) = 1 - \ell^{-(k^2-k)/(2^{l_h+1})}. \tag{37}$$

Вираз (37) після простих перетворень та логарифмування можна подати у такому вигляді [12]:

$$k^2 - k + 2^{l_h+1} ln\,(1 - p_\rho(N,k)) = 0 \tag{38}$$

При застосуванні λ-Полларда методу, по аналогії з (37) та (38), ймовірність виникнення колізії *P(N,k)* в *k* спробах, а значить знаходження ключа [11]:

$$p_\lambda(N,k) = 1 - \ell^{-(k^2-1)/(2^{l_h})}, \tag{39}$$

$$k^2 - 1 + 2^{l_h} ln\left(1 - p_\lambda\left(N,k\right)\right) = 0\ . \tag{40}$$

У цілому (37) – (40) є параметричними співвідношеннями, що зв'язують три параметри – $p\left(N,k\right)$, k та $N = 2^{l_h}$. Тому, задаючи два параметри, можна обчислити значення третього тощо. В нашому випадку необхідно визначити $p(N,k)$, тобто ймовірність визначення однієї послідовності секретного ключа відповідно (1), (11), (17) чи (26).

Зведемо (37) та (39) до основ 2 та 10.Тоді, використовуючи властивості степені та логарифму для (37), отримаємо

$$P(N,k)=1-2^{-\log_2 e\cdot(2^{2V}-2^{V})\cdot 2^{-(\ln+1)}}=1-2^{-(\log_2 e\cdot(2^{2V-\ln-1}-2^{V-\ln-1}))} \quad (41]$$

При $k^2 >> k$

$$P(N,k)=1-2^{-\log_2 e\cdot 2^{2V-\ln-1}}. \quad (42]$$

Зрозуміло, що (16) потрібно застосувати коректно.

Для основи степені 10 по аналогії із (41) та(42) маємо

$$P(N,k)=1-10^{-\log_{10} 2\cdot 10^{\log_{10}(2V-\ln-1-v+\ln+1)}} \quad (43)$$

або при $k^2 >> k$

$$P(N,k)=1-10^{-\log_{10} 2\cdot 10^{\log_{10}(2V-\ln-1)}}. \quad (44]$$

Формули (41) – (46) важливі тим, що вони дозволяють отримувати та робити інтерпретацію стійкості зразу в бітах(чи кубітах).

В табл. 1 наведені значення ймовірностей (37) при таких вихідних даних:

- l_h – довжина геш-значення;
- k – кількість спроб підібрати одну послідовність секретного ключа;
- $N = 2^{l_h}$ – кількість можливих (допустимих) секретних ключів (послідовностей);
- n кількість послідовностей, що можуть бути використані в якості одноразового секретного ключа;

В табл. 1 показана ймовірність модифікації $P_{\rho}(N,k)$ ЕП при n=1.

Таблиця 1

l_h\k	2^{32}	2^{64}	2^{128}	2^{256}	2^{512}
256	$7.965*10^{-59}$	$1.469*10^{-39}$	0,393	1	1
512	$6.879*10^{-136}$	$1.269*10^{-116}$	$4.318*10^{-78}$	0.393	1

Наведені в табл. 1 дані не протирічять теорії колізій, вони характерні в точках $2^{128}, 2^{256}$.

2.3. Метод оцінки складності атаки створення хибного повідомлення та ЕП

Розглянемо атаку на ЕП, сутність якої зводиться до аналізу можливості створити та нав'язати хибне повідомлення при застосуванні розглянутих в розд. 1 механізмів ЕП з одноразовими ключами. Будемо вважати, що в системі ЕП забезпечується довіра за рахунок третьої довірчої сторони чи між підписувачами безпосередньо. Аналіз будемо вести в узагальненому вигляді для усіх чотирьох механізмів, що наведені вище.

В якості моделі порушника та моделі загроз приймемо відповідні моделі, що наведені в підрозд. 2.1. За виключенням п.8 будемо розглядати атаку зі створення хибних повідомлення та ЕП.

Попередній аналіз показує, що атака зі створенням хибного з ЕП повідомлення може здійснюватись за таких умов: криптоаналітику відомий відкритий ключ, наприклад у вигляді сертифікату відкритого ключа; відкритий ключ криптоаналітику невідомий, але він володіє повною інформацією про ЕП, в тому числі може накопичувати та аналізувати одноразові ЕП. Розглянемо та проведемо аналіз стійкості ЕП проти атаки повне розкриття при відомому відкритому ключі, коли криптоаналітику достовірно є відомими відкриті ключі (2), (12), (18) та (29). Дану атаку називаємо атакою «повне розкриття» тому, що в результаті її реалізації компрометується секретний ключ, наприклад (Х, У), для випадку (2), (18) та (29).

Вирішення задачі «повне розкриття» в такій постановці зводиться до застосування методу створення колізій для тих *n* послідовностей, що будуть використані при підписуванні засобом шифрування бітів геш-значення хибного повідомлення, наприклад для (29) та (28). Тобто, знаючи $H(x_i)$, необхідно знайти x_i для усіх *n*, які згідно з геш-значенням хибного повідомлення M^* повинні бути використаними. Повний опис таких подій можна отримати з використанням формул (37) – (41). Наприклад, при застосуванні методу ρ-Полларда ймовірність створення колізій визначається (37), але вимагає виконання $k_{\min}$ групових операцій для того, щоб забезпечити мінімально допустиму ймовірність створення колізій $P_{\min}(N,k)$. Тобто, в цьому випадку формули (37) та (38) мають такий вигляд

$$p_{\min_\rho}(N,k_{\min}) = 1-\ell^{-(k^2{}_{\min}-k_{\min})/(2^{l_h+1})}, \quad (45)$$

$$k^2{}_{\min} - k_{\min} + 2^{l_h+1} ln\,(1- p_{\min_\rho}(N,k)) = 0\ . \quad (46)$$

Зробимо детально аналіз та постановку цієї задачі.

По перше, криптоаналітик обмежений часом створення колізій для усіх *n* відкритих ключів, наприклад часом $t_{\min}$. Знаючи час $t_{\min}$, визначимо через необхідну складність допустиме значення $k_{\min}$. Далі, знаючи, *N* (l_{min}) та $k_{\min}$, знаходимо конкретне значення *P**. Також будемо вважати, що створення колізій відбувається для усіх *n* паралельно.

В табл. 2 наведено значення часової складності T обернення (створення колізій) (років) для таких вихідних даних: потужність криптоаналітичної системи відповідно до $I = 10^{12}, 10^{14}, 10^{16}$ *гр.оп./сек* , $t_{\min} = 3.15 \bullet 10^7$ сек/рік, $l_h = 256(512)$ бітів.

Таблиця 2

$l \backslash I$	10^{12}	10^{14}	10^{16}
256	10^{19}	10^{17}	10^{15}
512	10^{56}	10^{54}	10^{52}

В табл. 3 наведено оціночні значення ймовірності визначення однієї послідовності секретного ключа з один рік.

Таблиця 3

$l \backslash I$	10^{12}	10^{14}	10^{16}
256	10^{-19}	10^{-17}	10^{-15}
512	10^{-56}	10^{-54}	10^{-52}

Аналіз даних табл. 2 та 3 дозволяє зробити висновок про практичну неможливість створення хибного повідомлення методом обернення навіть одного відкритого ключа. При оберненні *n* ключів та послідовному криптоаналізі відповідно часова складність збільшується в *n* разів, а ймовірність успішного криптоаналізу зменшується в *n* разів.

Розглянемо атаку за умови, що відкритий ключ невідомий. Дану атаку також назвемо атакою «повне розкриття» тому, що в результаті її реалізації компрометується секретний ключ, наприклад (X, У) для випадку (2), (18) та (29). Спроба вирішення задачі «повне розкриття» в такій постановці зводиться до наступного. При підписуванні засобом шифрування бітів геш-значення робиться спроба створити хибне повідомлення, наприклад при повному розкритті (28). Вона вирішується методом Гровера послідовно для кожної із послідовностей секретного ключа. За даних умов ймовірність створення колізії для розкриття однієї послідовності секретного ключа можна оцінити, використовуючи формули (37) та (38), але при цьому також задатись і допустимими значеннями $k_{\min}$ та $t_{\min}$.

2.4. Оцінка та порівняння розмірів одноразових ключів та ЕП

При розгляді та аналізі механізму Лампорта, Лампорта – Діффі, Вінтерніц та удосконаленого механізму ЕП з разовими ключами однією із задач, що ставилась та вирішувалась, є зменшення розмірів ключів та ЕП. На наш погляд, найбільш продуктивним є удосконалений механізм з одноразовими ключами. Орієнтуючись на викладене в підрозд.1.1 – 1.4 та 2.1 – 2.3, зробимо оцінки розмірів ключів та ЕП для усіх розглянутих механізмів.

В якості вихідних даних приймемо такі.

1. В механізмі Лампорта та Лампорта – Діффі використовуються значення $n = l_h$, w=1, а l_h =256 та 512 бітів.
2. В механізмі Вінтерніц використовуються значення $n = l_h$, w=2, 4, 6, 8, 16, причому аналіз будемо проводити для l_h =256 та 512 бітів.
3. В удосконаленому механізмі $l_h = \mu \times n, \mu = 2,4,8,16,32,128,256.$, а W використовується у змісті (19).

В табл. 4 наведено результати оцінки розмірів секретних та відкритих одноразових ключів та розмірів ЕП для механізмів Лампорта та Лампорта – Діффі. Довжина секретного та відкритого ключів визначається як $2 \times l_h \times n$, довжина ЕП $l_h \times n$.

Таблиця 4

Розміри даних l_h, n			Розмір секретного ключа	Розмір відкритого ключа	Розмір ЕП
256	256		2^{17}	2^{17}	2^{16}
512	512		2^{19}	2^{19}	2^{18}

В табл. 5 наведено результати оцінки розмірів секретних та відкритих одноразових ключів та розмірів ЕП для механізму Вінтерніц. Довжина секретного та відкритого ключів визначається як $2 \times w^2 \times n_i \times l_h$, довжина ЕП $n_i \times l_h$

Таблиця 5

Розміри даних\ l_h, n_i	w_i		Розмір секретного ключа	Розмір відкритого ключа	Розмір ЕП
256	128	2	2^{18}	2^{18}	2^{15}
	64	4	2^{19}	2^{19}	2^{14}
	32	8	2^{23}	2^{23}	2^{16}
512	256	2	2^{20}	2^{20}	2^{17}
	128	4	2^{21}	2^{21}	2^{16}
	64	8	2^{25}	2^{25}	2^{17}

В табл. 6 наведено результати оцінки розмірів секретних та відкритих одноразових ключів та розмірів ЕП для удосконаленого механізму. Довжина секретного та відкритого ключів визначається як $2 \times \mu_i \times l_h$, довжина ЕП $\mu_i \times l_h$

Таблиця 6

Розміри даних\ l_h,	w_i		Розмір секретного ключа	Розмір відкритого ключа	Розмір ЕП
256	μ_i	2	2^{10}	2^{10}	2^{9}
		4	2^{11}	2^{11}	2^{10}
		8	2^{12}	2^{12}	2^{11}
		16	2^{13}	2^{13}	2^{12}
		32	2^{14}	2^{14}	2^{13}
		128	2^{16}	2^{16}	2^{14}
		256	2^{17}	2^{17}	2^{16}
		2	2^{11}	2^{11}	2^{10}

512	μ_i	4	2^{12}	2^{12}	2^{11}
		8	2^{13}	2^{13}	2^{12}
		16	2^{14}	2^{14}	2^{13}
		32	2^{15}	2^{15}	2^{14}
		128	2^{17}	2^{17}	2^{16}
		256	2^{18}	2^{18}	2^{17}
		512	2^{19}	2^{19}	2^{18}

У цілому результати порівняння досліджених та запропонованого механізму дозволяють зробити такі висновки.

Розміри секретних та відкритих одноразових ключів та розмірів ЕП для механізму Вінтерніц у порівнянні з механізму Лампорта вимагають збільшення розмірів секретних та відкритих одноразових ключів від 2 до 64 разів, але розміри ЕП зменшуються в 2 рази.

Розміри секретних та відкритих одноразових ключів та розмірів ЕП для удосконаленого механізму у порівнянні з механізмом Лампорта можуть бути зменшені для довжини ЕП 256 від 2 до 128 разів, а для довжини ЕП 512 ві 2 до 512 разів. При цьому для μ_i 256 та 512 відповідні значення табл. 8 співпали з відповідними даними табл 6.

Розміри секретних та відкритих одноразових ключів та розмірів ЕП для удосконаленого механізму у порівнянні з механізмом Вінтерніц для довжини 256 можуть бути зменшені для довжини ЕП 256 від 64 до 256 разів, а для довжини ЕП 512 – в 64 до 512 разів. При цьому довжини ЕП зменшуються від 8 до 32 разів (для довжини геш-значення 256), а для довжини 512 – від 2 до 128 разів.

У цілому необхідно зробити висновок, що удосконалений механізм забезпечує зменшення розмірів секретних та відкритих ключів, а також розмір ЕП.

Але необхідно ще дослідити та порівняти криптографічну стійкість та визначити розміри параметрів, при яких можна застосовувати удосконалений алгоритм одноразового ЕП.

2.5. Оцінка стійкості ЕП на основі геш-функцій з одноразовими ключами

Для визначення ймовірності успішного знаходження усіх n із $2n$ секретних послідовностей $P(n, p(N,k))$ секретного ключа покладемо, що усі послідовності є випадковими, рівно ймовірними та незалежними (див. п.1 – 7), тому ймовірність правильного визначення усіх послідовностей є добутком n подій. З урахуванням (37) отримуємо, що ймовірності успішного знаходження усіх n із $2n$ секретних послідовностей

$$P(n, p_\lambda(N,k)) = (1-\ell^{-(k^2-k)/(2^{l_h+1})})^n. \quad (47)$$

По аналогії з (41) для λ-Полларда методу для загального випадку отримаємо

$$P(n, p_\rho(N,k)) = (1-\ell^{-(k^2-1)/(2^{l_h})})^n. \quad (48)$$

Ймовірності (47) та (48) можна трактувати як ймовірності несанкціонованого доступу в систему ЕП на основі одноразових ключів. Також необхідно відмітити, що на перший погляд формули (47) та (48) дають різні результати. Насправді це не так, оскільки складність групових операцій в обох випадках є різною. При застосуванні ρ-Полларда методу використовується один процес реалізації групових операцій, а при застосуванні λ-Полларда методу застосовуються два процеси. Хоча з цього питання ще потрібні деталізації.

Для зручності обчислень формули (47) та (48) наведемо для основ 2 та 10. В результаті маємо для (47)

$$P(n, p(N,k)) = (1-2^{-\log_2 e\cdot(2^{2V}-2^{V})\cdot 2^{-(\ln+1)}})^n = (1-2^{-(\log_2 e\cdot(2^{2V-\ln-1}-2^{V-\ln-1}))})^n \quad (49]$$

При $k^2 >> k$

$$P(n, p(N,k)) = (1-2^{-\log_2 e\cdot 2^{2V-\ln-1}})^n \quad (50]$$

Для основи степені 10 аналогічно (49) та(50) маємо

$$P(n, p(N,k)) = (1-10^{-\log_{10} 2\cdot 10^{\log_{10}(2V-\ln-1-v+\ln+1)}})^n, \quad (51)$$

а при $k^2 >> k$

$$P(n, p(N,k)) = (1-10^{-\log_{10} 2\cdot 10^{\log_{10}(2V-\ln-1)}})^n \quad (52]$$

В табл. 7 наведено значення ймовірності вироблення хибного ЕП згідно запропонованої методики при n = 2 та при k= $2^{32}, 2^{64}, 2^{128}, 2^{256}, 2^{512}$. Пояснимо, що при n=2 згідно з нашою моделлю геш-значення зашифровується всього двома одноразовими парами ключів.

В табл. 8 наведено значення ймовірності вироблення хибного ЕП при n = 8 та при k= $2^{32}, 2^{64}, 2^{128}, 2^{256}, 2^{512}$.

Таблиця 7

lh\k	2^{32}	2^{64}	2^{128}	2^{256}	2^{512}
256	6 *10-117	2*10-78	0.154	1	1
512	6.8 *10-136	1.6*10-232	1.8*10-155	0.154	1

Таблиця 8

lh\k	2^{32}	2^{64}	2^{128}	2^{256}	2^{512}
256	1.6*10-465	2.2*10-311	0.0237	1	1
512	2.2 *10-1087	6.7*10-928	1.2*10-619	0.154	1

При реалізації механізмів Ломпарта, Ломпарта – Діффі та Вінтерніц при n=256 для реальних значень k практично отримуємо, що $P(n, p_\rho(N,k)) = 0.$ Так, навіть при $k = 2^{128}$ та $l_h = 256$ згідно (49) та (50) отримуємо $P(n, p_\rho(N,k)) = (0.393)^{256} = \leq 10^{-52}$.

Висновки

1. Можливо, механізми ЕП на основі геш-функцій є найбільш перспективними. Стійкість таких механізмів грунтується на використанні однонаправлених та криптографічних геш-функцій та генераторів випадкових та/чи псевдовипадкових послідовностей. Щодо більшості механізмів на основі геш-функцій є докази криптографічної стійкості як до класичних так і квантових атак. Також ЕП на основі геш-функцій практично є прийнятними як щодо складності (швидкодії), так і довжини ЕП та ключів.

2. Особливістю механізмів ЕП на основі геш-функцій є те, що їх криптографічна стійкість грунтується на колізійній стійкості функцій гешування та/чи стійкості до знаходження прообразу. Так як нині уже розроблено та прийнято у якості геш-функцій ряд, по суті стандартизованих постквантових геш-функцій, то такі механізми, на наш погляд, є перспективними. Але залишається ряд проблемних питань, що повинні певною мірою бути вирішеними до початку етапу стандартизації, в тому числі: доведення криптографічної

стійкості; мінімізація розмірів загальних параметрів та ключів; мінімізація довжини підпису та підвищення швидкодії; вироблення та перевіряння ЕП тощо відповідно до вимог.

3. Механізми ЕП з одноразовими ключами Лампорта, Лампорта – Діффі та Вінтерніц дійсно забезпечують як практичну, так і теоретичну криптографічну стійкість проти існуючих атак і можуть бути віднесені до криптоперетворень з бездоганним ЕП. Основними недоліками, що характерні для вказаного класу ЕП з одноразовими ключами, є великі довжини секретних та відкритих ключів, а також довжина ЕП.

4. На наш погляд, запропоноване удосконалення ЕП на основі алгоритму Вінтерніц має суттєві переваги щодо зменшення довжин ключів та ЕП, в той же час при обґрунтованому виборі параметрів механізму забезпечується криптографічна стійкість до атак модифікації та створення хибних підписів. Але вказаний клас ЕП з одноразовими ключами є складним, та скоріше може бути застосований для захисту повідомлень критичного рівня.

5. Розміри секретних та відкритих одноразових ключів та розмірів ЕП для механізму Вінтерніц у порівнянні з механізму Лампорта вимагають збільшення розмірів секретних та відкритих одноразових ключів від 2 до 64 разів, але розміри ЕП зменшуються в 2 рази.

6. Розміри секретних та відкритих одноразових ключів та розмірів ЕП для удосконаленого механізму у порівнянні з механізмом Лампорта можуть бути зменшені для довжини ЕП 256 від 2 до 128 разів, а для довжини ЕП 512 – від 2 до 512 разів. При цьому для μ_i 256 та 512 відповідні значення табл. 8 співпали з відповідними даними табл 6.

7. Розміри секретних та відкритих одноразових ключів та розмірів ЕП для удосконаленого механізму у порівнянні з механізмом Вінтерніц для довжини 256 можуть бути зменшені для довжини ЕП 256 від 64 до 256 разів, а для довжини ЕП 512 – від 64 до 512 разів. При цьому довжини ЕП зменшуються ввід 8 до 32разів (для довжини геш-значення 256), а для довжини 512 від 2 до 128 разів.

8. У цілому необхідно зробити висновок, що удосконалений механізм забезпечує зменшення розмірів секретних та відкритих ключів, а також розмір ЕП. Удосконалення механізму Лампорта зводиться до застосування іншого правила зашифрування (31). Якщо значення b_i блоку знаходиться в інтервалі 931), то b_i блок зашифровується (заміняється) послідовно
секретним ключем із множини (28) X, інакше заміняється послідовно секретним ключем із множини (28) Y.

9. Попередній аналіз показав, що основними загрозами щодо ЕП Лампорта, Лампорта – Діффі, Вінтерніц та удосконаленого з разовими ключами є підробка ЕП та створення хибного ЕП. Вказані загрози можуть бути потенційно реалізованими засобом здійснення силових атак типу «брутальна сила» та аналітичного типу на основі квантових алгоритмів, наприклад Гровера.

10. Обґрунтовані співвідношення (37) – (44) отримані на основі математичних методів Полларда та, по суті, є реалізацією алгоритму Гровера. На основі їх використання отримані аналітичні оцінки складності криптоаналізу типу повне розкриття для усього класу ЕП з одноразовими ключами.

Список літератури: 1. *A RIDDLE* WRAPPED IN AN ENIGMA. NEAL KOBLITZ AND ALFRED J. MENEZES Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A. 2. *Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone.* Report on Post – Quatum Cryptography. NISTIR 8105 (DRAFT). https://www.google.com.ua/search? 3. DRAFT – DRAFT – DRAFT. Proposed Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. http://www.nist.gov/pqcrypto. 4. *Горбенко, І.Д., Кузнєцов, О.О., Потій, О.В., Горбенко, Ю.І., Ганзя, Р.С., Пономар, В.А.* Постквантова криптографія та механізми її реалізації // Радиотехника. – 2016. – Вып. 186. – С. 32–52. 5. *Leslie Lamport.* Constructing digital signatures from a one way function. Technical. Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979. 6. *Andreas Hülsing.* W-OTS+ – shorter signatures for hash-based signature schemes. In Amr Youssef, Abderrahmane Nitaj, and Aboul-Ella Hassanien, editors, Progress. in

Cryptology – AFRICACRYPT 2013, volume 7918 of LNCS, pages 173–188.Springer, 2013. 7. *Ralph Merkle*.
A certified digital signature. In Gilles Brassard, editor, Advances inCryptology – CRYPTO ’89, volume 435 of LNCS, pages 218–238. Springer, 1990. 8. *Daniel J. Bernstein;, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O’Hearn*. SPHINCS: practical stateless hash-based Signatures. djb@cr.yp.to. daira@leastauthority.com, zooko@leastauthority.com 9. *Gorbenko, I., Ponomar, V.* Examining a possibility to use and the benefits of post-quantum algorithms dependent on the conditions of their application // Eastern European Journal of Enterprise Technologies, Vol.2, Issue 9-86, 2017, Pages 21-32. http://journals.uran.ua/eejet/article/view/96321/94881. 10. *ETSI GR* QSC 001 V.1.1.1 (2016-07). Quntum-Safe Cryptography (QSC); Quantum-safe algorithmic framework. 11. *Горбенко, Ю.І.* Методи побудування та аналізу, стандартизація та застосування криптографічних системи ; за заг. ред. Горбенка І.Д. – Харків : Форт, 2016. – 958с. 12. *Gorbenko, I., Kuznetsov, A., Gorbenko, Yu., Kavun, S., Kachko, O., Yesina, M.* Electronic Signature Mechanisms. The Current State, the Existing Contradictions and Prospects of Practical Use for the Post-Quantum Period: Monograph. – ASC Academic Publishing, USA, 2017. – 165 p. 13. *Grover’s Quantum* Search Algorithm [електронний ресурс] – Режим доступу: URL: http://twistedoakstudios.com/blog/Post2644_grovers-quantum-search-algorithm. 14. *Кузнецов, А.А., Пушкарев, А.И., Сватовский, И.И., Шевцов, А.В.* Несимметричные криптосистемы на алгебраических кодах для пост-квантового периода // Радиотехника. – 2016. – Вып. 186. – С. 70-90. 15. *Кузнецов, О., Горбенко, Ю., Шевцов, О., Кузнецова, Т.* Дослідження криптографічних атак на схеми електронного цифрового підпису в фактор-кільцях зрізаних поліномів // Захист інформації. – Київ : Національний авіаційний університет, 2016. – Т. 18, №4, жовтень-грудень 2016. – С. 293-300.

Харківський національний університет
імені В.Н. Каразіна
Надійшла до редколегії 20.04.2017

РАДИОТЕХНИЧЕСКИЕ И ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ

УДК 621.396.967.2

І.І. ОБОД, д-р техн. наук, І.В. СВИД, канд. техн. наук, І.А. ШТИХ

СИНТЕЗ ТА АНАЛІЗ ОПТИМАЛЬНОГО ВИЯВЛЮВАЧА СИГНАЛІВ ЗАПИТУ У ЛІТАКОВИХ ВІДПОВІДАЧАХ ВТОРИННИХ СИСТЕМ СПОСТЕРЕЖЕННЯ

Постановка проблеми й аналіз літератури

Основою інформаційного забезпечення споживачів системи контролю використання повітряного простору є первині системи спостереження (СС), тобто системи, що працюють з ехо-сигналами, та вторинні СС, тобто системи, що працюють за сигналами відповіді (СВ). Основним елементом, який істотно знижує завадостійкість вторинних СС, є літаковий відповідач (ЛВ) [1 – 4]. ЛВ призначений для роботи з радіолокаційними системами (РЛС), які входять до системи управління повітряним рухом, і забезпечує автоматичну видачу цим РЛС інформації про координати літака, бортовий номер, барометричну висоту польоту, а також сигнали індивідуального опізнавання та аварії. Головною функцією ЛВ є приймання сигналів від наземних РЛС, комбінування інформаційного повідомлення – відповіді та передавання знову назад на обладнання РЛС. Саме принцип побудови останнього, принцип обслуговування сигналів запиту (СЗ) знижують завадостійкість як ЛВ, так і вторинних СС в цілому. Наявність багатоканальності в прийомі СЗ розширює структурні можливості при побудові виявлювачів СЗ, зокрема, в варіантах об'єднання попередніх рішень каналів виявлення. Однак в існуючих ЛВ реалізований квазіоптимальний виявлювач СЗ при багатоканальному прийомі з об'єднанням канальних рішень виявлення СЗ.

Мета роботи

Синтезування та аналіз оптимального виявлювача сигналів запиту в літакових відповідачах вторинної системи спостереження.

Основна частина

Виявлювач СЗ в ЛВ є багатоканальним. Це обумовлено наявністю декількох антенних систем, що працюють як на прийом СЗ, так і випромінювання СВ [4]. Після порогових пристроїв і дешифраторів сигнали підсумовуються елементом об'єднання. Однак слід враховувати, що параметри СЗ, прийняті різними каналами, істотно відрізняються, що не враховується при побудові виявлювачів сигналів в існуючих ЛВ. Крім того, в існуючих ЛВ об'єднанню підлягають попередні рішення про виявлення СЗ, здійснені, як правило, дешифратором, тобто квазіоптимальним виявлювачем. Однак СЗ, як відомо [5], містять кілька простих сигналів без внутріімпульсной модуляції, часова розстановка яких і визначає код СЗ. Ці обставини дозволяють синтезувати оптимальний виявлювач СЗ в двох різних постановках:

- виявлення СЗ з ваговим міжканальним об'єднанням канальних рішень про виявлення СЗ;
- виявлення СЗ з ваговим міжканальним об'єднанням канальних імпульсів СЗ.

Будемо вважати, що число каналів прийому СЗ дорівнює m, а число імпульсів в СЗ становить n (значность коду). Отримаємо загальний алгоритм виявлення сукупності одиничних рішень і на підставі отриманого алгоритму розглянемо структури виявлювачів СЗ в ЛВ при зазначених вище постановках.

У кожному з каналів обробки ЛВ прийняті сигнали після оптимальної лінійної обробки і детектування порівнюються в ПП з порогом. Після ПП на подальшу обробку надходить реалізація $x_{ij}=1$, якщо в елементі часового дозволу $(i=\overline{1,m})$ і $(j=\overline{1,n})$, відповідному аналізує-

мому просторовому дозволу, відбулося перевищення порога; якщо ж не сталося – то $x_{ij}=0$. Для прийняття рішення про наявність або відсутність сигналу при спільній міжканальній обробці піддається сукупність нулів і одиниць x_{ij}. Очевидно, що x_{ij} – випадкова величина, що підкоряється розподілу Бернуллі

$$P(x_{ij})=P_{ij}^{x_{ij}}\left(1-P_{ij}\right)^{1-x_{ij}}, \tag{1}$$

де P_{ij} – ймовірність перевищення порога в ij-му каналі обробки. У відсутності сигналу $P_{ij}=F_{ij}$ – ймовірність хибної тривоги, а при впливі сигналу $P_{ij}=D_{ij}$ – ймовірність виявлення.

Задачу оптимальної обробки сигналів можна розглядати в різних постановках. Дійсно, в розглянутому виявлювачі можливе управління напругою порога спрацьовування вихідного ПП, а також напругою порога канальних ПП. Розглянемо характеристики виявлювача при управлінні величиною порога тільки на вихідному ПП. Ймовірності хибної тривоги і правильного виявлення сигналів в каналах обробки будемо вважати заданими (хоча і довільними).

Припустимо, що на вхід пристрою спільної обробки прийнятих сигналів надходить сукупність випадкових величин x_{ij}. Спільні розподілу ймовірностей всіх можливих комбінацій x_{ij} як у відсутність, так і при наявності сигналу (гіпотези H_0 та H_1), тобто $P\left(x_{ij}\middle|H_0\right)$ та $P\left(x_{ij}\middle|H_1\right)$ довільні, але відомі. Для кожної конкретної сукупності x_{ij} сформуємо відношення правдоподібності

$$\Lambda=P\left(x_{ij}\middle|H_1\right)\Big/P\left(x_{ij}\middle|H_0\right). \tag{2}$$

Порівняння Λ з порогом, визначеним за допустимої ймовірності хибної тривоги, забезпечує оптимальне за критерієм Неймана – Пірсона рішення про наявність або відсутність сигналу.

Через незалежності шумів в каналах обробки можна записати

$$P(x_{ij}\mid H_0)=\prod_{i=1,j=1}^{m,n}F_{ij}^{x_{ij}}\left(1-F_{ij}\right)^{1-x_{ij}}. \tag{3}$$

Легко бачити, що при впливі сигналу перевищення порогів в каналах обробки – незалежні події. Тоді можна записати

$$P(x_{ij}\mid H_1)=\prod_{i=1,j=1}^{m,n}D_{ij}^{x_{ij}}\left(1-D_{ij}\right)^{1-x_{ij}}. \tag{4}$$

З урахуванням (3) і (4) вираз (2) можна записати як

$$\Lambda=\prod_{i=1,j=1}^{m,n}D_{ij}^{x_{ij}}\left(1-D_{ij}\right)^{1-x_{ij}}\Bigg/\prod_{i=1,j=1}^{m,n}F_{ij}^{x_{ij}}\left(1-F_{ij}\right)^{1-x_{ij}}. \tag{5}$$

Прологарифмував (5), отримуємо

$$L=\ln\Lambda=\sum_{i=1,j=1}^{m,n}x_{ij}\left(\ln D_{ij}-\ln F_{ij}\right)+\left(1-x_{ij}\right)\left[\ln\left(1-D_{ij}\right)-\ln\left(1-F_{ij}\right)\right] \tag{6}$$

Якщо позначити множники при x_{ij}

$$Q_{ij} = \ln D_{ij} - \ln F_{ij} - \ln(1 - D_{ij}) + \ln(1 - F_{ij}) = D_{ij}(1 - F_{ij})/(1 - D_{ij})F_{ij}. \quad (7)$$

і відкинути доданки, які не залежать від x_{ij}, отримуємо оптимальний за критерієм Неймана – Пірсона алгоритм виявлення сигналів запиту при об'єднанні попередніх рішень виявлення сигналів або імпульсів всіх каналів обробки ЛВ

$$L = \sum_{i=1,j=1}^{m,n} Q_{ij} x_{ij} \gtrless z_0, \quad (8)$$

де z_0 – поріг, який визначається вихідною ймовірністю F.

Отже, оптимальна спільна обробка СЗ зводиться до вагового підсумовування одиниць і нулів x_{ij}, що відображають прийняті в каналі обробки попередні рішення. Вагові коефіцієнти (7) підвищують роль того каналу, де вища ймовірність D_{oij} і нижча ймовірність F_{oij} Вагові коефіцієнти (7) залежать як від відношення с/ш, так і від рівня шумів в різних каналах обробки ЛВ.

Оскільки x_{ij} дорівнює 0 чи 1, то ліва частина (8) представляє собою суму $k = mn$ вагових коефіцієнтів Q_{ij}, а значить, може приймати тільки певні дискретні значення. Значення порогу z_0 в цьому випадку може лежати в межах $0 < z_0 < \sum_{i=1,j=1}^{m,n} Q_{ij}$, щоб, з одного боку, не приймалося завжди тривіальне рішення про виявлення, а з іншого – тривіальне рішення про невиявлення.

Якщо всі Q_{ij} різні і сума будь-якої групи Q_{ij} не збігається з сумою будь-якої іншої їх групи, то при різних комбінаціях значень x_{ij} для розглядаємого нами випадку можливі $2^m - 1$ різних правил виявлення.

Слід зазначити, що підсумовування імпульсів сигналу запиту в каналах обробки здійснюється без ваг, через однакові відносини с/ш і рівня завад в каналі, що спрощує алгоритм обробки. Зокрема, виявлювач сигналів в каналах для першої ситуації або крайовий виявлювач сигналів для другої ситуації може бути виконаний у вигляді дешифратора з цілої логікою обробки («n з n»). Безвесове підсумовування нулів та одиниць в каналах обробки і заміна виявлювача СЗ дешифратором не приводять до істотних втрат в пороговому відношенні с/ш.

В цьому випадку, для розглянутих ситуацій вираз (8) можна записати:

- при міжканальному об'єднанні результатів виявлення СЗ

$$L = \sum_{i=1}^{m} Q_i \times \left(x_i = \prod_{j=1}^{n} x_j \right) \gtrless z_0, \quad (9)$$

- при міжканальному об'єднанні результатів виявлення імпульсів СЗ

$$L = \prod_{j=1}^{n} \left(x_j = \sum_{i=1}^{m} Q_i x_i \gtrless z_0 \right). \quad (10)$$

Отримані алгоритми (9) і (10) дозволяють викласти структурні схеми виявлювачів сигналів запиту, для розглянутих ситуацій між канального об'єднання попередніх канальних рішень про виявлення сигналів або імпульсів. В синтезованих виявлювачах є три порогових пристрої: перший – пороговий пристрій з аналоговим порогом, де відбувається виявлення

імпульсів сигналів запиту, другий – в дешифраторі (цифровий поріг) і третій – при виявленні об'єднаних імпульсів (сигналів) (цифровий поріг).

Таким чином, оптимізація виявлення сигналів запиту в літакових відповідачах зводиться до вибору для спільної обробки одного з вирішальних правил, що задовольняють алгоритму (8), (9) і (10), і до установки однакових відносних порогів в каналах обробки сигналів запиту в літакових відповідачах, що забезпечують такі значення F_i, які при вибраному вирішальному правилі дають необхідне значення результуючої ймовірності F.

Розрахунок показників виявлення сигналів запиту за наведеними вище виразами досить складний через необхідність розгляду відмінності завадових коливань і відносини с/ш в каналах обробки. Припустимо, що число каналів обробки сигналів запиту складає m. У кожному каналі обробки однакове ставлення с/ш. В цих умовах вагові коефіцієнти внутриканального і міжканального об'єднання однакові, а розрахункові вирази для показників виявлення спрощуються.

Розрахунки виявлення сигналів запиту в літакових відповідачах для $m = 2$ представлені на рис.1, а для $m = 3$ – на рис. 2.

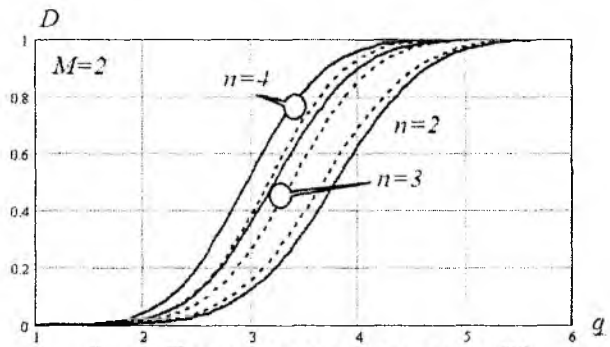


Рис.1. Показники якості виявлення СЗ

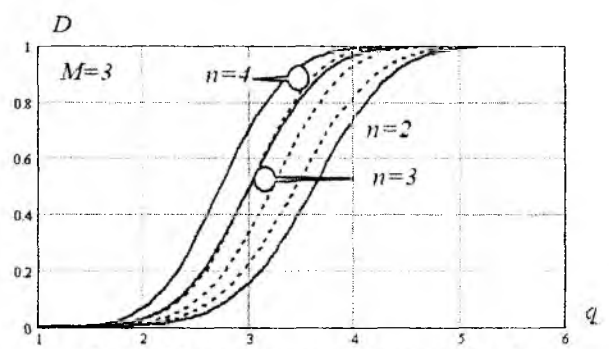


Рис.2. Показники якості виявлення СЗ

Висновки

Міжканальне об'єднання результатів виявлення імпульсів дозволяє отримати переваги в пороговому відношенні с/ш (близько 1 дБ) порівняно з міжканальним об'єднанням результатів виявлення сигналів запиту.

Збільшення значності використовуваних сигналів запиту вторинних систем спостереження дозволяє підвищити ймовірність виявлення їх в літакових відповідачах.

Список літератури: 1. *Автоматизированные* системы управления воздушным движением: Новые информационные технологии в авиации ; под ред. С.Г. Пятко и А.И. Краснова. – СПб. : Политехника, 2004. 2. *Агаджанов, П.А., Воробьев, В.Г., Кузнецов, А.А.* Автоматизация самолетовождения и управления воздушным движением. – М. : Транспорт, 1980. – 342 с. 3. *Фарина, А.* Цифровая обработка радиолокационной информации / А.Фарина, Ф.Студер. – М. : Радио и связь, 1993. – 319 с. 4. *Давыдов, П.С., Жаворонков, В.П., Кащеев, Г.В.* Радиолокационные системы летательных аппаратов. – М. : Транспорт, 1977. – 356 с. 5. *Обод, І.І.* Завадозахищеність вторинних систем спостереження повітряного простору / І.І. Обод, І.В. Свид, І.А. Штих. – Х. : ХНУРЕ, 2014. – 310 с.

Харківський національний університет радіоелектроніки

Надійшла до редколегії 05.04.2017