ISSN 1727-3366

KHARKIV NATIONAL UNIVERSITY OF RADIO ELECTRONICS

ACADEMY OF SCIENCES OF APPLIED RADIO ELECTRONICS

APPLIED RADIO ELECTRONICS

Scientific and Technical Journal 2018 Volume 17 № 3, 4

Special issue devoted to problems of ensuring information security

CONTENTS

ПЕРСПЕКТИВНЫЕ МЕТОДЫ ГЕНЕРАЦИИ КЛЮЧЕЙ И КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

УДК 004.056.55

УСКОРЕННЫЙ МЕТОД ВЫЧИСЛЕНИЯ АЛГЕБРАИЧЕСКОЙ ИМУННОСТИ НЕЛИНЕЙНЫХ УЗЛОВ ЗАМЕНЫ СИММЕТРИЧНЫХ ШИФРОВ

К. Е. ЛИСИЦКИЙ, А. А. КУЗНЕЦОВ, Ю. И. ГОРБЕНКО, В. В. ОНОПРИЕНКО, И. В. СТЕЛЬНИК

Рассматривается ускоренный метод вычисления алгебраической иммунности нелинейных узлов замены симметричных шифров по Жан-Шарлю Фожеру (Jean-Charles Faugère). Он основан на поиске аннигилирующей функции к полиному Жегалкина, построенному от исходного нелинейного узла замены. Приводится анализ быстродействия известного алгоритма подсчёта алгебраической иммунности. Обсуждаются детали реализации алгоритма и приводится описание ускоренного алгоритма вычисления алгебраической иммунности, оптимизированного по времени вычисления и по ресурсам задействования ОЗУ

Ключевые слова: симметричный шифр, алгебраический иммунитет, нелинейный узел замены, булева функция, производительность алгоритма.

ВВЕДЕНИЕ

Важное место среди криптографических алгоритмов защиты информации занимают симметричные (блочные и поточные) шифры. Благодаря своей простоте, быстродействию и надежности они широко используются во многих криптографических приложениях [1–4].

Исследование и анализ симметричных криптосистем, а также их отдельных составных узлов, является актуальной и важной научно-технической задачей. В этой статье речь пойдёт об одном из важнейших примитивов любого симметричного шифра – нелинейном узле замены или, так называемом, S-блоке. Процедура прохождения S-блокового преобразования в шифре обеспечивает дополнительное рассеивание и перемешивание байтов текста, что существенным образом усложняет реализацию различных криптоаналитических атак [3]. А точнее, речь пойдёт об алгебраической иммунности, как об одном из криптографических показателей S-блоков, позволяющих аналитически оценить стойкость симметричных шифров к алгебраическим атакам [5–10]. Основная идея алгебраических атак строится на поиске возможности описания шифрующего преобразования с помощью системы уравнений, связывающих между собой биты открытого текста, ключа и шифртекста [5–7].

Как показывает анализ, расчёт алгебраической иммунности является нетривиальной задачей [11-13]. В частности, для расчета алгебраической иммунности в терминах, введенных Жан-Шарлем Фожером (Jean-Charles Faugère) [11], необходимо оценить минимальную степень полинома из минимального редуцированного базиса Грёбнера идеала, заданного системой уравнений, описывающих нелинейный узел замен. В [13] приведены примеры расчета алгебраической иммунности нелинейных узлов некоторых блочных симметричных шифров. Построение минимального редуцированного базиса Грёбнера при этом реализовано с помощью системы компьютерной алгебры «Magma» [14].

Другой подход к расчету алгебраической иммунности S-блока (векторного отображения) может быть реализован через нахождение минимальной степени ненулевых аннигиляторов булевой функции, описывающей этот S-блок [13]. Действительно, в работе [15, с. 337] показано, что произвольный S-блок с n входами и m выходами может быть однозначно представлен булевой функцией от $n+m$ переменных и алгебраическая иммунность векторного отображения (S-блока) совпадает с минимальной степенью ненулевых аннигиляторов этой функции. Расчет алгебраической иммунности через поиск ненулевых аннигиляторов связан с решением большеразмерной системы уравнений, требующей значительных вычислительных ресурсов и объёма памяти, выходящих за практически приемлемые границы.

В данной работе ставится задача изучить метод расчёта алгебраической иммунности, основанный на решении систем линейных уравнений для нахождения ненулевых аннигиляторов к функции в виде полинома Жегалкина, построенной по исходному S-блоку [13, 15]. В работе предлагается алгоритм расчёта алгебраической иммунности, дается оценка сложности в сравнении с известными результатами.

1. ОБЗОР ЛИТЕРАТУРЫ

Интерес, вызванный в своё время к алгебраическим методам криптоанализа блочных и поточных шифров в 2003 г. работами N. Courtois и W. Meier [1]

не утихает и сегодня. Результатом повышения внимания к этим методам стало появление нового криптографического показателя эффективности S-блоков – алгебраической иммунности. В работе [6] напоминается, что понятие алгебраической иммунности для булевых функций было введено в 2004 г. W. Meier, E. Pasalic и C. Carlet в [7]. Алгебраической иммунностью $AI(f)$ булевой функции $f: Z_2^n \to Z_2$ называется минимальное число d такое, что существует булева функция g степени d, не тождественно равная нулю, для которой $fg = 0$ или $(f \oplus 1)g = 0$ [3, 16]. Булева функция $g \in V_n$ называется *аннигилятором* функции $f \in V_n$, множество различных аннигиляторов образует линейное пространство $Ann(f) = \{g \in V_n \mid f \cdot g = 0\}$ [16].

Далее процитируем ещё одну выдержку из работы [3]. Понятие алгебраической иммунности различными способами было обобщено на векторный случай. Так, в работе [9] F. Armknecht и M. Krause, а также G. Ars и J.-C. Faug`ere в [10] рассмотрели алгебраическую иммунность S-блоков и ввели понятия базовой $AI(F)$ и графической $AI\mathrm{gr}(F)$ алгебраической иммунности векторных булевых функций. При этом базовая алгебраическая иммунность больше 1 только при малых значениях m, поэтому данный параметр анализируется у S-блоков, которые используются в поточных шифрах. Графическая алгебраическая иммунность используется для изучения сопротивляемости алгебраическим атакам блочных шифров. Следующее обобщение, которое воспринимается многими исследователями одним из наиболее естественных с криптографической точки зрения, это компонентная алгебраическая иммунность. Компонентной алгебраической иммунностью $AI_{\mathrm{comp}}(F)$ векторной булевой функции F: $Z_2^n \to Z_2^m$ называется минимальная алгебраическая иммунность компонентных функций $b\,F$ ($b \in Z_2^m$, $b \neq 0$), т. е. $AI_{\mathrm{comp}}(F) = \min\{AI(b\,F) : b \in Z_2^m,\ b \neq 0\}$, где $b\,F = b_1 f_1 \oplus \ldots \oplus b_m f_m$ [10]. В случае компонентной алгебраической иммунности в [10] также получено, что $AI_{\mathrm{comp}}(F) \le \lfloor n/2 \rfloor$.

Ещё один метод определения алгебраической иммунности нелинейных узлов замены по Жан-Шарлю Фожеру, строится с применением базисов Грёбнера [17]. Предположим, что S-блок задается системой алгебраических уравнений над двоичным полем:

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = y_1, \\ f_2(x_1, x_2, \ldots, x_n) = y_2, \\ \ldots \\ f_m(x_1, x_2, \ldots, x_n) = y_m, \end{cases} \quad (1)$$

т.е. совокупностью булевых многочленов

$$\begin{aligned} & y_1 - f_1(x_1, x_2, \ldots, x_n), \\ & y_2 - f_2(x_1, x_2, \ldots, x_n), \\ & \ldots, \\ & y_m - f_m(x_1, x_2, \ldots, x_n) \end{aligned} \quad (2)$$

в кольце $K[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m]$ от переменных $x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m$ с коэффициентами над полем $K = GF(2)$.

С системой уравнений (1), алгебраически задающих структуру S-блока, свяжем идеал $I(S)$ в кольце $K[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m]$ (обозначается как $I(S) \lhd K[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m]$), порожденный многочленами (2):

$$I(S) = (y_1 - f_1, y_2 - f_1, \ldots, y_m - f_m) = \left\{ \begin{matrix} (y_1 - f_1) \cdot r_1 + (y_2 - f_1) \cdot r_2 + \ldots + (y_m - f_m) \cdot r_m; \\ r_1, r_2, \ldots, r_m \in K[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m] \end{matrix} \right\}.$$

Алгебраическая иммунность $AI(S)$ нелинейного узла (S-блока) определяется как минимальная степень многочлена P из идеала $I(S)$ [11]:

$$AI(S) = \min\{\deg(P), P \in I(S) \lhd K[x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m]\}, \quad (3)$$

причем минимальный редуцированный базис Грёбнера идеала $I(S)$ при степенном обратном словарном упорядочении (degrevlex) содержит линейный базис полиномов P из $I(S)$, таких, что $AI(S) = \deg(P)$. Другими словами, для вычисления алгебраической иммунности $AI(S)$ достаточно построить минимальный редуцированный базис Грёбнера идеала $I(S)$, заданного уравнениями (2) и найти многочлен минимальной степени среди элементов этого базиса.

Связь алгебраической иммунности S-блока и булевой функции показана в [15, с. 337]. Рассмотрим булевую функцию

$$f_S(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) : GF(2)^{n+m} \to GF(2),$$

значения которой определим следующим образом:

$$f_S(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) = \begin{cases} 1, \forall i, j : f_i(x_1, x_2, \ldots, x_n) = y_j, \\ 0, \forall i, j : f_i(x_1, x_2, \ldots, x_n) \neq y_j. \end{cases} \quad (4)$$

Множество решений уравнения

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

совпадает с множеством решений системы (1). Следовательно, имеем различные базисы $(f_S - 1)$ и $(y_1 - f_1, y_2 - f_1, \ldots, y_m - f_m)$ одного идеала эквивалентных систем, т.е.

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

Идеал пространства аннигиляторов $Ann(f_S)$ в кольце $K[x_1, x_2, ..., x_n, y_1, y_2, ..., y_m]$ совпадает с идеалом $I(f_S - 1)$, следовательно, алгебраическая иммунность (3) булевого отображения $S: GF(2)^n \to GF(2)^m$ совпадает с минимальной степенью ненулевых полиномов, принадлежащих аннигилятору функции f_s: $AI(S) = \min\{D\deg(g) \mid g \in Ann(f_S)\}$ [13].

Исследованию различных алгоритмов расчета алгебраической иммунности нелинейных узлов замены и посвящена данная работа.

2. АЛГОРИТМЫ РАСЧЕТА АЛГЕБРАИЧЕСКОГО ИММУНИТЕТА

Прежде чем привести конкретные алгоритмы вычисления алгебраической иммунности, приведём некоторые обозначения, которые вводятся в работах [8, 16].

Моном (одночлен) относительно переменных $x_1, ..., x_n$ запишем в виде

$$x^u = \prod_{i=1}^{n} x_i^{u_i} = \begin{cases} x_i, u_i = 1, \\ 1, u_i = 0, \end{cases} \quad (5)$$

где вектора $x, u \in V_2^n, x = (x_1, ..., x_n), u = (u_1, ..., u_n)$.

Степень одночлена x^u определяется весом Хемминга (числом ненулевых координат) $w_h(u)$ вектора $u = (u_1, ..., u_n)$, т.е.

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

С учетом этих обозначений булеву функцию $f(x)$ в алгебраической нормальной форме (в форме полинома Жегалкина) запишем в виде

$$f(x) = \sum_{u \in GF(2)^n} a_u x^u, \; a_u \in GF(2). \quad (6)$$

Функцию (аннигилятор) $g \in A_d^n(f)$ также представим в виде полинома Жегалкина

$$g(x) = \sum_{v \in GF(2)^n : w_h(v) \le d} b_v x^v, \quad (7)$$

где $b_v \in GF(2)$ – неизвестные коэффициенты аннигилятора, $w_h(v)$ – вес Хемминга вектора $v = (v_1, ..., v_n)$.

Линейное пространство аннигиляторов степени $\le d$ обозначим

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

Функция $g(x)$ принадлежит пространству $A_d^n(f)$ только в том случае, если для любого $x \in GF(2)^n$ выполняется равенство $f(x) \cdot g(x) = 0$.

Приведём алгоритм вычисления алгебраической иммунности булевой функции [16] и оценим его возможную стандартную реализацию по объёму вычислений.

Алгоритм вычисления алгебраической иммунности булевой функции

Вход: $n \in N$, функция $f(x)$ (заданная списком одночленов x^u с ненулевыми коэффициентами a_u в (6)).

Выход: Значение алгебраической иммунности $AI(f)$.

Шаг 1. Присваиваем $d = 1$.

Шаг 2. Вычисляем пространство аннигиляторов $A_d^n(f)$ и $A_d^n(f + 1)$.

Шаг 3. Если $A_d^n(f) = 0$ и $A_d^n(f + 1) = 0$ присваиваем $d = d + 1$ и переходим к шагу 2.

Шаг 4. Если $A_d^n(f) \ne 0$ и/или $A_d^n(f + 1) \ne 0$ присваиваем $AI(f) = d$ и подаем на выход алгоритма.

Для использования приведенного алгоритма для расчета алгебраической иммунности векторного отображения (S-блока) необходимо перевести (отобразить) нелинейный узел в булеву функцию $f(x)$ в соответствии с (4), например, в виде полинома Жегалкина (6).

Алгоритм преобразования последовательности (4) в полином Жегалкина в общем виде сводится к решению уравнений, слагаемыми в которых, являются все возможные “составные” мономы, относительно данного и сам искомый моном, в правой части уравнения стоит значение булевой функции, относительно искомого монома. Стандартное решение может выглядеть как поиск “вхождения” каждого следующего монома в искомый, т. е. (65536×65536) операций, без учёта вспомогательных.

Для оптимизации необходимо найти зависимость искомого монома от позиций возможных мономов “вхождения”. Если рассмотреть искомый моном и сохранить позиции значения 0, то можно найти закономерность позиций, на которых точно не будут находиться мономы, “входящие” в искомый. Следовательно, пропуская априори, неподходящие нам позиции, мы можем двигаться только по позициям возможного вхождения мономов в искомый, что существенно сокращает поиск мономов “вхождения” и повышает быстродействие алгоритма преобразования к полиному Жегалкина.

Теперь дополним предыдущий алгоритм двумя новыми первыми пунктами по преобразованию S-блока в полином Жегалкина. В итоге этот модифицированный алгоритм будет выглядеть следующим образом.

Алгоритм вычислении алгебраической иммунности S-блоков

Вход: $n \in N$, S-блок $V_n \to V_n$, заданный в виде квадратной таблицы.

Выход: Значение алгебраической иммунности $AI(f)$.

Шаг 1. Вычисляем отображение S-блока (4) в виде таблицы истинности соответствующей булевой функции с удвоенным числом переменных $f_s : V_{2n} \to \{0,1\}$.

Шаг 2. Вычисляем алгебраическую нормальную форму булевой функции $f_s : V_{2n} \to \{0,1\}$. Задаём её списком одночленов (5) с ненулевыми коэффициентами a_u, $f(x) = \sum_{u \in GF(2)^n} a_u x^u$, $a_u \in GF(2)$.

Присваиваем $d = 1$.

Шаг 3. Если $A_d^n(f) = 0$ и $A_d^n(f+1) = 0$ присваиваем $d = d + 1$ и переходим к шагу 2.

Шаг 4. Если $A_d^n(f) \neq 0$ и/или $A_d^n(f+1) \neq 0$ присваиваем $AI(f) = d$ и подаем на выход алгоритма.

Как видно при построении этого алгоритма уже предприняты шаги по сокращению объёма вычислений. Сначала вычисляется аннулирующий многочлен первой степени, затем второй, третей и вычисления заканчиваются, как только найден аннулирующий многочлен минимальной степени.

Приведём здесь результаты оценки вычислительной сложности этого алгоритма при его реализации без каких-либо упрощений.

Поиск пространства аннигиляторов осуществляется путём решения системы линейных уравнений, полученных на основе группирования неизвестных коэффициентов по всем различным мономам. Рассмотрим пример, приведенный в работе [13], где $f(x)$ – полученный многочлен в виде полинома Жегалкина; $g(x)$ – искомый аннулирущий многочлен.

Пример. Для $n = 2$ и $d = 1$ имеем:

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

$$g(x) = b_{00} + b_{10}x_1 + b_{01}x_2.$$

После подстановки в $f(x) \cdot g(x) = 0$ получим

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

откуда имеем систему линейных однородных уравнений:

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

относительно неизвестных $b_{00}, b_{10}, b_{01}, \ldots$ – коэффициентов функции $g(x)$.

Тогда, например, для функции $f(x) = x_1 + x_2$ (т.е. при $a_{00} = a_{11} = 0$ и $a_{10} = a_{01} = 1$) получим систему:

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

Для решения системы уравнений, например, методом Гаусса строится таблица размером $2^{2n} \times 2^{2n}$, где n – степень S-блока. Ячейки таблицы – сгруппированный результат умножения всех мономов полинома $f(x)$ на полином $g(x)$. Группировка происходит так, что строки представляются как все возможные варианты мономов (всего их 2^{2n}), а столбцы – это неизвестные коэффициенты (их также 2^{2n}). Таким образом, значение ячейки таблички может быть = 1, если коэффициент b_i присутствует в конкретном мономе и 0, если такой коэффициент отсутствует. Таблица 1 иллюстрирует этот процесс для примера $f(x) = x_1 + x_2$.

Таблица 1

Результирующая таблица сгруппированных коэффициентов

$b_{i,j} / x$	b_{00}	b_{01}	b_{10}	b_{11}
1	0	0	0	0
x_1	1	1	0	0
x_2	1	0	1	0
x_1x_2	0	1	1	0

Для байтового S-блока (n = 8), получаем таблицу размерами 65536×65536. Для хранения одной вспомогательной таблицы, которая содержит результат умножения всех возможных мономов полинома $f(x)$ на мономы полинома $g(x)$, необходимо приблизительно 8.5 Гб ОЗУ с учётом того, что каждое число в таблице может занимать 2 байта (максимально возможное значение 65535), это без учёта дополнительных вспомогательных таблиц, которые могут также понадобится в ходе вычислений.

Если говорить о временных затратах на вычисления, цифры нас также не порадуют, нам понадобятся неоднократные проходы по всей таблице (4294967296 операций на один проход) для решения систем линейных уравнений.

3. ОПТИМИЗИРОВАННЫЙ АЛГОРИТМ

Известно, что значение алгебраической иммунности S-блока, не может быть больше, чем $n/2$ [6], следовательно, для байтового S-блока значение алгебраической иммунности не может быть больше 4-х. Следовательно, при построении таблицы перемножения всех мономов полинома $f(x)$ на мономы полинома $g(x)$ мы можем брать только те мономы полинома $g(x)$, у которых степень будет меньше, либо равна $n/2$ (для байтового S-блока это четвёрка). Количество подходящих нам мономов полинома $g(x)$ в худшем случае, нетрудно посчитать по формуле

$$k = \sum_{i=1}^{n/2} C_{2n}^{i} . \qquad (8)$$

Для байтового S-блока получим:

$$k = C_{16}^{1} + C_{16}^{2} + C_{16}^{3} + C_{16}^{4} = 2516 .$$

Итого, 2516 мономов в худшем случае, вместо 65536 для байтового S-блока. Далее все вычисления при расчёте максимально возможного значения алгебраической иммунности будут происходить уже с таблицей размерами (65536×2516), а это уже 164888576 операций для прохода по всей таблице, вместо 4294967296. И около 0.3 Гб ОЗУ на ПК, вместо 8.5 для хранения такой таблицы. Возникает только одна проблема – мономы полинома никак не упорядочены по степеням, и хранятся в произвольном порядке относительно степеней, а точнее в порядке, который задаёт арифметика цифр (0...65536). Например, моном первой степени можно встретить в позициях (1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768), где позиции > 2516 не учитываются нашей оптимизацией. Для решения этой проблемы, необходимо сгруппировать полином $g(x)$ таким образом, чтобы мономы были упорядочены по степеням. Например, можно хранить позиции интересующих нас мономов отдельно и обращаться только к нужным позициям. Это позволит работать с таблицей (65536×2516), вместо (65536×65536) и во много раз сократит временные затраты на вычисления и ОЗУ на ПК.

4. ЭКСПЕРИМЕНТЫ

В качестве тестов производительности оптимизированного и стандартного решения были взяты узлы нелинейной замены шифра AES и шифра Калина (рис. 1).

S-блок AES:

```
0x63 0x7c 0x77 0x7b 0xf2 0x6b 0x6f 0xc5 0x30 0x01 0x67 0x2b 0xfe 0xd7 0xab 0x76
0xca 0x82 0xc9 0x7d 0xfa 0x59 0x47 0xf0 0xad 0xd4 0xa2 0xaf 0x9c 0xa4 0x72 0xc0
0xb7 0xfd 0x93 0x26 0x36 0x3f 0xf7 0xcc 0x34 0xa5 0xe5 0xf1 0x71 0xd8 0x31 0x15
0x04 0xc7 0x23 0xc3 0x18 0x96 0x05 0x9a 0x07 0x12 0x80 0xe2 0xeb 0x27 0xb2 0x75
0x09 0x83 0x2c 0x1a 0x1b 0x6e 0x5a 0xa0 0x52 0x3b 0xd6 0xb3 0x29 0xe3 0x2f 0x84
0x53 0xd1 0x00 0xed 0x20 0xfc 0xb1 0x5b 0x6a 0xcb 0xbe 0x39 0x4a 0x4c 0x58 0xcf
0xd0 0xef 0xaa 0xfb 0x43 0x4d 0x33 0x85 0x45 0xf9 0x02 0x7f 0x50 0x3c 0x9f 0xa8
0x51 0xa3 0x40 0x8f 0x92 0x9d 0x38 0xf5 0xbc 0xb6 0xda 0x21 0x10 0xff 0xf3 0xd2
0xcd 0x0c 0x13 0xec 0x5f 0x97 0x44 0x17 0xc4 0xa7 0x7e 0x3d 0x64 0x5d 0x19 0x73
0x60 0x81 0x4f 0xdc 0x22 0x2a 0x90 0x88 0x46 0xee 0xb8 0x14 0xde 0x5e 0x0b 0xdb
0xe0 0x32 0x3a 0x0a 0x49 0x06 0x24 0x5c 0xc2 0xd3 0xac 0x62 0x91 0x95 0xe4 0x79
0xe7 0xc8 0x37 0x6d 0x8d 0xd5 0x4e 0xa9 0x6c 0x56 0xf4 0xea 0x65 0x7a 0xae 0x08
0xba 0x78 0x25 0x2e 0x1c 0xa6 0xb4 0xc6 0xe8 0xdd 0x74 0x1f 0x4b 0xbd 0x8b 0x8a
0x70 0x3e 0xb5 0x66 0x48 0x03 0xf6 0x0e 0x61 0x35 0x57 0xb9 0x86 0xc1 0x1d 0x9e
0xe1 0xf8 0x98 0x11 0x69 0xd9 0x8e 0x94 0x9b 0x1e 0x87 0xe9 0xce 0x55 0x28 0xdf
0x8c 0xa1 0x89 0x0d 0xbf 0xe6 0x42 0x68 0x41 0x99 0x2d 0x0f 0xb0 0x54 0xbb 0x16
```

S-блок Калина:

```
0xA8 0x43 0x5F 0x06 0x6B 0x75 0x6C 0x59 0x71 0xDF 0x87 0x95 0x17 0xF0 0xD8 0x09
0x6D 0xF3 0x1D 0xCB 0xC9 0x4D 0x2C 0xAF 0x79 0xE0 0x97 0xFD 0x6F 0x4B 0x45 0x39
0x3E 0xDD 0xA3 0x4F 0xB4 0xB6 0x9A 0x0E 0x1F 0xBF 0x15 0xE1 0x49 0xD2 0x93 0xC6
0x92 0x72 0x9E 0x61 0xD1 0x63 0xFA 0xEE 0xF4 0x19 0xD5 0xAD 0x58 0xA4 0xBB 0xA1
0xDC 0xF2 0x83 0x37 0x42 0xE4 0x7A 0x32 0x9C 0xCC 0xAB 0x4A 0x8F 0x6E 0x04 0x27
0x2E 0xE7 0xE2 0x5A 0x96 0x16 0x23 0x2B 0xC2 0x65 0x66 0x0F 0xBC 0xA9 0x47 0x41
0x34 0x48 0xFC 0xB7 0x6A 0x88 0xA5 0x53 0x86 0xF9 0x5B 0xDB 0x38 0x7B 0xC3 0x1E
0x22 0x33 0x24 0x28 0x36 0xC7 0xB2 0x3B 0x8E 0x77 0xBA 0xF5 0x14 0x9F 0x08 0x55
0x9B 0x4C 0xFE 0x60 0x5C 0xDA 0x18 0x46 0xCD 0x7D 0x21 0xB0 0x3F 0x1B 0x89 0xFF
0xEB 0x84 0x69 0x3A 0x9D 0xD7 0xD3 0x70 0x67 0x40 0xB5 0xDE 0x5D 0x30 0x91 0xB1
0x78 0x11 0x01 0xE5 0x00 0x68 0x98 0xA0 0xC5 0x02 0xA6 0x74 0x2D 0x0B 0xA2 0x76
0xB3 0xBE 0xCE 0xBD 0xAE 0xE9 0x8A 0x31 0x1C 0xEC 0xF1 0x99 0x94 0xAA 0xF6 0x26
0x2F 0xEF 0xE8 0x8C 0x35 0x03 0xD4 0x7F 0xFB 0x05 0xC1 0x5E 0x90 0x20 0x3D 0x82
0xF7 0xEA 0x0A 0x0D 0x7E 0xF8 0x50 0x1A 0xC4 0x07 0x57 0xB8 0x3C 0x62 0xE3 0xC8
0xAC 0x52 0x64 0x10 0xD0 0xD9 0x13 0x0C 0x12 0x29 0x51 0xB9 0xCF 0xD6 0x73 0x8D
0x81 0x54 0xC0 0xED 0x4E 0x44 0xA7 0x2A 0x85 0x25 0xE6 0xCA 0x7C 0x8B 0x56 0x80
```

Рис. 1.

При построении полинома Жегалкина с помощью стандартного алгоритма реализации необходимо было около 20 минут. Оптимизированный алгоритм построения полинома Жегалкина справился с поставленной задачей менее чем за 2 секунды. Замеры производились на ПК с Windows 10, Intel Core i7-3630QM 2.4 Ггц.

Алгебраическая иммунность S-блока шифра AES равна 2. Воспользовавшись выражением (8), имеем: $k = C_{16}^1 + + C_{16}^2 = 136$. Алгебраическая иммунность S-блока шифра Калина равна 3, т.е., имеем: $k = C_{16}^1 + C_{16}^2 + C_{16}^3 = 696$. Именно такие значения получены с помощью системы «Magma» в работе [13].

С помощью оптимизированного алгоритма заведомо выбирая только необходимые коэффициенты для подсчёта алгебраической иммунности шифров AES и Калина, получаем таблицы предвычислений размерами (65536×136) и (65536×696) соответственно, вместо (65536×65536).

Время выполнения для стандартного алгоритма выходит за приемлемые сроки. Подсчёт алгебраической иммунности одного узла нелинейной замены занимал бы более одной недели. Оптимизированный алгоритм вычислений выполнил поставленную задачу за 4 секунды для S-блока AES и за 20 с для S-блока Калина-2 соответственно. Замеры производились на ПК с Windows 10, Intel Core i7-3630QM 2.4 Ггц.

5.ОБСУЖДЕНИЕ

Предложен ускоренный алгоритм расчёта алгебраической иммунности, байтовых S-блоков, в котором используются реально существующие практические и теоретические ограничения, характерные для оцениваемого показателя. Он позволяет существенно сократить объёмы обрабатываемой информации. В качестве таких ограничений использовано то, что в соответствии с теорией алгебраическая иммунность для байтовых S-блоков не может превышать значения 4, т.е. степень аннулирующего многочлена не может быть выше четвёртой. Последующая операция приведения сокращённой по числу столбцов матрицы коэффициентов к диагональному виду позволяет сократить размеры матрицы коэффициентов и по числу строк (появляются строки из одних нулей). В результате действительно удаётся существенно ускорить процедуру выполнения вычислений.

ВЫВОДЫ

Таким образом, основным результатом работы является обоснование ускоренного метода расчёта алгебраической иммунности S-блоков. Исключение при формировании программы множеств данных, не участвующих на каждом из этапов её работы в формировании результата, а также учёт априорно известных данных относительно конечного результата позволили существенно сократить объёмы промежуточных вычислений и добиться повышения производительности программы в сотни раз. Время работы программы при расчёте алгебраической иммунности S-блока со значением этого показателя равного трём составляет около 20-ти с.

Литература

[1] *A.J. Menezes, P.C. van Oorschot, S.A. Vanstone*. Handbook of Applied Cryptography. CRC Press, 1997, 794 p.

[2] *N. Ferguson and B. Schneier.* Practical Cryptography. John Wiley & Sons, 2003, 432 p.

[3] *Горбенко І.Д., Горбенко Ю.І.* Прикладна криптологія. Теорія. Практика. Застосування: монографія.– Харків: Видавництво «Форт», 2012. – 870 с.

[4] ISCI'2018: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2018, 360 p.

[5] *Courtois N. and Meier W.* Algebraic attacks on stream ciphers with linear feedback // Eurocrypt'2003. LNCS. 2003. V. 2656. P. 345–359.

[6] *Покрасенко Д.П.* Об алгебраической иммунности векторных булевых функций / Д.П. Покрасенко // Прикладная дискретная математика. Приложение, 2014, № 7, 43–48.

[7] *Meier W., Pasalic E., and Carlet C.* Algebraic attacks and decomposition of Boolean functions // Eurocrypt'2004. LNCS. 2004. V. 3027. P. 474–491. 3. Armknecht F. and Krause M. Constructing single- and multi-output Boolean functions with maximal immunity // ICALP'2006. LNCS. 2006. V. 4052. P. 180–191.

[8] *Armknecht F. and Krause H.* Constructing single- and multi-output Boolean functions with maximal immunity // ICALP'2006/ V/4052. P. 180–191.

[9] *Ars G. and Faug`ere J.-C.* Algebraic immunities of functions over finite fields // Proc. Conf. BFCA. 2005. P. 21–38.

[10] *Carlet C.* On the algebraic immunities and higher order nonlinearities of vectorial Boolean functions // Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes. Amsterdam: IOS Press, 2009. P. 104–116.

[11] *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.

[12] *Покрасенко Д.П.* Компонентная алгебраическая иммунность S-блоков, использующихся в некоторых блочных шифрах // Прикладная дискретная математика. Приложение, 2017, № 10, 49–51.

[13] *Кузнецов О.О.* Алгебраїчний імунітет нелінійних вузлів симетричних шифрів / О.О. Кузнецов, Ю.І. Горбенко, І.М. Білозерцев та інші // Радиотехника. – 2017. – Вып. 189. –С. 47–58.

[14] Magma Computational Algebra System. Available at: http://magma.maths.usyd.edu.au/magma.

[15] *Massimiliano Sala, Teo Mora, Ludovic Perret, Shojiro Sakata, Carlo Traverso Gröbner* Bases, Coding, and Cryptography. Springer-Verlag Berlin Heidelberg. – 426 p.

[16] *Баев Владимир Валерьевич*. Эффективные алгоритмы получения оценок алгебраической иммунности булевых функций: диссертация на соискание ученой степени кандидата физико-математических наук: 01.01.09 – Москва, 2008. – 101 с.

[17] *Аржанцев И.В.* Базисы Грёбнера и системы алгебраических уравнений. Летняя школа. Современная математика. Дубна, июль 2002. – Москва: МЦНМО, 2003. – 68 с.

Поступила в редколлегию 20.11.2018

Лисицкий Константин Евгеньевич, аспирант кафедры безопасности информационных систем и технологий Харьковского национального университета им. В.Н. Каразина. Область научных интересов – криптография, технологии блочного симметричного шифрования.

Кузнецов Александр Александрович, доктор технических наук, профессор, заместитель главного конструктора АТ «ИИТ», профессор кафедры безопасности информационных систем и технологий Харьковского национального университета им. В.Н. Каразина. Область научных интересов – криптография и аутентификация, алгебраическая теория кодов, обработка, передача и защита информации.

Горбенко Юрий Иванович, кандидат технических наук, исполнительный директор АТ «ИИТ», старший научный сотрудник кафедры Харьковского национального университета имени В.Н. Каразина. Область научных интересов – криптография и аутентификация, инфраструктура открытых ключей.

Оноприенко Виктор Васильевич, кандидат технических наук, доцент, генеральный директор АТ «Институт информационных технологий». Область научных интересов – криптография и аутентификация, инфраструктура открытых ключей, теория защиты информации, информационная и кибербезопасность государства.

Стельник Игорь Валерьевич, заместитель директора департамента защиты информации Администрации Государственной службы специальной связи и защиты информации Украины. Область научных интересов – криптография и аутентификация, теория защиты информации.

УДК 004.056.55

Лисицький К.Є. **Прискорений метод реалізації обчислення алгебраїчної імунності нелінійних вузлів заміни блокових симетричних шифрів** / К.Є. Лисицький, О.О. Кузнецов, Ю.І. Горбенко, В.В. Оноприєнко, І.В. Стельник // Прикладна радіоелектроніка: наук.–техн. журнал. – 2018. –Том 17. № 3, 4. – С. 81–87.

Розглядається прискорений метод обчислення алгебраїчної імунності нелінійних вузлів заміни симетричних шифрів по Жан-Шарлю Фожеру (Jean-Charles Faugère). Він заснований на пошуку анігілюючої функції до полінома Жегалкина, побудованої з вихідного нелінійного вузла заміни. Проводиться аналіз швидкодії відомого алгоритму підрахунку алгебраїчної імунності. Обговорюються деталі реалізації алгоритму і надається опис покращеного алгоритму обчислення алгебраїчної імунності, оптимізованого за часом і обсягом затрат і ресурсів.

Ключові слова: симетричний шифр, алгебраїчний імунітет, нелінійний вузол заміни, булева функція, продуктивність алгоритму.

Табл.: 01. Іл.: 01. Бібліогр.: 17 найм.

UDC 004.056.55

Lisitsky K. **An accelerated method of calculating the algebraic immunity of nonlinear nodes of replacing symmetric ciphers** / K. Lisitsky, A. Kuznetsov, Yu. Gorbenko, V.Onopryenko, I.Stelnik // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 81–87.

An accelerated method for calculating the algebraic immunity of nonlinear replacement nodes of symmetric ciphers by Jean-Charles Faugère is considered. It is based on the search for the annihilating function to the Zhegalkin polynomial constructed from the original nonlinear node of substitution. An analysis of the speed of a known algorithm for calculating algebraic immunity is given. The details of the implementation of the algorithm are discussed and a description of the accelerated algorithm for calculating algebraic immunity, optimized in terms of computation time and for the resources of the RAM is provided.

Keywords: symmetric cipher, algebraic immunity, non-linear replacement node, Boolean function, algorithm performance.

Tab. 01. Fig. 01. Ref.: 17 items.

УДК 004.056.55

ДОСЛІДЖЕННЯ МЕТОДІВ ФОРМУВАННЯ ВИПАДКОВИХ НЕЛІНІЙНИХ ВУЗЛІВ ЗАМІН СИМЕТРИЧНИХ ШИФРІВ

О. О. КУЗНЕЦОВ, І. М. БІЛОЗЕРЦЕВ, А. І. ПУШКАРЬОВ, Ю. І. ГОРБЕНКО, В. В. ОНОПРІЄНКО

Обгрунтовуються основні показники ефективності криптографічних булевих функцій та векторних відображень, які застосовуються у якості вузлів ускладнення симетричних криптоперетворень. Досліджуються евристичні методи формування криптографічних булевих функцій та нелінійних S-блоків симетричних шифрів, що відповідають встановленим вимогам безпеки. Обгрунтовуються перспективні напрямки подальших досліджень з метою удосконалення евристичних методів синтезу випадкових вузлів заміни.

Ключові слова: симетричні криптоперетворення, випадкові нелінійні вузли заміни, евристичні методи генерації, показники криптографічної стійкості.

ВСТУП

Для забезпеченні послуг інформаційної безпеки (конфіденційності, цілісності, неспростовності тощо) зазвичай застосовують симетричні шифри [1, 2]. З погляду на забезпечувану стійкість, швидкодію та надійність симетричні криптоперетворення мають певну перевагу та зручність у практичному використанні [2].

Випадкові нелінійні вузли заміни (криптографічні булеві функції, векторні відображення або S-блоки) відіграють суттєву роль у забезпеченні безпеки симетричних перетворень. Зокрема їх криптографічні властивості безпосередньо впливають на стійкість шифрів до різних криптоаналітичних атак [3–19]. Отже питання синтезу випадкових вузлів заміни з необхідними криптографічними характеристиками є безумовно актуальним та важливим завданням [20–23].

1. КРИТЕРІЇ ТА ПОКАЗНИКИ ЕФЕКТИВНОСТІ НЕЛІНІЙНИХ ВУЗЛІВ УСКЛАДНЕННЯ

Розглянемо критерії та показники ефективності нелінійних вузлів замін, що безпосередньо впливають на рівень стійкості сучасних симетричних шифрів до різних криптоаналітичних атак [3–19].

У більшості відомих робіт в області аналізу і синтезу нелінійних вузлів замін сучасних симетричних шифрів використовується математичний апарат криптографічних булевих функцій [3–12]. При цьому векторні відображення (S-блоки) подаються сукупністю компонентних булевих функцій, властивості яких характеризують ефективність нелінійного вузла замін.

Як основні критерії і показники ефективності використовують [3–12]: збалансованість і нелінійність компонентних булевих функцій; кореляційний імунітет; критерій поширення; алгебраїчний степінь; значення функції автокореляції.

Булевою функцією f від n змінних є функція, що здійснює відображення з поля $GF(2^n)$ усіх двійкових векторів $x = (x_1, \ldots, x_n)$ довжини n у полі $GF(2)$.

Зазвичай булеві функції подаються в нормальній алгебраїчній формі.

Поле $GF(2^n)$ складається з 2^n векторів α_i: $\alpha_0 = (0,\ldots,0,0)$, $\alpha_1 = (0,...,0,1),...,$ $\alpha_{2^n-1} = (1,...,1,1)$, $\alpha_i \in V_n$ де V_n – векторний простір в $GF(2^n)$.

Послідовністю функції f називається (1, - 1) -послідовність, визначена як $((-1)^{f(\alpha_0)}, (-1)^{f(\alpha_1)}, \ldots, (-1)^{f(\alpha_{2^n-1})})$.

Таблицею істинності функції f називається (0,1) -послідовність, визначена як $(f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{2^n-1}))$.

Послідовність функції f є *збалансованою*, якщо її (0,1)-послідовність ((1, -1)-послідовність) містить однакову кількість нулів і одиниць (одиниць і мінус одиниць). Функція f є збалансованою, якщо збалансована її послідовність.

Еквівалентне визначення: функція f над $GF(2^n)$ є збалансованою, якщо її вихідні значення є рівноімовірними:

$$|\{x \mid f(x) = 0\}| = |\{x \mid f(x) = 1\}| = 2^n - 1.$$

Афінною функцією f називається функція виду $f = a_1x_1 \oplus \ldots \oplus a_nx_n \oplus c$, де a_j, $c \in GF(2)$, $j = 1, 2,..., n$.. Функція f називається лінійною, якщо $c = 0$.

Вагою Хемінга вектора α, що позначається як $W(\alpha)$, є кількість одиниць у векторі (послідовності). *Відстанню Хемінга d(f, g)* між послідовностями двох функцій f і g називається кількість позицій, в яких відрізняються послідовності цих функцій.

Нелінійність N_S перетворення – мінімальна відстань Хемінга між вихідною послідовністю S і усіма вихідними послідовностями афінних функцій над деяким полем:

$$N_s = \min\{d(S,\varphi)\},$$

де φ – безліч афінних функцій.

Нелінійність функції N_f – мінімальна відстань Хемінга N_f між функцією f і усіма афінними функціями над $GF(2^n)$:

$$N_f = \min\{d(f,\varphi)\},$$

де φ – безліч афінних функцій.

Для довільної функції f нелінійність N_f над $GF(2^n)$ може досягати

$$N_f \le 2^{n-1} - 2^{n/2-1}.$$

Для збалансованої функції f над $GF(2^n)$ ($n \ge 3$) нелінійність N_f може досягати:

$$N_f \le \begin{cases} 2^{n-1} - 2^{\frac{n}{2}-1} - 2, n = 2k \\ \left\lfloor\left\lfloor 2^{n-1} - 2^{\frac{n}{2}-1} \right\rfloor\right\rfloor, n = 2k = 1, \end{cases}$$

де – $\lfloor\lfloor x \rfloor\rfloor$ максимальне парне ціле, менше або рівне x.

Функція f має *кореляційний імунітет* порядку k, якщо вихідна послідовність функції $y \in Y$ статистично не залежить від будь-якої підмножини з k вхідних координат :

$$\forall \{x_1,...,x_k\} \;\; P(y \in Y \mid \{x_1,...,x_k\} \in X) \;=\; P(y \in Y).$$

Еквівалентне визначення кореляційного імунітету в термінах перетворення Уолша: функція f над полем $GF(2^n)$ має кореляційний імунітет порядку k, $KI(k)$, якщо її перетворення Уолша задовольняє рівності $F(\omega) = 0$ для усіх $\omega \in V_n$ такий, що $1 \le W(\omega) \le k$:

$$\forall \omega \in V_n \;\; F(\varpi) \;=\; 0 \quad KI(f) = k \;.$$

Перетворення Уолша $F(\omega)$ функції f над полем $GF(2^n)$ визначається тоді, як що набуває дійних значень функція

$$F(\omega) = 2^{-n} \sum\nolimits_x (-1)^{f(x) \oplus \langle \varpi, x \rangle},$$

де $\omega \in V_n$, $f(x)$, $\langle \omega, x \rangle \in N$ ($\langle \omega, x \rangle$ – скалярний добуток $w_1x_1 \oplus \ldots \oplus w_nx_n$.

Кореляційно-імунна функція k-го порядку – функція, що має кореляційний імунітет порядку k. Збалансовані кореляційно-імунні функції називаються еластичними функціями.

Функція f над полем $GF(2^n)$ відповідає:

– *критерію поширення* відносно вектора a, КР(a), якщо функція $f(x) \oplus f(x \oplus a)$ є збалансованою, $x \in V_n$, де $x = (x_1, x_2, \ldots, x_n)$

$$P(f(x) = f(x \oplus \alpha)) = \frac{1}{2};$$

– *критерію поширення* міри k, КР(k), якщо задовольняється критерій поширення відносно усіх векторів $a \in V_n$ при $1 \le W(a) \le k$:

$$P(f(x) = f(x \oplus \alpha)) = \frac{1}{2} \quad \forall \alpha : 1 \le W(\alpha) \le k;$$

– *суворому лавинному критерію* (СЛК), якщо f задовольняє критерій поширення степеня 1:

$$P(f(x) = f(x \oplus \alpha)) = \frac{1}{2} \quad \forall \alpha : W(\alpha) = 1 \,.$$

Алгебраїчний степінь deg(f) є степінь найдовшого доданка функції, поданої в алгебраїчній нормальній формі.

Автокореляційна функція $\hat{r}(s)$ для $s \in 0 \ldots 2^n - 1$ визначена як

$$\hat{r}(s) = \sum_{x=0}^{2^n-1} \hat{f}(x) \hat{f}(x \oplus s).$$

Говорять, що функція f задовольняє *характеристику поширення* m, якщо

$$(1 \le |s| \le m) \Rightarrow |\hat{r}(s)| = 0.$$

Аналогічно, автокореляція $AC(f)$ функції f визначається як модуль найбільшого значення $\hat{r}(s)$. :

$$AC(f) = \max_{s \ne 0} \left| \sum_x \hat{f}(x) \hat{f}(x \oplus s) \right| = \max_{s \ne 0} |\hat{r}(s)|.$$

Автокореляція забезпечує витік інформаційного потоку з входу на вихід функції. У деяких випадках дуже сильний взаємозв'язок може бути представлений як лінійна структура (для яких справедлива рівність $f(x) \oplus f(x \oplus s) = 1$ або $f(x) \oplus f(x \oplus s) = 0$). Вони, як правило, уникаються.

Розглянуті критерії і показники ефективності S-блоків відображають здатність нелінійного вузла протистояти атакам певного типу. Нелінійність, критерій поширення і кореляційна імунність характеризують здатність протистояти кореляційним атакам, алгебраїчний ступінь і автокореляція – аналітичним атакам, збалансованість – статистичним.

Найбільшого розвитку в ході формування та аналітичного опису властивостей векторних відображень (S-блоків) набули методи булевої алгебри та, зокрема, булевих функцій [3–16]. Дійсно, оскільки булева функція $f(x)$ від n змінних – це відображення

$$f(x) : GF(2)^n \to GF(2),$$

де $x = (x_1, \ldots, x_n)$, тоді векторну функцію можна подати через множину булевих функцій так:

Векторна (n,m) *булева функція* (S-блок) – це відображення

$$F = F(x_1,...,x_n) = (f_1(x_1,...,x_n),...,f_m(x_1,...,x_n)) :$$

$$GF(2)^n \to GF(2)^m. \qquad (1)$$

Зрозуміло, що кожна компонентна функція $f_i, i = 1,...,m$ – це булева функція від n змінних.

Оптимізація окремих криптографічних показників однієї із компонентних функцій $f_i, i = 1,...,m$ із (1) може призвести до суттєвого погіршення показників інших функцій і навпаки. Отже необхідно розробити математичний апарат та відповідне програмне забезпечення, які при прийнятній обчислювальній складності забезпечать комплексне підвищення основних криптографічних показників усіх компонентних булевих функцій $f_i, i = 1,...,m$ із виразу (1) та всіх їхніх лінійних комбінацій.

Очевидно, що перевірка усіх можливих критеріїв значно зменшить швидкодію алгоритму генерації, тому до переліку критеріїв слід включати лише найбільш критичні, з точки зору стійкості до основних видів криптоаналітичних атак. Сукупність усіх можливих критеріїв оцінки розділятимемо на *безумовні* та *умовні* [2].

До безумовних критеріїв належать ті критерії, виконання яких є обов'язковим, тобто безумовним, для забезпечення генерованою підстановкою криптографічної стійкості від основних видів крипто-аналітичних атак [2]. У свою чергу до умовних критеріїв належать додаткові критерії ефективності, які можна застосувати для порівняння між собою нелінійних вузлів, що відповідають безумовним критеріям [2].

Далі розглянемо перелік обраних безумовних критеріїв, яким має відповідати байтова бієктивна підстановка, тобто при $n = m = 8$.

1. *Нелінійність.* Даний критерій забезпечує стійкість до лінійного криптоаналізу [3–19]. Дійсно, для S-блоку S розмірністю $n \times m$ співвідношення між його нелінійністю N_F і максимумом таблиці лінійних апроксимацій $LAT_{\max}$ записується в такий спосіб:

$$N_F = 2^{n-1} - LAT_{\max}.$$

Позначимо цей критерій $W_{\delta 1}$.

2. *Алгебраїчний степінь.* Максимальне значення алгебраїчного степеня забезпечує стійкість до статистичних атак [3–19]. Позначимо цей критерій $W_{\delta 2}$.

3. *Максимум таблиці диференційних різниць (або δ - рівномірність).* Мінімізація значення нетривіального максимуму таблиці різниць забезпечує високий рівень стійкості до диференційного критоаналізу [3–19]. Для S-блоку S розмірністю $n \times m$ співвідношення між його автокореляцією AC і максимумом *таблиці диференційних різниць* $DDT_{\max}$ має вигляд [8]:

$$DDT_{\max} \geq 2^{n-m} + 2^{-m} AC.$$

Позначимо цей критерій $W_{\delta 3}$.

4. *Алгебраїчний імунітет.* Значення алгебраїчного імунітету $AI = 3$ значно підвищує її стійкість до алгебраїчних атак [11, 12]. Позначимо цей критерій як $W_{\delta 4}$.

5. *Відсутність фіксованих точок та циклів менше за 4.* Даний критерій S-блоків застосовується у більшості шифрів для захисту від статистичних атак [3–19]. Відповідно, позначимо цей критерій $W_{\delta 5}$.

Докладне визначення та порядок розрахунку окремих криптографічних показників наведено, наприклад, у [3–8].

З урахуванням наведених вище безумовних критеріїв $W_{\delta 1} - W_{\delta 5}$ відповідно до [2], можна сформувати *функцію відповідності* підстановки:

$$f_{фв}(\ ) = W_{\delta 1} \wedge W_{\delta 2} \wedge W_{\delta 3} \wedge W_{\delta 4} \wedge W_{\delta 5}.$$

Функція відповідності становить безумовний інтегральний критерій, за допомогою якого можна оцінити якість згенерованого S-блоку та прийняти рішення про його відповідність вимогам криптографічної стійкості.

До умовних належать ті критерії, виконання яких є бажаним, але не обов'язковим результатом [2]. Наприклад, складність або час генерації підстановки бажано зменшити, бо це прискорить досягнення мети – формування S-блоку із необхідними (безумовними) показниками стійкості.

Отже, мету та завдання розробки методів генерації нелінійних вузлів заміни можна формально подати у вигляді безумовного інтегрального критерію при мінімізації складності (часу):

$$S \to \min \left| \left\{ f_{фв}(\ ) = W_{\delta 1} \wedge W_{\delta 2} \wedge W_{\delta 3} \wedge W_{\delta 4} \wedge W_{\delta 5} \right\}.$$

Вирішення зазначеного завдання полягає у пошуку сукупності певних перетворень (комплементацій) над компонентними булевими функціями $f_i, i = 1,...,m$ із (1), які за кінцеве число кроків дозволять досягти встановлених (бажаних) значень криптографічних показників.

Проведемо аналіз та порівняльні дослідження різних методів генерації нелінійних вузлів ускладнення.

2. МЕТОДИ ГЕНЕРАЦІЇ НЕЛІНІЙНИХ ВУЗЛІВ УСКЛАДНЕННЯ

Методи генерації криптографічних булевих функцій можна умовно розділити на три типи: методи випадкової генерації, алгебраїчні методи та евристичні методи [12, 20, 22, 23].

Методи випадкової генерації, попри свою простоту, не можуть бути використані для ефективної генерації булевих функцій з високими показниками криптографічної стійкості при $n \geq 8$. Алгебраїчні методи, зі свого боку, є високоефективними та не вимагають потужних обчислювальних ресурсів, але булеві функції, сформовані із застосуванням цих методів, мають низьку алгебраїчну складність і є потенційно вразливими до деяких сучасних криптоаналітичних атак. Еврис-

тичні методи [3 – 14], серед усіх вищезгаданих, є найсучаснішими та дозволяють генерувати булеві функції з криптографічними характеристиками, що наближені до максимально можливих.

Таким чином серед різних методів генерації нелінійних вузлів заміни [14] найбільш перспективними вважаються евристичні методи [12, 13, 20–23]. Порівняно із методами випадкової генерації вони значно простіше в реалізації, а порівняно із алгебраїчними дозволяють оптимізувати різні криптографічні показники та позбутися детермінованих, тобто, заздалегідь прогнозованих конструкцій [22, 23].

В основі більшості евристичних методів лежать ітеративні процедури підвищення певних показників криптографічної стійкості нелінійного вузла заміни. Наприклад, у відомому методі градієнтного підйому [15] шляхом комплементації деякої позиції в таблиці істинності початкової функції підвищується нелінійність. Критерієм градієнтного пошуку є максимізація відстані за Хемінгом між формованою послідовністю і послідовностями лінійних функцій.

2.1. Метод градієнтного підйому

Суть методу полягає в підвищенні нелінійності випадкової булевої функції шляхом комплементації деякої позиції в таблиці істинності початкової функції. Кожна позиція таблиці істинності відповідає унікальним вхідним даним. Метод дозволяє створити повний список/перелік таких вхідних даних функції, що комплементація будь-якої відповідної даному входу вихідної позиції в таблиці істинності збільшуватиме нелінійність даної функції. Список/перелік таких позицій в таблиці істинності позначається як 1 - *Improvement Set* функції $f(x)$, або $1-IS_f$ [42].

Визначення 1. [15]. Нехай $g(x)=f(x)\oplus 1$ для $x=x_a$ та $g(x)=f(x)$ для всіх інших x. Якщо $N_g > N_f$, то $x_a \in 1-IS_f$.

В [15] подано швидкий систематичний метод визначення множини $1-IS_f$ заданої булевої функції шляхом використання її таблиці істинності і перетворень Уолша-Адамара. Для знаходження множини $1-IS_f$ заданої булевої функції необхідно спочатку визначити значення коефіцієнтів перетворення Уолша-Адамара, які відповідали б величинам, близьким до абсолютного значення максимального коефіцієнта, $WHT_{\max}$.

Визначення 2. Нехай $f(x)$, є булевою функцією з перетворенням Уолша-Адамара $F(w)$, де $WHT_{\max}$ позначає максимальне абсолютне значення $F(w)$. Тоді існуватиме одна або більше лінійних функцій $L_w(x)$, що мають мінімальну відстань до функції $f(x)$, і для даних w буде справедлива рівність $|F(w)|=WHT_{\max}$.

Визначається так:

$$W_1^+ = \{w : F(w) = WHT_{\max}\} \text{ та}$$

$$W_1^- = \{w : F(w) = -WHT_{\max}\}.$$

Також визначаються множини w, для яких значення WHT наближені до максимуму:

$$W_2^+ = \{w : F(w) = WHT_{\max} - 2\},$$

$$W_2^- = \{w : F(w) = -(WHT_{\max} - 2)\}$$

$$W_3^+ = \{w : F(w) = WHT_{\max} - 4\} \text{ та}$$

$$W_3^- = \{w : F(w) = -(WHT_{\max} - 4)\}.$$

Коли таблиця істинності змінюється рівно в одній позиції, всі значення WHT змінюються на +2 або -2. З цього випливає, що для збільшення нелінійності всі значення WHT у множині W_1^+ мають бути змінені на -2, всі значення WHT у множині W_1^- мають бути змінені на 2, а також всі значення WHT у множині W_2^+ мають бути змінені на -2, всі значення WHT у множині W_2^- мають бути змінені на 2. Якщо перші дві умови є очевидними, то наступні дві умови потрібні для того, щоб всі інші значення $|F(w)|$ залишалися меншими, ніж $WHT_{\max}$. Дані умови можуть бути подані у вигляді простих тестів.

Теорема 1 [15]. Нехай дана деяка булева функція $f(x)$ з $WHT\ F(w)$, і визначені множини $W^+ = W_1^+ \cap W_2^+$ та $W^- = W_1^- \cap W_2^-$. Тоді для деякого входу x існує елемент з *Improvement Set* і виконуються такі дві умови:

$$f(x) = L_w(x) \text{ та для усіх } w \in W^+, \text{ та}$$

$$f(x) \neq L_w(x) \text{ та для усіх } w \in W^-.$$

Якщо функція $f(x)$ не збалансована, зниження незбалансованості може бути досягнуто використанням додаткового обмеження:

якщо $F(0) > 0$, $f(x) = 0$, інакше $f(x) = 0$.

Критерієм градієнтного пошуку є максимізація відстані за Хемінгом між формованою послідовністю і послідовностями лінійних функцій. Після поновлення алгебраїчної форми булевої функції виробляються аналогічні операції: виконується перетворення Уолша-Адамара WHT і знаходяться максимальні значення коефіцієнтів перетворення; формується множина *Improvement Set*; знаходяться елементи послідовності функції, що збігаються з елементами послідовності найближчої лінійної форми; інвертування елементів,

що збіглися, і підвищення нелінійності функції, за допомогою «віддалення» від найближчої лінійної функції. Далі виконуються чергові ітерації, аналогічні розглянутим вище.

Проведені у [16, 17, 22] дослідження показали, що метод градієнтного підйому з [15] обчислювально витратний і, з великою кількістю аргументів булевої функції, вимагає виконання значного числа повторюваних ітерацій. Для зниження обчислювальної складності в [16, 17, 22] запропоновано метод градієнтного спуску з бент-послідовностями як вхідні дані. За своєю сутністю цей метод наслідує окремі перетворення методу градієнтного підйому з [15], але, як показано у [16, 17, 22], вимагає значно менших обчислювальних витрат.

2.2. Метод градієнтного спуску

Даний метод заснований на комплементації позицій бент-послідовностей для градієнтного пошуку збалансованих булевих функцій за критерієм максимізації відстані Хеммінга між сформованими послідовностями і послідовностями всіх лінійних функцій. Це дозволяє знизити обчислювальні витрати на пошук булевих функцій з необхідними криптографічними властивостями [16, 17, 22].

Основною ідеєю методу градієнтного спуску є ефективне зниження нелінійності заданих бент-послідовностей при кожній з $2^{n/2-1}$ обов'язкових комплементацій. Для досягнення заданої верхньої межі нелінійності необхідно із загального числа позицій x таблиці істинності, що підлягають комплементації, визначити те число позицій y, зміна яких має наслідком зміну WHT на +2, і те число позицій z, зміна яких має наслідком зміну WHT на -2, $x = y + z$.

Після розрахунку необхідного числа комплементацій бент-послідовності на першому кроці евристичного пошуку виконується перетворення Уолша-Адамара WHT і визначається максимальна відстань за Хеммінгом до однієї або декількох послідовностей лінійних функцій $L_i(x)$. Ця операція відповідає вибору нульового значення коефіцієнтів перетворення Уолша-Адамара WHT, після чого формується безліч лінійних функцій, що становлять *Improvement Set*. Далі проводиться інвертування елементів послідовності бент-функції, які збігаються з елементами послідовностей лінійних функцій з безлічі *Improvement Set*. У результаті незбалансованість функції знижується, але знижується так само і нелінійність, тобто послідовність функції не є вже максимально віддаленою від послідовностей лінійних функцій $L_i(x)$. На наступній ітерації всі операції повторюються. Таким чином, як критерій градієнтного пошуку криптографічних функцій, пропонованим методом є максимізація мінімальної відстані за Хеммінгом формованої послідовності і послідовностей лінійних функцій [22].

2.3. Переставний метод пошуку випадкових S-блоків

У дисертації [12] та статтях [18, 19] набули подальшого розвитку евристичні методи. Зокрема, методи градієнтного пошуку булевих функцій було розширено та застосовано для генерації векторних відображень (S-блоків) з необхідними властивостями.

У розробленому в [12, 18, 19] методі пропонується використовувати той самий підхід, як і у розглянутому вище методі градієнтного спуску, однак з двома істотними відмінностями [12]:

– замість булевих функцій використовувати векторні булеві функції;

– замість бент-функцій (послідовностей) використовувати векторні булеві функції (підстановки) з максимальними показниками δ -рівномірності або з максимальним значенням нелінійності.

Зміна необхідної кількості біт у бент-послідовності не гарантує досягнення нелінійності, близької до максимальної. Для векторного випадку в [21] було доведено твердження, що при обміні місцями двох значень в деякій перестановці значення, нелінійності і δ-рівномірність відрізнятимуться на невелике значення (± 2 та ± 4, відповідно). Ґрунтуючись на цьому і було розроблено більшість відомих евристичних методів пошуку нелінійних вузлів ускладнення.

Метод, що запропоновано в [12] та досліджено в [18, 19], приймає на вхід перестановки векторну функцію F з мінімальним показником δ-рівномірності і кількість значень (*NP*), які необхідно поміняти місцями для досягнення оптимальних криптографічних показників.

Основні кроки методу наведено нижче [12].

а) Генерація підстановки S на основі обраної переставної векторної булевої функції F.

б) Випадковий обмін місцями *NP* значень підстановки S і формування підстановки S_t.

в) Послідовне обчислення показників залежно від їх обчислювальної складності. Якщо підстановка S_t задовольняє всі критерії, крім циклових, тоді застосовується лінійно-еквівалентне перетворення для досягнення критерію відсутності фіксованих точок. За невідповідності хоча б одного з критеріїв перехід в п. б).

У результаті сформована підстановка зберігатиметься в S_t.

Подальші дослідження показали, що практична реалізація методу з [12] вимагає значних обчислювальних витрат – один байтовий S-блок формується в середньому за 3.5 години на однопроцесорному персональному комп'ютері (ПК) [12, 18, 19].

2.4. Переставний метод пошуку S-блоків із фіксованою кількістю замін

Інший метод, запропонований у [18] та розвинений у [19], що є модифікацією попередньо методу з [12], дозволяє формувати підстановки за значно мен-

ший час, або з більш високими криптографічними показниками. Сутність методу полягає в одноразовому застосуванні фіксованої кількості перестановок $NP = 22$. Тобто на кроці в) попереднього методу (див. п. 3.3) немає повернення на крок в).

В ході застосування як вхідні дані підстановки S з високою нелінійністю (наприклад, S-блока з алгебраїчної конструкцією $F(x) = x^{254}$) вдається зберегти високі криптографічні показники та, за рахунок обміну місцями NP значень підстановки, надати векторному відображенню псевдовипадкового вигляду. Обране значення $NP = 22$ дозволяє з високою ймовірністю отримати максимально можливе значення алгебраїчного імунітету (для байтового відображення дорівнює 3) та високих значень інших криптографічних показників.

Дійсно, як показано у [19] запропонований метод дозволяє значно скоротити час генерації і, наприклад, на однопроцесорному ПК за 10 хвилин сформувати збалансовану (бієктивну) байтову підстановку з такими характеристиками:

- нелінійність $N_F = 104$;
- алгебраїчний степінь $AD = 7$;
- алгебраїчний імунітет $AI = 3$;
- максимум таблиці диференціалів $DDT_{\max} = 8$;
- відсутність фіксованих точок та циклів довжиною $l \le 3$.

Проте у процесі генерації цей метод передбачає застосування фіксованої кількості перестановок елементів початкової алгебраїчної підстановки [18, 19]. Тобто нелінійні вузли, сформовані у такий спосіб, у своїй суті є детермінованими (за винятком незначної кількості пар елементів). Отже, такий метод не реалізує повною мірою випадкову генерацію, його практичне застосування є обмеженим.

ВИСНОВКИ ТА ОБГОВОРЕННЯ

Аналізуючи розглянуті методи формування нелінійних вузлів заміни слід, перш за все, відмітити значний прогрес в їхньому вдосконаленні протягом останніх 10–15 років.

Розроблений понад 20 років тому метод градієнтного підйому [15] декілька разів вдосконалювався та відтворювався із іншими процедурами та новими ідеями евристичного пошуку. Зокрема, ідея застосування як вхідні дані високонелінійних послідовностей та бент-функцій була запропонована спершу в роботі [16], продовжена у [20] та в дисертації [22]. Замість пошуку шляхів підвищення нелінійності випадкової послідовності в методі градієнтного підйому значно швидше можна знайти збалансовану криптографічну функцію шляхом градієнтного спуску, як це показано, наприклад, у [17]. Тобто цей крок у розвитку евристичних методів синтезу криптографічних булевих функцій був надзвичайно вдалим та ефективним. На жаль, як було показано в дисертації [23], застосувати цей підхід до побудови векторних відображень (S-блоків) шляхом, наприклад, ітеративного формування компонентних функцій $f_i, i = 1,...,m$ з (1) практично неможливо. Дійсно, результати розрахунків з [23] та з інших пошукових робіт свідчать, що побудова байтових (і більшого розміру) векторних відображень через підбір компонентних функцій $f_i, i = 1,...,m$ є обчислювально недосяжною задачею.

Інший вдалий крок було зроблено в дисертації [12] та наукових статтях [18, 19]. Ідею градієнтного спуску було поширено на векторні відображення не через компонентні функції $f_i, i = 1,...,m$ з (1), а шляхом застосування ітеративних процедур пошуку до вхідного високонелінійного S-блоку. Наприклад, застосувавши алгебраїчну конструкцію $F(x) = x^{254}$ можна сформувати бієктивне векторне відображення із нелінійністю $N_F = 112$, алгебраїчним степенем $AD = 7$, алгебраїчним імунітетом $AI = 2$ та максимумом таблиці диференціалів $DDT_{\max} = 4$. Далі, поступово змінюючи перестановкою окремі позиції таблиці замін, можна досягти значення $AI = 3$ без суттєвого погіршення інших показників, наприклад, нелінійності та δ-рівномірності. Генерація S-блоків у такий спосіб вимагає значного часу, але вона доступна навіть із використанням однопроцесорного ПК (потребує декілька годин на генерацію однієї байтової підстановки). Звісно, час формування буде значно меншим за випадкову генерацію [23].

Подальший розвиток методів генерації було отримано в роботах [18, 19]. Шляхом одноразового застосування фіксованої кількості перестановок $NP = 22$ вдається досить швидко (за декілька хвилин) сформувати байтовий нелінійний вузол із заданими криптографічними властивостями (навість із $N_F = 104$). Однак випадкова заміна місцями лише 22 позицій у таблиці підстановок робить цей метод обмеженим для практичного використання. Фактично, із 256 позицій в таблиці підстановок лише 44 можуть бути замінені місцями, тобто у

$$\frac{256-44}{256}\cdot 100\% \approx 83\%$$

сформовані вузли формуватимуть таке саме значення на виході, як і векторне відображення, яке використовувалося як початкове (стартове) для алгоритму генерації. Відповідно, всі формовані таким чином відображення будуть на 83% подібні вихідному і задавати не випадкову підстановку.

Як підсумок, слід зазначити, що завдання генерації випадкових підстановок із необхідними криптографічними властивостями в повному обсязі не вирішено. Отримано лише часткові, обмежені для широкого застосування результати. Метою подальших досліджень в цій галузі є розробка методів та обчислювальних алгоритмів формування випадкових нелінійних вузлів замін з необхідними (близькими до верхніх теоретичних

меж) криптографічними характеристиками. Тобто з одного боку ставиться завдання зі зниження обчислювальної складності формування S-блоку, з іншого – необхідно забезпечити виконання заданих високих показників стійкості (збалансованості, нелінійності, автокореляції, алгебраїчної імунності, циклової структури та ін.).

Література

[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: монографія.– Харків: Видавництво «Форт», 2012. – 870 с.

[3] *Bart Preneel.* Analysis and Design of Cryptographic Hash Functions. [Електронний ресурс] – Режим доступа: homes.esat.kuleuven.be/~preneel/phd_preneel_feb1993.pdf

[4] *Carlet C.* Boolean functions for cryptography and error correcting codes // Cambridge Univ. Press, Cambridge. – 2007. – 148 p. [Електронний ресурс] – Режим доступа: www1.spms.ntu.edu.sg/~kkhoongm/chap-fcts-Bool.pdf

[5] *Carlet C.* Vectorial Boolean functions for Cryptography // Cambridge Univ. Press, Cambridge. – 95 p. [Електронний ресурс] – Режим доступа: www.math.univ-paris13.fr/~carlet/chap-vectorial-fcts-corr.pdf

[6] *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.

[7] *Zhuo Zepeng, Zhang Weiguo.* On correlation properties of Boolean functions // Chinese Journal of Electronics. Jan, Vol.20, 2011, №1, 143–146 pp.

[8] *X.-M. Zhang, Y. Zheng, and H. Imai.* Relating Differential Distribution Tables to Other Properties of Substitution Boxes. Des. Codes Cryptography, 19(1), pp. 45-63, 2000.

[9] *Ann Braeken.* Cryptographic Properties of Boolean Functions and S-Boxes. PhD thesis, Katholieke Universiteit Leuven (KUL), 2006, 221 p.

[10] *Fuller J.E.* Analysis of Affine Equivalent Boolean Functions for Cryptography: PhD Thesis / J.E. Fuller // Queensland University of Technology, 2003. – 187 p.

[11] *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.

[12] *Казимиров А. В.* Методы и средства генерации нелинейных узлов замены для симметричных криптоалгоритмов : диссертация на соискание учёной степени кандидата технических наук : 05.13.21 – системы защиты информации – Харьков, 2013. – 190 с.

[13] *Burnett L.* Heuristic Optimization of Boolean Functions and Substitution Boxes for Cryptography: PhD Thesis / L. Burnett. – Queensland University of Technology, 2005. – 204 p.

[14] *C. Easttom,* "A generalized methodology for designing nonlinear elements in symmetric cryptographic primitives," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2018, pp. 444-449.

[15] *W. Millan, A. Clark, E. Dawson,* "Smart Hill Climbing Finds Better Boolean Functions", Proceedings of the Workshop on Selected Areas on Cryptography SAC 97, Springer-Verlag, pp. 50-63, 1997.

[16] *Y. Izbenko, V. Kovtun and A. Kuznetsov,* "The Design of Boolean Functions by Modified Hill Climbing Method," 2009 Sixth International Conference on Information Technology: New Generations, Las Vegas, NV, 2009, pp. 356361.

[17] *Kuznetsov, I. Moskovchenko, I. Bilozertsev, S. Kavun, T. Kuznetsova.* Heuristic Methods for the Design of Cryptographic Boolean Functions. In.: ISCI'2018: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2018, pp. 45–74.

[18] *O. Kazymyrov, V. Kazymyrova, R. Oliynykov.* A Method For Generation Of High-Nonlinear S-Boxes Based On Gradient Descent. [Електронний ресурс] – Режим доступа: https://eprint.iacr.org/2013/578.pdf

[19] *M. Rodinko and R. Oliynykov.* Optimization of High Nonlinearity S-boxes Generation Method. Tatra Mountains Mathematical Publications, September 2017, 70 (1): pp. 93–105.

[20] *Кузнецов А. А.* Метод построения криптографически стойких булевых функций на основе градиентного спуска / А. А. Кузнецов, Ю. А. Избенко, И. Московченко // Зб. наук. пр. Харк. ун-ту Повітр. Сил. – Х.: ХУПС, 2007. – С. 63–66.

[21] *Yu Y.* Constructing Differentially 4 Uniform Permutations from Known Ones / Yuyin Yu, Mingsheng Wang, Yongqiang Li // Chinese Journal of Electronics. – 2013. – Vol. 22, № 3. – P. 495–499.

[22] Математичні моделі та обчислювальні методи імовірнісного формування нелінійних вузлів замін симетричних криптографічних засобів захисту інформації [Текст] : автореф. дис... канд. техн. наук : 01.05.02 / Московченко Ілларіон Валерійович ; Харківський національний ун-т ім. В.Н.Каразіна. – Х., 2009. – 20 с.

[23] Обчислювальні методи синтезу нелінійних вузлів замін для підвищення ефективності симетричних криптоперетворень : автореф. дис ... канд. техн. наук: 05.13.21 / Сергій Олександрович Ісаєв. – Харків, 2013. – 22 с.

Надійшла до редколегії 25.12.2018

Кузнєцов Олександр Олександрович, доктор технічних наук, професор, заступник головного конструктора ПАТ «ІІТ», професор кафедри безпеки інформаційних систем і технологій Харківського національного університету ім. В.Н. Каразіна. Галузь наукових інтересів – криптографія та автентифікація, алгебраїчна теорія кодів, обробка, передача та захист інформації.

Білозерцев Іван Микитович, науковий співробітник ПАТ «ІІТ», здобувач кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – криптографія, блокові симетричні шифри.

Пушкарьов Андрій Іванович, директор департаменту захисту інформації Адміністрації державної служби спеціального зв'язку та захисту інформації України. Галузь наукових інтересів – теорія захисту інформації, інформаційна та кібербезпека держави.

Горбенко Юрій Іванович, кандидат технічних наук, виконавчий директор ПАТ «ІІТ», старший науковий співробітник кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – криптографія та автентифікація, інфраструктура відкритих ключів.

Онопрієнко Віктор Васильович, кандидат технічних наук, доцент, генеральний директор ПАТ «Інститут інформаційних технологій». Галузь наукових інтересів – криптографія і автентифікація, інфраструктура відкритих ключів, теорія захисту інформації, інформаційна та кібербезпека держави.

УДК 004.056.55

Кузнецов А.А. **Исследование методов формирования случайных нелинейных узлов замен симметричных шифров** / А.А. Кузнецов, И.М. Белозерцев, А.И. Пушкарев, Ю.И. Горбенко, В.В. Ониприенко // Прикладная радиоэлектроника: науч. – техн. журнал. – 2018. – Том 17, №. 3, 4. – С. 88–95.

Обосновываются основные показатели эффективности криптографических булевых функций и векторных отображений, которые применяются в качестве узлов усложнения симметричных криптопреобразований. Исследуются эвристические методы формирования криптографических булевых функций и нелинейных S-блоков симметричных шифров, соответствующих установленным требованиям безопасности. Обосновываются перспективные направления дальнейших исследований с целью усовершенствования эвристических методов синтеза случайных узлов замены.

Ключевые слова: симметричные криптопреобразования, случайные нелинейные узлы замены, эвристические методы генерации, показатели криптографической стойкости.

Библиогр.: 23 наим.

UDC 004.056.55

Kuznetsov A. **Investigation of methods for forming random nonlinear nodes of replacing symmetric ciphe** / A. Kuznetsov, I.M. Belozertsev, A.I. Pushkarev, Yu. Gorbenko, V.Onopryenko // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 88–95.

The main indicators of the effectiveness of cryptographic Boolean functions and vector mappings, used as nodes of the complexity of symmetric crypto-transformations, are substantiated. Heuristic methods for the formation of cryptographical Boolean functions and nonlinear S-blocks of symmetric ciphers that meet the established security requirements are investigated. Prospects for further research with the aim of improving heuristic methods for the synthesis of random replacement nodes are substantiated.

Keywords: symmetric cryptotransformations, random nonlinear replacement nodes, heuristic methods of generation, indicators of cryptographic robustness.

Ref.: 23 items.

УДК 004.056.55

ПЕРІОДИЧНІ ВЛАСТИВОСТІ КРИПТОГРАФІЧНО СТІЙКИХ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

О. О. КУЗНЕЦОВ, А. С. КІЯН, Д. І. ПРОКОПОВИЧ-ТКАЧЕНКО, В. П. ЗВЄРЄВ, Е. В. КОТУХ, Т. Ю. КУЗНЕЦОВА

У цій роботі розглянуто доказово стійкі генератори псевдовипадкових послідовностей, завдання криптоаналізу яких зводиться до вирішення добре відомої і надзвичайно складної математичної задачі, що належить до класу NP-складних. Зокрема, розглянуто генератори Blum-Blum-Shub, Rivest-Shamir-Adleman, Dual Elliptic Curve і генератор на синдромному декодуванні (Pseudo-Random Generator Provably as Secure as Syndrome Decoding). Досліджено періодичні властивості формованих псевдовипадкових послідовностей. Показано, що розглянуті генератори не дозволяють сформувати послідовності максимального періоду. Крім того, для кожного генератора існують початкові стану (слабкі ключі), що призводять до катастрофічно малих довжин періодів формованих послідовностей.

Ключові слова: модель доказової безпеки, генератор псевдовипадкових чисел, періодичні властивості.

ВСТУП

Важливим напрямком сучасної криптографії є побудова криптографічно стійких (англ. Cryptographically Strong) генераторів псевдовипадкових послідовностей (ПВП), які відповідають вимогам моделі доказової безпеки (англ. Provable Security Model) [1]. Сутність цієї моделі полягає у зведенні задачі криптоаналізу до вирішення добре відомої і надзвичайно складної математичної проблеми (що належить до класу NP-складних), наприклад, факторизації, дискретного логарифмування тощо. [1]. Криптографічні примітиви, які відповідають такій моделі безпеки, прийнято називати доказово безпечними, тому що їх криптоаналіз можна порівняти з рішенням NP-складної математичної проблеми.

Обґрунтування безпеки доказово стійких генераторів базується на прийняття припущення про існування так званих односторонніх функцій [1, 2]. Одностороння функція $f: x \to y$, задана на безлічі x з областю значень в безлічі y володіє двома властивостями:

– існує поліноміальний алгоритм обчислення $f(x)$;

– не існує поліноміальною алгоритму інвертування функції $f(x)$, тобто розв'язання рівняння $f(x) = y$.

Виконання другої властивості на сьогоднішній день не доведено ні для однієї з можливих функцій $f(x)$, тобто не доведено саме існування односторонніх функцій (як є бездоказовим і припущення $P \neq NP$ в теорії складності). Водночас, на використанні різних кандидатів на односторонню функцію будуються практично всі відомі криптосистеми з відкритим ключем [2]. До претендентів на односторонню функцію належать розкладання цілих чисел на множники, проблему обчислення дискретних логарифмів або обчислення квадратного кореня за модулем складеного числа, задачу синдромного декодування, дискретного логарифмування в групі точок еліптичної кривої тощо [1–7]. Метою цієї статті є дослідження періодичних властивостей криптографічно стійких ПВП, які формуються доказово безпечними генераторами. Зокрема, в цій роботі досліджуються такі генератори: Blum-Blum-Shub (BBS) [5], Micali-Schnorr та Rivest-Shamir-Adleman (RSA) [3, 4], Dual Elliptic Curve Deterministic Random Bit Generator [6], Code-based Pseudorandom Generator [7].

2. ДОКАЗОВО СТІЙКІ ГЕНЕРАТОРИ ПВП

2.1. Генератор BBS

Найбільш важлива одностороння функція, використовувана в ході побудови генераторів ПВП, – є факторизація цілих чисел [1, 2]. Широко відомим криптопримітивом, заснованим на цій проблемі, є генератор BBS [5], запропонований в 1986 р. Ленором Блюмом, Мануелем Блюмом і Майклом Шубом.

Обчислення ПВП у генераторі BBS описується виразом:

$$x_n = x_{n-1}^2 \bmod N,$$

де $N = pq$ є добутком двох великих простих p і q і які можуть бути обидва порівнянні з числом 3 за модулем 4.

На кожному кроці алгоритму формують один біт ПВП шляхом взяття біта парності числа x_n (або одного найменш значущого біту).

Головною перевагою генератору BBS є те, що для отримання числа x_n необов'язково обчислювати всі $n-1$ попередніх чисел. Необхідно лише знати початковий стан генератора, тобто число x_0 (яке задається, наприклад, секретним ключем), а також числа p і q. Будь-який елемент послідовності описується виразом:

$$x_n = x_0^{2^n \bmod ((p-1)(q-1))} \bmod N.$$

2.2. Генератори Micali-Schnorr та RSA

Стійкість генераторів Micali-Schnorr та RSA заснована на теоретико-складній задачі обчислення *дискретних логарифмів* [1, 2]. Кожен елемент ПВП у генераторі *Micali-Schnorr* описується відповідно до виразу [2–4]:

$$x_n = x_{n-1}^e \bmod N \,. \qquad (1)$$

Початковий стан генератора, тобто число x_0, задається, наприклад, секретним ключем. На кожному кроці формується r біт ПВП шляхом зчитування r найменш значущих біт числа x_n, причому [2–4]:

$$r = \lfloor \lg(pq) \rfloor + 1 - \left\lfloor (\lg(pq)+1)(1-\frac{2}{e}) \right\rfloor$$

де

$$e \in \begin{cases} 1 < e(p-1)(q-1); \\ НОД(e,(p-1)(q-1) = 1; \\ 80e \le \lfloor \lg(pq) \rfloor + 1; \end{cases}$$

p і q – прості числа.

У генераторі *RSA* кожен елемент ПВП описується виразом (1), але на відміну від генератора *Micali-Schnorr* на вихід поступає один найменш значущий біт (біт парності) числа x_n [2–4].

2.3. Генератор Dual Elliptic Curve

У національному стандарті США NIST Special Publication 800-90A [6] визначено рекомендації щодо побудови генераторів ПВП із застосуванням різних математичних методів, у тому числі, із застосуванням перетворень у групі точок еліптичної кривої. І хоча в оновленні версії стандарту [8] цей генератор було виключено через певні недоліки, ми розглянемо алгоритм формування ПВП з метою дослідження його періодичних властивостей.

Метод формування псевдовипадкових послідовностей із використанням перетворень на еліптичних кривих, який запропоновано в рекомендаціях NIST SP 800-90, засновано на застосуванні двох скалярних множень точок еліптичної кривої та відображенні відповідних x-координат отриманих результатів у ненульове ціле значення.

Перше скалярне множення на фіксовану (базову) точку P виконується для формування проміжного стану s_i, яке циклічно оновлюється на кожній ітерації в ході функціонування відповідного генератору. Таким чином значення стану s_i залежить від значення попереднього стану s_{i-1} (на попередній ітерації) та від значення базової точки P:

$$s_i = \phi(x(s_{i-1}P)), \qquad (2)$$

де $x(A)$ – є x-координатою точки A, $\phi(x)$ – функція відображення елементів поля у ненульові цілі числа.

Початкове значення параметра s_0 формується із використанням процедури ініціалізації, яка включає введення секретного ключа (*Key*), що задає початкову ентропію (невизначеність), та хешування введеного ключа із форматуванням отриманого результату до визначеної довжини бітів. Отримане таким чином значення *Seed* засіює (ініціює) початкове значення параметра: $s_0 = Seed$.

Друге скалярне множення на фіксовану (базову) точку Q виконується для формування проміжного стану r_i, яке після відповідного перетворення і задає значення формованих псевдовипадкових бітів. Значення параметру r_i залежить від сформованого у результаті першого скалярного множення параметра s_i та від значення базової точки Q:

$$r_i = \phi(x(s_i Q)), \qquad (3)$$

Отримане таким чином значення r_i є вихідним для формування псевдовипадкових бітів, які формуються шляхом зчитування блоку з найменш значущих (правих) бітів числа r_i. ПВП формується шляхом конкатенації зчитаних бітів формованих чисел r_i.

Значення фіксованих (базових) точок задаються у вигляді констант і під час формування ПВП не змінюються.

Таким чином, розглянутий метод формування псевдовипадкових послідовностей застосовує перетворення у групі точок еліптичної кривої для формування проміжних станів s_i і r_i. Причому зворотна дія, тобто формування s_{i-1} за відомим s_i, та/або формування s_i та відомим r_i пов'язано з вирішенням теоретико-складного завдання дискретного логарифмування у групі точок еліптичної кривої. Схему формування проміжних станів генератора подано на рис. 1.

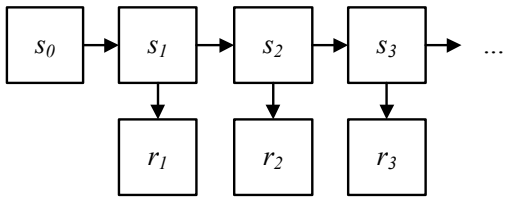

Рис. 1. Схема формування проміжних станів генератору

Як видно з рис. 1 послідовність станів $\ldots s_{i-1}$, s_i, $\ldots s_{i+1} \ldots$ формується із початкового значення $s_0 = Seed$, яке в свою чергу формується із даних секретного ключа. Кожне наступне значення s_i залежить від попереднього значення s_{i-1} і формується за допомогою скалярного множення базової точки еліптичної кривої за формулою (2).

Окремі біти ПВП формуються шляхом зчитування бітів послідовності чисел … r_{i-1}, r_i, r_{i+1},…, тобто шляхом зчитування даних, отриманих у результаті скалярного множення іншої базової точки на відповідні значення станів … s_{i-1}, s_i, … s_{i+1} … за формулою (3).

Оскільки таємний ключ *Key*, який задає правило формування послідовностей, після певних перетворень визначає початкове значення параметру s_0, відповідна стійкість розглянутого генератора базується на зведенні завдання відновлення секретних ключових даних до вирішення добре відомого і надзвичайно складного математичного завдання дискретного логарифмування у групі точок еліптичної кривої. Крім того окремі фрагменти псевдовипадкової послідовності також пов'язані між собою скалярним множенням точки еліптичної кривої, тобто, для того, щоб відновити будь-який фрагмент псевдовипадкової послідовності за якимось іншим, відомим фрагментом, потрібно вирішити завдання дискретного логарифмування у групі точок еліптичної кривої. І навпаки, якщо для розглянутого генератора за відомим фрагментом псевдовипадкової послідовності вдається відновити інший, будь-який невідомий фрагмент, або вдається відновити значення секретного ключа (або хоча б значення елементів послідовності … s_{i-1}, s_i, … s_{i+1} …) це означає, що вдається вирішити завдання дискретного логарифмування в групі точок еліптичної кривої, тобто інвертована функція (2) або (3).

2.4. Генератор синдромного декодування

Побудову цього генератора засновано на використані блокового (n,k,d) коду, який заданий своєю перевірною матрицею H розміром n стовпців і $n-k$ рядків. У теорії кодування відома NP-складна проблема синдромного декодування [9, 10]:

– за відомим вектором-синдромом s довжини $n-k$ і відомою матрицею H знайти такий вектор помилки e довжини n, що $s = e \cdot H^T$, причому вага Хеммінга (число ненульових елементів) вектору e дорівнює $w(e) = t = \left\lceil \frac{d-1}{2} \right\rceil$, де $\lceil x \rceil$ – найменше ціле число, що не менше x.

Величина t визначає здатність (n,k,d) коду, тобто це гарантоване число помилок, які можливо виправити, застосувавши метод максимальної правдоподібності. Для деяких кодів (зі спеціальною структурою матриці H) відомі швидкі алгоритми алгебраїчного декодування, тобто знаходження вектора e є поліноміально вирішуване завдання. Однак для кодів загального положення (без спеціальної структури матриці H) завдання знаходження вектора e є надзвичайно складним, найкращі алгоритми засновані на переборному пошуку.

В роботі [7] запропоновано генератор ПВП, стійкість якого заснована на вирішенні проблеми синдромного декодування (Pseudo-Random Generator Provably as Secure as Syndrome Decoding). Для формування ПВП в цьому генераторі використовується двійковий (n,k,d) код і наступне рекурентне правило: $s_i = e_i \cdot H^T$, де: e_i – двійковий вектор довжини n, $w(e_i) = t = \left\lceil \frac{d-1}{2} \right\rceil$; s_i – двійковий вектор довжини $n-k$; H – двійкова перевірна матриця (n,k,d) коду. Початковий стан e_0 генератора задається за допомогою рівноважного кодування ініційованої послідовності y_0 довжини $m = \left\lfloor \log_2 \left(\frac{n!}{t!(n-t)!} \right) \right\rfloor$ біт.

Наприклад, за допомогою секретного ключа, тобто $y_0 = Key$. Рівноважне кодування перетворює двійковий вектор y_0 довжини m у двійковий вектор e_0 довжини n, причому $w(e_0) = t$.

Черговий стан генератора e_{i+1} також формується за допомогою рівноважного кодування. Для цього двійковий вектор s_i розбивається на дві частини: $s_i = y_{i+1} \| z_{i+1}$ (тут $\|$ – символ конкатенації), причому довжина двійкового вектора y_{i+1} дорівнює m. Решта $n-k-m$ біт утворюють вектор z_{i+1}, який подається на вихід генератора як елемент ПВП. Рівноважне кодування вектора y_{i+1} дозволяє сформувати стан e_{i+1} і обчислення повторюються.

Таким чином, на кожному кроці алгоритму формуються $n-k-m$ біт ПВП, причому завдання знаходження стану e_i генератора за відомим фрагментом ПВП пов'язане з вирішенням теоретико-складної проблеми синдромного декодування.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

В ході експериментальних досліджень ставилося завдання оцінити період формованих ПВП. Для цього розглянуті вище генератори були програмно реалізовані для невеликих параметрів і виконано повний перебір всіх можливих векторів ініціалізації (секретних ключів). Для кожної ініціалізації сформована ПВП, оцінено її період. В результаті ми маємо повний набір всіх довжин періодів ПВП, які можуть бути породжені кожним генератором для відповідних вхідних параметрів.

3.1. Періодичні властивості генератора BBS

Для проведення досліджень періодичних властивостей розглянутих генераторів, експериментальні дослідження полягали у повному переборі всіх можливих значень вектора x_0, оцінці відповідних довжин пері-

оду L формованих ПВП та порівнянні з максимальною довжиною періоду $L_{\max} = \min\left(2^M - 1, n-1\right)$, де M – бітова довжина ключових даних (бітова довжина вектору x_0).

Під час проведення експериментальних досліджень генератора BBS було обрано такі вихідні дані.

Експеримент 1. Вихідні дані: $p = 19$, $q = 43$, $n = 817$. Як вектор x_0 обиралися всі цілі числа від 2 до 816.

Експеримент 2. Вихідні дані: $p = 131$, $q = 163$, $n = 21353$. Як вектор x_0 обиралися всі цілі числа від 2 до 21352.

Експеримент 3. Вихідні дані: $p = 523$, $q = 1031$, $n = 539213$. Як вектор x_0 обиралися всі цілі числа від 2 до 539212.

Експеримент 4. Вихідні дані: $p = 2039$, $q = 4099$, $n = 8357861$. Як вектор x_0 обиралися всі цілі числа від 2 до 8357860.

Отримані результати експериментальних досліджень узагальнені та наведені як діаграми на рис. 2–5. На діаграмах наведено розподіл кількостей K таких векторів x_0, які дають відповідне значення періоду L.

Таким чином, проведені експериментальні дослідження довели, що генератор BBS володіє істотним недоліком. Період сформованих ПВП менший за максимальний на 2–5 порядків (для введених загальних параметрів). При збільшенні бітового ініціалізуючого вектора різниця між максимальним та фактичним періодом формованих ПВП також збільшується. Слід також зазначити, що для кожного розглянутого випадку існують такі значення векторів x_0, при введені яких відповідний генератор формує катастрофічно погані послідовності з довжиною періоду, що дорівнює одиниці. Тобто можна стверджувати, що для цих значень (т .з. слабких ключів) генератор формує детерміновані послідовності і не виконує свої функції.

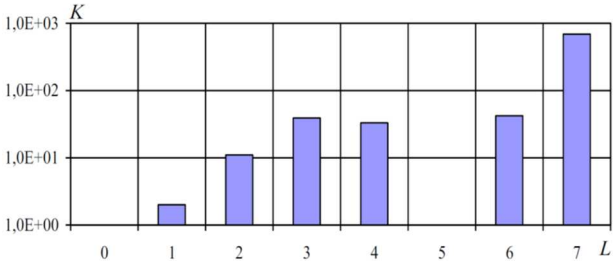

Рис. 2. Розподіл кількості K векторів x_0 по довжинах періодів формованих послідовностей (експеримент 1, $L_{\max} = 816$)

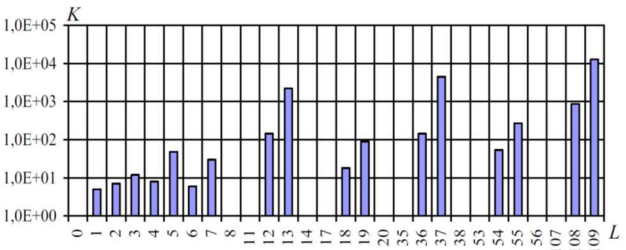

Рис. 3. Розподіл кількості K векторів x_0 по довжинах періодів формованих послідовностей (експеримент 2, $L_{\max} = 21352$)

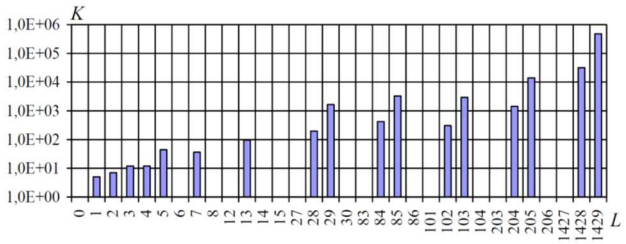

Рис. 4. Розподіл кількості K векторів x_0 по довжинах періодів формованих послідовностей (експеримент 3, $L_{\max} = 539212$)

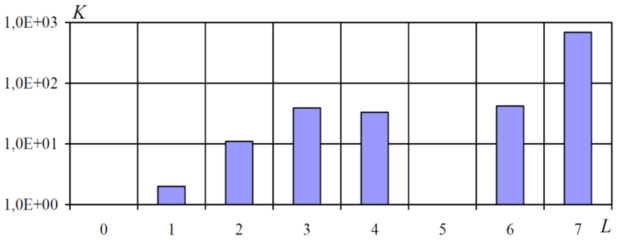

Рис. 5. Розподіл кількості K векторів x_0 по довжинах періодів формованих послідовностей (експеримент 4, $L_{\max} = 8357860$)

3.2. Періодичні властивості генератора RSA

Під час проведення експериментів обрано такі вихідні дані (рис. 6–9).

Експеримент 1. Вихідні дані: $p = 17$, $q = 37$, $n = 629$, $e = 257$. Як вектор x_0 обиралися всі цілі числа від 2 до 628.

Експеримент 2. Вихідні дані: $p = 131$, $q = 163$, $n = 21353$, $e = 1031$. Як вектор x_0 обиралися всі цілі числа від 2 до 21352.

Експеримент 3. Вихідні дані: $p = 521$, $q = 1031$, $n = 537151$, $e = 2029$. Як вектор x_0 обиралися всі цілі числа від 2 до 537150.

Експеримент 4. Вихідні дані: $p = 2053$, $q = 4099$, $n = 8415247$, $e = 2051$. Як вектор x_0 обиралися всі цілі числа від 2 до 8415246.

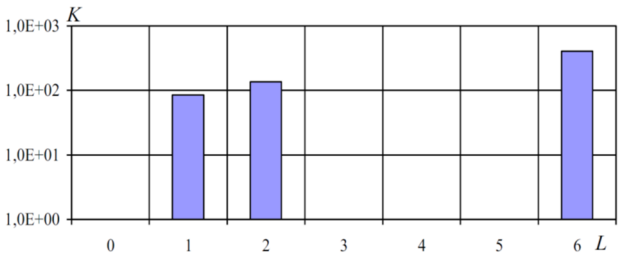

Рис. 6. Розподіл кількості K векторів x_0 по довжинах періодів формованих послідовностей (експеримент 1, $L_{\max} = 628$)

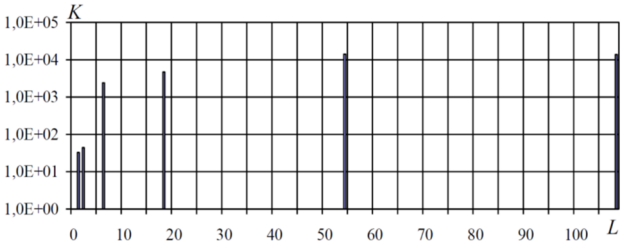

Рис. 7. Розподіл кількості K векторів x_0 по довжинах періодів формованих послідовностей (експеримент 2, $L_{\max} = 21352$)

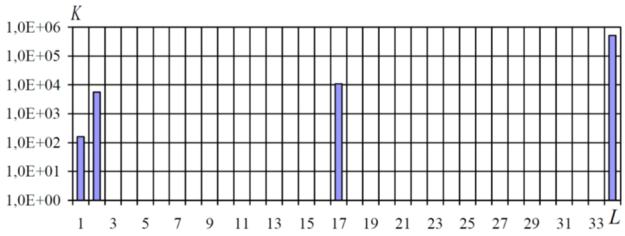

Рис. 8. Розподіл кількості K векторів x_0 по довжинах періодів формованих послідовностей (експеримент 3, $L_{\max} = 537150$)

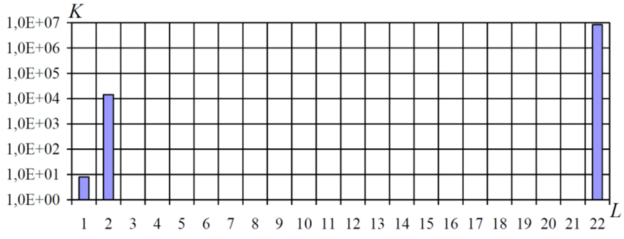

Рис. 9. Розподіл кількості ключів по довжинах періодів формованих послідовностей (експеримент 4, $L_{\max} = 8415248$)

За результатами експериментів з генератором RSA з'ясовано, що період сформованих ПВП також значно менший за максимальний. Наприклад, судячи із результатів експерименту 1 при максимальній довжині періоду $L_{\max} = 628$ фактична довжина періоду формованих послідовностей лежить у межах $L = 1..6$, тобто різниця між максимальним та фактичним періодом щонайменше у 100 разів. Судячи із експерименту 2 відповідні значення дорівнюють $L_{\max} = 21352$ і $L = 1..108$, тобто різниця між максимальним та фактичним періодом складає вже у 200 разів. У четвертому експерименті різниця між максимальним і фактичним періодом складає вже понад п'ять порядків. Генератор RSA також має слабкі ключі (вектори x_0), які призводять до катастрофічно низьких значень періоду сформованих послідовностей. Формована за допомого генератора Мікалі-Шнора ПВП за своїми періодичними властивостями не може бути кращою за ПВП, які формуються генератором RSA.

3.3. Періодичні властивості генератора Dual Elliptic Curve

Проаналізуємо роботу генератора ПВП, який використовує перетворення групи точок еліптичної кривої. Як приклад розглянемо випадок еліптичної кривої, яка задана рівнянням

$$y^2 \equiv x^3 - 3x + 4 \pmod 7,$$

причому виконується умова

$$4a^3 + 27b^2 = 2 (\text{mod } 7) \neq 0 (\text{mod } 7).$$

Ненульові точки (x_i, y_i) цієї кривої наведено у таблиці 1.

Таблиця 1

Множина ненульових точок еліптичної кривої

i	1	2	3	4	5	6	7	8	9
(x_i, y_i)	(0,2)	(0,5)	(1,3)	(1,4)	(3,1)	(3,6)	(4,0)	(5,3)	(5,4)

Припустимо, що як базові точки P і Q використовуються точки максимального порядку, наприклад, точки $P = (3,1)$ і $Q = (0,1)$. Побудуємо послідовність внутрішніх станів (2) та (3) та оцінимо періодичність цих послідовностей. Для спрощення вважатимемо, що функцію $\varphi(x)$ відображення елементів поля x у ненульові цілі числа задано як $\varphi(x) = x + 1$. Це припущення не накладає певних обмежень щодо кількості можливих неоднакових результатів відображення φ, оскільки за визначенням маємо функціональне співвідношення аргументу (елементи поля) та значення функції $\varphi(x)$ (деяке ціле число), тобто відображення є бієктивним і воно може бути подане як звичайна перестановка елементів поля. Додавання одиниці виключає формування нульового значення, виникнення якого переводить роботу генератора у вироджений стан (формується детермінована послідовність тільки нульових значень).

Отримані результати роботи генератора (значення внутрішніх станів) для всіх можливих початкових значень $s_0 = Seed$ наведено у табл. 2. Значення станів наводяться до першого повторення, бо решта значень є циклом. В останній колонці наведено період L фор-

мованих послідовностей станів, тобто найменша кількість елементів ПВП, через яку починається повторення.

Як видно із наведених даних періодичні властивості генератора ПВП незадовільні. Дійсно, отримані послідовності мають дуже малі значення періодів, в більшості послідовностей період дорівнює $L=2$, одна із послідовностей має період $L=1$, тобто на виході генератора формується одне й те саме значення. Хоча з огляду на порядок базових точок P і Q очікуване значення періоду дорівнює $L_{\max}=10$.

Таблиця 2

Внутрішні стани генератора та довжини періодів

$s_0 \setminus i$	i	**1**	**2**	**3**	**4**	**5**		***L***
$s_0=1$	s_i	4	2	6	2			2
	r_i	2	6	2	6			2
$s_0=2$	s_i	6	2	6				2
	r_i	2	6	2				2
$s_0=3$	s_i	1	4	2	6	2		2
	r_i	4	2	6	2	6		2
$s_0=4$	s_i	2	6	2				2
	r_i	6	2	6				2
$s_0=5$	s_i	5	5					1
	r_i	5	5					1
$s_0=6$	s_i	2	6	2				2
		6	2	6				2
		1	4	2	6	2		2
		4	2	6	2	6		2
		6	2	6				2
		2	6	2				2
		4	2	6	2			2
		2	6	2	6			2

3.4. Періодичні властивості генератора синдромного декодування

Для проведення експериментів були використані циклічні коди з наступними параметрами.

Експеримент 1. Вихідні дані: перевірна матриця довічного (15, 7, 5) коду. Як ініційована послідовність обиралися всі двійкові вектори довжини біт.

Експеримент 2. Вихідні дані: перевірна матриця довічного (15, 5, 7) коду. Як ініційована послідовність обиралися всі двійкові вектори довжини біт.

Експеримент 3. Вихідні дані: перевірна матриця довічного (31, 16, 7) коду. Як ініційована послідовність обиралися всі двійкові вектори довжини біт.

Експеримент 4. Вихідні дані: перевірна матриця довічного (31, 11, 5) коду. Як ініційована послідовність обиралися всі двійкові вектори довжини біт.

Отримані результати наведено як діаграми на рис. 10–13. На діаграмах наведено розподіли числа K різних векторів , які дають відповідне значення періоду ПВП. Максимальна (очікувана) довжина періоду становить біт.

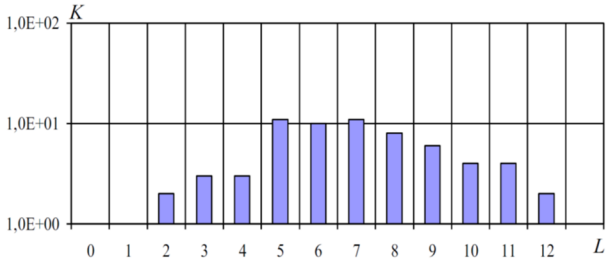

Рис. 10. Розподіл кількості ключів по довжинах періодів формованих послідовностей (експеримент 1, $L_{\max}=64$)

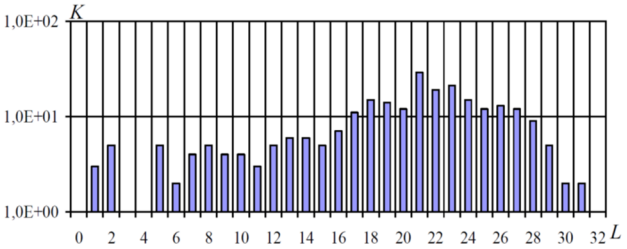

Рис. 11. Розподіл кількості ключів по довжинах періодів формованих послідовностей (експеримент 2, $L_{\max}=256$)

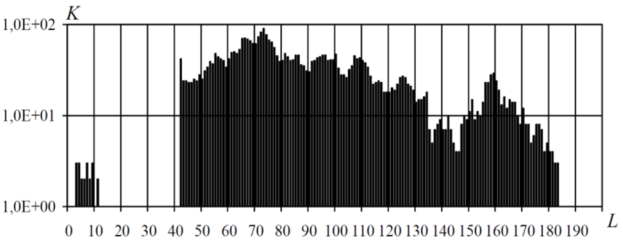

Рис. 12. Розподіл кількості ключів по довжинах періодів формованих послідовностей (експеримент 3, $L_{\max}=4096$)

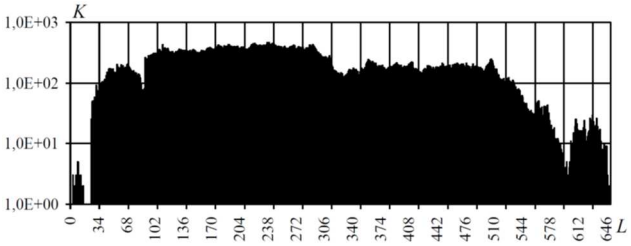

Рис. 13. Розподіл кількості ключів по довжинах періодів формованих послідовностей (експеримент 4, $L_{\max}=131072$)

Отримані результати показують, що генератор ПВП, заснований на проблемі синдромного декодування, також формує послідовності, період яких істотно нижче максимального. Це спостерігається у всіх розглянутих випадках і зі збільшенням довжини ініційованого вектора розбіжності між очікуваним і фактичним періодом збільшуються. Наприклад, для останнього випадку фактичний період менше максимального більш ніж в 200 разів.

4. ВИСНОВКИ

За результатами проведених експериментальних досліджень встановлено, що відомі доказово стійкі генератори ПВП володіють певними недоліками. Зок-

рема періоди формованих ПВП значно менші за максимальні. З підвищенням довжини вектора ініціалізації підвищується і розходження між очікуваною (максимальною) довжиною періоду і фактично спостерігаючими у формованих ПВП.

Таким чином, з огляду на високі криптографічні властивості доказово стійких генераторів актуальним є питання їх подальшого удосконалення з метою формування ПВП максимального періоду. Найбільш привабливим виглядає удосконалення генератора на синдромному декодуванні, оскільки він здатен протистояти квантовим методам криптографічного аналізу [11].

Література

[1] "Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption", April 19, 2004 – Version 0.15 (beta), Springer-Verlag, 829 p.

[2] *Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone.* Handbook of Applied Cryptography, CRC Press, 1997, 794 p.

[3] *A. Shamir.* "On the generation of cryptographically strong pseudorandom sequences". ACM Transactions on Computer Systems, vol. 1., 1983, pp. 38–34.

[4] *M. Blum, S. Micali.* "How to generate cryptographically strong sequences of pseudo-random bits". SIAM Journal on Computing, vol. 13, 1984, pp. 850–864.

[5] *L. Blum, M. Blum, M. Shub.* "A simple unpredictable pseudorandom number generator". SIAM Journal on Computing, vol. 15, 1986, pp. 364–383.

[6] *Elaine Barker and John Kelsey.* "Recommendation for random number generation using deterministic random bit generators". National Institute of Standards and Technology, January 2012, 124 p. [On-line]. Internet: https://doi.org/10.6028/NIST.SP.800–90A

[7] *Jean-Dernard Fisher, Jacques Stern.* "An efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding". EUROCRYPT'96 Proceeding, LNCS 1070, p. 245–255.

[8] *Elaine Barker and John Kelsey.* "Recommendation for random number generation using deterministic random bit generators". National Institute of Standards and Technology, June 2015, 101 p. [On-line]. Internet: https://doi.org/10.6028/NIST.SP.800-90Ar1

[9] *F. J. MacWilliams and N. J. A. Sloane.* The theory of error-correcting codes. North-Holland, Amsterdam, New York, Oxford, 1977, 762 p.

[10] *R.E. Blahut.* Theory and Practice of Error Control Codes. Addison Wesley Publishing Company, Inc., Reading, Massachusetts, 1983, 1983, 500 p.

[11] *D. Bernstein, J. Buchmann and E. Dahmen*. Post-Quantum Cryptography. Springer-Verlag, Berlin-Heidleberg, 2009, 245 p.

Надійшла до редколегії 10.10.2018

Кузнецов Олександр Олександрович, доктор технічних наук, професор, заступник головного конструктора ПАТ «ІІТ», професор кафедри безпеки інформаційних систем і технологій Харківського національного університету ім. В.Н. Каразіна. Галузь наукових інтересів – криптографія та автентифікація, алгебраїчна теорія кодів, обробка, передача та захист інформації.

Кіян Анастасія Сергіївна, магістрант кафедри безпеки інформаційних систем і технологій Харківського національного університету ім. В.Н. Каразіна. Галузь наукових інтересів – криптографія та автентифікація, алгебраїчна теорія кодів та кодові криптосистеми.

Прокопович-Ткаченко Дмитро Ігоревич, кандидат технічних наук, завідувач кафедри кібербезпеки Університету митної справи та фінансів. Заступник Голови Державної служби спеціального зв'язку та захисту інформації України (2013 – 2014 р.). Галузь наукових інтересів – інформаційна та кібербезпека держави, автентифікація та безпека безпроводових мереж.

Звєрєв Володимир Павлович, кандидат технічних наук, старший науковий співробітник, Помічник Голови Національної поліції України, Голова Державної служби спеціального зв'язку та захисту інформації України (2014 – 2015 р.). Галузь наукових інтересів – інформаційна та кібернетич-на безпека держави.

Котух Євген Володимирович, кандидат технічних наук, доцент, доцент кафедри кібербезпеки Університету митної справи та фінансів. Галузь наукових інтересів – кодова криптографія і автентифікація, обробка, передача та захист інформації.

Кузнецова Тетяна Юріївна, науковий співробітник кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – криптографія і автентифікація, обробка, передача та захист інформації.

УДК 004.056.55

Кузнецов А. А. **Периодические свойства криптографически стойких псевдослучайных последовательностей /** А. А.Кузнецов, А. С. Киян, Д. И. Прокопович-Ткаченко, В. П. Зверев, Е. В. Котух, Т. Ю. Кузнецова // Прикладная радиоэлектроника: науч. – техн. журнал. – 2018. – Том 17, №. 3, 4. – С. 96–103.

В данной работе рассмотрены доказуемо стойкие генераторы псевдослучайных последовательностей, задача криптоанализа которых сводится к решению хорошо известной и очень сложной математической задачи, относящейся к

классу NP-сложных. В частности, рассмотрены генераторы Blum-Blum-Shub, Rivest-Shamir-Adleman, Dual Elliptic Curve и генератор на синдромном декодировании (Pseudo-Random Generator Provably as Secure as Syndrome Decoding). Исследованы периодические свойства формируемых псевдослучайных последовательностей. Показано, что рассмотренные генераторы не позволяют сформировать последовательности максимального периода. Кроме того, для каждого генератора существуют начальные состояния (слабые ключи), которые приводят к катастрофически малым длинам периодов формируемых последовательностей.

Ключевые слова: модель доказуемой безопасности, генератор псевдослучайных чисел, периодические свойства.

Табл. 2. Ил. 13. Библиогр.: 11 наим.

UDC 004.056.55

Kuznetsov O. O. **Periodic properties of cryptographically secure pseudorandom sequences** / O. O. Kuznetsov, A. S. Kian, D. I. Prokopovich-Tkachenko, V. P. Zverev, E. V. Kotuh, T. Yu. Kuznetsova // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 96–103.

This paper considers evidentially secure pseudorandom sequences generators whose problem of cryptanalysis is reduced to solving a well-known and extremely complex mathematical problem that belongs to a NP-complex class. In particular, Blum-Blum-Shub, Rivest-Shamir-Adleman, Dual Elliptic Curve generators and that which is based on syndrome decoding (Pseudo-Random Generator Provably as Secure as Syndrome Decoding) are considered. The periodic properties of molded pseudorandom sequences are investigated. It is shown that the considered generators do not enable to form maximum period sequences. In addition, for each generator there are initial states (weak keys), which lead to catastrophically small lengths of periods of molded sequences.

Keywords: proof-security model, pseudorandom number generator, periodic properties.

Tabl. 2, Fig. 13. Ref.: 11 items.

УДК 004.056.55

ГЕНЕРАЦІЯ КЛЮЧІВ З БІОМЕТРИЧНИХ ОБРАЗІВ РАЙДУЖНОЇ ОБОЛОНКИ ОКА

М. С. ЛУЦЕНКО, О. О. КУЗНЕЦОВ, Ю. І. ГОРБЕНКО, А. І. ПУШКАРЬОВ, А. О. УВАРОВА

Розглядаються найпоширеніші підходи для створення біометричних криптосистем, зокрема, систем з генерацією ключа. Розробляється нова схема формування ключа методом нечітких екстракторів з біометричних даних райдужної оболонки ока. Запропонована програмна реалізація та проведено експериментальні дослідження алгоритму генерації ключів на основі біометричних даних, отриманих за розробленим методом нечітких екстракторів з райдужної оболонки ока.

Ключові слова: біометрія, біометричні криптосистеми, генерація криптографічних ключів, райдужна оболонка ока.

ВСТУП

На разі актуальним є питання поєднання класичної криптографії з технологією біометрії [1, 2]. Математичні моделі та методи захисту інформації, що засновані на використанні біометричних образів, стають одними з основних елементів у забезпеченні високо надійних ідентифікаційних та верифікаційних систем [3–16]. Наразі, вже існують методи створення біометричних ключів на основі обрисів тіла людини, обличчя, райдужної оболонки ока, голосу, геометрії долоні, відбитків пальців, обрисах судин долоні, динаміки машинного почерку і, навіть, ДНК тощо.

Біометричні дані – це унікальне цифрове представлення (модель) певної біометричної характеристики особи, яке отримане зі зчитувального біометричного пристрою (сканеру). Біометрія має декілька важливих переваг [3, 4]:

- біометрія однозначно ідентифікує осіб;
- отримані біометричні дані більш складні та випадкові, порівняно зі звичайними паролями, тому, вірогідно, матимуть більший запас криптографічної стійкості;
- біометричні дані є практично невід'ємною частиною особи, принаймні їх не можливо просто загубити, як, наприклад, носій – токен або смарт-картку – з ключем;
- біометричні образи при накладенні певних апаратних обмежень у системі, яка їх використовує, не можуть бути скопійовані та відтворені сторонньою особою.

Біометричні дані можуть бути використані у системах з генерацією ключа, у системах з відтворенням ключа та у системах зі зв'язуванням ключа. Залежно від обраного методу біометричні дані можуть як зберігатися в захищеному сховищі як шаблон для порівняння, слугувати доповненням до секретної інформації, що зберігається на певному носії або використовуватися для генерації криптографічного ключа [3–16]. Остання галузь застосування біометричних даних – генерація ключа з біометрії – не вимагає великого об'єму захищеного сховища та взагалі, його існування, має широкі можливості для використання у поєднанні з існуючими криптографічними методами. Наприклад, біометричні дані можуть бути використані як ключ для відомого алгоритму симетричного шифрування.

У біометричній криптосистемі з генерацією ключа псевдовипадкова послідовність (ключ) формується безпосередньо з біометричних даних користувача і не зберігається в системі. Це є незаперечною перевагою порівняно з іншими існуючими методами. Дійсно, такі системи є більш безпечними, але їх важко застосовувати через навіть незначну мінливість біометричних характеристик, оскільки необхідно з приблизно схожих даних згенерувати той самий ключ знову і знову. Також недоліком таких систем є неможливість (або суттєва обмеженість) сформувати новий ключ. Отже, якщо криптографічний ключ коли-небудь буде скомпрометований, то використання цього конкретного біометричного образу та конкретного алгоритму генерації ключа буде неможливе. У системі, де потрібне періодичне оновлення криптографічного ключа, це неприйнятно.

Найпоширенішою технологією, на якій базуються біометричні криптографічні системи з генерацією ключа, є нечіткі екстрактори. Метою цієї роботи є розробка та дослідження методу (нечіткого екстрактора) генерації ключів з біометричних образів райдужної оболонки ока. Розпізнавання райдужної оболонки ока – це автоматизований метод біометричної ідентифікації, який використовує математичні методи розпізнавання образів на фото та відео зображеннях [3–16]. Ця характеристика є складною, унікальною та стабільною. Розпізнавання райдужної оболонки використовує технологію відеокамери з інфрачервоним підсвічуванням для отримання зображень детальних та складних структур райдужної оболонки, які є видимими зовні.

1. РОЗРОБКА СХЕМИ ВИЛУЧЕННЯ БІОМЕТРИЧНИХ ДАНИХ МЕТОДОМ НЕЧІТКИХ ЕКСТРАКТОРІВ З РАЙДУЖНОЇ ОБОЛОНКИ ОКА

Розглянемо сучасний стан існуючих методів вилучення біометричних образів та створення криптосистем з генерацією ключів.

Дагман (англ. John Daugman) запропонував інтегро-диференціальний оператор для локалізації областей райдужки ока разом із видаленням можливих шумів повік [5, 6]. Проте автор не зазначив чи обробляються також шуми від зіниці та вій. Уайлдс (англ. Richard P. Wildes) виконує сегментацію райдужки за допомогою простих операцій фільтрування та гістограми [7]. За даною методикою можливо виявити шуми від повік, проте не від вій або зіниці. Боулз (англ. W. W. Boles) і Боаша (англ. B. Boashah) [8], Лім (англ. Lim) і співавт. [9] і Нох (англ. Noh) [10] головним чином зосереджувалися в своїх розробках на представленні зображення райдужки та узгодженні функції, і не вводили інформацію про сегментацію. Тиссе (англ. Tisse) та інші [11] запропонували метод сегментації на основі інтегро-диференціальних операторів з перетворенням Хафа. Це зменшило час обчислень і виключило потенційні центри поза знімком очей. Вії та зіниця також не враховувались у цьому методі. Ме (англ. Ma) та інші [12] запропонували оброблення райдужки шляхом сегментації та простої фільтрації.

Схема нечіткого екстрактора була модифікована Бойєном (англ. Boyen), в якій до хешування виконується фіксована перестановка біт двійкових послідовностей, отриманих з райдужної оболонки ока. Таким чином, передбачається формування декількох різних криптографічних ключів на основі одних і тих самих біометричних даних, але з використанням різних перестановок. Додатково секретним ключем в такому випадку може слугувати перестановка. Також, передбачається, що компрометація одного ключа не призведе до компрометації біометричних даних особи чи неможливості певної особи користуватися системою.

Хао (англ. Hao), Андерсон (англ. Anderson) and Дагман (англ. Daugman) представили метод генерації криптографічного ключа на основі біометричного методу, який використовує дані райдужки ока та дворівневу методику корекції помилок даних, що об'єднує коди Адамара та Рида-Соломона [6].

1.1. Метод нечітких екстракторів

Метричний простір – це множина M з функцією відстані

$$dis : M \times M \to \mathbb{R}^+ = [0, \infty).$$

У термінах даної роботи, нехай M завжди буде кінцевою множиною, а функція відстані прийматиме лише цілі значення та задовольнятиме такі вимоги:

– виконується аксіома тотожності:

$$dis(x, y) = 0 \text{ при } x = y;$$

– виконується аксіома симетрії:

$$dis(x, y) = dis(y, x);$$

виконується нерівність трикутника:

$$dis(x, z) \le dis(x, y) + dis(y, z).$$

Зосередимось на наступних показниках.

Відстань Хеммінга (англ. Hamming metric). Нехай, якщо $M = F^n$ для деякого алфавіту F, тоді $dis(w, w')$ – це кількість позицій, в яких рядки w та w' відрізняються.

Множина показників різниці (англ. Set difference metric). Нехай M складається з усіх підмножин універсальної множини U. Для двох множин w, w' їхня симетрична різниця дорівнює

$$w \Delta w' \overset{def}{=} \{x \in w \cup w' \mid x \notin w \cap w'\}.$$

Отже, відстань між двома множинами w, w' – це $|w \Delta w'|$.

Відстань редагування (англ. Edit metric). Нехай $M = F^n$, та відстань між w та w' визначається як найменша кількість додавань та видалень символів, необхідних для перетворення w в w'. (Ця метрика відрізняється від відстані Хеммінга, оскільки тут відбувається зсув символів).

Як вже згадувалося, всі три показники належать до біометричних даних.

Передбачуваність випадкової величини A визначається як $\max_a \Pr[A = a]$ і, відповідно, мінімальна ентропія $H_\infty(A)$ дорівнює

$$H_\infty(A) = \min_a - \log_2 \Pr(A = a) =$$

$$= -\log_2 \left(\max_a \Pr[A = a]\right) = \min_a \left\{ \log_2 \frac{1}{\Pr[A = a]} \right\},$$

тим самим мінімальна ентропія може розглядатися як найгірший випадок ентропії.

Мінімальна ентропія розподілу послідовності визначає можливість отримати майже однакові випадкові біти.

Статистична відстань є мірою розрізнення. Статистична відстань між двома ймовірними розподілами A і B становить

$$SD(A, B) = \frac{1}{2} \sum_v |\Pr(A = v) - \Pr(B = v)|.$$

Для будь-якої системи, якщо A замінюється на B, вона вестиме себе як оригінальна система з ймовірністю принаймні $1 - SD[A, B]$.

За визначенням, (M,m,l,t,e) -нечітким екстрактором є пара рандомізованих процедур, процедури генерації *Gen* і процедури відтворення *Rep* з такими властивостями [10]:

1. Процедура генерації *Gen* на вході отримує $w \in M$ та на виході формує рядок $R \in \{0,1\}^l$ та допоміжні дані $P \in \{0,1\}^*$.

Процедура відтворення *Rep* приймає на вхід елемент $w' \in M$ і бітовий рядок $P \in \{0,1\}^*$. Властивість коректності нечітких екстракторів гарантує, що, якщо $dis(w,w') \le t$ та R, P були згенеровані за допомогою $(R, P) \leftarrow Gen(w)$, то $Rep(w', P) = R$. Якщо $dis(w,w') > t$, тоді ніякої гарантії не надається щодо вихідних даних. Властивість безпеки гарантує, що для будь-якого розподілу W на M мінімальної ентропії m, рядок R має приблизно рівномірний розподіл навіть у випадку якщо зловмиснику відомо P, якщо $(R, P) \leftarrow Gen(W)$, то $SD((R, P),(U_l, P)) \le e$.

Нечіткий екстрактор ефективний, якщо *Gen* і *Rep* виконуються за менше, ніж поліноміальний час.

Іншими словами, нечіткі екстрактори дозволяють витягнути деяку випадковість R з w, а потім успішно відтворити R з будь-якого рядка w', близького до w. У відтворенні використовуються допоміжні дані P, що утворюються під час початкового вилучення; однак P не обов'язково мають зберігатися в таємниці, оскільки R близькі до дійсно випадкової послідовності, навіть, якщо відомі P. Сильні екстрактори дійсно можуть розглядатися як "чіткі" аналоги нечітких екстракторів, що відповідають якщо припустити, що $t = 0$, $P = X$ та $M = \{0,1\}^n$.

1.2. Вилучення біометричних даних з райдужної оболонки ока

Райдужна оболонка ока визнана одним із найнадійніших, унікальних та неінвазійних біометричних методів для отримання біометричних даних особи. Вважається, що ця метрика особи немає залежності від генетики людини, а отже гарантує унікальну ідентифікацію людини. Розпізнавання райдужної оболонки ока розглядається як найбільш надійна та точна біометрична система ідентифікації, яка використовується в сучасну епоху. Більшість комерційних систем розпізнавання оболонок ока використовують запатентовані алгоритми, розроблені Дагманом, і ці алгоритми мають високий рівень коректності розпізнавання особи.

Як фізіологічний параметр розглядається райдужна оболонка ока – кругла пластинка з кришталиком в центрі, одна з трьох складових судинної оболонки ока. Райдужна оболонка знаходиться між рогівкою і кришталиком і виконує функцію своєрідної природної діафрагми, регулюючої надходження світла в око. Райдужна оболонка пігментована. За своєю структурою райдужна оболонка складається з еластичної матерії – трабекулярної мережі. Трабекулярна мережа – сітчасте утворення, що складається з поглиблень, гребінчастих стяжок, борозен, кілець, зморшок, веснянок, судин і інших рис. Завдяки такій кількості складових «візерунок» мережі досить випадковий (вважається, що ймовірність співпадіння двох райдужок різних людей дорівнює 2^{-78}), що веде до великої ймовірності унікальності райдужної оболонки. Візерунок траберкулярної мережі залишається незмінним протягом усього життя людини. Винятком вважається отримання серйозної травми і хірургічне втручання.

У даній роботі усі зображення були взяті з бази даних райдужних оболонок ока CASIA Iris Image Database. Зображення цієї бази даних CASIA-Iris-Interval були захоплені камерою, що є новою розробкою компанії CASIA для захоплення райдужки ока. Завдяки новому дизайну ця камера може захоплювати дуже чіткі зображення райдужки.

Сегментація. Сегментація – це процес пошуку найбільш корисної частини зображення райдужки для подальшої обробки. Це робиться шляхом локалізації зіниці та межі райдужної оболонки, вії та повік. У випадку відсутності належної сегментації, наступні етапи розпізнавання райдужки будуть некоректними через велику кількість сторонніх даних (шум), у результаті будуть сформовані помилкові дані як шаблон. Для швидкої сегментації райдужної оболонки [13–15] можливо локалізувати її простим поєднанням гауссівської фільтрації (англ. Gaussian filtering), виявлення краю райдужки оператором Кенні (англ. Canny edge detection operator) та трансформації Хафа (англ. Hough transform). Трансформація Хафа [16] використовується для визначення радіуса та центру зіниці та райдужки. Оператор Кенні використовується для виявлення країв зображення райдужки, він є достатньо ефективним.

Оператор Кенні. Алгоритм виявлення меж райдужної оболонки ока за допомогою оператора Кенні можна розділити на 5 етапів:

1. Застосувати фільтр Гаусса для згладжування зображення, щоб вилучити шум та вилучити небажані деталі зображення:

$$g(m,n) = G_\sigma(m,n) \cdot f(m,n),$$

де

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{m^2+n^2}{2\sigma^2}}.$$

2. Знайти градієнти інтенсивності зображення. Для цього можна використати будь-який з існуючих

градієнтних операторів: Робертса (англ. Roberts), Собеля (англ. Sobel), Прейвіта (англ. Prewitt), тощо. У результаті

$$M(n,n)=\sqrt{g_m^2(m,n)+g_n^2(m,n)}$$

та

$$\Theta(m,n)=\frac{1}{\tan\left(\frac{g_n(m,n)}{g_m(m,n)}\right)}.$$

3. Застосувати заглушення немаксімумів, щоб позбутися від помилкових даних на виявленому краї. Тобто на основі заданого порогового значення формуються приблизні межі райдужки:

$$M_T(m,n)=\begin{cases}M(m,n), & if\ M(m,n)>T\\ 0, & if\ M(m,n)\le T\end{cases},$$

де T обирається таким, що всі елементи краю збереглися, тоді як більша частина шуму була вилучена.

4. Застосувати подвійний поріг для визначення потенційних меж. Межу, отриману на попередньому етапі, потрібно уточнити. Подамо $M_T(m,n)$ у вигляді масиву даних довжиною k елементів, тобто

$$M_T(m,n)=\{M_{T0}(m,n),M_{T1}(m,n),\ldots,M_{Tk}(m,n)\}.$$

У такому випадку до кожного з таких елементів застосовуватимуться:

$$M_{Ti}(m,n)\begin{cases}M_{Ti}(m,n), & if\ M_{Ti}(m,n)>M_{Ti-1}(m,n),\\ & M_{Ti}(m,n)>M_{Ti+2}(m,n)\\ 0, & otherwise\end{cases}$$

де $M_{Ti-1}(m,n)$ та $M_{Ti+1}(m,n)$ – сусідні елементи за напрямом градієнту $\Theta(m,n)$.

5. Виділити край райдужки, пригнічуючи всі інші ребра, які слабкі і не пов'язані з сильними краями райдужки. Виділити дані, отримані на попередньому етапі за двома пороговими значеннями τ_1 та τ_2 ($\tau_1<\tau_2$) для отримання двох двійкових зображень T_1 та T_2. Слід зауважити, що T_2 зі збільшенням τ_2 містить менше шуму та помилкових меж, проте збільшуються прогалини поміж сегментами межі, відповідно це справедливо і для T_1 зі зменшенням τ_1. Потім необхідно поєднати сегменти T_2, щоб утворити суцільну межу. Для цього, необхідно знайти кінець сегменту T_2 та відстежити його сусідів у T_1, поєднати ці два елементи, а потім продовжити шукати сусідів елементу з T_1 у T_2.

Трансформація Хафа. Зазвичай перетворення Хафа використовується для виявлення певних ліній, проте можливе використання і для виявлення кругової області зображення, в даному випадку, зіниці ока. Отже алгоритм складається з наступних етапів.

1. Створити аккумулятивний простір, який за розміром співпадає з розміром вхідного зображення. Спочатку для кожної комірки встановлено значення 0.

2. Для кожної точки межі (i,j) на зображенні, інкрементуються усі точки, які згідно з рівнянням

$$(i-a)^2+(j-b)^2=r^2$$

можуть бути центром кругу. Такі точки у рівнянні позначені як a.

3. Для кожного можливого значення a, знайденого на попередньому кроці, виконується пошук усіх можливих значень b, які задовольняють рівняння.

4. Виконується пошук локальних максимумів в акумулятивному просторі. Ці точки становлять кола, які були виявлені алгоритмом.

Якщо не відомо радіус кола, який необхідно знайти заздалегідь, можливе використання тривимірного простору акумулятора для пошуку кругів з довільним радіусом. Проте це значно погіршить швидкодію алгоритму.

Цей метод також може виявити кола, які частково знаходяться за межами акумулятивного простору, якщо в цьому просторі ще є достатня кількість області кола.

Нормалізація. Після визначення меж зображення райдужної оболонки необхідно нормалізувати. Це необхідний крок, покликаний компенсувати зміни розмірів зіниці. Локалізована райдужка нормалізується до прямокутного блоку з фіксованим розміром, що відповідає ширині блоку, а кутове зміщення Θ відповідає довжині блоку, як показано на рис. 1.

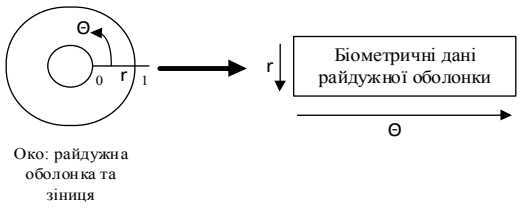

Рис. 1. Схематичне подання нормалізації зображення райдужної оболонки ока

Найпоширенішим способом нормалізації є побудова моделі розгортки Дагмана (англ. Daugman's rubbersheet model) [5, 6]. Формально нормалізація це лінійна модель, яка формується з відповідних кожному пікселю райдужної оболонки, незалежно від її розміру і стану розширення зіниці, пару полярних координат (r,Θ), де r знаходиться на одиничному інтервалі $r\in[0,1]$ і Θ є кут в діапазоні $\Theta\in[0,2\pi]$. Відображення зображення райдужної оболонки (x_i,y_i) та зіниці

(x_p, y_p) від декартової системи координат до безрозмірної неконцентричної полярної системи координат (r,Θ) можна подати так:

$$I\big(x(r,\Theta), y(r,\Theta)\big) \to I(r,\Theta),$$

що складається з

$$x(r,\Theta) = (1-r)\,x_p(\Theta) + x_i(\Theta),$$

$$y(r,\Theta) = (1-r)\,y_p(\Theta) + y_i(\Theta),$$

де:

$x(r,\Theta)$ та $y(r,\Theta)$ визначаються як лінійні комбінації множини точок зорієнтованої межі зіниці $\big(x_p(\Theta), y_p(\Theta)\big)$, так множини кінцевих точок зовнішнього периметру райдужної оболонки $\big(x_i(\Theta), y_i(\Theta)\big)$,

$I\big(x(r,\Theta), y(r,\Theta)\big)$ – область райдужної оболонки ока у декартовій системі кооординат,

$I(r,\Theta)$ – це область райдужної оболонки ока, що представлення в полярних координатах.

Приклад нормалізації зображення райдужки ока наведено на рис. 2.

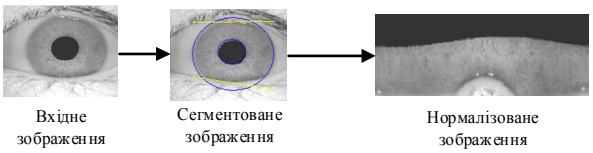

Рис. 2. Формування нормалізованого зображення райдужної оболонки ока

Формування даних. Формування даних – це процес вилучення інформації з зображення райдужки. В ході вилучення даних райдужної оболонки з нормалізованого зображення виділяють контрольну область. До кожної точки обраної області застосовують фільтри Габора (англ. Gabor filter) [13–15], для того, щоб витягти фазову інформацію. Можна застосовувати й інші фільтри, але принцип залишається таким самим. Існує декілька показників, за якими можна порівняти фільтри, а саме [3–16]:

1. FAR (англ. False Accept Rate) – ймовірність помилкової ідентифікації, тобто ймовірність того, що система ідентифікації помилково визнає справжність користувача, якого не зареєстровано в системі.
2. FRR (англ. False Reject Rate) – ймовірність того, що система біоідентифікації не визнає справжність біометричних даних зареєстрованого в ній користувача.
3. EER (англ. Equal error rate) – коефіцієнт рівного рівня помилок, при якому обидві помилки (помилка прийому і помилка відхилення) еквівалентні. Чим менше EER, тим точнішою буде система.

Безсумнівним плюсом фазової складової є те, що вона, на відміну від амплітудної інформації не залежить від контрасту зображення і освітлення. Слід зазначити, що ці дані не можуть бути використані для реконструкції зображень ока. Фільтр Габора дають змогу сформувати інваріантну систему для вилучення біометричних ознак. Фільтр Габора – це лінійний фільтр, який є згорткою перетворень Фур'є гармонійної функції і функції Гауса.

Нехай f позначає частоту синусоїдальної плоскої хвилі, а α і β – просторові константи Гаусової оболонки по осі x' і y', а Θ – орієнтація фільтра Габора. В такому випадку, двовимірний фільтр Габора визначається як:

$$G(x,y,\Theta) = \frac{1}{2\pi\alpha\beta} e^{-\left(\frac{x'^2}{2\alpha^2}\right)-\left(\frac{y'^2}{2\beta^2}\right)} \cos(2\pi f x'),$$

$$x' = x\sin\Theta + y\cos\Theta,$$

$$y' = -x\cos\Theta + y\sin\Theta.$$

Відповідне подання в просторово-частотній області:

$$H(u,v) = e^{-2\pi^2\left[(u-F)^2\alpha^2 + v^2\beta^2\right]}.$$

Обробка біометричних даних. Залежно від обраних алгоритмів сегментації, нормалізації та формування даних, можна отримати різний об'єм біометричних даних. Наприклад, лише фільтр Габора може формувати блоки даних довжиною від 1 до 256 біт. Використання блоків певної довжини залежить від потреб системи, так наприклад, використання блоку довжиною, наприклад, 16 біт надає можливість ігнорувати деякий шум зображення. Оскільки передбачається формування криптографічного ключа з біометричних даних, використання блоку довжиною 256 біт має більший запас стійкості перед криптографічними атаками типу груба сила, проте у таких системах більш високі критерії до відсутності шумів у біометричних даних. Зазвичай, параметри перетворення фільтра Габора обирають компромісно залежно від вимог до рівня криптографічної безпеки системи.

Отже, процедури етапу формування біометричних даних можна подати як це зображено на рис. 3.

Після етапу формування біометричних даних починається етап генерації криптографічного ключа.

1.3. Генерація криптографічного ключа з біометричних даних райдужної оболонки ока методом нечітких екстракторів

Етап генерації ключа складається з двох складових:

- виконання завадостійкого кодування біометричних даних;
- хешування отриманої послідовності.

Завадостійке кодування потрібно, безпосередньо, як реалізація методу нечітких екстракторів.

Процедура хешування опціональна. Вона необхідна для того, щоб компрометація сформованого ключа не призводила до компрометації біометричного образу або не унеможливлювала (не обмежувала) можливості щодо використання певною особою системи. Тобто у разі компрометації ключа, у системі буде на основі тих самих біометричних даних сформовано новий ключ лише з використанням іншого набору параметрів для хешування.

Рис. 3. Запропонована схема обробки зображення для вилучення біометричних даних з райдужної оболонки ока

Завадостійке кодування кодами Ріда-Соломона. Коди Ріда-Соломона – це циклічні коди [1–2], що виправляють помилки, з елементами – блоками довільної довжини. Усі елементи належать полю $GF(2^k)$, усі операції також виконуються в цьому полі. В даній роботі було використано поле $GF(2^8)$ за породжуючим поліномом:

$$y = x^8 + x^4 + x^3 + x^2 + 1.$$

Дамо визначення основним змінним перетворень:

– Нехай $\left\{\alpha^0, \alpha^1, \alpha^2, ..., \alpha^{|F|-2}\right\}$ будуть елементами поля $GF\left(2^8\right)$;

– k означатиме довжину біометричних даних, що підлягають кодуванню, $1 \le k < |F|$;

– m – це кількість помилок, які можуть бути виправлені, $1 \le m < |F| - k$;

– Нехай $n = k + m$ означатиме довжину блоку даних, який отримано після кодування, $2 \le n < |F|$.

Процедура кодування має такий вигляд [1–3].

Вхідні дані: послідовність $M = \left(M_0, M_1, ..., M_{k-1}\right)$, $M_i \in GF\left(2^8\right)$.

Вихідні дані: послідовність $s = \left(s_0, s_1, ..., s_{n-1}\right)$, $s_i \in GF\left(2^8\right)$.

1. Визначимо поліном генератора як:

$$g(x) = \prod_{i=0}^{m-1}\left(x - \alpha^i\right) = \left(x - \alpha^0\right)\left(x - \alpha^1\right)...\left(x - \alpha^{m-1}\right).$$

2. Визначимо поліном послідовності:

$$M(x) = \sum_{i=0}^{k-1} M_i x^i = M_0 x^0 + M_1 x^1 + ... + M_{k-1} x^{k-1}.$$

3. Кодові слова коду Ріда-Соломона визначається як:

$$s(x) = M(x)x^m - \left[\left(M(x)x^m\right) \bmod g(x)\right],$$

$$s(x) = \sum_{i=0}^{n-1} s_i x^i = s_0 x^0 + s_1 x^1 + ... + s_{n-1} x^{n-1}.$$

Процедура декодування виконується в декілька етапів, а саме [1–3]:

1) Обчислення синдрому помилки;
2) Побудова поліному помилки;
3) Знаходження коренів поліному помилки;
4) Визначення характеру помилки та виправлення її.

На сьогодні коди Ріда - Соломона мають дуже широку сферу застосування завдяки їхній здатності знаходити і виправляти багаторазові помилки.

Хешування даних алгоритмом Купина (ДСТУ 7564:2014). Купина (англ. Kupyna) – ітеративна криптографічна функція хешування, прийнята як національний стандарт України ДСТУ 7564:2014 «Інформаційні технології. Криптографічний захист інформації. Функція хешування» [17].

ДСТУ 7564:2014 визначає функцію хешування, передбачається використання двох основних реалізацій «Купина-256» або «Купина-512», що формують хеш довжиною 256 та 512 біт відповідно.

Алгоритм генерації хешу має наступний вигляд. В ході формування хешу повідомлення доповнюється, далі поділяється на l -бітні блоки $m_0,...,m_t$, після чого виконується обробка кожного блоку шляхом ітеративного виконання функції стиснення ϕ. При цьому формуються значення $h_i = \varphi(h_{i-1}, m_i)$, де $i = 1,...,t$, а початкове значення $h_0 = IV$. Після обробки останнього блоку повідомлення результуюче хеш-значення обчислюється як $H(M) = \Omega(h_t)$, де Ω – завершальне перетворення, що повертає n-бітне значення, кратне 8 ($0 < n \le \frac{l}{2}$).

Основні перетворення. Функція стиснення ϕ складається з перетворень l -бітного блока P і Q та визначається так:

$$\varphi(h,m) = P(h \oplus m) \oplus Q(m) \oplus h .$$

Завершальне перетворення Ω визначається так:

$$\Omega(x) = trunc_n(P(x) \oplus x) .$$

Перед виконанням перетворень P і Q вхідна послідовність подається як внутрішній стан функції хешування довжиною l біт ($l = 512$ або $l = 1024$ залежно від розміру внутрішнього стану). Після завершення виконання перетворень P і Q внутрішній стан знову трансформується у послідовність байт, яка подається на вхід наступної ітерації функції стиснення ϕ або на завершальне перетворення для формування результуючого хеш-значення.

Перетворення P і Q є варіантами блокового шифру, в яких замість циклових ключів використовуються визначені константи. Кількість циклів (r) залежить від розміру внутрішнього стану.

Функція хешування, визначена в ДСТУ 7564:2014, формує хеш-значення для повідомлення, що складається з бітової послідовності довжини від 0 біт (порожній рядок) до $2^{96} - 1$ біт. У даній роботі було використано алгоритм хешування для формування послідовності довжиною 512 біт.

Результати проведених ретельних досліджень [1] криптографічної стійкості ДСТУ 7564:2014 дозволяють зробити висновок про її високий рівень захищеності. Тому алгоритм, що реалізований в ДСТУ 7564:2014, забезпечує стійкість до атак на алгоритми хешування, зокрема до атак знаходження прообразів та здійснення колізій.

Формування криптографічного ключа. Формування ключа відбувається методом нечітких екстракторів за схемою приватного шаблону. Алгоритм генерації ключа схематично подано на рис. 4.

Для покращення властивостей роботи алгоритму використовуватимуться допоміжні дані – біти перевірки завадостійкого коду Ріда-Соломона для виправлення помилок. Нижче наведено алгоритм формування криптографічного ключа запропонованої схеми.

Алгоритм генерації ключа складається із виконання наступних кроків.

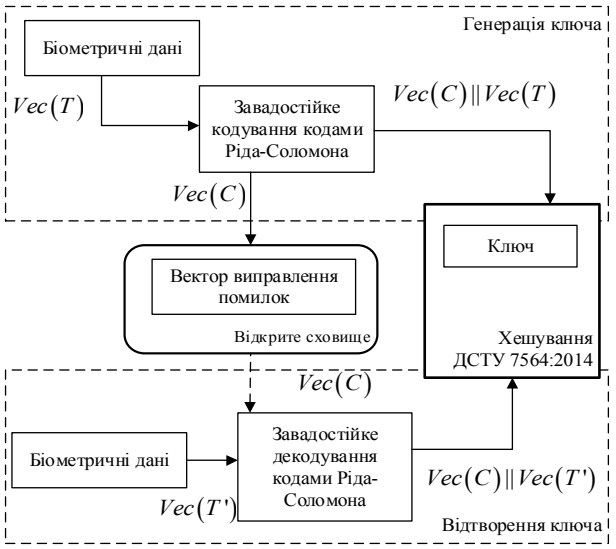

Рис. 4. Схематичне зображення запропонованої біометричної криптографічної системи з генерацією ключа

1. До біометричного образу T довжиною M біт, застосовується кодування кодами Ріда-Соломона. У результаті формується вектор корекції помилок $Vec(C)$

$$Vec(C) = (C_1, C_2, ..., C_n),$$

для n-бітного коду, визначений як $Vec(c_i) = (c_{i,1}, c_{i,2}, ..., c_{i,n})$.

Потім, вектор контрольної суми поєднується з біометричними даними

$$Vec(C) \| Vec(T) .$$

Загалом, послідовність після кодування має вигляд

$$Vec(C,T) = (C_1, C_2, ..., C_n \| T_1, T_2, ..., T_M) .$$

2. Отриманий після кодування вектор корекції помилок $Vec(C)$ може зберігатися у будь-якому відкритому сховищі, наприклад, на смарт-карті або токені.

3. Для формування ключа виконується хешування за алгоритмом хешування Купина-512 послідовності $Vec(C) \| Vec(T)$

$$Key = Hash[Vec(C) \| Vec(T)] .$$

4. У результаті сформована послідовність довжиною 512 біт, яка придатна для використання як криптографічний ключ.

Алгоритм відтворення ключа:

1. До біометричного образу T' довжиною M біт додається вектор корекції помилок, що було сформовано (3.33):

$$Vec'(C,T') = (C_1, C_2, ..., C_n \| T_1', T_2', ..., T_M').$$

2. До отриманої послідовності $Vec'(C,T')$ застосовується декодування алгоритмом Ріда-Соломона. Після успішного декодування формується вектор $Vec(T')$. У разі неможливості правильного декодування послідовності $Vec'(C,T')$ зчитується новий біометричний образ, заново застосовується алгоритм відтворення ключа з п.1.

3. Для формування ключа виконується хешування $Vec(C) \| Vec(T')$ та, залежно від реалізації, з допоміжними даними

$$Key = Hash\left[Vec(C) \| Vec(T')\right].$$

Відтворений ключ можна використовувати за призначенням.

2. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

2.1. Вхідні та вихідні дані алгоритму генерації

Умовно розділимо алгоритм генерації на два етапи: вилучення біометричних даних та генерацію ключа.

На етапі вилучення біометричних даних вхідними даними буде зображення (двовимірний масив даних) ока певного розміру. Розмір зображення залежить виключно від фізичного обладнання, що було використано для захоплення зображення ока людини. Якщо використовувати зображення ока, що містяться у бібліотеці CASIA, то розмір зображення ока дорівнюватиме 320×280 пікселів.

Розмір вихідних даних також може варіюватися. Використання у алгоритмі фільтрів, таких як фільтр Габора, дає змогу варіювати (квантувати) значення елементів результуючої послідовності біометричних даних у діапазоні від 1 до 256 біт. Кодування блоку пікселів меншою бітовою послідовністю дає змогу зневажити вплив яскравості зображення або інших завад на формування даних, проте знижується ймовірність коректного формування ключа.

Загалом, результатом застосування фільтра Габора є вектор $Vec_{GABOR}(BD)$

$$Vec_{GABOR}(BD) = (BD_1, BD_2, ..., BD_m)$$

довжиною $m = 8192$ елементи, які

$$BD_i \in GF\left(2^k\right), i = 1,2,...,m,\ k = 1,2,...,8,$$

де $GF\left(2^k\right)$ – розширення двійкового поля Галуа ступеня k, де k може приймати значення від ступеня квантування.

Проте зважаючи на те, що отримати повне зображення райдужки ока досить важка задача, оскільки у нормальному стані око людини відкрито неповністю, райдужку можуть частково перекривати верхнє та нижнє повіки, вії. Отже, приблизно ¾ райдужної оболонки ока можливо захопити без спричинення незручностей для користувача системи. У контексті алгоритму це означає, що більш доречно враховувати лише 6144 елементи вектора, отриманого після застосування фільтра Габора, тобто $M = 6144$.

Отже, вихідними даними етапу вилучення даних є вектор

$$Vec_{GABOR}(BD) = (BD_1, BD_2, ..., BD_M)$$

з елементами

$$BD_i \in GF\left(2^8\right), i = 1,2,...,M.$$

Вхідними даними для етапу генерації ключа є бітовий масив довжиною $L = M \times k = 6144 \times 8 = 49152$ біт, що було сформовано з вектора $Vec_{GABOR}(BD)$. Вихідними даними є бітова послідовність довжиною 512 біт – криптографічний ключ.

2.2. Приклад використання програмного забезпечення

Етап вилучення даних з зображення ока подано на рис. 5.

Найчастіше, нормалізоване зображення райдужної оболонки фільтрується набором двовимірних фільтрів Габора з $\Theta = 0^o$, $\Theta = 45^o$, $\Theta = 90^o$ та $\Theta = 135^o$. На рис. 5 подано усі чотири варіанти застосування фільтра. Проте, для того щоб обрати певний фільтр для формування даних у реалізації, була проведена перевірка відповідності даних отриманих після фільтрування до рівномірного розподілу. Сформовані вектори $Vec_{GABOR}(BD)_\Theta$ були перетворені у двійкові послідовності.

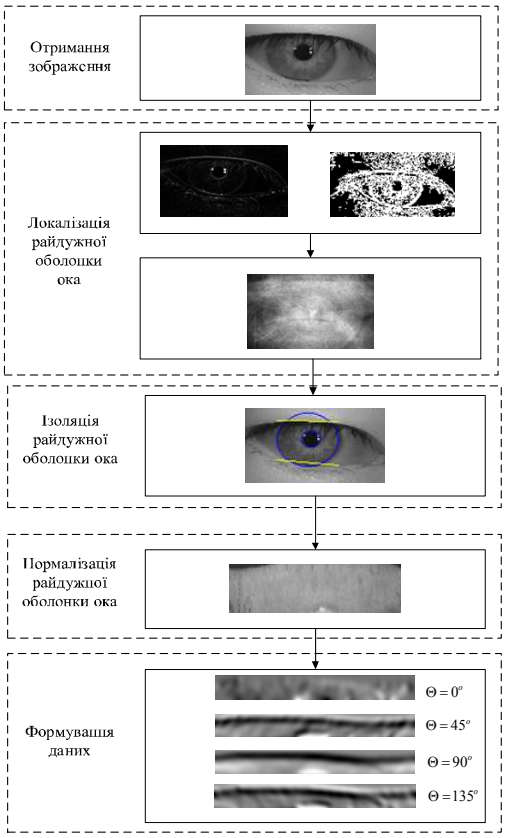

Рис. 5. Результати застосування запропонованої схеми обробки зображення для вилучення біометричних даних з райдужної оболонки ока на етапі вилучення даних

2.3. Експериментальні дослідження алгоритму генерації ключів

Для основних перетворень на обох етапах виконання реалізації було отримано наступні показники швидкодії, як показано у таблиці 1.

Таблиця 1

Оцінка швидкодії запропонованої програмної реалізації алгоритму генерації ключів з райдужної оболонки ока

Операція	Швидкодія, мс
Локалізація райдужної оболонки ока - оператор Кенні - перетворення Хафа	 43,64 826
Ізоляція райдужної оболонки ока	2,9
Нормалізація райдужної оболонки ока	6,58
Формування даних	113,09
Коди Ріда-Соломона - Кодування - Декодування	 1,56 2,14
Хешування алгоритмом Купина	1,02

Для оцінки швидкодії було виконано обчислення часу виконання операції під час обробки кожного з 756 зображень з бази CASIA. Слід зазначити, що не дивлячись на високі показники швидкодії, ці результати можуть бути оптимізовані. Аналіз швидкодії проводилися на обчислювальній платформі з ОС Windows 10 x64, Intel Core i7, 4.7 ГГц. Оцінимо запропонований алгоритм за такими параметрами, як ймовірність помилкової ідентифікації (FAR) та ймовірність того, що система не визнає справжність біометричних даних зареєстрованого в ній користувача (FRR).

В ході аналізу цих параметрів, слід зазначити, що отримані практичні результати доводять, що райдужна оболонка ока дійсно носить унікальний характер. Оскільки, раніше було зазначено, що послідовності, отримані на виході з фільтра Габора, досить схожі з випадковими послідовностями, подальші оцінки наведені для даних, подібних до вихідних послідовностей Габора, проте створених генератором випадкових чисел.

Отже, отримано, що ймовірність помилкової ідентифікації для запропонованого алгоритму досить низка FAR = 0,14%. Більш цікавим з точки зору дослідження є показник ймовірності відхилення справжніх біометричних даних користувача, оскільки в запропонованому алгоритмі застосовується метод нечітких екстракторів. Таким чином, встановлено, що рівень FRR для запропонованого алгоритму дорівнює 19.5%. Це досить високий показник ефективності алгоритму, проте актуальним питанням є підвищення рівня надійності роботи реалізації.

На рис. 6 наведено залежність рівня FRR від виправляючої можливості коду для запропонованої реалізації.

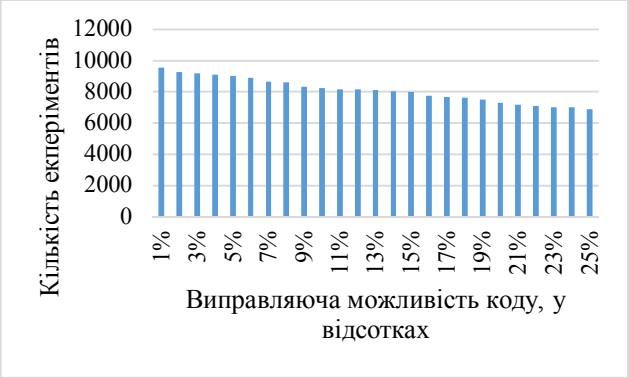

Рис. 6. Залежність ймовірності вдалого декодування від виправляючої можливості кодів Ріда-Соломона

Для виконання даного аналізу формувалася випадкова двійкова послідовність, потім вона підлягала кодуванню кодами Ріда-Соломона, коректуючий вектор запам'ятовувався, а потім вхідна послідовність зазнавала змін відповідно до можливостей виправляючого коду (розглянуто коди, які можуть виправляти від 1% до 25% помилок). Потім відбувалося декодування послідовності. Така процедура повторювалася 10000 разів. На гістограмі наведено кількість успішних декодувань пошкоджених послідовностей. За отриманими результатами можна зробити висновок, що пошук оптимального алгоритму завадостійкого кодування є також перспективним напрямом подальших досліджень.

ВИСНОВКИ

У даній роботі запропоновано алгоритм вилучення біометричних даних з райдужної оболонки ока на основі схеми нечітких екстракторів. Запропонова-

ний алгоритм складається з двох етапів: вилучення даних та генерації криптографічних ключів. На етапі вилучення даних застосовуються такі перетворення: оператор Кенні, перетворення Хафа, фільтр Гауса, модель розгортки Дагмана, двовимірний фільтр Габора. На етапі генерації використовуються алгоритми класичної криптографії: алгоритм кодування Ріда-Соломона, який за свою досить довгу історію існування добре себе зарекомендував, та український національний стандарт хешування «Купина», який було прийнято у 2014 році після детальних досліджень, з високими криптографічними показниками.

Також у роботі було проведено оцінку показників ефективності розробленого алгоритму. Виконання повної процедури вилучення даних та генерації ключа займає менше 1 секунди. Показники ймовірності помилкової ідентифікації FAR та ймовірність помилкового відхилення даних FRR дорівнюють, відповідно, 0,14% та 19.5%.

Література

[1] *Горбенко І.Д.* Прикладна криптологія. Теорія. Практика. Застосування: монографія. / І.Д. Горбенко, Ю.І. Горбенко. – Харків: «Форт», 2012. – 870 с.

[2] *Есин В.І.* Безпека інформаційних систем і технологій./ Есин В.І., Кузнецов О.О., Сорока Л.С.. – Харків: ХНУ ім. В.Н. Каразіна, 2013. – 632 с.

[3] *U. Uludag, S. Pankanti, S. Prabhakar, A. Jain.* Biometric Cryptosystems: Issues and Challenges. Proceedings of the IEEE. – June 2004. – Vol. 92, NO. 6.

[4] *Anil K. Jain, Arun Ross.* An Introduction to Biometric Recognition. IEEE Transactions on circuits and systems for video technology, 2004, Vol. 14, NO. 1, pp. 4–20.

[5] *J. Daugman* How iris recognition works / J. Daugman . – Circuits and Systems for Video Technology, IEEE Transactions, 2004, Vol 14, NO 1, pp. 21–30

[6] *F. Hao, R. Anderson, J. Daugman.* Combining crypto with biometrics effectively. IEEE Transactions on Computers. – 2006. – Vol. 55. – pp. 1081-1088.

[7] *Richard P. Wildes*. Iris recognition: An emerging biometric technology. Proceedings of the IEEE, 1997, Vol. 85, pp. 1348–1363.

[8] *W. W. Boles, B. Boashash.* A human identification technique using images of the iris and wavelet transform, IEEE Trans. Signal Process. – 1998. – Vol. 46, NO. 4. – pp. 1185–1188

[9] *S. L. Lim, K. L. Lee, O. B. Byeon, T. K. Kim.* Efficient Iris Recognition through Improvement of Feature Vector and Classifier. ETRI J. – 2001. – Vol. 23, NO. 2, pp.61–70 .

[10] *K. Bae, S. Noh, J. Kim.* Iris Feature Extraction using Independent Component Analysi. 4th International Conference on Audio-and Video-based Biometric Person Authentication, Guildford, UK. – 2003. – pp. 838–844.

[11] *C. Tisse, L. Martin, L. Torres, M. Robert.* Person identification technique using human iris recognition. Proc. Vis Interface. – 2002. – pp.294–299.

[12] *L. Ma, T. Tan, Y. Wang, D. Zhang*. Efficient iris recognition by characterizing key local variations. IEEE. Image Process. – 2004. – Vol. 13. – pp. 739–750.

[13] *C. Rathgeb, A. Uhl, P. Wild.* Iris-biometrics: from segmentation to template security. Advances in Information Security, Springer. – 2013.

[14] *M.R. Ogiela, L. Ogiela.* Image based crypto-biometric key generation. 2011 Third International Conference on Intelligent Networking and Collaborative Systems, Fukuoka, Japan. – 2011. – pp. 673–678.

[15] *L. Wu, X. Liu, S. Yuan, P. Xiao.* A novel key generation cryptosystem based on face features. In Signal Processing (ICSP), 2010 IEEE 10th International Conference, 2010, pp. 1675–1678.

[16] *Sunil Chawla, Aashish Oberoi.* A Robust Algorithm for Iris Segmentation and Normalization using Hough Transform. Global Journal of Business Management and Information Technology. – 2011. – Vol. 1, No. 2. – pp.69–76

[17] Національний Стандарт України ДСТУ 7564:2014. Інформаційні технології. Криптографічний захист інформації. Функція гешування. – К: 2014. – 41 с.

Надійшла до редколегії 20.12.2018

Луценко Марія Сергіївна, науковий співробітник ПАТ «ІІТ», здобувач кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – біометрична криптографія, блокові симетричні шифри.

Кузнецов Олександр Олександрович, доктор технічних наук, професор, заступник головного конструктора ПАТ «ІІТ», професор кафедри безпеки інформаційних систем і технологій Харківського національного університету ім. В.Н. Каразіна. Галузь наукових інтересів – криптографія та автентифікація, алгебраїчна теорія кодів, обробка, передача та захист інформації.

Горбенко Юрій Іванович, кандидат технічних наук, виконавчий директор ПАТ «ІІТ», старший науковий співробітник кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – криптографія та автентифікація, інфраструктура відкритих ключів.

Пушкарьов Андрій Іванович, директор департаменту захисту інформації Адміністрації державної служ-би спеціального зв'язку та захисту інформації України. Галузь наукових інтересів – теорія захисту інформації, інформаційна та кібербезпека держави.

Уварова Анна Олександрівна, провідний інженер Конструкторського бюро «Південне» ім. М. К. Янгеля», здобувач кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – біометрична криптографія, блокові симетричні шифри.

УДК 004.056.55

Луценко М. С. **Генерация ключей из биометрических образов радужной оболочки глаза** / М. С. Луценко, А. А. Кузнецов, Ю. И. Горбенко, А. И. Пушкарев, А. А. Уварова // Прикладная радиоэлектроника: науч. – техн. журнал. – 2018. – Том 17, №. 3, 4. – С. 104–114.

Рассматриваются наиболее распространенные подходы для создания биометрических криптосистем, в частности, систем с генерацией ключа. Разрабатывается новая схема формирования ключа методом нечетких экстракторов из биометрических данных радужной оболочки глаза. Предложена программная реализация и проведены экспериментальные исследования алгоритма генерации ключей на основе биометрических данных, полученных разработанным методом нечетких экстракторов из радужной оболочки глаза.

Ключевые слова: биометрия, биометрические криптосистемы, генерация криптографических ключей, радужная оболочка глаза.

Табл. 1. Ил. 6. Библиогр.: 17 наим.

UDC 004.056.55

Lutsenko M. **Generation of keys using biometric images of the iris of the eye** / M. Lutsenko, A. Kuznetsov, Yu. Gorbenko, A. I. Pushkarev, A. Uvarova // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 104–114.

The most common approaches for creating biometric cryptosystems, in particular, systems with key generation, are considered. A new key generation scheme is being developed using fuzzy extractors from the biometric data of the iris. A software implementation is proposed and experimental studies of the key generation algorithm based on biometric data obtained by the developed method of fuzzy iris extractors are carried out.

Keywords: biometrics, biometric cryptosystems, cryptographic key generation, iris.

Tab. 1. Fig. 6. Ref.: 17 items.

УДК 004.056.55

ДОКАЗУЕМО СТОЙКИЙ ГЕНЕРАТОР ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ПОСТКВАНТОВОГО ПРИМЕНЕНИЯ

А. А. КУЗНЕЦОВ, А. С. КИЯН, Д. И. ПРОКОПОВИЧ-ТКАЧЕНКО, В. П. ЗВЕРЕВ, Е. В. КОТУХ, Т. Ю. КУЗНЕЦОВА

В данной работе рассматривается доказуемо стойкий генератор псевдослучайных последовательностей, задача криптоанализа которого сводится к решению хорошо известной и чрезвычайно сложной математической проблеме синдромного декодирования (относящейся к классу NP-сложных). Установлено, что формируемые псевдослучайные последовательности не обладают максимальным периодом, фактический период значительно ниже ожидаемого. Предлагается новая схема генератора, которая сохраняет все позитивные свойства прототипа, однако формируемые последовательности обладают максимальным периодом.

Ключевые слова: модель доказуемой безопасности, генератор псевдослучайных чисел, кодовые криптосистемы, постквантовая криптография.

ВВЕДЕНИЕ

Важным направлением в развитии постквантовых методов защиты информации является криптография, основанная на кодах, исправляющих ошибки (Code-based Cryptography) [1, 2]. В работах [3–9] показано, что использование кодовых криптосистем позволяет обеспечить высокую стойкость как к классичес-кому, так и к квантовому криптоанализу.

Первая кодовая криптосистема была предложена 40 лет назад [3] и, при соответствующих параметрах, остается стойкой по сегодняшний день [4–9]. Несмотря на многочисленные попытки криптоанализа [5–9] схема McElice на основе кодов Гоппы [10] считается надежной альтернативой современным криптосистемам с открытым ключом.

Дальнейшее развитие кодовой криптографии получило в работах [11–20]. В частности, в [11] предложена эквивалентная по стойкости криптосистема Niederreiter, которая положена в основу схем электронной цифровой подписи [14, 15]. В [16] предложен новый вариант подписи, использующий криптосистему McElice.

На сегодняшний день National Institute of Standards and Technology (NIST) США проводит открытый конкурс постквантовой криптографии [1, 2, 21–23], где анализируется 64 конкурсных предложения (из 82 предварительно поданных) по трем основным направлениям: шифрование с открытым ключом (public-key encryption), механизмы инкапсуляции ключей (key encapsulation mechanism - KEM), и электронные цифровые подписи (digital signature) [22]. Из общего числа конкурсных предложений третью часть занимает кодовая криптография [23]. Ожидается [1, 23], что в ближайшие десятилетия проект NIST PQC завершиться приятием серии стандартов постквантовой криптографии с открытым ключом.

Еще одним направлением в развитии кодовой криптографии является построение доказуемо стойких генераторов псевдослучайных последовательностей [25–27]. Суть модели доказуемой безопасности (Provable Security Model) состоит в сведении задачи криптоанализа к решению хорошо известной и чрезвычайно сложной математической задачи (относящейся к классу NP-сложных), например, факторизации, дискретного логарифмирования, и пр. [28]. Криптографические примитивы, соответствующие такой модели безопасности, принято называть доказуемо безопасными, т.к. их криптоанализ сопоставим с решением NP-сложной математической задачи. В контексте развития постквантовой криптографии построение и анализ доказуемо стойких генераторов несомненно является важным и актуальным.

Целью данной работы является анализ доказуемо стойкого генератора псевдослучайных последовательностей, задача криптоанализа которого сводится к решению проблемы синдромного декодирования (относящейся к классу NP-сложных) [25], исследование периодических свойств формируемых последовательностей. В работе показано, что формируемые последовательности не обладают максимальным периодом, фактический период значительно ниже ожидаемого. Предлагается новая схема генератора, которая сохраняет все позитивные свойства прототипа, однако формируемые псевдослучайные последовательности обладают максимальным периодом.

1. ДОКАЗУЕМО СТОЙКИЙ ГЕНЕРАТОР, ОСНОВАННЫЙ НА СИНДРОМНОМ ДЕКОДИРОВАНИИ

Доказуемо безопасный генератор, основанный на синдромном декодировании (Pseudo-Random Generator Provably as Secure as Syndrome Decoding), был впервые предложен в работе [25], его исследование и дальнейшее развитие получило в работах [26,27].

Построение генератора основано на использовании блокового (n,k,d) кода, который задан своей проверочной матрицей H размером n столбцов и $n-k$ строк. В теории кодирования известна NP-полная проблема синдромного декодирования [29, 30]:

– по известному вектору-синдрому s длины $n-k$ и известной матрице H найти такой вектор ошибки e длины n, что $s = e \cdot H^T$, причем вес Хемминга (число ненулевых элементов) вектора e равен $w(e) = t = \left\lceil \frac{d-1}{2} \right\rceil$, где $\lceil x \rceil$ – наименьшее целое число, не меньшее x.

Величина t определяет исправляющую способность (n,k,d) кода, т.е. гарантированное число ошибок, которое возможно исправить, применив метод максимального правдоподобия. Для некоторых кодов (со специальной структурой матрицы H) известны быстрые алгоритмы алгебраического декодирования, т.е. нахождение вектора e полиномиально разрешимая задача. Однако для кодов общего положения (без специальной структуры матрицы H) нахождение вектора e является чрезвычайно сложным, наилучшие алгоритмы основаны на переборном поиске.

Для формирования псевдослучайной последовательности используется двоичный (n,k,d) код и следующее рекуррентное правило: $s_i = e_i \cdot H^T$, где: e_i – двоичный вектор длины n, $w(e_i) = t = \left\lceil \frac{d-1}{2} \right\rceil$; s_i – двоичный вектор длины $n-k$; H – двоичная проверочная матрица (n,k,d) кода.

Начальное состояние генератора e_0 задается посредством равновесного кодирования инициализирующей последовательности (*Seed*) y_0 длины

$$m = \left\lfloor \log_2 \left(\frac{n!}{t!(n-t)!} \right) \right\rfloor \text{ бит},$$

где $\lfloor x \rfloor$ – наибольшее целое число, не превосходящее x.

Равновесное кодирование преобразует двоичный вектор y_0 длины m в двоичный вектор e_0 длины n, причем $w(e_0) = t$. Очередное состояние генератора e_{i+1} также формируется посредством равновесного кодирования. Для этого двоичный вектор s_i разбивается на две части:

$$s_i = y_{i+1} \| z_{i+1},$$

(здесь $\|$ – символ конкатенации), причем длина двоичного вектора y_{i+1} равна m. Оставшиеся $n-k-m$ бит образуют вектор z_{i+1}, который подается на выход генератора как элемент псевдослучайной последовательности. Равновесное кодирование вектора y_{i+1} позволяет сформировать состояние e_{i+1} и вычисления повторяются. Алгоритмы равновесного кодирования предлагаются во многих источниках, например, в [31].

Формирование псевдослучайных последовательностей осуществляется итерационной процедурой с использованием проверочной матрицы кода H для формирования вектора-синдрома s_i (см. рис. 1). На каждом шаге алгоритма формируются $n-k-m$ бит последовательности z_{i+1}, причем задача нахождения состояния генератора e_i по известному фрагменту последовательности z_{i+1} и/или вектору-синдрому s_i сопряжена с решением теоретико-сложностной задачи синдромного декодирования.

Для проведения экспериментальных исследований периодических свойств псевдослучайных последовательностей разработана программная реализация генератора. Для небольших параметров выполнен полный перебор всех возможных векторов инициализации (*Seed*). Для каждой инициализации сформирована псевдослучайная последовательность, оценен ее период. В результате мы имеем полный набор всех длин периодов, которые могут быть порождены каждым генератором для соответствующих входных параметров.

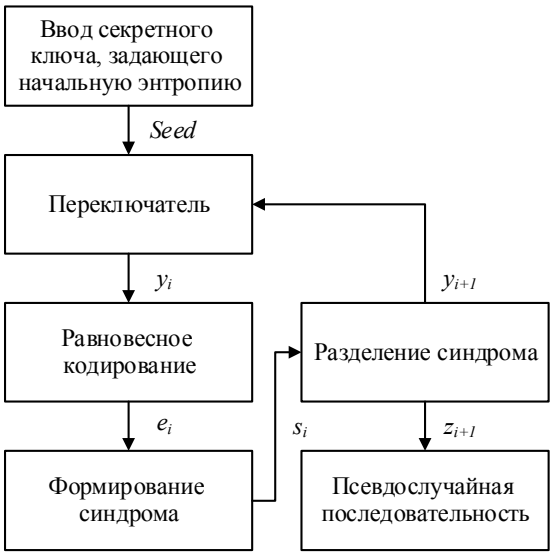

Рис. 1. Структурная схема генератора из [25]

На рисунке 2 приведены распределения числа ключей по длинам периодов в случае использования двоичного (31, 16, 7) кода. В качестве инициализирующей последовательности y_0 выбирались все двоичные вектора длины $m = 12$ бит. На рисунке 3 приведены соответствующие распределения для двоичного (31, 11,

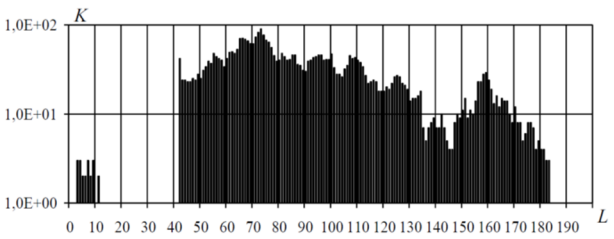

Рис. 2. Распределение количества ключей по длинам периодов формируемых последовательностей, $L_{\max} = 4095$

5) кода с $m = 17$ бит. Максимальная (ожидаемая) длина периода формируемых последовательностей составляет $L_{\max} = 2^m - 1$ бит.

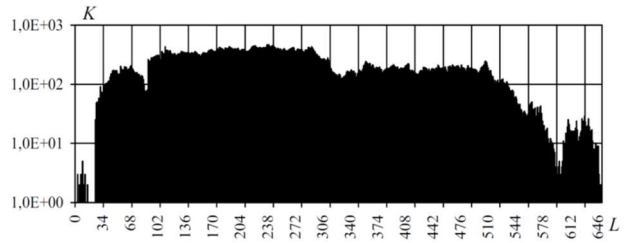

Рис. 3. Распределение количества ключей по длинам периодов формируемых последовательностей, $L_{\max} = 131073$

Полученные результаты показывают, что рассмотренный генератор формирует последовательности, период которых существенно ниже максимального. С увеличением длины инициализирующего вектора расхождения между ожидаемым и фактическим периодом увеличиваются. Например, для последнего случая фактический период меньше максимального более чем в 200 раз.

Выявленный недостаток предлагается устранить добавлением в схему генератора рекуррентных преобразований, гарантирующих максимальный период $L_{\max} = 2^m - 1$.

3. ДОКАЗУЕМО СТОЙКИЙ ГЕНЕРАТОР МАКСИМАЛЬНОГО ПЕРИОДА

В основе предлагаемой схемы генератора, как и в методе-прототипе, лежит использование проблемы синдромного декодирования. Однако правило формирования псевдослучайных последовательностей изменено. Для обеспечения максимального периода предлагается дополнительно использовать рекуррентные преобразования, например, регистры сдвига с линейной обратной связью (РСЛОС, англ. linear feedback shift register, LFSR). При размере регистра m бит и использовании обратных связей, заданных коэффициентами примитивного полинома, будет гарантирован максимальный период $L_{\max} = 2^m - 1$ выходной последовательности [29, 30].

Структурная схема предлагаемого генератора представлена на рис. 4. Цветом выделены дополнительно внесенные блоки преобразований.

Начальное состояние генератора инициализируется последовательностью $y_0 = Seed$, которая после равновесного кодирования преобразуется в вектор e_0. Последовательность $Seed$ задает также начальное состояние u_0 рекуррентного преобразования (например, РСЛОС), обозначим его $\varphi(u)$.

На каждой итерации вычисляется состояние

$$u_{i+1} = \varphi(u_i),$$

которое поступает на сумматор (см. рис. 4).

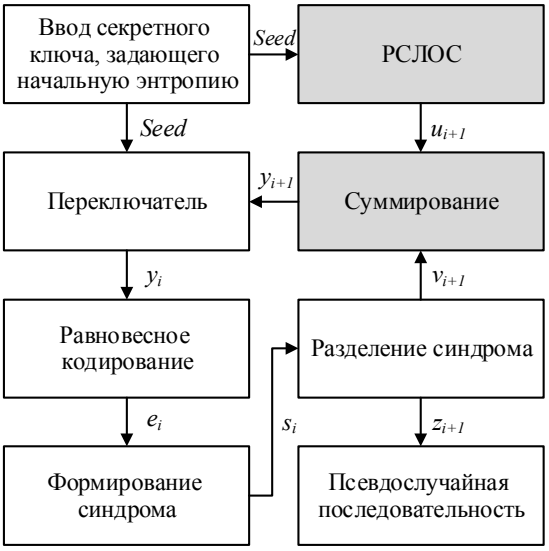

Рис. 4. Структурная схема предлагаемого генератора

Остальная часть генератора функционирует также, как и в методе-прототипе. С использованием двоичного (n,k,d) кода по правилу

$$s_i = e_i \cdot H^T$$

формируется вектор-синдром s_i, который разбивается на две части:

$$s_i = v_{i+1} \| z_{i+1},$$

причем длина двоичного вектора v_{i+1} равна m.

Оставшиеся $n-k-m$ бит образуют вектор z_{i+1}, который подается на выход генератора как элемент псевдослучайной последовательности.

Вектор v_{i+1} складывается с вектором u_{i+1} для формирования очередного значения y_{i+1}:

$$y_{i+1} = u_{i+1} + v_{i+1}$$

и вычисления повторяются.

Таким образом, за счет добавления рекуррентного преобразования (например, РСЛОС) удается обеспечить максимальный период формируемых последовательностей, причем задача нахождения состояния генератора e_i по известному фрагменту псевдослучайной последовательности z_{i+1} и/или вектору-синдрому s_i, как и в методе-прототипе, сопряжена с решением теоретико-сложностной проблемы синдромного декодирования.

Для подтверждения заявленных характеристик разработана программная реализация предложенного генератора. Для выбранных в разделе 2 параметров выполнен полный перебор всех возможных векторов инициализации (*Seed*). Для каждой инициализации сформирована псевдослучайная последовательность,

оценен ее период. Полученные результаты показывают, что все вводимые вектора инициализации приводят к формированию последовательностей максимального периода:

– при использовании двоичного (31, 16, 7) кода с инициализирующей последовательностью y_0 длины $m=12$ бит период всех формируемых последовательностей равен $L_{\max}=2^{12}-1=4095$;

– при использовании двоичного (31, 11, 5) кода с инициализирующей последовательностью y_0 длины $m=17$ бит период всех формируемых последовательностей равен $L_{\max}=2^{17}-1=131073$.

Предлагаемое улучшение генератора сопряжено с повышением вычислительной сложности. Фактически, на каждой итерации при формировании блока псевдослучайной последовательности необходимо дополнительно вычислить очередное состояние рекуррентного преобразования. В тоже время, в случае использования РСЛОС вычислительная сложность повысится не значительно – на один такт регистра сдвига с обратными связями.

ВЫВОДЫ

В данной работе исследованы доказуемо безопасные генераторы, криптоанализ которых основан на решении проблемы синдромного декодирования (относящейся к классу NP-сложных). Ожидается, что этот класс криптопримитивов будет надежным и безопасным даже в условиях применения квантовых методов криптографического анализа.

Рассмотренный генератор, предложенный в работе [25], был реализован программно, для небольших параметров кодов исследованы периодические свойства формируемых псевдослучайных последовательностей. Установлено, что для всех вводимых векторов инициализации генератор формирует последовательности с очень малыми длинами периодов, которые меньше ожидаемого (максимального) периода на несколько порядков.

Для устранения выявленных недостатков предложено усовершенствовать генератор посредством дополнительного выполнения рекуррентных преобразований, гарантирующих максимальный период формируемых последовательностей (например, РСЛОС). Экспериментальные исследования подтвердили заявленные характеристики. Кроме того, задача нахождения состояния генератора по известному фрагменту последовательности, как и в методе-прототипе, сопряжена с решением теоретико-сложностной задачи синдромного декодирования. Следовательно, предлагаемый генератор, как и генератор из [25], будет устойчив к атакам квантового криптоанализа.

Вычислительная сложность реализации предлагаемого генератора незначительно превосходит прототип. На каждой итерации (для формирования каждого блока выходной последовательности) необходимо дополнительно вычислять очередное состояние рекуррентного преобразования. В случае использования РСЛОС вычислительная сложность повысится не значительно (на один такт регистра).

Литература

[1] *D. Moody.* “Post-Quntum Cryptography: NIST’s Plan for the Future.” The Seventh International Conference on Post-Quntum Cryptography, Japan, 2016. [On-line]. Internet: https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf [March 8, 2016].

[2] *Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta*, Ray Perlner and Daniel Smith-Tone. “NISTIR 8105. Report on Post-Quantum Cryptography”, National Institute of Standards and Technology, Internal Report 8105, April 2016, 10 p.

[3] *R.J. McEliece* “A public-key cryptosystem based on algebraic coding theory”. DSN Progress Report 42-44, Jet Propulsion Lab., Pasadena, CA, January-February, 1978, pp. 114-116.

[4] *D. Bernstein, J. Buchmann and E.Dahmen*. Post-Quantum Cryptography. Springer-Verlag, Berlin-Heidleberg, 2009, 245 p.

[5] *Сидельников В.М.* Криптография и теория кодирования. Материалы конференции «Московский университет и развитие криптографии в России», МГУ. – 2002. – 22 с.

[6] *Anne Canteaut and Nicolas Sendrier*. “Cryptanalysis of the original McEliece cryptosystem”. In Kazuo Ohta and Dingyi Pei, editors, Advances in cryptology - ASIACRYPT’98, volume 1514 of Lecture Notes in Computer Science, pp. 187–199.

[7] *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.

[8] *L.Minder and A. Shokrollahi.* “Cryptanalysis of the Sidelnikov Cryptosystem”, Advances in Cryptology - EUROCRYPT 2007: 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings, Springer Berlin Heidelberg, 2007, pp. 347–360.

[9] *D.J.Bernstein, T.Lange and C.Peters*. “Attacking and Defending the McEliece Cryptosystem”. In: Buchmann J., Ding J. (eds) Post-Quantum Cryptography. PQCrypto 2008. Lecture Notes in Computer Science, vol 5299. Springer, Berlin, Heidelberg, pp. 31–46.

[10] *Гонна В. Д.* Новый класс линейных корректирующих кодов // Проблемы передачи информации. – 1970. – Т. 6, вып.3. – С. 24–30.

[11] *Niederreiter H.* “Knapsack-type cryptosystems and algebraic coding theory”. Problem Control and Inform Theory, 1986, v. 15. pp. 19–34.

[12] *A. Kuznetsov, A. Kiian, M. Lutsenko, I. Chepurko and S. Kavun,* "Code-based cryptosystems from NIST PQC," 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2018, pp. 282–287.

[13] *T. R. N. Rao and K. H. Nam*. “Private-key algebraic-coded cryptosystem”. Advances in Cryptology - CRYPTO 86, New York, NY: Springer, pp. 35–48.

[14] *Courtois, N., Finiasz, M., and N.Sendrier*. “How to achieve a McEliece-based digital signature scheme”. In Advances in

Cryptology - ASIACRYPT 2001, volume 2248, pp. 157–174.

[15] *M.Finiasz.* “Parallel-CFS: Strengthening the CFS McEliece-based signature scheme”. In Biryukov, A., Gong, G., Stinson, D., eds.: Selected Areas in Cryptography. Volume 6544 of LNCS., Springer (2010), pp. 159–170.

[16] *Alexandr Kuznetsov, Andriy Pushkar’ov, Nastya Kiyan and Tetiana Kuznetsova.* “Code-Based Electronic Digital Signature”, The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT’2018, 24-27 May, 2018, Kyiv, Ukraine.

[17] *A. Kuznetsov, I. Svatovskij, N. Kiyan and A. Pushkar'ov*, "Code-based public-key cryptosystems for the post-quantum period," 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, 2017, pp. 125–130.

[18] *Yu. V. Stasev, A. A. Kuznetsov.* “Asymmetric Code-Theoretical Schemes Constructed with the Use of Algebraic Geometric Codes”. Cybernetics and Systems Analysis, Volume 41, Issue 3, pp. 354–363, May 2005.

[19] *A. Kuznetsov, R. Serhiienko and D. Prokopovych-Tkachenko,* "Construction of cascade codes in the frequency domain," 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, 2017, pp. 131–136.

[20] *Yu.V. Stasev, A.A. Kuznetsov*. “Asymmetric code-theoretical schemes constructed with the use of algebraic geometric codes”. Kibernetika i Sistemnyi Analiz, No. 3, pp. 47–57, May-June 2005.

[21] “Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process”, National Institute of Standards and Technology, 25 p. [On-line].Internet: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[22] *Ray A. Perlner and David A. Cooper.* “Quantum Resistant Public Key Cryptography: A Survey”, IDtrust ’09, April 14-16, 2009, Gaithersburg, MD, pp. 85-93. [On-line]. Internet: https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=901595

[23] *Dustin Moody.* “Let’s Get Ready to RumbleThe NIST PQC “Competition”, National Institute of Standards and Technology, April 18, 2018, 37 p. [On-line]. Internet: https://csrc.nist.gov/CSRC/media/Presentations/Let-s-Get-Ready-to-Rumble-The-NIST-PQC-Competiti/images-media/PQCrypto-April2018_Moody.pdf

[24] “Computer Security Resource Center”. Round 1 Submissions. Created January 03, 2017, Updated June 25, 2018. [On-line] Internet: https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions

[25] *Jean-Dernard Fisher, Jacques Stern*. “An efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding”. EUROCRYPT’96 Proceeding, LNCS 1070, p. 245–255.

[26] *P. Gaborit, C. Lauradoux and N. Sendrier*, "SYND: a Fast Code-Based Stream Cipher with a Security Reduction," 2007 IEEE International Symposium on Information Theory, Nice, 2007, pp. 186–190.

[27] *T. Wu and R. Wang*, "Stream cipher by reed-solomon code," 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2017, pp. 422–427.

[28] “Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption”, April 19, 2004 - Version 0.15 (beta), Springer-Verlag, 829 p.

[29] *F. J. MacWilliams and N. J. A. Sloane*. The theory of error-correcting codes. North-Holland, Amsterdam, New York, Oxford, 1977, 762 p.

[30] *R.E. Blahut.* Theory and Practice of Error Control Codes. Addison Wesley Publishing Company, Inc., Reading, Massachusetts, 1983, 1983, 500 p.

[31] Методи та алгоритми адаптивного рівноважного кодування на основі біноміальних чисел для інформаційних систем: автореф. дис. / О.В. Бережная; Харк. нац. ун-т радіоелектрон. – Х., 2002. – С. 19.

Поступила в редколлегию 25.12.2018

Кузнецов Александр Александрович, доктор технических наук, профессор, заместитель главного конструктора АТ «ИИТ», профессор кафедры Харьковского национального университета им. В.Н. Каразина. Область научных интересов – криптография и аутентификация, обработка, передача и защита информации.

Киян Анастасия Сергеевна, магистрант кафедры безопасности информационных систем и технологий Харьковского национального университета им. В.Н. Каразина. Область научных интересов – криптография и аутентификация, алгебраическая теория кодов и кодовые криптосистемы.

Прокопович-Ткаченко Дмитрий Игоревич, кандидат технических наук, заведующий кафедрой кибербезопасности Университета таможенного дела и финансов. Заместитель Председателя Государственной службы специальной связи и защиты информации Украины (2013-2014 г.). Область научных интересов – аутентификация и безопасность беспроводных сетей.

Зверев Владимир Павлович, кандидат технических наук, старший научный сотрудник, Помощник Председателя Национальной полиции Украины, Председатель Государственной службы специальной связи и защиты информации Украины (2014-2015 г.). Область научных интересов – информационная и кибернетическая безопасность государства.

Котух Евгений Владимирович, кандидат технических наук, доцент, доцент кафедры кибербезопасности Университета таможенного дела и финансов. Область научных интересов – информационная и кибербезопасность государства, кодовая криптография и аутентификация.

Кузнецова Татьяна Юрьевна, научный сотрудник кафедры безопасности информационных систем и технологий Харьковского национального университета имени В.Н. Каразина. Область научных интересов – криптография и аутентификация, блочные симметричные шифры.

УДК 004.056.55

Кузнецов О. О. **Доказово стійкий генератор псевдовипадкових послідовностей для постквантового застосування** / О. О. Кузнецов, А. С. Кіян, Д. І. Прокопович-Ткаченко, В. П. Звєрєв, Е. В. Котух, Т. Ю. Кузнецова // Прикладна радіоелектроніка: наук.–техн. журнал. – 2018. –Том 17. № 3, 4. – С. 115–120.

У даній роботі розглядається доказово стійкий генератор псевдовипадкових послідовностей, завдання криптоаналізу якого зводиться до вирішення добре відомої і надзвичайно складної математичної проблеми синдромного декодування (що належить до класу NP-складних). Встановлено, що формовані псевдовипадкові послідовності не володіють максимальним періодом, фактичний період значно нижчий за очікуваний. Пропонується нова схема генератора, яка зберігає всі позитивні властивості прототипу, проте формовані послідовності мають максимальний період.

Ключові слова: модель доказової безпеки, генератор псевдовипадкових чисел, кодові криптосистеми, постквантова криптографія.

Іл.: 4. Бібліогр.:31 наім.

UDC 004.056.55

Kuznetsov A. A. **Provably strong pseudorandom sequence generator for post-quantum applications** / A. A. Kuznetsov, A. S. Kiian, D. I. Prokopovich-Tkachenko, V. P. Zverev, E. V. Kotukh, T. Yu. Kuznetsova // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 115–120.

This paper considers a provably secure pseudorandom sequence generator whose task of cryptanalysis is reduced to solving a well-known and extremely complex mathematical problem of syndromic decoding (which belongs to a NP-complex class). It is found that formed pseudorandom sequences do not have the maximum period, the actual period is much lower than an expected one. A new generator scheme is proposed which retains all positive properties of the prototype, but formed sequences have a maximum period.

Keywords: proof-security model, pseudorandom number generator, code-based cryptosystems, post-quantum cryptography.

Fig. 4. Ref.: 31 items.

УДК 621.3.06

ДИФФЕРЕНЦИАЛЬНЫЙ КРИПТОАНАЛИЗ БЛОЧНОГО ARX-ШИФРА «КИПАРИС-256»

Р. Ю. ЕЛИСЕЕВ, М. Ю. РОДИНКО, Р. В. ОЛЕЙНИКОВ

В статье представлены результаты дифференциального криптоанализа симметричного блочного шифра «Кипарис-256», выполненного с применением ряда методов, в частности, с помощью алгоритма Мацуи и использованием частичных таблиц распределения разностей. В ходе исследований был найден ряд дифференциальных характеристик вплоть до пяти циклов шифрования. Кроме того, был обнаружен ряд характеристик с вероятностями от 2^{-2} до 2^{-5}, входы и выходы которых имеют малый вес Хэмминга.

Ключевые слова: малоресурсная криптография, блочный симметричный шифр, дифференциальный криптоанализ, дифференциальная характеристика.

ВВЕДЕНИЕ

Метод построения симметричных криптографических преобразований на основе ARX (Addition-Rotation-XOR) конструкций [1] привлекает все большее внимание разработчиков. С одной стороны, метод дает возможность создавать очень простые в описании и реализации преобразования. С другой стороны, возникают проблемы при попытках криптоанализа ARX-преобразования классическими методами.

На сегодняшний день ARX-шифры активно исследуются как с точки зрения поиска универсальных алгоритмов и подходов к криптоанализу [2], так и построения примитивов с заданными криптографическими свойствами [1]. Тем не менее, даже при исследовании произвольного алгоритма на основе ARX преобразований все еще сложно однозначно говорить о доказуемой криптостойкости: почти под каждый алгоритм необходимо разрабатывать свою доказательную базу.

Последнее связано с тем, что большая часть существующей доказательной базы для блочных шифров создана для алгоритмов, чья цикловая функция основана на чередовании линейных и нелинейных преобразований с известными математическими свойствами [3]. Такие шифры хорошо зарекомендовали себя с точки зрения криптостойкости и эффективности реализации на широком спектре платформ и элементных баз. Известными примерами подобных шифров являются DES [4], [5], ГОСТ 28147-89 [6], Camellia [7], AES [8], Калина [9]. Все они имеют доказательную стойкость, однако достаточно сложны в оптимизированной программной реализации и сильно зависят от качества (и главное наличия) кэширования данных на уровне процессора для хранения предварительно рассчитанных Т-таблиц [10].

В связи с вышесказанным, малоресурсные алгоритмы, в частности, основанные на ARX, привлекают все больше внимания разработчиков. В Украине разработан малоресурсный блочный шифр «Кипарис», обеспечивающий компактную реализацию и высокую скорость преобразований на различных платформах [11].

Целью статьи является поиск дифференциальных характеристик в перспективном блочном ARX-шифре «Кипарис-256» [11] и изучение принципов распространения дифференциальных разностей через циклы шифрования. Несмотря на малоресурсный дизайн (с точки зрения программной реализации на широком спектре платформ общего назначения), алгоритм использует нелинейные преобразования с достаточно большими размерами входов-выходов (сложение по модулю 2^{32} в 256-битной версии), что заметно осложняет его исследование классическими методами и алгоритмами дифференциального криптоанализа, ориентированными на SPN-архитектуру цикловой функции с небольшими табличными нелинейными преобразованиями.

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Дифференциальный криптоанализ

Дифференциальный криптоанализ [4] – статистическая атака на симметричные криптопреобразования, изучающая изменения разности между двумя парами тестов по мере их прохождения через компоненты преобразования.

При анализе шифра рассматриваются, в первую очередь, дифференциальные пути (характеристики), т.к. на сегодняшний день только характеристики могут быть эффективно вычислены для современных шифров.

Под дифференциальной характеристикой понимают набор разностей между двумя текстами в определенные моменты вычислений (на входе, выходе и между циклами, например), в то время как дифференциал состоит лишь из пары входной разности и выходной. В общем случае один дифференциал состоит из множества дифференциальных характеристик, сумма вероятностей которых и составляет его вероятность.

1.2. Описание блочного шифра «Кипарис»

Алгоритм шифрования «Кипарис» [11] выполняет преобразования блоков данных размером 256 и 512 бит с использованием ключа шифрования такой же

длины. Длина ключа совпадает с размером блока. Таким образом, алгоритм поддерживает два варианта шифрования: «Кипарис-256» и «Кипарис-512».

К входным данным алгоритма принадлежат открытый текст и ключ шифрования, представленные в виде строк длиной $8 \times l$ бит. С исходными данными алгоритма принадлежит шифртекст, представленный в виде строки длиной $8 \times l$ бит.

Для шифра «Кипарис-256» $l = 32$, количество циклов шифрования $N_r = 10$ для шифра «Кипарис-512» $l = 64$, количество циклов шифрования $N_r = 14$.

«Кипарис-256» ориентирован на использование на 32-битных платформах, «Кипарис-512» – на применение на 64-битных платформах.

На вход процедуры шифрование подается блок открытого текста в виде одномерного массива из восьми l-битных слов (word) и цикловые ключи. После окончания процедуры шифрование полученный шифртекст представляется в виде последовательности l-битных слов.

В основе шифра «Кипарис» лежит сеть Фейстеля, один цикл которой изображен на рис. 1.

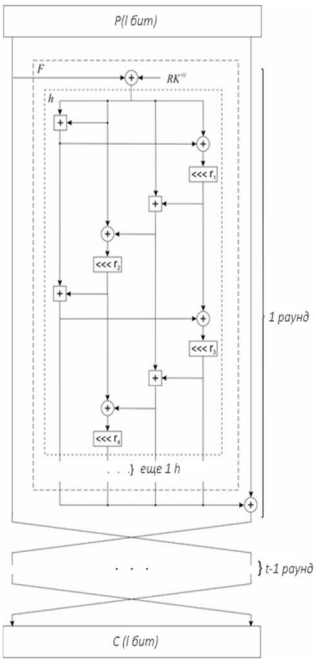

Рис. 1. Графическое представление шифра «Кипарис»

Блок открытого текста делится на два подблока длиной $4 \times l$ бит. Левый подблок поступает на вход циклового преобразования F.

Сначала подблок составляется по модулю 2 с цикловым ключом, а затем дважды обрабатывается функцией HalfRound (обозначена h на рис. 1).

На вход функции HalfRound подается четыре l-битных слова (l_0, l_1, l_2, l_3). Значения циклических сдвигов (rot0, rot1, rot2, rot3) зависят от длины блока и практически равны:

1) для шифра «Кипарис-256» $(\text{rot}_0, \text{rot}_1, \text{rot}_2, \text{rot}_3) = (16,12,8,7)$.

2) для шифра «Кипарис-512» $(\text{rot}_0, \text{rot}_1, \text{rot}_2, \text{rot}_3) = (32,24,16,15)$.

Для простоты последующего анализа в одном применении функции HalfRound можно выделить 4 применения более простой «элементарной» функции (рис. 2):

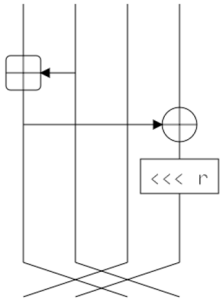

Рис. 2. Элементарный цикл «Кипарис-256»

Такое изображение позволяет упростить анализ цикловой функции благодаря тому, что каждый из таких повторяющихся циклов содержит в себе лишь одну нелинейную операцию.

1.3. Анализ модульного сложения

Вероятность преобразования разности по модулю 2 (xor difference probability) представляет собой вероятность того, что при входных разностях α, β на выходе сумматора окажется разность γ. Реализация на практике может быть построена на основе битовых преобразований, что позволяет существенно ускорить и упростить процесс расчетов [12].

$$XDP^{+}(\alpha,\beta\to\gamma)=\begin{cases}0 & if \quad eq(\alpha<<1,\beta<<1,\gamma<<1)\wedge \\ & \quad \wedge(\alpha\oplus\beta\oplus\gamma\oplus(\alpha<<1))\neq 0\\ 2^{-w_h\left(\neg eq(\alpha,\beta,\gamma)\wedge mask(n-1)\right)} & else,\end{cases}$$

где $eq(\alpha, \beta, \gamma)$ – функция побитового сравнения, возвращающая «1» в случае, если все 3 бита на соответствующих позициях равны и «0» в противном случае; w_h – вес Хемминга или количество ненулевых бит в слове, $mask(n)$ – функция, возвращающая слово из n единиц в младших позициях, остальные заполняются нулями.

1.4. Частичная таблица распределения разностей

Большинство современных стандартизированных и используемых на практике блочных шифров основаны на биективных блоках подстановки 8×8 бит или 4×4 бит. Такие размеры удобны не только в различных видах реализации, но и для дифференциального криптоанализа, так как позволяют построить полную таблицу распределения разностей – отображение пары входная-выходная разность в вероятность такого перехода по всем возможным наборам данных.

Общий размер полной таблицы составляет 2^{m+n} элементов, где m – количество входов S-блока, а n – количество выходов.

В случае же ARX преобразований, где обычно используются сумматоры достаточно большой разрядности (в случае «Кипарис-256» – 32-битовые), размеры полных таблиц становятся слишком большими для современных компьютеров, а время их построения – не практичным.

В связи с этим был предложен подход [13] на основе т.н. неполных таблиц распределения разностей (англ. partial difference distribution table, pDDT), которые включают в себя не все множество возможных входов-выходов, а только ту его часть, вероятность элементов которой превышает определенный заданный предел. Такие таблицы содержат гораздо меньшее количество элементов и как следствие легко могут быть использованы для «крупных» операций. Более того, в случае сложения (и некоторых других операций), по мере увеличения разрядности общая вероятность монотонно убывает, что приводит к возможности достаточно эффективно отсекать бесперспективные (с точки зрения вероятности преобразования) наборы разностей на ранних этапах.

1.5. Частичная таблица распределения разностей для сложения

Алгоритм построения неполной таблицы распределения разностей для сложения представлен на рис. 3. Основная идея состоит в рекурсивном поиске в глубину всех возможных комбинаций входных-выходных значений, которые собираются, бит за битом. На каждом шаге осуществляется проверка текущей вероятности и если она ниже заданного граничного значения – поиск в поддереве не производится, так как вероятность не может увеличиться.

1.6. Алгоритм Мацуи

В 1994 был предложен алгоритм [14], часто называемый в литературе алгоритмом Мацуи (англ. Matsui), предназначенный для эффективного поиска лучших дифференциальных путей или линейных аппроксимаций для DES (применим в неизменном виде к любой сети Фейстеля с двумя ветвями). Тем не менее, в общем случае алгоритм не гарантирует, что найденная характеристика или аппроксимация будет иметь наибольшую возможную вероятность.

```
Procedure compute_pddt (n, p_theshold, k, α_k, β_k, γ_k) :
  If n == k:
      Add (α_k, β_k, γ_k) to D
      Return
  For x,y,z ∈ {0, 1}:
      α_{k+1} ← x|α_k
      β_{k+1} ← y|β_k
      γ_{k+1} ← z|γ_k
      p_{k+1} = xdp+(α, β, γ)
      If p_{k+1} ≥ p_threshold :
          compute_pddt(n, p_theshold, k + 1, α_{k+1}, β_{k+1}, γ_{k+1})
```

Рис. 3. Алгоритм построения pDDT для сложения

Идея заключается в рекурсивном поиске на заданном количестве циклов n. Входными данными являются известные лучшие вероятности первых $n-1$ циклов, обозначаемые как $B_1, B_2, \ldots, B_{n-1}$, а также предположительная вероятность цикла n, обозначаемая как $\underline{B_n}$. В результате своей работы алгоритм возвращает B_n. Важным условием является выполнение $\underline{B_n} \le B_n$, так же $\underline{B_n}$ должен быть как можно ближе к B_n, чтобы ускорить выполнение алгоритма.

Псевдокоды этапов рекурсивного поиска отображены в алгоритмах 1–4. При окончании каждого из них управление передается предыдущему. После окончания всех итераций первого шага переменная $\underline{B_n}$ содержит лучшую найденную вероятность дифференциального пути или линейной аппроксимации (нотация ориентирована на поиск дифференциальных характеристик).

Алгоритм 1. Первый этап поиска.

- Выполнить для всех ΔX_1:
  - $p_1 = \max\limits_{\Delta Y}(\Delta X_1, \Delta Y)$;
  - если $[p_1, B_{n-1}] \ge \underline{B_n}$:
    - переход на второй этап.

Алгоритм 2. Второй этап поиска.

- Для всех возможных ΔX_2 и ΔY_2:
  - $p_2 = (\Delta X_2, \Delta Y_2)$;
  - если $[p_1, p_2, B_{n-2}] \ge \underline{B_n}$:
    - переход на третий этап.

Алгоритм 3. i-й этап поиска, где $3 \le i \le n-1$

- Для всех ΔY_i:
  - $\Delta X_i = \Delta X_{i-2} \oplus \Delta Y_{i-1}$;
  - $p_i = (\Delta X_i, \Delta Y_i)$;
  - если $[p_1, p_2, \ldots, p_i, B_{n-i}] \ge \underline{B_n}$:
    - переход на $i+1$ этап.

Алгоритм 4. Последний этап поиска

- $\Delta X_n = \Delta X_{n-2} \oplus \Delta Y_{n-1}$;
- $p_n = (\Delta X_n, \Delta Y)$;
- если $[p_1, p_2, \ldots, p_n] \ge \underline{B_n}$:
  - $\underline{B_n} = [p_1, p_2, \ldots, p_n]$.

Повышение быстродействия в алгоритме достигается благодаря отсечению на каждом из шагов разностей, на основе которых невозможно построить дифференциальный путь не хуже текущего лучшего известного. Именно с этим связаны необходимость выбирать начальную оценку максимально близкой к лучшей возможной вероятности, но не большей ее ($\underline{B_n} \le B_n$). Стоит также заметить, что оценка вероятности конечной характеристики на каждом из этапов обладает погрешностью, поэтому в общем случае

невозможно доказать, что результаты будут действительно оптимальными.

2. МЕТОДЫ ИССЛЕДОВАНИЙ

2.1. Основной анализ на основе алгоритма Мацуи и pDDT

Дифференциальный криптоанализ блочного ARX-шифра «Кипарис-256» состоит из следующих шагов:

1) построение pDDT для операции сложения;
2) построение pDDT для «элементарного» цикла;
3) построение pDDT для цикловой функции;
4) использование pDDT цикловой функции в алгоритме Мацуи для поиска оптимальных дифференциальных характеристик через несколько циклов.

2.2. Дополнительный анализ на основе SMT-решателя

Также, независимо проводились исследования с использованием SMT-решателя Z3 [15]. С его помощью производился:

1) поиск дополнительных маршрутов прохождения дифференциальных путей через цикловую функцию в алгоритме Мацуи. Маршруты рассчитывались таким образом, чтобы «выводить» алгоритм на входы предварительно подготовленной pDDT цикловой функции;

2) поиск одноцикловых характеристик с фиксированной вероятностью.

2.3. Частичная таблица распределения разностей сложения

Частичная таблица распределения разностей сложения строилась по приведенному выше алгоритму и для ускорения процесса не включает записи с вероятностями ниже 2^{-3}. Такое граничное значение позволяет, с одной стороны, очень эффективно рассчитывать и хранить в оперативной памяти таблицу, с другой же стороны дает очень высокие погрешности при поиске дифференциальных характеристик, т.к. большая часть возможных дифференциальных переходов не принимается во внимание.

pDDT для «элементарного» цикла (рис.2) легко выводится из таблицы сложения, поэтому в ходе расчетов не хранилась и рассчитывалась «на лету».

2.4. Частичная таблица распределения разностей цикловой функции

В первую очередь были исследованы 2^{16} разностей, в старших разрядах входных слов цикловой функции, которые последовательно проходили через всю цикловую функцию.

После чего все те же входные разности были использованы как промежуточный результат выполнения цикловой функции, полученный между двумя применениями HalfRound преобразований «Кипарис-256». Более подробно процесс такого поиска на исходных данных, построенных с учетом весов Хэмминга, описан в [16].

Так же был использован адаптивный выбор входных разностей с минимизацией битовых весов, были исследованы все комбинации из 1, 2 и 3 активных бит. Однако поиск велся не из «середины» цикловой функции, а со всех точек между «элементарными» циклами.

Во всех трех случаях вероятности дифференциальных характеристик не ограничивались какими-либо граничными значениями (в силу небольших объемов входных данных). Вероятность полноциклового пути оценивалась как произведение вероятностей переходов на отдельных нелинейных преобразованиях.

2.5. Поиск дифференциальных характеристик

После получения частичной таблицы распределения разностей для одного применения цикловой функции она может быть использована в алгоритме Мацуи. Поскольку предварительные результаты по шифру «Кипарис-256» отсутствовали, в качестве начальных вероятностей были взяты 0, что лишь немного замедляет алгоритм (по сути, первая найденная характеристика считается лучшей известной на момент начала анализа).

В качестве рабочих одноцикловых вероятностей в основном использовалась pDDT цикловой функции, отсортированная по убыванию вероятности, что позволяет сразу же получить достаточно хорошее приближение к наилучшему дифференциальному пути (в ходе расчетов первый найденный путь всегда был лучшим). Начальный набор вероятностей был расширен за счет расчета «на лету» характеристик для входов, не известных pDDT, что позволяет продолжать поиск в ситуациях, когда разность выхода предыдущего цикла имеет большое количество активных бит. Ограничений на вероятности таких путей не налагалось.

Для повышения быстродействия, эти промежуточные характеристики хранились в кэше небольшого размера (1000 элементов) до его переполнения. При переполнении кэш полностью очищался.

3. РЕЗУЛЬТАТЫ ИССЕДОВАНИЙ

В результате работы была найдена характеристика с вероятностью 2^{-2} через один цикл, что подтверждает предыдущие результаты, дополнительно был найден ряд характеристик с вероятностями 2^{-6}, 2^{-9}, 2^{-10} и ниже. Все эти дифференциальные пути использовались при поиске многоцикловых характеристик.

Лучшие результаты и их вероятности:

- 3 цикла: 80000000 00000000 80000000 80008000 -> 88000000 40404404 00808088 00800088 с вероятностью 2^-12;
- 4 цикла: 80000000 00000000 80000000 80008000 -> 68208626 75211214 CA4A2004 6AC4EA4C с вероятностью 2^-106;

– 5 циклов: 80000000 00000000 80000000 80008000 -> AF6A6C6F 1C9496F1 2F6EE961 5ACFAE08 с вероятностью 2^-253.

Приведенные выше данные говорят о том, что, несмотря на достаточно простое цикловое преобразование, сложность атаки быстро растет по мере увеличения количества циклов. Это связано с хорошим лавинным эффектом, благодаря которому очень сложно получить дифференциальный путь через один цикл, который, имея большую вероятность сам по себе, на выходе имел разность, позволяющую в следующем цикле также получить переход с большой вероятностью.

С другой же стороны, алгоритм имеет одноцикловые характеристики с высокой вероятностью и низким весом Хэмминга, как входа, так и выхода. Примеры таких характеристик:

– 00000000 80000000 00800000 80008080 -> 80000000 00004000 00000080 00000080 с вероятностью 2^{-2};
– 00000000 80000000 00800000 80008080 -> 80000000 0000C000 00000180 00000080 с вероятностью 2^{-3};
– 00000000 80000000 01800000 80008080 -> 80000000 0000C000 00000180 00000080 с вероятностью 2^{-4};
– 81181000 80081000 00000000 01008000 -> 00000000 00000040 80000000 00000000 с вероятностью 2^{-5}.

Эти переходы не использовались в расчетах и не присутствовали в pDDT цикловой функции.

Использование же SMT-решателя для поиска дополнительных путей в алгоритме Мацуи эффекта не дало по той причине, что найденные «короткие пути» имели низкую вероятность, хотя были построены на основе pDDT с высокими вероятностями.

ВЫВОДЫ

На основе полученных данных можно утверждать, что шифр «Кипарис-256» является устойчивым к дифференциальному криптоанализу после 6 циклов, однако уже после 4 циклов анализ имеет большую сложность данных и не может быть осуществлен на практике.

Таким образом, запас стойкости алгоритма «Кипарис-256» по отношению к рассмотренной атаке составляет 4 цикла из 10.

Литература

[1] *Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., & Biryukov, A.* (2016). Design Strategies for ARX with Provable Bounds: SPARX and LAX (Full Version). IACR Cryptology ePrint Archive.

[2] *Mouha, N., Velichkov, V., Canniere, C. De., Preneel B.* Toolkit for the Differential Cryptanalysis of ARX-based Cryptographic Constructions, Workshop on Tools for Cryptanalysis, 2010.

[3] *Heys, M.* The Design of Substitution-Permutation Networks Resistant to Differential and Linear Cryptanalysis. Journal of Cryptology - JOC. 9. 148–155. 10.1007/BF02254789.

[4] *Biham, E. and A. Shamir.* "Differential cryptanalysis of DES-like cryptosystems." In Menezes and Vanstone 90, 2–21.

[5] National Institute of Standards and Technology, "FIPS-46-3: Data Encryption Standard." Oct. 1999.

[6] ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – М.: Изд-во стандартов, 1989. – 20 с.

[7] *Aoki, K., Ichikawa, T., and Kanda. M. Camellia*: A 128-Bit Block Cipher Suitable for Multiple Platforms-Design and Analysis-. 2000, http://www.cryptonessie.org.

[8] International Organization for Standardization. ISO/IEC 18033-3:2010. Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers, 2010.

[9] ДСТУ 7624:2014. Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення [Текст]. – Введ. 01–07–2015. – К.: Мінекономрозвитку України, 2015. – 119 с.

[10] *Daemen, J.* (1999). AES Proposal: Rijndael.

[11] Родінко, М. Ю. Постквантовий малоресурсний симетричний блоковий шифр «Кипарис» / М. Ю. Родінко, Р. В. Олійников // Радиотехника. – 2017. – Вып. 189. – С. 100–107.

[12] *Wallén, J.* (2003). On the Differential and Linear Properties of Addition.

[13] *Biryukov, A., and Velichkov, V.* "Automatic search for differential trails in arx ciphers," in Topics in Cryptology (CT-RSA'14), pp. 227–250, Springer, 2014.

[14] *Matsui, M.* On correlation between the order of S-boxes and the strength of DES. In: De Santis A. (eds) Advances in Cryptology — EUROCRYPT'94. EUROCRYPT 1994. Lecture Notes in Computer Science, vol 950. Springer, Berlin, Heidelberg.

[15] *Leonardo Mendonça de Moura and Nikolaj Bjørner.* Z3: An Efficient SMT Solver. In Proceedings of the 14th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS 2008), Budapest, Hungary. 337– 340.

[16] *Rodinko, M., Oliynykov, R., and Eliseev, R.* "Search for one-round differential characteristics of lighweight block cipher Cypress-256", 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kiev, 2018, pp. 312–315.

Поступила в редколлегию 03.12.2018

Елисеев Роман Юрьевич, студент, ХНУРЭ. Область научных интересов – анализ и синтез симметричных криптографических преобразований.

Родинко Мария Юрьевна, аспирантка кафедры безопасности информационных систем и технологий ХНУ им. В.Н. Каразина. Область научных интересов – анализ и синтез блочных симметричных шифров.

Олейников Роман Васильевич, доктор технических наук, профессор кафедры безопасности информационных систем и технологий ХНУ им. В.Н. Каразина. Область научных интересов – анализ и синтез симметричных криптографических преобразований, безопасность программного обеспечения, сетевая безопасность, блокчейн.

УДК 621.3.06

Єлисєєв Р. Ю. **Диференційний криптоаналіз блокового ARX-шифру «Кипарис-256»** / Р. Ю. Єлисєєв, М. Ю. Родінко, Р. В. Олійников // Прикладна радіоелектроніка: наук. – техн. журнал. – 2018. – Том 17, № 3, 4. – С. 121–126.

У статті наведено результати диференційного криптоаналізу симетричного блокового шифру «Кипарис-256», виконаного із застосуванням ряду методів, зокрема, за допомогою алгоритму Мацуї і використанням часткових таблиць розподілу різниць. В ході досліджень було знайдено ряд диференційних характеристик включно до п’яти циклів шифрування. Крім того, був виявлений ряд характеристик з ймовірністю від 2^{-2} до 2^{-5}, входи і виходи яких мають малу вагу Хемінга.

Ключові слова: малоресурсна криптографія, блоковий симетричний шифр, диференційний криптоаналіз, диференційна характеристика.

Іл.: 3. Бібліогр.: 16 назв.

UDC 621.3.06

Eliseev R. Yu. **Differential cryptanalysis of ARX block cipher Cypress-256** / R. Yu. Eliseev, M. Yu. Rodinko, R. V. Oliynykov // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 121–126.

This paper presents the results of differential cryptanalysis of the block cipher Cypress-256 performed by several methods, in particular, with the help of Matsui algorithm and application of partial difference distribution tables. As a result a number of differential characteristics up to five rounds of ciphering were found. In addition, a number of differential characteristics with low probabilities (2^{-2}, 2^{-3}, 2^{-4} and 2^{-5}) whose inputs and outputs have small Hamming weight were found.

Keywords: lightweight cryptography, block symmetric cipher, differential cryptanalysis, differential characteristic.

Fig.3. Ref.: 16 items.

УДК 004.056.55

АНАЛІЗ СУТНОСТІ ТА МОДЕЛІ ПРОТОКОЛУ ІНКАПСУЛЯЦІЇ КЛЮЧІВ У КІЛЬЦІ ПОЛІНОМІВ НАД СКІНЧЕНИМ ПОЛЕМ

І. Д. ГОРБЕНКО, О. Г. КАЧКО, В. А. ПОНОМАР, М. В. ЄСІНА, О. С. АКОЛЬЗІНА, В. А. КУЛІБАБА

У роботі розглядається аналіз сутності та моделі протоколу інкапсуляції ключів у кільці поліномів над скінченим полем. Наводяться основні положення стосовно протоколів. Наводяться результати порівняння механізмів інкапсуляції ключів. Наводиться криптографічний протокол інкапсуляції та декапсуляції ключа в NTRU Prime Ukraine.

Ключові слова: інкапсуляція ключів, кільце поліномів, протокол.

ВСТУП

На сьогодні основні зусилля світової криптографічної спільноти зосереджені на створенні практичних квантово-стійких механізмів електронного підпису (ЕП), асиметричного шифрування (АСШ) та протоколів інкапсуляції ключів (ПІК) [1–7]. Одним з механізмів, що може бути застосований для побудови АСШ та ПІК для постквантового періоду, є різні варіанти застосування криптографічних перетворень в кільцях поліномів, випробуваним варіантом його є NTRU криптосистема [1]. У [3] запропоновано механізми побудови АСШ та ПІК, що можуть забезпечити 5 рівень квантової криптографічної стійкості (128 біт квантової та 256 класичної криптостійкості). Але, на наш погляд, важливою як теоретичною, так і практичною є проблема забезпечення включно до 7 рівня криптографічної стійкості (256 біт квантової та 512 класичної криптостійкості). З точки зору постквантових ПІК на сьогодні важливим є: аналіз стану розроблення та стандартизації ПІК; обґрунтування та розробка формального опису протоколу інкапсуляції ключів, у якому можна було б закласти формально вимоги, незалежно від математичних перетворень, що застосовуються, а також розробка пропозицій щодо побудови ПІК, включно до 7 рівня безпеки. При цьому необхідно, щоби ПІК будувався на основі тієї ж математичної бази, що і АСШ.

Метою цієї статі є аналіз стану розроблення та стандартизації ПІК взагалі, і для постквантового періоду, розробка та формальний опису ПІК, в який добре вписувались би як існуючі ПІК [2, 3], так і перспективні постквантові на основі кілець поліномів над скінченими полями, а також аналіз властивостей такого виду ПІК.

З огляду на суттєву важливість застосування алгоритмів направленого шифрування (АСШ) на міжнародному рівні під час виконання Європейського проекту NESSIE, особливу увагу було приділено реалізації висунутих вимог щодо ПІК. У подальшому на основі отриманих результатів, пропозицій та рекомендацій було прийнято міжнародний стандарт ISO/IEC 18033-2 «Інформаційна технологія – Методи захисту – Алгоритми захисту – Частина 2: Асиметричні шифри» [2]. В процесі підготовки та оголошення конкурсу NIST США як основний примітив визначив ПІК.

Аналіз механізмів та безпосередньо ПІК, що наведені в [1–6], дозволив зробити такі висновки.

– Основні зусилля на світовому рівні зосереджені на створенні механізмів інкапсуляції ключів та ПІК.

– Розроблено та подано на різні конкурси, але в основному на конкурс постквантових криптопримітивів, механізми реалізації ПІК.

– Основним призначенням ПІК є генерування та передача відправником отримувачу інкапсульованого ключа та ключових даних в інкапсульованому (захищеному) вигляді та декапсуляцію їх отримувачем відповідно.

Механізм інкапсуляції ключів у запропонованій термінології призначений для інкапсуляції та декапсуляції ключів, а також обчисленні (генеруванні) та використанні секретних ключів, під час застосування, наприклад, режимів роботи симетричних блокових і потокових шифрів [4]. У таблиці 1 наведено перелік основних кандидатів на постквантові механізми та ПІК, які розглядаються та порівнюються на конкурсі NIST США [3, 5, 6].

Таким чином, усього подано 40 кандидатів на постквантові стандарти механізмів інкапсуляції ключів. Під час їхнього розроблення використано різні математичні основи, в тому числі: алгебраїчні решітки; коди, мультиваріативні перетворення у квадратичних полях, перетворення типу ПІК тощо.

В процесі досліджень, деякі результати яких наведені в цій статті, було проведено аналіз та висунуті вимоги до механізмів ПІК та запропоновано конкретні реалізації, з урахуванням [3] та механізмів ПІК, що наведені у [2].

Також враховано, що у стандарті ISO/IEC 18033-2 наведено матеріали щодо обґрунтування нової структури для асиметричного направленого шифрування KEM-DEM [2].

Вказаний механізм реалізує АСШ, оскільки інкапсуляція виконується на відкритому ключі (кортежі) іншого абонента, а декапсуляція на особистому ключі (кортежі) абонента. При декапсуляції перевіря-

Таблиця 1

Механізми постквантових кандидатів на ПІК

Математичні методи	Кандидати в стандарти	Число кандидатів
Алгебраїчні решітки	CRYSTALS-KYBER, Ding Key Exchange, HILA5, EMBLEM та R.EMBLEM, FrodoKEM, KINDI, LAC, LIMA, Lizard, NewHope, NTRU Encrypt, NTRU-HRSS-KEM, NTRU Prime, Odd Manhattan's, KCL (pka OKCN/AKCN/CNKE), Round2, SABER, Titanium	18
Коди	BIG QUAKE, BIKE, Classic McEliece, DAGS, HQC, LAKE, LEDEkem, Lepton, NTS-KEM, QC-MDPC KEM, RLCE-KEM, RQC	12
Мульти варіативні перетворення	CFPKM, DME	2
Геш-перетворення	==========	0
Інші перетворення	Mersenne-756839, Post-quantum RSA-Encryption, Ramstake, SIKE, Three Bears	5
Відкликані кандидати	HK17, RVB, Edon-K	3
Атаковані але виправлені		0
Усього кандидатів		40

ється цілісність і справжність відкритого ключа та перевіряється справжність (автентичність) іншого абонента.

Можна говорити про неспростовність абонента, що генерує особисті ключі. Наприклад, у нашому випадку неспростовність абонента В може бути реалізована способом передачі відкритого кортежу С через третю довірену сторону тощо.

У цій статті наводяться початкові результати досліджень щодо механізмів та ПІК, у тому числі:

1. Попередні результати порівняння механізмів та протоколів інкапсуляції ключів. При порівняльному аналізі використовувалася сукупність безумовних та умовних критеріїв.

2. Формальний опис механізму та протоколу інкапсуляції ключів.

3. Обґрунтування та аналіз моделі безпеки щодо постквантових механізмів і ПІК.

4. Попередні дослідження дозволили як найбільш перспективні виділити механізм NTRUPrime.

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕХАНІЗМІВ ТА ПРОТОКОЛІВ ІНКАПСУЛЯЦІЇ КЛЮЧІВ

1.1. Попередні результати порівняння механізмів та протоколів інкапсуляції ключів

Під час порівняльного аналізу використовувалася сукупність безумовних та умовних критеріїв. Як безумовні критерії запропоновано наступні [7, 8]:

1) Іст. – рівень криптографічної стійкості;

2) lв.к – довжина відкритого ключа;

3) lо.к – довжина особистого ключа;

4) lрез. – довжина результату криптоперетворення;

5) Тпр. – швидкість прямого криптоперетворення;

6) Тзв. – швидкість зворотного криптоперетворення.

Експертні оцінки використовувалися для оцінки важливості кожної з наведених характеристик, а безпосередньо при порівнянні алгоритмів використовувалися об'єктивні числові значення, шкала оцінки та вагові коефіцієнти важливості характеристик, що були отримані в ході експертного оцінювання (таблиця 2).

Таблиця 2

Експертні оцінки характеристик криптоалгоритмів

Показники / Експерти	$I_{ст.}$	$l_{в.к}$	$l_{о.к}$
1	0,070	0,047	0,033
2	0,066	0,066	0,032
3	0,081	0,039	0,039
4	0,094	0,048	0,026
5	0,078	0,078	0,036
W	0,078	0,056	0,033
Показники / Експерти	**$l_{рез.}$**	**$T_{пр.}$**	**$T_{зв.}$**
1	0,283	0,283	0,283
2	0,202	0,316	0,316
3	0,367	0,237	0,237
4	0,424	0,204	0,204
5	0,420	0,193	0,193
W	0,339	0,247	0,247

Оскільки в NIST було викладено велику кількість алгоритмів, що використовують перетворення в решітках числових полів та математичні коди (порівняно з іншими), то було вирішено спочатку порівняти алгоритми кожного з цих типів, а потім алгоритми, що займуть 1–2 місця з іншими. В даному дослідженні при виборі алгоритмів висувалися додаткові безумовні вимоги:

1) алгоритм має гарантувати, що найменше 3 рівень безпеки за класифікацією NIST;

2) якщо існує декілька варіантів наборів параметрів для одного алгоритму, то порівняно бере участь варіант, що гарантує найбільшу безпеку.

В таблиці 3 наведено характеристики обраних для порівняння алгоритмів, що засновані на використані перетворень в алгебраїчних решітках.

Таблиця 3

Характеристики алгоритмів інкапсуляції ключів на алгебраїчних решітках

Алгоритми	$I_{ст.}$	$I_{в.к}$	$I_{о.к}$	$I_{рез.}$	$T_{пр.}$	$T_{зв.}$
CRYSTALS-KYBER	5	5 422	5 422	5 422	2 112 734	15 843 611
Ding Key Exchange	5	4 964	4 964	4 964	9 541 851	13 413 988
FrodoKEM	3	10 449	15 673	6 763	7 419 629	2 933 863
HILA5	5	2 758	2 758	2 758	7 921 744	5 408 625
KINDI	5	11 616	2 973 704	2 144	1 035 852	151 282 021
LAC.CCA KEM	5	2 064	3 072	2 176	998 494	7 617 506
LIMA	5	1 680	32	1 568	914 137	1 698 920
Lizard	5	840	32	1 184	1 734 600	397 902 189
NewHope	5	9 616	19 872	9 736	1 525 623	4 428 250
NTRUEncrypt	5	7 989	8 029	15 962	299 133	456 199
Odd Manhattan’s	5	5 884	5 924	11 752	132 351 915	3 827 502
Round2	5	1 456	1 712	2 544	767 892	780 283
SABER	5	1 184	1 472	1 824	1 825 775	518 385
ThreeBears	5	1 056	2 080	1 536	241 051	912 083
Titanium	5	544	1 056	1 024	3 469 480	257 284
NTRUPrime_AVX	5	1 600	1 218	1 047	99 149	125 820

На рис. 1 відображено гістограму відносної переваги алгоритмів. Як видно найбільшу перевагу має алгоритм NTRUPrime_AVX, на другому місці – Titanium, на третьому ThreeBears, на четвертому NTRUEncrypt, на п’ятому SABER тощо.

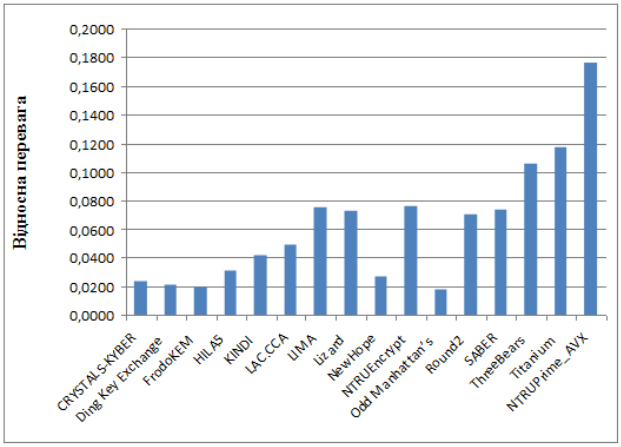

Рис. 1. Відносна перевага алгоритмів на основі перетворень в алгебраїчних решітках

На рис. 2 відображено гістограму відносної переваги алгоритмів. Як видно, найбільшу перевагу має алгоритм LAKE, на другому місці з невеликим відривом – RLCE-KEM.

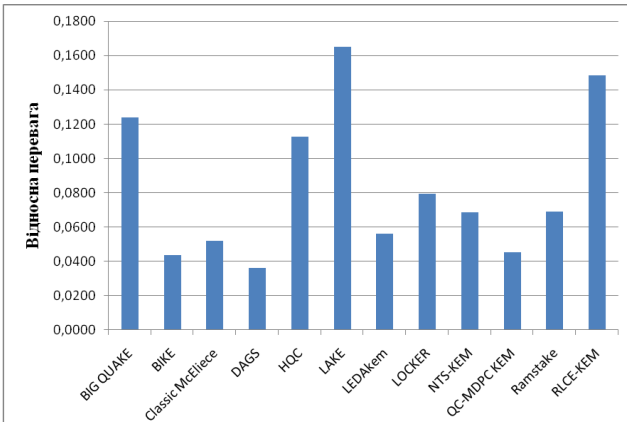

Рис. 2. Відносна перевага алгоритмів на основі математичних кодів

В таблиці 4 наведено характеристики обраних для порівняння алгоритмів, що засновані на використанні математичних кодів.

Таблиця 4

Характеристики алгоритмів інкапсуляції ключів на математичних кодах

Алгоритми	$l_{ст.}$	$l_{в.к}$	$l_{о.к}$	$l_{рез.}$	$T_{пр.}$	$T_{зв.}$
BIG QUAKE	5	149 800	41 804	492	12 772 072	51 333 539
BIKE	5	9 034	9 034	9 034	1 967 128	43 842 551
Classic McEliece	5	8 188	8 188	8 188	1 786 760	37 247 437
DAGS	5	4 964	4 964	4 964	109 932 700	15 652 888
HQC	5	1 440	3 168	1 504	3 529 138	2 703 872
LAKE	5	1 040	1 536	1 088	604 031	4 774 104
LEDAkem	5	2 256	32	3 120	106 848 011	28 161 595
LOCKER	5	9 523	19936	10 023	850 000	5 372 000
NTS-KEM	5	7 417	7 457	14 818	1 081 765	5 564 976
QC-MDPC KEM	5	5 115	5 155	10 214	28 662 874	3 254 700
Ramstake	5	2 819	2 859	5 622	159 129 068	1 575 641
RLCE-KEM	5	1 696	1 664	2 083	21 850 039	168 952

В таблиці 5 наведено характеристики алгоритмів, що засновані на різних криптографічних перетвореннях.

Таблиця 5

Характеристики алгоритмів інкапсуляції ключів

Алгоритми	$l_{ст.}$	$l_{в.к}$	$l_{о.к}$
Titanium (решето числового поля)	5	544	1 056
NTRUPrime_AVX (решето числового поля)	5	1 600	1 218
LAKE (математичні коди)	5	1 040	1 536
RLCE-KEM (математичні коди)	5	1 696	1 664
Edon-K (геш)	3	10 449	15 673
SIKE (ізогенії)	3	1 023	1 173
Алгоритми	$l_{рез.}$	$T_{пр.}$	$T_{зв.}$
Titanium (решето числового поля)	1 024	3 469 480	257 284
NTRUPrime_AVX (решето числового поля)	1 047	99 149	125 820
LAKE (математичні коди)	1 088	604 031	4 774 104
RLCE-KEM (математичні коди)	2 083	21 850 039	168 952
Edon-K (геш)	6 251	419 330	691 377
SIKE (ізогенії)	1 023	565 483 974	767 267

На рис. 3 відображено гістограму відносної переваги алгоритмів. Як видно, найбільшу перевагу має алгоритм NTRUPrime_AVX, на другому місці – Titanium, тобто перші місця посіли алгоритми на решеті числового поля.

Таким чином, за результатами порівняльного аналізу можна зробити такий висновок щодо переваг кандидатів на постквантовий стандарт ПІК на першому місці – NTRUPrime_AVX, на другому – Titanium, на третьому – SIKE, на четвертому – RLCE-KEM, на п'ятому – LAKE, на шостому – Edon-K.

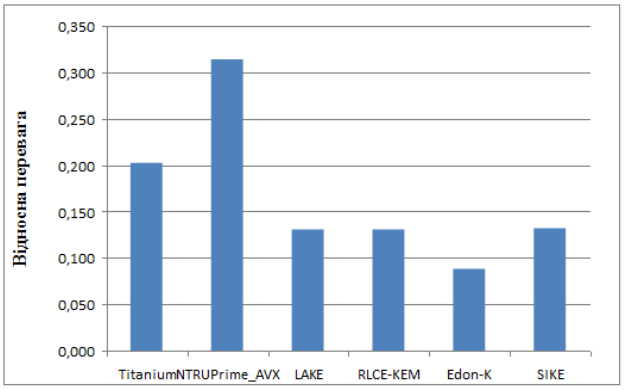

Рис. 3. Відносна перевага алгоритмів інкапсуляції ключів

2. ФОРМАЛЬНИЙ ОПИС ПРОТОКОЛУ ІНКАПСУЛЯЦІЇ КЛЮЧІВ

Призначення ПІК – інкапсуляція ключа відправником та передача його отримувачу в інкапсульованому (захищеному) вигляді.

Для реалізації протоколу інкапсуляції необхідно:

– узгодити стандарт асиметричного шифрування та інших необхідних стандартів криптографічних перетворень;

– узгодити чи відповідним чином згенерувати та захищеним способом розподілити загальні для подальшого здійснення захищеного зв'язку типу клієнт–клієнт чи клієнт–сервер;

– встановити загально-системі чи загальні параметри відповідної асиметричної криптосистеми;

– отримувач асиметрично зашифрованих повідомлень та інкапсульованих ключів від інших клієнтів генерує свою асиметричну ключову пару – секрет-

ний (особистий) та відкритий ключі. Особистий (секретний) зберігається і доступний тільки цьому отримувачу, а відкритий робиться доступним захищеним способом відправнику;

– клієнт-відправник генерує сеансів ключ інкапсуляції у вигляді асиметричної пари (можливо асиметричних пар) – секретного та відкритого ключів інкапсуляції та вводить їх у дію;

– клієнт-відправник захищає відкритий ключ інкапсуляції засобом його безпосередньої інкапсуляції та підготовляє його для відправки у захищеному вигляді клієнту-отримувачу;

– клієнт-отримувач приймає захищене повідомлення та здійснює його декапсуляцію і одночасно визначає його цілісність та справжність;

– клієнт-відправник використовує асиметричний ключ інкапсуляції – секретний та відкритий ключі обчислює секретний ключ для його використання при зашифруванні та автентифікації тощо повідомлень для клієнта чи сервера;

– клієнт-отримувач використовує декапсульований ключ та, можливо, свої загальні параметри та секретний ключ обчислює секретний ключ для його використання при розшифруванні та автентифікації отриманих зашифрованих чи одного зашифрованого повідомлення;

– за необхідності робиться підтвердження отриманих захищених повідомлень.

Попередні результати досліджень з урахуванням [2, 3, 7] дозволили у найбільш загальному випадку запропонувати механізм та ПІК. Сутність ПІК у наступному.

В ході генерації (обчисленні) асиметричної ключової пари отримувача необхідно:

1. Обчислити (згенерувати) асиметричну пару асиметричного шифрування – відкритий P_k та секретний (особистий) S_k ключі.

2. Зробити доступними для відправника компоненти відкритого ключа, наприклад, у вигляді:

$$P_k := (ЗП, W, \lambda), \quad (1)$$

де *ЗП* – значення загальних параметрів АСШ; W – відкритий ключ (можливо його сертифікат); λ – рівень безпеки.

3. Записати та зберігати для використання особистий ключ S із забезпеченням його конфіденційності, цілісності, справжності та доступності тощо, наприклад, у вигляді:

$$S_k := (S, P_k), \quad (2)$$

де S – секретний ключ розшифрування отримувача, а P_k – відкритий ключ (1).

Відправник (наприклад, клієнт-сервер чи клієнт) може інкапсулювати свій сеансовий асиметричний ключ, якщо він має доступ до відкритого ключа отримувача (1). Для цього він виконує такі операції.

Перед генерацією асиметричної ключової пари відправник повинен:

– узгодити та налаштувати механізм та засіб асиметричного шифрування;

– узгодити та налаштувати загальні параметри засобів механізму шифрування, в тому числі алгоритм вироблення ключа $KDF(\bullet)$ на основі спільної таємниці.

Алгоритм інкапсуляції відправником виконується в такій послідовності (при невідомому S_k):

1. Згідно з прийнятим механізмом генерується ключ сеансу r відправника.

2. Згідно з прийнятим механізмом обчислюється відкритий ключ сеансу C відправника, наприклад, у вигляді

$$C := F(r, ЗП). \quad (3)$$

3. Відправником обчислюється спільна таємниця інкапсуляції ключа $Q \neq 0$ відправника

$$Q := F(C, r, ЗП, W). \quad (4)$$

4. Відправником обчислюється значення секретного (особистого) ключа

$$K := KDF(Q), \quad (5)$$

наприклад, для подальшого його застосування як ключ симетричного шифрування.

5. На останок відправник має секретний сеансів ключ K_C для подальшого використання, наприклад, у вигляді симетричного ключа сеансу, а також, інкапсульований у відкритий ключ C свій секретний ключ сеансу r.

Далі відправник свій відкритий ключ сеансу C, забезпечуючи його цілісність, справжність та доступність, робить доступним отримувачу. Використовуючи відкритий ключ сеансу C та свій секретний (особистий) ключ S, отримувач може зробити декапсуляцію, а також обчислити секретний ключ наступного сеансу K_C.

Вхідними даними для алгоритму декапсуляції є відкритий ключ C, що отриманий від відправника, та особистий (секретний) ключ S_k, конкретно секретний ключ S із (2).

Декапсуляція ключа отримувачем здійснюється в такій послідовності:

1. Отримувач, використовуючи свій секретний ключ S, обчислює спільну таємницю

$$Q := F(C, ЗП, S). \quad (6)$$

За необхідності перевіряється, що $Q \neq 0$.

2. Відправником обчислюється значення секретного ключа

$$K := KDF(Q). \quad (7)$$

3. Відправник та отримувач мають секретний ключ K_C для подальшого використання його, наприклад, у вигляді симетричного ключа сеансу шифрування та автентифікації.

3. МОДЕЛЬ БЕЗПЕКИ ЩОДО ПЕРСПЕКТИВНИХ ТА ПОСТКВАНТОВИХ МЕХАНІЗМІВ І ПРОТОКОЛІВ ІНКАПСУЛЯЦІЇ КЛЮЧІВ

Модель безпеки щодо перспективних та постквантових механізмів і протоколів інкапсуляції ключів розглядається з урахуванням основних положень та пропозицій, що викладені за даною тематикою в [9, 10].

За основу було взято модель безпеки Canetti-Krawczyk (CK) використовується для автентифікації обміну ключами (AKE) [9].

Також використано неформально поняття "досконала пряма секретність (безпека)" (PFS) [9, 10] зазначається як властивість, що "компрометування довгострокових ключів не компрометує минулі ключі сеансу".

Коли доводять, що протокол має бути SK-безпечним, автоматично отримується доказ того, що цей протокол гарантує PFS.

Також вважатимемо, що протокол KE задовольняє SK-безпеку без PFS, якщо він використовує SK-безпеку відносно будь-якого KE-зловмисника в UM, що не допускає припинення дії ключів [9, 10].

На основі CK моделі були визначені загальні положення щодо вимог до ПІК та до моделі безпеки. Їхня сутність у тому, що [7]:

– *повнота* протоколу, сутність якої полягає в тому, що якщо *S* істинне, то суб'єкт/об'єкт *P* з великою ймовірністю переконає про це перевірника і той згодиться, що *S* істинне;

– *коректність* протоколу, яка проявляється в тому, що, якщо *S* хибне, то суб'єкт *V* виявить це з великою ймовірністю, тобто *P* не переконає *V*, що *S* істинне.

Крім вимог повноти й коректності, протокол з нульовим розголошенням має задовольняти ще й таку умову – нульове розголошення інформації про *S* – у результаті виконання протоколу (*P,V,S*) перевірник не зменшує свою апріорну невизначеність відносно твердження *S*, тобто він не отримує ніякої інформації про те, чому *S* істинне.

Дослідження показали, що забезпечення нульового розголошення на практиці може бути здійснене, якщо [7]:

– кожен суб'єкт, який має намір діяти або як пред'явник *P*, або як перевірник *V*, повинен мати засоби генерації випадкових чисел;

– абоненти групи мають дійти згоди щодо того, яка функція гешування використовуватиметься;

– кожен об'єкт, який має намір діяти як пред'явник, має бути забезпечений асиметричною ключовою парою, обраною згідно з рекомендаціями, які наведено вище;

– кожен об'єкт, який має намір діяти як перевірник, має бути забезпечений засобами обчислення довірчих копій відкритих ключів перевірки для об'єктів, чия ідентичність перевіряється.

Сутність моделі CK. Модель безпеки CK стосується безпеки ключа сеансу, що використовується на сеансі зв'язку. В ході її оцінки використовується формальна модель для протоколів обміну ключами та можливостей криптоаналітика (зловмисника). Поняття безпеки, яке називається *безпекою ключа сеансу* (або *SK-security*), направлене на забезпеченні безпеки окремих ключів сеансу. Її порушення є компрометацією сеансового ключа. У випадку безпечності ключа зловмисник "нічого не дізнається про значення ключа", коли він перехоплює дані протоколу обміну ключами та здійснює атаки на інші сеанси та сторони, що взаємодіють. Такий підхід є стандартним для моделі семантичної безпеки, коли криптоаналітик не може відрізнити реальне значення ключа від незалежного випадкового значення. В даному випадку говорять про реалізацію криптографічного протоколу «нульових знань». Водночас таке визначення SK-безпеки необхідно використовувати обережно, тому що забезпечення необхідного рівня стійкості під час встановлення та використання протоколів обміну ключами для реалізації захищених каналів, мета може досягатися без складних вимог.

Детальний аналіз, що проведений з метою визначення критеріїв та показників оцінки, та порівняння ПІК, дозволив обґрунтувати та запропонувати поділ наведених критеріїв на безумовні та умовні критерії оцінки безпечності ПІК.

Таким чином, до безумовних критеріїв оцінки ПІК належатимуть.

1. Рівень реалізації моделі безпеки IK-CPA/CCA2.

2. Криптографічна стійкість (складність криптографічного аналізу) щодо криптоперетворення АСШ – $W_{асш}$ (ЕП – $W_{еп}$), що застосовуються у протоколі ПІК.

3. Криптографічна стійкість (складність криптографічного аналізу) щодо криптоперетворення інкапсуляції – $W_{пік}$, що застосовуються у протоколі інкапсуляції;

4. Криптоживучість ключів щодо криптоперетворення АСШ – $G_{асш}$ (ЕП – $G_{еп}$), що застосовуються у протоколі ПІК.

5. Криптоживучість ключів, що застосовуються у протоколі ПІК – $G_{пік}$;

6. Захищеність криптопротоколу від раніше переданих повідомлень – $W_{рпп}$.

7. Неспростовність криптоперетворень АСШ – $N_{асш}$, що встановлені для криптографічного захисту;

8. Неспростовність криптоперетворень ЕП – $N_{еп}$, що встановлені для криптографічного захисту;

9. Новизна ключів АСШ(ЕП) – $W_{кл}$, що застосовуються в протоколі інкапсуляції ПІК;

10. Характеристика степеня нерозрізнюваності для ключів АСШ, ЕП та ПІК.

4. АНАЛІЗ ТА РОЗРОБКА ПІК NTRU PRIME

4.1. Загальні положення

У цьому розділі наводяться результати аналізу та обґрунтування перспективного підходу до реалізації протоколу інкапсуляції ключів на основі застосування NTRU Prime. По суті пропонується модернізований нами NTRUPrime з використанням кілець поліномів.

Використовуватимемо такі математичні перетворення [1, 3, 8–10]:

1. «Класичний NTRU»: кільця виду $(Z/q)[x]/\left(x^p-1\right)$, де p – просте число і q – є степенем 2, використовуються в класичній криптосистемі NTRU [1]. Він не рекомендується для застосування, оскільки на нього існують атаки.

2. «NTRU NTT»: кільця форми $(Z/q)[x]/\left(x^p+1\right)$, де p – є степенем 2 і $q \in 1+2pZ$ – просте. Він використовується в типових криптосистемах на основі Ring-LWE, забезпечує захист від атак сторонніми каналами, але також не рекомендується до застосування [3].

3. «NTRU Prime» та «Модифікована NTRU Prime»: в ньому також використовуються поля виду $(Z/q)[x]/\left(x^p-x-1\right)$, де p просте. Він удосконалений та використовується в «Модифікованому NTRU Prime». Обґрунтування переваг та застосування наведені в [3].

Конкретно в [3] запропоновано та використовується криптосистема з відкритим ключем «Модернізований NTRU Prime 4591^{761}», там також доводиться, що з цим полем забезпечується модель безпеки IND-CCA2 зі стійкістю з постквантовим рівнем стійкості 2^{128}. Причому, в модернізованому NTRU Prime 4591^{761} інкапсуляція (зашифрування) 256-бітного сеансового ключа вимагає 59600 циклів, а декапсуляція (розшифрування) – лише 97452 цикли.

Модернізований NTRU Prime має, порівнянно з вибором кільця NTRU, декілька переваг щодо реалізації та криптографічної стійкості. Наприклад, відсутня небажана вірогідність виникнення помилок розшифрування, яка з'являється у більшості криптосистем на решітках. Крім того модернізований NTRU Prime 4591^{761} насправді забезпечує великий запас стійкості, що важливо в перспективі, коли можуть з'явитися більш ефективні атаки.

В таблиці 6 наведено результати порівняння множення поліномів – основної операції в NTRU Prime.

Таблиця 6

Порівняння результатів множення

Рек	Константа	Цикли	Кільце	Метод
ні	Так	11722	$(Z/8192)[x]/\left(x^{701}-x-1\right)$	Karatsuba та ін.
так	Так	28682	$(Z/4591)[x]/\left(x^{761}-x-1\right)$	Karatsuba та ін.
ні	Так	31000	$(Z/12289)[x]/\left(x^{1024}+1\right)$	NTT
ні	Ні	>91056	$(Z/2048)[x]/\left(x^{743}-1\right)$	Розріджений вхід (sprase input)

Загальноприйнятим є те, що кільця наведеного типу особливо ефективні. Так, кільця NTT допускають множення за рахунок трьох теоретико-числових перетворень, тобто швидких перетворень Фур'є над скінченими полями, причому з невеликими витратами ресурсів для «множення». Якщо відмовитись від NTT та перемножувати по-іншому, то можна використати оптимізовану комбінацію з декількох частин методу Карацуби та методу Тоома. Наведено більш ефективний алгоритм множення поліномів кілець над скінченим полем.

Аналіз показав, що перехід до криптографії на решітках з простими модулями запропоновано у 2014 року [3]. В 2016 році було опубліковано чорновий варіант [3], у якому запропоновані варіанти стандартного NTRU з використанням кілець виду $A = Z[X]/(X^N - X - 1)$.

4.2. Параметри механізму NTRU Prime Ukraine

Модернізований механізм NTRU Prime Ukraine [9–10] механізм належить до криптосистем, що задаються додатними цілими (p, q, t), причому:

– p – просте ціле таке, що

$$p \geq \max\{2t, 3\}; \tag{8}$$

– q – просте ціле таке, що

$$q \geq 48t + 1; \tag{9}$$

– $x^p–x–1$ – незвідний у кільці поліномів

$$(Z/q)[x]. \tag{10}$$

Позначимо кільце виду $Z[x]/(x^p - x - 1)$, кільце з модулем 3 виду $(Z/3)[x]/(x^p - z - 1)$, та поле з модулем q $(Z/q)[x]/(x^p–x–1)$ як відповідно R, $R/3$ та R/q. Причому, вважатимемо, що елемент R є малим, якщо всі його коефіцієнти приймають значення {–1, 0, 1}.

Крім того, говоритимемо, що елемент є t-малим, якщо точно $2t$ його коефіцієнтів є ненульовими. Вага Хеммінга буде $2t$.

Предметом наших подальших досліджень є удосконалений механізм NTRU Prime Ukraine 5-го та 7-го рівнів стійкості.

4.3. Особливості генерування ключа шифрування в NTRU Prime Ukraine

Довгостроковий ключ АСШ генерує отримувач. В удосконаленому механізмі NTRU Prime Ukraine отримувач генерує ключ наступним чином.

1. Генерація рівноймовірного, випадкового «маленького» (у змісті t ненульових елементів) елементу $g \in R$. Цей крок повторюється доти, доки g буде оборотним у $R/3$.

2. Генерація постійного випадкового також t-маленького елементу – секретного ключа $f \in R$. Слід відмітити, що f – ненульовий, та оборотний у R/q, причому $t \geq 1$.

3. Обчислення відкритого ключа у вигляді

$$h=g/(3f) \text{ у } R/q. \qquad (11)$$

В (11) q – просте та більше за 3, тому 3 – оборотне в R/q, а також і $3f$ – оборотне в R/q.

4. Закодувати h у рядок $\underline{h}$. Відкритий ключ буде – $\underline{h}$.

5. Зберегти секретні поліноми f в R, та $1/g$ в $R/3$ надійним чином.

4.4. Загальні положення щодо протоколу інкапсуляції ключів

Механізм (протокол) інкапсуляції ключів подається у такій послідовності:

- побудування загальних параметрів протоколу;
- генерування асиметричних пар ключів протоколу;
- протокол інкапсуляції ключів;
- протокол декапсуляції ключів.

Побудування загальних параметрів [9, 10]. Механізм АСШ NTRU Prime Ukraine є NTRU-подібною криптосистемою, що є перспективною для застосування у постквантовий період. Загальні параметри, призначення та формули для обчислень визначені у таблицях 7, 8.

Якщо елемент R належить до $R/3$ – він є «малим», це означає, що усі його коефіцієнти належать множині $\{-1,0,1\}$.

Малий елемент є t-малим, якщо в нього рівно $2t$ коефіцієнтів є ненульовими.

Далі як параметри (n, q, t) обираються параметри з [3], які відповідають стійкості k>=200.

Генерування ключів для NTRU Prime Ukraine [8–9].

Генерування ключів для NTRU PRIME Ukraine, 5-го рівня здійснюється для параметрів згідно з таблицею 7.

Таблиця 7

Загальні параметри АСШ NTRU Prime Ukraine

Позначення	Призначення	Формула
$(Z/q)[x]$	Кільце поліномів. Кожен елемент Z/q зазвичай кодується у $\lceil \log_2 q \rceil$ біт.	
n	Порядок поліному. Визначає кількість його коефіцієнтів. Просте число, для якого поліном $x^n - x - 1$ є незвідним.	$n \geq \max\{3, 2t\}$
R	Поле поліномів $Z[x]$ з модулем $x^n - x - 1$.	$Z[x]/(x^n - x - 1)$
$R/3$	Поле поліномів $(Z/q)[x]$ з модулем $x^n - x - 1$.	$(Z/3)[x]/(x^n - x - 1)$
R/q	Поле поліномів $(Z/q)[x]$ з модулем $x^n - x - 1$.	$(Z/q)[x]/(x^n - x - 1)$
p	Менший модуль	$p = 3$
q	Більший модуль, просте число, за яким зводяться усі коефіцієнти поліному R/q.	$q \geq 48t + 3$
t	Натуральне число, кількість ненульових елементів поліному залежить від цього параметру.	$t \geq 1$
k	Рівень криптостійкості.	256, 512
m	Таємне повідомлення. Кількість 0, 1 та -1 не менше, ніж t.	$m \in R/3$
M	Повідомлення після доповнення випадкового рядка b та іншої інформації.	
$octL$	Поле для завдання довжини повідомлення	1 байт
e	Зашифроване повідомлення.	$e \in R/q$

Таблиця 8

Параметри для генерації ключа

Позначення	Призначення	Формула
$G(g)$	Випадковий малий елемент (поліном). Кількість ненульових елементів дорівнює $2n/3+1$. Є секретним параметром, що використовується для обчислення відкритого ключа.	$g \in R/3$
F	Випадковий t-малий елемент (поліном). Кількість ненульових елементів полінома дорівнює $2t$. Є секретним параметром, що використовується для обчислення секретного ключа.	

Продовження таблиці 8

Позначення	Призначення	Формула
f	Малий елемент (поліном), є секретним (особистим) ключем: f=p*F+1.	f=(1+3F)mod$q$ $f \in R/q$
h	Відкритий ключ відправника. Оборотний елемент у R/q.	h=p*G/f
$\underline{h}$	Значення h, що перетворюється в рядок октетів. Довжина $\underline{h}$ дорівнює $n\lceil \log_2 q \rceil$.	

Кількість 1 та -1 у секретному ключі та у засліплюючому поліномі (значення t) визначається n та криптостійкістю k, яку потрібно забезпечити. Значення параметрів n, q, t для режимів роботи (256/128) Додаткові параметри NTRUPrime Ukraine наведено в [9].

Додаткові параметри використовуються під час зашифрування та розшифрування, залежать тільки від загальних параметрів k, n, t, q.

Генерація асиметричних ключів. Для визначення асиметричних ключів необхідні такі параметри:

n – порядок поліному, визначає кількість його коефіцієнтів. Просте число, для якого поліном $x^n - x - 1$ є незвідним, просте число,

q – більший (великий) модуль, просте число за яким зводяться усі коефіцієнти поліному R/q,

t – натуральне число, що визначає кількість ненульових коефіцієнтів поліному.

За означенням особистим ключем є поліном f:

$$f=(1+3F)\mathrm{mod}q,\ \mathrm{F,G}\in R/3,\ ||F||_1=2t,\ ||G||=2n/3+1.$$

Вихідними даними для генерації асиметричного ключа є такі:

- особистий ключ – поліном f у R/q,
- відкритий ключ – поліном h у R/q.

$$\mathrm{mod}(q) \text{ в полі } (Z/q)[x]/(x^n - x - 1).$$

5. КРИПТОГРАФІЧНИЙ ПРОТОКОЛ ІНКАПСУЛЯЦІЇ КЛЮЧА В NTRU PRIME UKRAINE

Протокол інкапсуляції NTRU Prime Ukraine-KEM є «механізмом інкапсуляції ключа». Під час виконання протоколу інкапсуляції отримувач повинен:

1. Отримати захищеним чином у закодованому вигляді відкритий ключ, наприклад, чинний сертифікат відкритого ключа $\underline{h}$ отримувача.

2. Згенерувати t-маленький секретний ключ сеансу поліном $r \in R$ та зберегти його як секретний елемент.

3. Декодувати відкритий ключ отримувача $\underline{h}$ поліном у поліном $h \in R/q$.

4. Обчислити секретний поліном $hr \in R/q$.

5. Зробити округлення кожного коефіцієнта поліному hr, які становлять цілі числа у межах від $-(q-1)/2$ до $(q-1)/2$, до найменшого числа кратного 3. Внаслідок множення h на r та округлення отримаємо поліном – криптограму $c \in R$.

6. Здійснити кодування криптограми c у рядок $\overline{c}$ згідно з прийнятим алгоритмом кодування.

Для отримання відкритого ключа сеансу C «узгодження ключа» та секретного ключа сеансу зашифрування K здійснити гешування секретного ключа сеансу r та взяти як C старшу половину $H(r)$, а як K – молодшу половину $H(r)$ (формат little endian).

7. На основі геш-значення секретного ключа $H(r)$ сформувати відкритий ключ сеансу C та секретний ключ сеансу K. Попередньо значення r може кодуватися.

8. Засобом конкатенації формується шифротекст $C\overline{c}$, а також секретний ключ сеансу K.

9. Ключ K зберігається в подальшому як секретний та має застосовуватися для зашифрування даних, що передаються від відправника до отримувача, наприклад, під час застосування блокового чи потокового алгоритму симетричного шифрування та автентифікації.

10. Інкапсульований ключ $C\overline{c}$, що сформований відправником, має передаватися отримувачу для використання при декапсуляції ключа отримувачем та застосування ключа K у подальшому при розшифруванні зашифрованих відправником даних.

Протокол інкапсуляції ключа відправником здійснюється так:

1) формується відкритий кортеж ключа p_k та секретний кортеж ключа s_k

$$pk: = (C\overline{c}, \lambda) \text{ й } sk: = (K, pk); \qquad (12)$$

2) алгоритм зашифрування даних. При зашифруванні даних можуть використовуватися узгоджені функції, які є доступними усім абонентам. Вони включають функцію автентифікації повідомлень $MAC(\cdot,\cdot)$ та симетричний алгоритм шифрування $(Sym.Encrypt, Sym.Decrypt)$, а також функції, що наведені в [4].

Алгоритм виконується у такій послідовності:

3) надалі K використовується як E_K і M_K, де E_K – ключ для симетричної схеми шифрування і M_K – підходящий розмір ключа для схеми автентифікації повідомлення;

4) повідомлення m зашифровується з використанням симетричного шифру на ключі E_K, тобто

$$C_1: = Sym.Encrypt(m, EK); \qquad (13)$$

5) для автентифікації шифротексту C_1 на ключі автентифікації M_K, обчислюється код автентифікації повідомлення (MAC) тобто

$$C_2: = MAC(C_1, MK); \qquad (14)$$

6) на останок відправник передає отримувачу

$$C\overline{c}\ ,\ C_1, C_2\ . \qquad (15)$$

Відмітимо, що кодування c у рядок $\overline{c}$ є ще одним примітивом удосконаленого NTRU Prime Ukraine, що дозволяє за необхідності скоротити довжину до 20%.

6. КРИПТОГРАФІЧНИЙ ПРОТОКОЛ ДЕКАПСУЛЯЦІЇ КЛЮЧА В NTRU PRIME UKRAINE

Розглянемо удосконалений NTRU Prime Ukraine механізм у частині декапсуляції ключа. Отримувач отримує відкритий ключ інкапсуляції $C\overline{c}$ та захищене на сеансових ключах повідомлення C_1, C_2, тобто

$$C\overline{c}\ ,\ C_1, C_2\ . \qquad (16)$$

Безпосередньо декапсуляція шифртексту $C\overline{c}$ здійснюється у такій послідовності:

1) отримувач декодує $\overline{c}$, внаслідок чого отримує $c \in R$;

2) отримувач множить отримане значення c на $3f$ в R/q;

3) отримувач розглядає кожний коефіцієнт $3fc$ у R/q як ціле у межах з $-(q-1)/2$ до $(q-1)/2$, а далі приводить за модулем 3, та отримує поліном e у $R/3$;

4) отримувач множить поліном e на $1/g$ в $R/3$;

5) знижує e/g в $R/3$ до малого поліному $r' \in R$;

6) обчислює за аналогією як і при інкапсуляції c', C', K' з r';

7) якщо r' є t-малим, $c'=c$ та $C'=C$, то виводиться та використовується ключ симетричної автентифікації та розшифрування на ключі K'. Інакше визначається помилка і від захищеного повідомлення здійснюється відмова.

ВИСНОВКИ

1. Модернізований NTRU Prime має, порівняно з вибором кільця NTRU, декілька переваг щодо реалізації та криптографічної стійкості. Наприклад, відсутня небажана вірогідність виникнення помилок розшифрування, яка з'являється у більшості криптосистем на решітках. Крім того модернізований NTRU Prime 4591^{761} насправді забезпечує великий запас стійкості, що важливо в перспективі, коли можуть з'явитись більш ефективні атаки.

2. Основним призначенням ПІК є генерування та передача відправником інкапсульованого ключа та ключових даних отримувачу в інкапсульованому (захищеному) вигляді та декапсуляцію їх отримувачем.

3. Формальний опис механізму та протоколу інкапсуляції ключів. Попередні результати досліджень з урахуванням дозволили у найбільш загальному випадку запропонувати механізм та протокол інкапсуляції ключів.

4. Термін *досконала пряма секретність* використовується для функції протоколів узгодження ключів, які дають гарантії того, що ключі останньої сесії не будуть скомпрометовані навіть, якщо секретний ключ сервера скомпрометований.

5. Удосконалений NTRU Prime Ukraine є «механізмом інкапсуляції ключа» (KEM), він ґрунтується на алгоритмах інкапсуляції та декапсуляції секретного ключа сеансу шифрування та автентифікації.

6. Відритий ключ сеансу $C\overline{c}$ (інкапсульований ключ), що обчислений відправником, має передаватися отримувачу для використання при декапсуляції ключа отримувачем та застосування його в подальшому.

7. Модернізований NTRU Prime має, порівняно з вибором кільця NTRU, декілька переваг щодо реалізації та криптографічної стійкості. Крім того модернізований NTRU Prime 4591^{761} насправді забезпечує великий запас стійкості, що важливо в перспективі, коли можуть з'явитись більш ефективні атаки.

Література

[1] American National Standard X9.98-2010. Lattice-based polynomial public key encryption algorithm, Part 1: key establishment, Part 2: data encryption. – 2010.

[2] ISO/IEC 18033-2:2015. Інформаційні технології. методи захисту. Алгоритми шифрування. Частина 2. Асиметричні шифри.

[3] «Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU Prime [Electronic resource]. – Access mode: https://ntruprime.cr.yp.to/ntruprime-20160511.pdf».

[4] [Проект ДСТУ ____:201_ Інформаційні технології Криптографічний захист інформації Алгоритм асиметричного шифрування NTRU Prime Ukraine.

[5] *Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman*. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings, volume 1423 of LNCS, pages 267–288. Springer, 1998.

[6] Циркулянт. – Електронний ресурс. – Режим доступу: https://uk.wikipedia.org/wiki/Циркулянт.

[7] *Горбенко І. Д.* Прикладна криптологія: Підручник / Горбенко І. Д., Горбенко Ю. І.; видання 2-ге. – Харків: Форт, 2013. – 878 с.

[8] *I. D. Gorbenko, O. G. Kachko, M. V. Yesina* Analysis of Asymmetric NTRU Prime IIT Ukraine Encryption Algorithm with Regards to Known Attacks, Telecommunications and Radio Engineering, Volume 77, 2018, Issue 9, pp. 799–816.

[9] *Горбенко І. Д.* Методи побудування загальних параметрів та ключів для NTRU Prime Ukraine 5-7 рівнів стійкості / І. Д. Горбенко, О. Г. Качко, Ю. І. Горбенко, І. В. Стельник, С. О. Кандій, М. В. Єсіна // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харьков: ХТУРЕ. – 2018. – Вып. 195. – С. 5–16.

[10] *I.D. Gorbenko, A.M. Oleksiychuk, O.H. Kachko, M.V. Yesina, V.A. Bobukh, S.O. Kandyi*. Calculation of general parameters for ntru prime ukraine of 6–7 levels of stability // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харьков: ХНУРЕ. – 2018. – Вып. 195. – С. 17–26.

Надійшла до редколегії 25.12.2018

Горбенко Іван Дмитрович, д-р. техн. наук, професор, професор кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – криптографія, криптоаналіз, постквантова криптографія, захист інформації.

Качко Олена Григорівна, канд. техн. наук, професор кафедри ПІ ХНУРЕ. Галузь наукових інтересів – криптографія, криптоаналіз, паралельні обчислення.

Пономар Володимир Андрійович, канд. техн. наук, науковий співробітник Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – криптографічні перетворення, безпечне програмування, методи багатофакторної автентифікації та їх застосування з метою захисту інформації, захист криптографічних засобів інформації.

Єсіна Марина Віталіївна, канд. техн. наук, старший викладач кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – захист інформації, постквантова криптографія.

Акользіна Ольга Сергіївна, науковий співробітник НДЧ кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – асиметричне шифрування, механізми інкапсуляції ключів, постквантова криптографія.

Кулібаба Владислав Андрійович, аспірант кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – моделі безпеки, протоколи інкапсуляції ключів, постквантова криптографія.

УДК 004.056.55

Горбенко И. Д. **Анализ сущности и модели протокола инкапсуляции ключей в кольце полиномов над конечным полем** / И. Д. Горбенко, Е. Г. Качко, В. А. Пономарь, М. В. Есина, О. С. Акользина, В. А. Кулибаба // Прикладная радиоэлектроника: науч.-техн. журнал. – 2018. – Том 17. № 3, 4. – С. – 127–137.

В работе рассматривается анализ сущности и модели протокола инкапсуляции ключей в кольце полиномов над конечным полем. Приводятся основные положения относительно протоколов. Приводятся результаты сравнения механизмов инкапсуляции ключей. Приводится криптографический протокол инкапсуляции и декапсуляции ключа в NTRU Prime Ukraine.

Ключевые слова: инкапсуляция ключей, кольцо полиномов, протокол.

Табл.: 8. Библиогр.: 10 назв.

UDC 004.056.55

Gorbenko I. D. **Analysis of the essence and models of the key encapsulation protocol in a polynomial ring over a finite field** / I. D. Gorbenko, E. G. Kachko, V. A. Ponomar, M. V. Yesina, O. S Akolzina, V. A. Kulibaba // Applied Radio Electronics: Sci. Journ. – 2018. Vol. 17. – № 3, 4. – P. 127–137.

The paper considers the analysis of the essence and models of the key encapsulation protocol in a polynomials ring over a finite field. The main provisions on the protocols are given. The results of the key encapsulation mechanisms comparison are presented. A cryptographic protocol of key encapsulation and decapsulation in NTRU Prime Ukraine is given.

Keywords: key encapsulation, polynomial ring, protocol.

Tab.: 8. Ref.: 10 items.

УДК 003.026:004.056

ПОРІВНЯННЯ КАНДИДАТІВ ЕЛЕКТРОННОГО ПІДПИСУ НА ПОСТКВАНТОВИЙ СТАНДАРТ NIST PQC НА БАЗІ MQ-ПЕРЕТВОРЕНЬ ТА ФУНКЦІЙ ҐЕШУВАННЯ

Ю. І. ГОРБЕНКО, І. С. КУДРЯШОВ, Д. С. НАУМЕНКО, В. В. ОНОПРІЄНКО

Наводяться результати порівняльного аналізу кандидатів на стандарти перспективних електронних підписів, що будуються на основі мультиваріативних квадратичних перетворень та функцій ґешування. Результати аналізу отримані в ході використання методики порівняння криптографічних механізмів на основі експертних оцінок за сукупністю умовних та безумовних критеріїв. Зроблено рекомендації щодо перспектив застосування кандидатів.

Ключові слова: MQ-перетворення, постквантовий алгоритм, електронний підпис, порівняльний аналіз, експертні оцінки, підпис на основі ґеш-функцій.

ВСТУП

Наприкінці 2016 року NIST США оголосив конкурс на нові стандарти постквантової асиметричної криптографії [1], зокрема, механізми електронного підпису (ЕП), направленого шифрування (НШ) та протоколи інкапсуляції ключів (ПІК). Необхідність їх розробки викликана суттєвим розвитком квантових обчислень – математичних квантових методів та квантових комп'ютерів, що можуть бути застосованими для криптоаналізу асиметричних криптоперетворень [2–22].

Серед поданих на конкурс кандидатів на стандарт ЕП значне число розроблено на основі застосування мультиваріативних квадратичних перетворень (Multivariate Quadratic Transformations, MQ-transformations) [2–10]. Перше за все механізми MQ-перетворень дозволяють забезпечити необхідні рівні стійкості, швидкодію та застосування в малоресурсних системах, а також можуть застосовуватися у загальному випадку. Властивості MQ-перетворень мають суттєве значення для практичних додатків, тому їхній аналіз та порівняння є важливою проблемною задачею, тим більше що вона вирішується NIST США на міжнародному рівні. Аналіз показав, що на конкурс NIST було подано 9 кандидатів ЕП на основі MQ-перетворень, а саме: LUOV [2], Gui [3], Rainbow [4], MQDSS [5], TPSig [6], DualModeMS [7], HiMQ-3 [8], GeMSS [9] та DME [10].

Також теоретичне та практичне визнання, як кандидати на стандарт отримали ЕП, що будуються на основі функцій ґешування та дереві Мерклі. Але проблемою є те, що реалізація таких криптосистем ЕП вимагає для створення нової інфраструктури відкритого ключа. Як показав аналіз конкурентними як кандидати на ЕП є Gravity-SPHINCS [11] та SPHINCS$^+$ [12].

Зрозуміло, що при такій значній наявності кандидатів на постквантовий стандарт ЕП, неохдно проводити їх порівняння за значною кількістю безумовних та умовних критеріїв [20–22].

Метою цієї статі є порівняльний аналіз кандидатів на постквантові стандарти ЕП, як всередині груп ЕП на основі певних математичних методів, так між ними, в даному випадку, що ґрунтуються на застосуванні мультиваріативних квадратичних перетворень та гешфункцій.

Таким чином, у цій статті наводяться початкові результати порівняльного аналізу кандидатів на постквантові стандарти ЕП. Під час досліджень за основу вибрані джерела [1–19], а також наша стаття [20].

1. СУТНІСТЬ ТА ЗАГАЛЬНА ХАРАКТЕРИСТИКА MQ-МЕХАНІЗМІВ

Серед кандидатів на асиметричні перетворення типу АСШ, ЕП та ПІК 10 грунтуються на механізмах багатовимірних MQ-перетворень [1–11, 20]. Аналіз показує, що багатовимірна MQ криптографія ґрунтується на складності вирішення задач, які пов'язані з багатовимірними поліномами над кінцевими полями та вирішенням систем багатовимірних поліноміальних рівнянь. Основними особливостями MQ-перетворень є невеликі, порівняно з іншими, складність асиметричних перетворень та невеликі обчислювальні ресурси здійснення перетворень. Як наслідок, вказане дозволяє реалізувати MQ-перетворення у відносно простих засобах ЕП.

Розглянемо сутність MQ-перетвореннння. Нехай F_q є скінченне поле з q елементами. Також нехай система мультиваріативних квадратичних поліномів $P=(P^{(1)},...,P^{(m)})$, з m рівняннями та n змінними визначена як:

$$P^{(k)}(x_1,...,x_n)=\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}^{(k)}x_ix_j+\sum_{i=1}^{n}\beta_i^{(k)}x_i+\alpha_0^{(k)},$$

$$k=1...m,\gamma_{ij}^{(k)},\beta_i^{(k)},\alpha_0^{(k)}\in F_q. \quad (1)$$

Основна ідея для конструкції MQ-схем полягає у тому, що необхідно обрати секретну систему $F=(F^{(1)},...,F^{(m)}):F_q^n\to F_q^m$ (так зване центральне відображення), яка складається з m мультиваріативних

квадратичних поліномів, n змінних, яка може бути інвертована з поліноміальною складністю.

Для того, щоб сховати структуру центрального відображення F у публічному ключі, необхідно також обрати два афінних лінійних відображення $S: F_q^m \to F_q^m$ та $T: F_q^n \to F_q^n$. Як публічний ключ використовується композиція квадратичних відображень $P = S \circ F \circ T$, яку важко відрізнити від випадкової системи і тому складно інвертувати. Як приватний ключ використовується сукупність відображень (S, F, T), знаючи які можна інвертувати публічний ключ P.

Послідовність (схема) генерації та перевірки ЕП [8], що базується на MQ-перетвореннях, наведено на рис. 1.

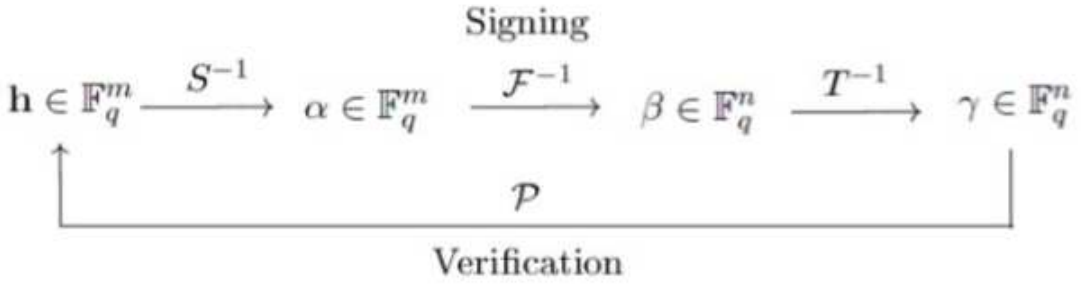

Рис. 1. Схеми створення та перевірки підпису на основі MQ-схеми

2. СУТНІСТЬ ТА ЗАГАЛЬНА ХАРАКТЕРИСТИКА ЕП НА ОСНОВІ ФУНКЦІЙ ҐЕШУВАННЯ

Спираючись на роботу Лемпорта [13], Діффі та Геллман запропонували одну з найпростіших схем підпису на основі ґеш-функцій[14–18]. В схемі задано параметр безпеки n та однонаправлену функцію $F: \{0,1\}^n \to \{0,1\}^n$. Сама схема використовується для підписання одного біту. Секретний ключ складається з випадкових значень $x_0, x_1 \in \{0,1\}^n$. Відкритий ключ складається з ґеш значень елементів секретного ключа – $(y_0, y_1) \coloneqq (F(x_0), F(x_1))$. Підпис σ біту b складається з відповідного значення секретного ключа: $\sigma = x_b$. Перевірка підпису виконується шляхом визначення ґеш значення підпису та перевірки виконання умови $y_b = F(\sigma)$.

Авторами роботи було також запропоновано використовувати m реалізацій описаної вище схеми для підписання повідомлення довжиною m біт. За допомогою такої схеми неможливо підписати повідомлення, довжина якого більше m біт. Для вирішення цієї проблеми було запропоновано таку конструкцію: стійка до колізій ґеш функція H, з довжиною вихідного значення m біт, застосовується до повідомлення M, у результаті чого отримується ґеш значення $h = H(M)$ довжиною m біт. Отримане значення підписується за допомогою схеми, яку було описаною вище.

Така схема є одноразовою, в ній кожна ключова пара може використовуватися для підписання лише одного повідомлення. Іншим прикладом схеми одноразового підпису є підпис Вінтерніца.

Основна ідея схеми одноразового підпису Вінтерніца (Winternitz one-time signature scheme – WOTS) вперше була запропонована Мерклєм. Базуючись на його роботі Вінтерніц удосконалив схему. Для n-бітного простору повідомлень обираються параметри ℓ та w такі, що $\ell \cdot \log_2 w = n$. Секретний ключ схеми є собою ℓ n-бітними рядками $(s_1, \dots, s_\ell)$, а відкритим ключем є $\left(F^{w-1}(s_1), \dots, F^{w-1}(s_\ell)\right)$, де F^{w-1} означає застосування функції F до секретного ключа $w-1$ раз. Цю конструкцію можна розглядати як ℓ ланцюгів, кожен з яких має довжину $w-1$. Для підписання повідомлення x, яке розбито на ℓ блоків довжиною $\log_2 w$ біт $(x_1, \dots, x_\ell)$, підписувач обчислює $\left(F^{x_i}(s_1)\right)_{1 \le i \le \ell}$. Перевірка виконується шляхом обчислення $F^{w-1-x_i}(y_i)$ для кожного елементу підпису y_i, та порівняння результату з відповідним елементом відкритого ключа $F^{w-1}(s_i)$.

Існують декілька варіантів ланцюгової функції F^i: WOTS^{CR}, WOTS^{PRF}, WOTS^{+}. В [14] було запропоновано WOTS^{+}, де в кожній ітерації використовується випадкова маска r_i, тобто $F_K^0(x) = x, F_K^i = F_K\left(F_K^{i-1}(\cdot) \oplus r_i\right)$. Ключ ПВГ K та маски $(r_1, \dots, r_w)$ є частиною відкритого ключа. Перевагою WOTS^{+} є те, що стійкість до колізій функції $\mathcal{F}_n$ не є обов'язковою, а достатньо стійкості до колізій та псевдовипадковості.

На практиці необхідно мати можливість створювати набагато більше підписів. У реальному житті ця цифра може досягати до 2^{50} підписів повідомлень за допомогою однієї ключової пари. Для конкурсу, NIST вимагає підписання 2^{64} повідомлень однієї ключовою парою.

Одним з варіантів створення схеми багаторазового підпису з схеми одноразового підпису є використання конструкції, запропонованої Мерклі в [15]. При заданих цілих чисел n, h та ґеш функції $H: \{0,1\}^{2n} \to \{0,1\}^n$, деревом Мерклє є двійкове дерево висотою h,

чиї вузли є $x \in \{0,1\}^n$, а значення вузла обчислюється як $x = H(y||z)$, де y та z є лівою та правою дитиною вузла відповідно. Листям дерева є значення особистого ключа. Корінь дерева r може публікуватися для подальшої автентифікації будь-якого з 2^h листів $v_1, \dots, v_{2^h}$. Для підтвердження того, що значення v є i-м листом, необхідно мати v, i та шлях автентифікації. Шлях автентифікації складається з усіх вузлів-сестер на шляху від i-го листа до кореня (всього h значень). Він дозволяє рекурсивно обчислити значення всіх внутрішніх вузлів до самого кореня, та порівняти його з r. Приклад шляху автентифікації для автентифікації i-го листа зображено на рисунку 2. Тобто шляхом автентифікації є сукупність вузлів, що зображені на рисунку сірим кольором.

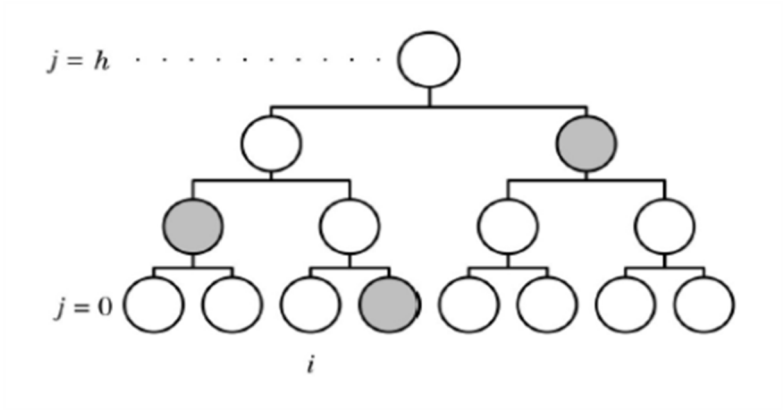

Рис. 2. Приклад шляху автентифікації i-го листа дерева Мерклє

У [16] Голдріх презентував конструкцію, яка базується на використанні двійкового дерева одноразових підписів. Для реалізації запропонованої схеми потрібно не просто ґешувати значення разом (як в стандартній конструкції дерева Мерклє), а замість цього прикріпляти ключову пару до кожного вузла дерева та використовувати її для підписання дочірніх вузлів. У цьому випадку немає необхідності повністю обчислювати дерево. Для цього необхідно, щоб ключі вузлів разом зі шляхом від випадкового вузла до кореня, були детерміновано згенерованими залежно від порядку. Це може бути досягнуто завдяки використанню псевдовипадкової функції, яка приймає на вхід секретне початкове значення та індекс вузла.

Хоча конструкція Голдріха дозволяє відмовитись від збереження стану, вона є досить неефективною. Покращеною версією такого підпису є конструкція SPHINCS [17].

По-перше, для листів дерева замість OTS (One Time Signature) використовується FTS (Few Time Signature), що дозволяє зменшити ймовірність виникнення колізії шляхів та зменшити висоту дерева. По-друге, внутрішні вузли дерева замінюються деревами Мерклє. Кожне з таких дерев підписує 2^h дітей, замість 2. Таким чином формується гіпер-дерево. За допомогою використання такої конструкції зменшується необхідний на генерацію підпису час та розмір підпису, адже до самого підпису входить менша кількість реалізацій OTS.

«Віртуальна» структура схеми SPHINCS повністю визначається ключовою парою . Основним елементом схеми є гіпер-дерево, яке має висоту h. Це дерево складається з d рівнів, кожен з яких складається з дерев висотою h/d. Кожне з цих дерев виглядає наступним чином. Листи дерев є $2^{h/d}$ коренями двійкового дерева. Кожен з коренів стискає відкритий ключ ключової пари WOTS$^+$. Тобто, дерево може розглядатися як ключова пара, кожна з яких може бути використана для підписання $2^{h/d}$ повідомлень. Всього а гіпер-дерево d рівнів. Рівень $d-1$ складається з одного дерева, рівень $d-2$ складається з $2^{h/d}$ дерев. Корені дерев на цьому підписуються за допомогою ключових пар WOTS$^+$ дерева на рівні $d-1$. В загальному випадку рівень i складається з $2^{(d-1-i)\binom{h}{d}}$ дерев, і, відповідно корені дерев на даному рівні підписуються за допомогою ключових пар WOTS$^+$ дерев на рівні $i+1$. На рівні 0 кожна з ключових пар WOTS$^+$ використовується для підписання відкритого ключа схеми HORST (модифікація схеми HORS (Hash to Obtain Random Subset) [18], яка була запропонована в [17]). Такою є так звана «віртуальна» структура схеми SPHINCS. Вона так називається через те, що всі значення встановлюються вибором початкового значення і біт-масок, а дерево повністю ніколи не обчислюється. Початкове значення є частиною секретного ключа і використовується для псевдовипадкової генерації ключів [17]. На рисунку 3 зображений один шлях в гіпер-дереві.

3. АНАЛІЗ КАНДИДАТІВ ЩОДО БЕЗУМОВНИХ КРИТЕРІЇВ

Для конкурсу NIST PQC було розроблено методику порівняння механізмів ЕП [20, 21], які мають протистояти загрозам постквантового періоду. При цьому кожний з алгоритмів ЕП має відповідати певним безу-

мовним критеріям, які наведено у таблиці 1. Ці критерії є обов'язковими, тобто якщо хоча б один з них не задовольняється, то кандидат відкидається.

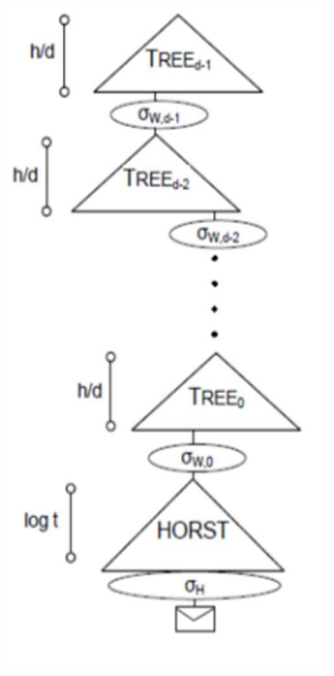

Рис. 3. Віртуальна структура підпису SPHINCS

Відповідно до прийнятих безумовних критеріїв проведено аналіз відповідності кожного з наведених алгоритмів ЕП, які приймають участь у конкурсі NIST PQC за такими умовами:

1. Наведені критерії вимагають чіткої відповідності, тому критерієм добору є логічна зміна так або ні (1 або 0). Тобто безумовний критерій можна подати у математичному поданні з урахуванням:

$$(W_{б1}, W_{б2}, W_{б3}, W_{б4}, W_{б5}, W_{б6}, W_{б7}) \in (1,0). \quad (2)$$

2. Використовуючи правило (2) функцію відповідності алгоритму вимогам, що викладені в таблиці 1, можна подати у вигляді інтегрального безумовного критерію:

$$W_б = W_1 \wedge W_2 \wedge W_3 \wedge W_4 \wedge W_5 \wedge W_6 \wedge W_7 .$$

Тобто, якщо $W_б$ відповідає значенню 0, то можна стверджувати, що криптоперетворення не відповідає безумовним критеріям, якщо 1 – то навпаки, відповідає.

Відповідно до поданих критеріїв проведений аналіз механізмів електронного підпису, який наведено у таблицях 2, 3.

Таблиця 1

Безумовні критерії оцінки постквантових криптографічних перетворень типу електронного підпису (ЕП)

№	Безумовні критерії	Позначення
1	Надійність, простота та прозорість математичної бази (математичних перетворень), що застосовуються в ході реалізації постквантових криптоперетворень ЕП.	$W_{б1}$
2	Практична захищеність криптоперетворення типу ЕП від відомих атак з використанням квантового комп'ютера та доступу криптоаналітика до 2^{64} обраних повідомлень, для моделі безпеки EUF – CMA	$W_{б2}$
3	Обгрунтованість реальної захищеності (стійкості) криптоперетворень типу ЕП від усіх відомих та потенційно можливих криптоаналітичних атак постквантового періоду на основі використання загальних параметрів та ключів з необхідними розмірами та властивостями (ключі 128 біт та більше класичної стійкості(безпеки)), включаючи статистичну безпеку.	$W_{б3}$
4	Теоретична захищеність криптографічних перетворень типу ЕП у постквантовий період проти існуючих силових, аналітичних та спеціальних атак для діючих моделей загроз (мінімум для моделі EUF – CMA для ЕП).	$W_{б4}$
5	Можливість заміни існуючих стандартизованих криптопримітивів на постквантові та застосування в діючих криптографічних системах та протоколах у певних умовах та обмеженнях.	$W_{б5}$
6	Обчислювальна ефективність – складність прямого $I_{пр}$ та зворотного $I_{зв}$ криптографічних перетворень ЕП, а також генерування асиметричних пар ключів $I_{кл}$ не вище за поліноміальну, забезпечення необхідних значень складності (швидкодії) $I_{пр}$, $I_{зв}$ та $I_{кл}$ при практичному застосуванні в додатках з апаратно-програмною та програмною їх реалізацією.	$W_{б6}$
7	Виконання обмежень на мінімальну та максимальну довжини особистих та відкритих ключів, розміри та збитковість ЕП, відсутність слабких особистих ключів для моделей безпеки постквантового періоду.	$W_{б7}$

У поданій специфікації DME[10] описані практичні досліди щодо безпеки алгоритму, але немає жодних уявлень щодо теоретичної захищеності від усіх відомих атак. Цей факт викликає підозру стосовно реальної захищеності механізму і відносно заявленого досягнутого рівня безпеки [19].

Відносно HiMQ3[8] наразі було знайдено недолік у доказі безпеки EUF-CMA[6], тому можливо визначити чи дійсно цей механізм задовольняє практичну та реальну захищеності відносно відомих та потенційно можливих криптоаналітичних атак.

Запропонований механізм TPSig[6] був відкликаний з конкурсу через те, що має лінійну складність відновлення секретного ключа[6]. Тому можна вважати, що три вищезгадані механізми отримали незадовільнений результат згідно з перевіркою на відповідність безумовним інтегральним критеріям.

Таблиця 2

Оцінка постквантових криптографічних перетворень типу електронного підпису (ЕП) на базі MQ-перетворень на відповідність безумовним критеріям

Scheme	$W_{\delta 1}$	$W_{\delta 2}$	$W_{\delta 3}$	$W_{\delta 4}$	$W_{\delta 5}$	$W_{\delta 6}$	$W_{\delta 7}$	W_{δ}
TPSig	1	0	0	1	1	1	1	0
HiMQ3	1	0	0	1	1	1	1	0
DME	1	1	0	1	1	1	1	0
LUOV	1	1	1	1	1	1	1	1
GUI	1	1	1	1	1	1	1	1
Rainbow	1	1	1	1	1	1	1	1
MQDSS	1	1	1	1	1	1	1	1
DualModeMS	1	1	1	1	1	1	1	1
GeMSS	1	1	1	1	1	1	1	1

Таблиця 3

Оцінка постквантових криптографічних перетворень типу електронного підпису (ЕП) на основі функцій гешування

Scheme	$W_{\delta 1}$	$W_{\delta 2}$	$W_{\delta 3}$	$W_{\delta 4}$	$W_{\delta 5}$	$W_{\delta 6}$	$W_{\delta 7}$	W_{δ}
Gravity-SPHINCS	1	1	1	1	0	1	1	0
SPHINCS^{+}	1	1	1	1	0	1	1	0

Надалі пропонується аналіз алгоритмів, які отримали позитивний результат перевірки, відносно умовних критеріїв та інтегрального умовного критерію, тобто LUOV[2], GUI[3], Rainbow[4], MQDSS[5], DualModeMs[7] та GeMSS[9].

За результатами, які наведено в таблиці 3 можна побачити, що запропоновані схеми відповідають всім умовним критеріям, окрім критерію $W_{\delta 5}$, який вимагає можливості заміни існуючих стандартизованих криптопримітивів на нові. Це є одним з основних недоліків постквантових схем електронного підпису на основі функцій гешування, адже для того, щоб застосовувати такі схеми, необхідно переробляти систему відкритого ключа. Але такі схеми все одно можуть знайти застосування, наприклад, у закритих спеціалізованих системах.

4. ПОРІВНЯННЯ АЛГОРИТМІВ ЩОДО УМОВНИХ КРИТЕРІЇВ

Для аналізу можливості використання тих чи інших алгоритмів, необхідно оцінити можливості кожного, а також оцінити перевагу одного алгоритму над іншим. Таку задачу можна вирішити за допомогою порівняння алгоритмів із застосуванням методу вагових коефіцієнтів. Далі наведено порівняння умовних оцінок алгоритмів.

Умовними оцінками виступали такі характеристики алгоритмів:

1) $I_{ст.}$ – рівень криптографічної стійкості;
2) lв.к – довжина відкритого ключа;
3) lо.к – довжина особистого ключа;
4) lрез. – довжина результату криптоперетворення (підпис);
5) Ткл. – швидкість створення ключової пари;
6) Тпр. – швидкість прямого криптоперетворення (створення підпису);
7) Тзв. – швидкість зворотного криптоперетворення (перевірка підпису).

Для поданих характеристик застосовується оцінка важливості, яку виставляють експерти в області криптографії. Числова шкала оцінки, числові значення, та вагові коефіцієнти важливості характеристик наведені у таблиці 4.

Таблиця 4

Експертні оцінки характеристик криптоалгоритмів методом ранжування

Показники / Експерти	Іст.	lв.к	lо.к	lрез.	$T_{кл.}$	Тпр.
1	7	5	3	2	1	4
2	6	7	1	3	2	4
3	5	6	1	2	3	4
4	5	6	1	4	2	3
5	6	2	1	4	3	5
W	**0,207**	**0,186**	**0,05**	**0,107**	**0,079**	**0,143**

Безпосередньо на конкурс NIST PQC було подано безліч механізмів із різними модифікаціями, тому було прийнято рішення для алгоритмів на базі MQ-перетворень порівнювати лише алгоритми, які мають позитивний інтегральний безумовний критерій. Для кожного такого алгоритму була обрана модифікація, яка відповідно до специфікації гарантує найвищий рівень безпеки.В таблицях 5, 6 наведено характеристики обраних для порівняння алгоритмів електронного підпису.

Всі значення розмірів ключів та підписів у таблицях 5 та 6 наведено в байтах, а швидкості виконання операцій – в циклах процесора. Для того, щоб проаналізувати ефективність виконуваних обчислень описаних алгоритмів, швидкість виконання операцій вимірюється в кількості циклів, які необхідно виконати процесору для реалізації операції. Цей показник залежить від характеристик системи, яка використовується для виконання операції. Обчислювальні можливості процесора вимірюються в Гц, що еквівалентно секунді в -1 ступені. Таким чином, значення кількості циклів, необхідних для виконання будь-якої операції можна отримати за формулою:

$$n = \nu \cdot t,$$

де n – кількість циклів процесора, ν – частота процесора, t – час виконання операції.

Таблиця 5

Характеристики алгоритмів електронного підпису на базі MQ-перетворень

Схема	$\mathbf{I_{ст}}$	$\mathbf{l_{в.к}}$	$\mathbf{l_{о.к}}$	$\mathbf{l_{рез}}$	$\mathbf{T_{кл.}}$	$\mathbf{T_{пр}}$	$\mathbf{T_{зв}}$
LUOV	256	100 967	32	521	276 912 036	144 203 736	84 564 465
GUI	256	5 928 141	159 642	83	239 502 000 000	872 949 000 000	2 004 155
Rainbow	256	1 351 373	892 109	118	45 064 000 000	3 916 000	2 897 000
GeMSS	256	3 603 792	82 056	104	18 174 000 000	12 740 000 000	160 420 000
MQDSS	192	88	48	67 800	6 680 606	776 183 461	571 665 382
DualModeMS	128	528	18 038 184	32 640	2 072 200 000 000	6 006 000 000	6 994 000

Таблиця 6

Характеристики алгоритмів електронного підпису на основі функцій ґешування

Схема	$\mathbf{I_{ст}}$	$\mathbf{l_{в.к}}$	$\mathbf{l_{о.к}}$	$\mathbf{l_{рез}}$	$\mathbf{T_{кл.}}$	$\mathbf{T_{пр}}$	$\mathbf{T_{зв}}$
Gravity-SPHINCS S	128	32	65 536	12 640	781 646 000	17 964 000	104 000
Gravity-SPHINCS M	128	32	262 144	28 929	24 229 712 000	18 900 000	252 000
Gravity-SPHINCS L	128	32	131 072	35 168	11 789 080 000	21 054 000	338 000
SPHINCS+ 128s	128	32	64	8 080	917 405 356	16 992 635 344	19 360 272
SPHINCS+ 128f	128	32	64	16 976	28 814 020	1 056 761 824	45 964 624
SPHINCS+ 192s	192	48	96	17 064	1 244 530 184	38 062 259 596	27 243 200
SPHINCS+ 192f	192	48	96	35 664	42 782 840	1 276 694 620	69 760 728
SPHINCS+ 256s	256	64	128	29 792	1 817 324 180	28 860 355 888	42 380 420
SPHINCS+ 256f	256	64	128	49 216	113 876 252	3 172 247 452	76 203 004

На рисунках 4 та 5 відображено гістограми відносної переваги алгоритмів, що базуються на MQ-перетвореннях та функціях ґешування.

Як видно з рисунку 4, серед алгоритмів на базі MQ-перетворень, відповідно до проведеної методики, кращими алгоритмами є Rainbow та LUOV. Останні місця займають MQDSS та DualModeMS. Серед алгоритмів, які базуються на функціях ґешування, кращими є Gravity-SPHINCS S, Gravity-SPHINCS M та SPHINCS$^+$ 128s. Останні місця займають SPHINCS$^+$ 192f та SPHINCS$^+$192s.

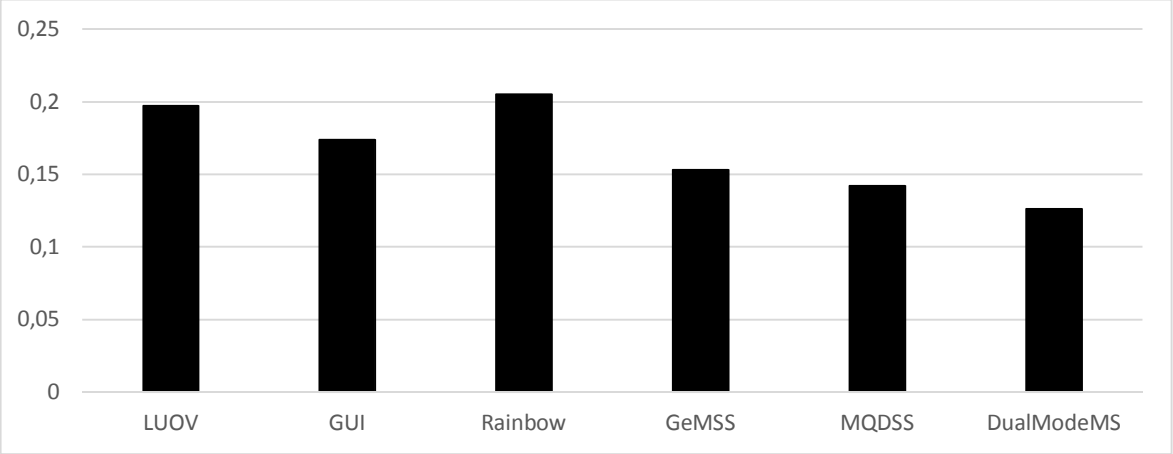

Рис. 4. Відносна перевага алгоритмів ЕП на базі MQ перетворень

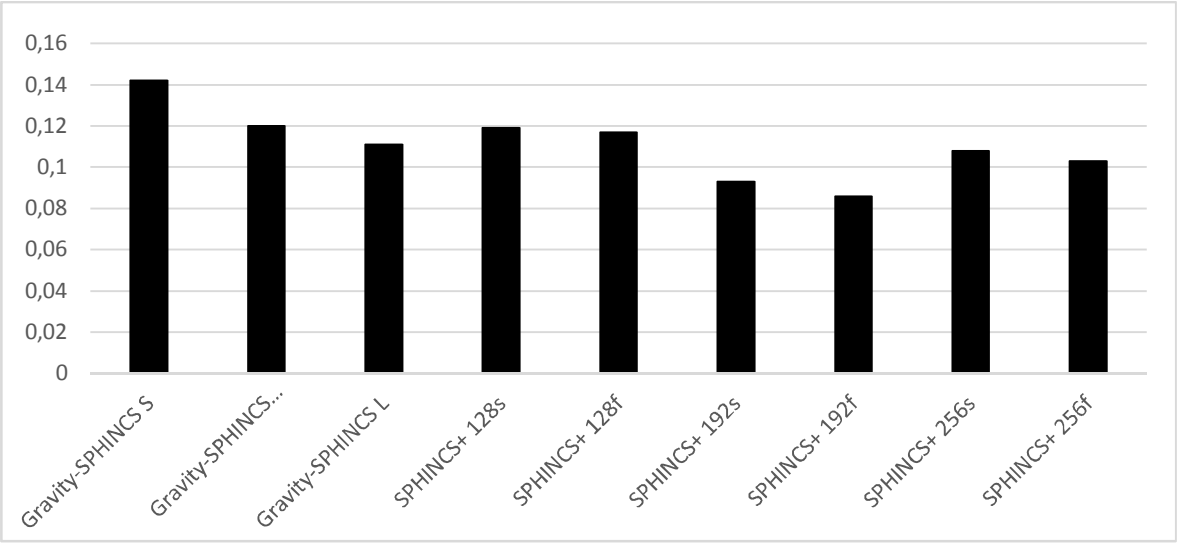

Рис. 5. Відносна перевага алгоритмів ЕП на основі функцій ґешування

5. ПОРІВНЯННЯ КРАЩИХ АЛГОРИТМІВ ВІДНОСНО УМОВНИХ КРИТЕРІЇВ

Далі наводиться порівняння найперспективніших алгоритмів ЕП, які було проаналізовано в розділі 4. Цими алгоритмами є LUOV, Rainbow. Gravity-SPHINCS S та SPHINCS$^+$ 128s. Алгоритм Gravity-SPHINCS M не було взято до порівняння, через те, що він є ще однією модифікацією алгоритму Gravity-SPHINCS, та має не дуже велику перевагу над SPHINCS$^+$ 128s.

Для порівняння було взято ті ж характеристики, а також показники з таблиць 4, 5, 6.

На рисунку 6 наведено результати цього порівняння.

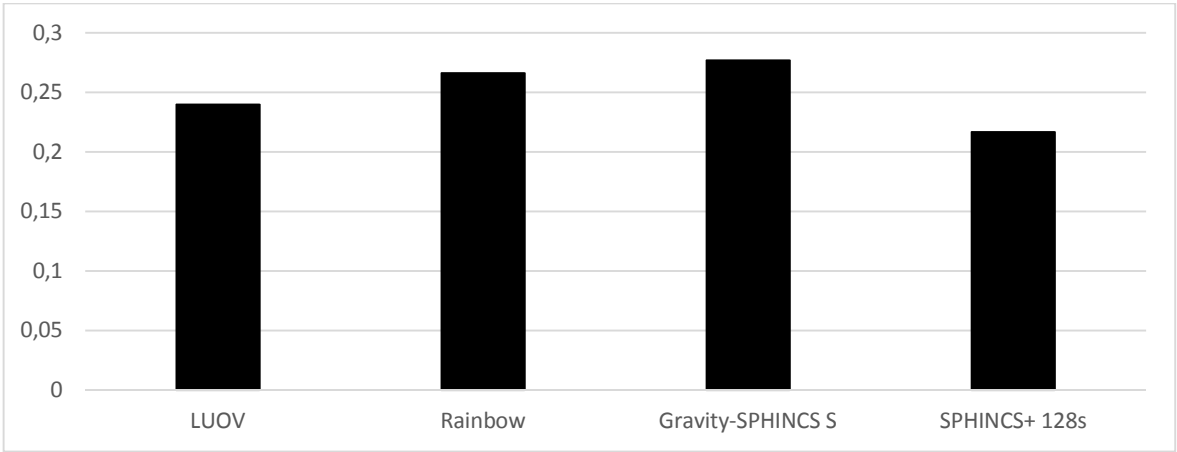

Рис. 6. Відносна перевага алгоритмів ЕП

Як видно з рисунка 6, алгоритм Gravity-SPHINCS S має найвищий показник, отже, є найперспективнішим з описаних алгоритмів. На другому місці є Rainbow. Як було зазначено вище, описані алгоритми на основі функцій ґешування, не можуть використовуватись як заміна існуючим національним стандартам, через необхідність перебудування існуючої інфраструктури відкритого ключа. Отже, алгоритм Gravity-SPHINCS S є дуже перспективним, але може використовуватися лише в спеціалізованих закритих системах, а алгоритм Rainbow є гарним кандидатом для використання на національному рівні.

ВИСНОВКИ

1. Первинний аналіз кандидатів, що представлені NIST США на конкурс постквантової криптографії, зроблено з використанням техніко-економічних показників, а саме розміру публічного та приватного ключа, рівнів криптографічної стійкості ЕП, розміру підпису, складності (швидкодії) генерування ключової пари, складності (швидкодії) обчислення та перевірки ЕП.

2. Значне число кандидатів на стандарт ЕП розроблено на основі застосування мультиваріативних квадратичних перетворень (Multivariate Quadra-tic Transformations, MQ-transformations). Механізми MQ-перетворень дозволяють забезпечити необхідні рівні стійкості, швидкодію та застосування в мало-ресурсних системах, а також можуть застосовуватися у загальному випадку.

3. Шість з дев'яти кандидатів на ЕП - LUOV, Rainbow, GUI, GeMSS, MQDSS, DualModeMS алгоритмів, що базуються на MQ-перетвореннях, відповідають безумовним критеріям.

4. Стосовно алгоритмів на основі функції ґешування слід зазначити, що такі алгоритми практично уже відповідають усім представленим безумовним критеріям. Проблемним, з точки зору складності, є реалізація сертифікації відкритих ключів.

5. Кандидати на стандарт Rainbow та LUOV мають найбільшу перевагу серед алгоритмів на базі MQ-перетворень, тому їх можна важати найбільш перспективними.

6. Алгоритм Gravity-SPHINCS S виявився найкращим в результаті порівняння відносно умовних критеріїв.

7. Як кандидат на національний стандарт найперспективнішим з описаних є алгоритм Rainbow.

8. Проект Gravity-SPHINCS S має найвищий показник, отже, є найперспективнішим з описаних алгоритмів. На другому місці є Rainbow.

Література

[1] Post-Quantum Cryptography, Round 1 Submissions, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[2] *Ward Beullens, Bart Preneel, Alan Szepieniec, Frederik Vercauteren.* LUOV: Lifted Unbalanced Oil and Vinegar, NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[3] *Jintai Ding, Ming-Shen Chen, Albrecht Petzoldt, Dieter Schmidt, Bo-Yin Yang.* Gui, NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[4] *Jintai Ding, Ming-Shen Chen, Albrecht Petzoldt, Dieter Schmidt, Bo-Yin Yang.* Rainbow. NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[5] *Simona Samardjiska, Ming-Shing Chen, Andreas Hulsing, Joost Rijneveld, Peter Schwabe.* MQDSS, NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[6] *Joseph Peretz, Nerya Granot.* TPSig, NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[7] *J.-C. Faugère, L Perret, J Ryckeghem.* DualModeMS: A Dual Mode for Multivariate-based Signature, NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[8] *Kyuang-Ah Shim, Cheol-Min Park, Aeyoung Kim.* HiMQ-3: A High Speed Signature Scheme based on Multivariate Quadratic Equations, NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[9] *A. Casanova, J.-C. Faugère, G. Macario-Rat, J Patarin, L Perret, J Ryckeghem.* GeMSS: A Great Multivariate Short Signature, NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions.

[10] *Ignacio Luengo, Martin Avendano, Michel Marco.* DME: DME a public key, signature and KEM system based on double exponentiation., NIST Submission, 2017. [On-line]. Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions. Unpublished.

[11] Jean-Philippe Aumasson. Gravity-SPHINCS v1, November 29, 2017.

[12] Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe. SPHINCS+. Submission to the NIST post-quantum project. November 30, 2017.

[13] *Leslie Lamport.* Constructing digital signatures from one-way functions. Technical report, Technical Report CSL-98, SRI International Palo Alto, 1979.

[14] *Andreas Hulsing.* W-OTS+ - shorter signatures for hash-based signature schemes. In Progress in Cryptology - AFRICACRYPT 2013, 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22–24, 2013. Proceedings, pages 173–188, 2013.

[15] *Ralph C. Merkle.* A certifed digital signature. In Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 1989, Proceedings, pages 218–238, 1989.

[16] *Oded Goldreich.* The Foundations of Cryptography - Volume 2, Basic Applications. Cambridge University Press, 2004.

[17] 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. 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 I, pages 368–397, 2015.

[18] *Leonid Reyzin, Natan Reyzin.* Better than BiBa: Short One-time Signatures with Fast Signing and Verifying. In Information Security and Privacy, 7th Australian Conference, ACISP 2002, Melbourne, Australia, July 3–5, 2002, Proceedings, pages 144–153, 2002.

[19] Post-Quantum Cryptography Lounge, 2018 [On-line]. Internet: https://www.safecrypto.eu/pqclounge/.

[20] *І.Д. Горбенко, І.С. Кудряшов, В.В. Онопрієнко.* Порівняльний аналіз пост квантових стандартів електронного підпису на основі мультиваріативних квадратичних перетворень // Радиотехника: всеукр. межвед. науч.-техн. сб. – Харьков: ХТУРЕ. – 2018. – Вып. 195. – С. 46–60.

[21] Yu. I. Gorbenko, T. V. Melnik, I.D. Gorbenko. "Analysis of Potential Post-Quantum Schemes of Hash-Based Digital Signatur" *Telecommunications and Radio Engineering*, Volume 77, 2018, Issue 7, pp. 603–626.

[22] Yu. I. Gorbenko, K. V. Isirova. "Improved Mechanism of One-Time Keys for Post-Quantum Period Based on the Hashing Functions" *Telecommunications and Radio Engineering*, Volume 77, 2018, Issue 14, pp. 1277–1296.

Надійшла до редколегії 25.12.2018

Горбенко Юрій Іванович, кандидат технічних наук, перший заступник головного конструктора АТ «ІІТ». Галузь наукових інтересів – системи, комплекси та засоби криптографічного захисту інформації.

Кудряшов Іван Сергійович, студент ХНУ ім. В. Н. Каразіна, факультет комп'ютерних наук, кафедра безпеки інформаційних систем і технологій. Галузь наукових інтересів – криптографічні властивості булевих функцій.

Науменко Данило Сергійович, студент факультету комп'ютерних наук Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – постквантовий електронний підпис.

Онопрієнко Віктор Васильович, кандидат технічних наук, генеральний директор АТ «ІІТ». Галузь наукових інтересів – системи, комплекси та засоби криптографічного захисту інформації.

УДК 003.026:004.056

Горбенко Ю. И. **Сопоставление кандидатов электронной подписи на постквантовый стандарт NIST PQC на базе MQ-преобразований и функций хеширования** / Ю. И. Горбенко, И. С. Кудряшов, Д. С. Науменко, В. В. Оноприенко // Прикладная радиоэлектроника: науч.-техн. журнал. –2018. – Том 17. № 3, 4. – С. 138–146.

Приводятся результаты сравнительного анализа кандидатов на стандарты перспективных электронных подписей, которые строятся на основе мультивариативних квадратичных преобразований и на функции гешування. Результаты анализа получены при использовании методики сравнения криптографических механизмов на основе экспертных оценок по совокупности условных и безусловных критериев. Сделаны рекомендации относительно перспектив применения кандидатов.

Ключевые слова: MQ-преобразования, постквантовый алгоритм, электронная подпись, сравнительный анализ, экспертные оценки, подпись на основе хеш-функций.

Табл.: 6. Ил.: 6. Библиогр.: 22 назв.

UDC 003.026:004.056

Gorbenko Yu. I. **Comparison of electronic signature candidates to the post quantum standard NIST PQC on the basis of MQ-transformations and functions of hashing** / Yu. I. Gorbenko, I. S. Kudryashov, D. S. Naumenko, V. V. Onoprienko // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17. № 3, 4. – P. 138–146.

The results of a comparative analysis of candidates for the standards of promising electronic signatures, which are based on multivariate quadratic transformations and the hashing function, are presented. The results of the analysis are obtained using the methodology for comparing cryptographic mechanisms based on expert estimates using a combination of conditional and unconditional criteria. Recommendations are made regarding the prospects for candidates application.

Keywords: MQ-transformations, post-quantum algorithm, electronic signature, comparative analysis, expert estimates, signature based on hash functions.

Tab.: 6. Fig.: 6. Ref.: 22 items.

УДК 621.3.06

АНАЛІЗ ЗАСТОСУВАННЯ ФУНЦІЇ ГЕШУВАННЯ У ТЕХНОЛОГІЇ BLOCKCHAIN

П. В. КРАВЧУК, І. Д. ГОРБЕНКО, А. І. ПУШКАРЬОВ

Наведено результати аналізу функцій гешування для їхнього застосування у системах, що використовують технологію Blockchain, а також результати порівняльного аналізу їх основних властивостей та рекомендації щодо застосування.

Ключові слова: геш-значення, електронний підпис, криптографічні механізми, криптографічна стійкість, послуги безпеки, технології блокчейн, складність криптопекретворень, функція гешування.

ВСТУП

Аналіз показав, що, під час розробки технологій Blockchain (далі – «блокчейн»), необхідно враховувати важливі аспекти та вимоги до технології блокчейн [у тому щодо інформаційної та кібербезпеки 1–4].

Одним із найважливіших криптографічних компонентів цієї технології, що суттєво визначає їх захищеність – є функції гешування відповідних даних. Тому, під час проектування технологій блокчейн геш-функція, що застосовуватиметься, має бути вибрана за основними безумовними та умовними критеріями – криптографічна стійкість проти класичних та квантових атак, складність (швидкодія) тощо. Важливо також при виборі функції гешування розглянути та порівняти основні альтернативи, у тому числі рівень їх стандартизації та тенденції застосування, а також популярність щодо застосування в сучасних мережах

Метою статті є аналіз та порівняльний аналіз функцій гешування за основними такими безумовними критеріями як криптографічна стійкість проти класичних та квантових атак, складністю криптографічних перетворень (швидкодії) та можливістю і умовами застосування в технологіях блокчейн.

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1. Сутність технології блокчейн

Блокчейн (англ. Blockchain, Block chain від block – блок, chain – ланцюг) [1] – це незмінні системи цифрових регістрів, реалізовані розподіленим чином (тобто, без центрального сховища) та зазвичай без центрального органу. На самому базовому рівні вони дозволяють спільноті користувачів записувати транзакції в загальнодоступному реєстрі цієї спільноти, так що ніяка транзакція не може бути змінена після опублікування.

Важливим компонентом технології блокчейн є використання криптографічних функцій ґешування (ФГ) для багатьох операцій, перше за все, таких як гешування вмісту блоку. Ґешування [2] це метод одно направленого відображення вхідних даних (файла, даних, деякого тексту або зображення тощо) довільної довжини в унікальне вихідне значення фіксованого розміру L_h (називається просто дайджест). З великою ймовірністю вихідне значення вважається випадковим, найменша зміна уведних даних (навіть на один біт) призводить до зміни вихідного, в середньому на один біт.

1.2. Функції гешування

Функції гешування *h(x)* (ФГ) [2,3] – це однонаправлена колізійна стійка функція відображення, що приймає на вході як аргумент інформаційну послідовність (рядок) *M* довільної довжини L_m і дає на виході практично випадкову послідовність (рядок) фіксованої довжини *Lh*. Результат гешування інформаційної послідовності *M* називають геш - образом *h(M)* Для відомих функцій гешування співвідношення між довжинами *M* і *h(M)* може бути довільним, тобто

$$|M| > |h(M)|, |M| < |h(M)|, |M| = |h(M)|.$$

Оскільки результат роботи функції гешування називається геш-образом, то масив даних *M* зазвичай називають прообразом (першим прообразом).

Наведемо формальне визначення функції гешування. Нехай *{0, 1}^m* – безліч всіх двійкових рядків довжини *m, {0, 1}** – безліч всіх двійкових рядків кінцевої довжини. Тоді геш функцією *h* називається перетворення виду:

$$h: \{0, 1\}^* \rightarrow \{0, 1\}^m,$$

де *m* – розрядність геш-образу. На рис. 1 як приклад показано схему гешування, PRNG (Pseudo-Random Number Generator) – генератор псевдовипадкових чисел; Q – елементи пам'яті PRNG; h_0 – вектор ініціалізації; $n=|m_i|$ – розрядність блоків інформаційної послідовності, $i = 1,\ldots, t$:

$$M = m_1, \ldots, m_t,$$

t – число блоків послідовності *M*; *N* – число елементів пам'яті PRNG. Для PRNG процес отримання функції гешування [4] можна спрощено розглядати як накладення псевдовипадкової послідовності (PRS, Pseudo-Random Sequence) на вхідну послідовність для подальшого її перетворення.

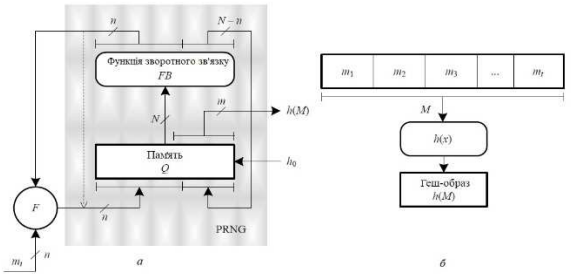

Рис. 1. Спрощений механізм функції гешування: а PRS

Однією із основних вимог до функцій гешування є їх колізійна стійкість. Фізично її можна визначити як можливість знайти колізію функції гешування, тобто складність знаходження двох різних випадкових даних (рядків) M1 і M2, таких що h(M1) = h (M2). Інакше кажучи, коли для двох різних аргументів M1 і M2 значення ФГ збігаються.

На рис. 2 приклад виникнення колізії при гешуванні даних M_3 і M_4.

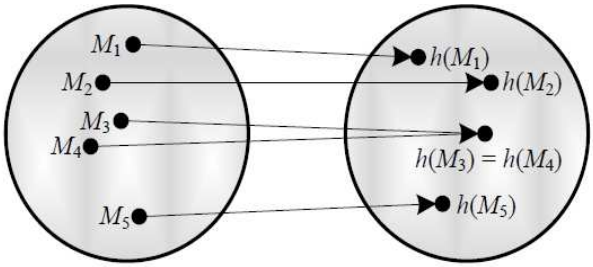

Рис. 2. Відображення безлічі прообразів і геш-образів

1.3. Вимоги до функцій гешування

ФГ знаходять широке застосування, в тому числі необхідно виділити такі застосування ФГ [2, 3,4]:

– механізми контролю цілісності та справжності інформації та ресурсів;

– протоколи електронного підпису, електронного штампу та мітки часу);

– алгоритми генерація псевдовипадкових послідовностей;

– криптографічні протоколи узгодження та встановлення ключів;

– контроль цілісності баз даних, у тому числі без використання спільного секрету тощо.

В технології блокчейн ФГ використовуються для контролю цілісності повідомлень (блоків) [5], які передаються по мережі або зберігаються поза межами захищеного середовища. Для ФГ, що використовуються у випадку технології блокчейн для контролю цілісності баз даних, мають виконуватися такі вимоги [5,2]:

– відновлюваність у просторі і часі;

– детермінованість – при однакових вхідних даних результат виконання ФГ буде однаковим (одне і те саме повідомлення завжди призводить до одного й того ж гешу);

– стійкість до знаходження колізій;

– стійкість до знаходження прообразу – неможливість знаходження невідомого прообразу для будь-яких заданих геш-значень;

– стійкість до атаки пошуку другого прообразу;

– атака збільшення довжини (length extension attack)

– атака фіксованих точок (fixed points).

Оцінки ідеальної стійкості для ФГ [2, 6] наведені на рис. 3.

Тип геш-функції	Ціль атаки	Ідеальна стійкість
Однонаправлена ФГ	Знаходження прообразу	2^l
	Знаходження другого прообразу	2^l
	Знаходження колізії	$2^{\frac{l}{2}}$
	Збільшення довжини	2^l
	Фіксовані точки	2^l
Функція створення MAC	Точне знайдення ключа	$\left\lceil \frac{k}{l} \right\rceil + \frac{2^k - 1}{1 - 2^{-l}}$
	Підробка повідомлення	$P_m = \max(2^{-k}, 2^{-l})$

Рис. 3. Необхідна стійкість ФГ до атак

1.4. Вибір перспективних ФГ

Розглянемо загальні дані щодо перспективних (сучасних) ФГ [7, 2]. У системах, що побудовані на технології блокчейн, сьогодні використовуються тільки деякі ФГ, що стандартизовані на міжнародному рівні та NIST США. Зокрема, найбільш поширені блокчейн розробки (крипто валюти), у яких використовують наступні стандартизовані алгоритми гешування: SHA-256, EtHash, Scrypt, X11, CryptoNight, EquiHash.

Оскільки більшість популярних криптовалют використовують ФГ SHA-2 та SHA-3 («Keccak») [8], то розглянемо їх порівняно з іншими відомими перспективними стандартизованими ФГ: російським ГОСТ 34.11-2012 («Стрибог»), Whirlpool та українським стандартом ДСТУ 7564:2014 («Kupyna»).

На рисунку 4 наведено порівняння ФГ за основними загальними параметрами.

Вибрані ФГ мають розмір вихідного геш-значення від 224 біт (доцільніше використовувати мінімальний вихід у 256 біт) до 512 біт. Інші показники, як розмір внутрішнього стану, розмір блоку та слова – лежать у схожих межах та залежать здебільшого від структури самого алгоритму гешування. За цими даними виділяти кращу ФГ не є правильним, адже це не говорить напряму про стійкість та швидкість алгоритму.

Алгоритм	Розмір виходу, біт	Розмір внутрішнього стану, біт	Розмір блоку, біт	Розмір слова, біт	Кількість раундів
ГОСТ 34.11-2012	256 (512)	256(512)	512	32	12
SHA-2 256	256/224	256	512	32	64
SHA-2 512	512/384	512	1,024	64	80
SHA-3 256	256	1600	1088	64	24
SHA-3 512	512	1600	576	64	24
Whirlpool	512	512	512	8	10
Kupyna 256	256	256	512	64	10
Kupyna 512	512	512	1024	64	14

Рис. 4. Загальні дані алгоритмів сучасних ФГ

2. МЕТОДИ ДОСЛІДЖЕНЬ ФГ

2.1. Статистичне тестування ФГ

Статистичне тестування ФГ, що аналізуються, із використанням методик тестування, що визначені в NIST STS 800- 22(2009) [9].

Тестування статистичних властивостей виконувалось у таких режимах:

1) висока збитковість вхідної послідовності;
2) у режимі генератора псевдовипадкових послідовностей відповідно до ISO/IEC 18031 із використанням функції гешування, з виконанням реініціалізації;
3) у режимі генератора псевдовипадкових послідовностей відповідно до ISO/IEC 18031 із використанням функції гешування, без виконання реініціалізації.

2.2. Порівняння складності (швидкодії) ФГ

Важливим критерієм для порівняння ФГ є складність (швидкодія) гешування.

Для оцінки складності реалізації ФГ вимірювались швидкість роботи стандартизованих реалізацій ФГ мовою Java. Для цього було використано компілятор IntelliJ IDEA 2016.2.4(64). Тестування швидкодії гешування проведено на комп'ютері Intel Core i5-4460 3.2 GHz, 8 GB RAM під управлінням операційної системи Microsoft Windows 10.

2.3. Порівняння стійкості ФГ щодо класичних атак

Під час дослідження стійкості функцій гешування були використані вже відомі широкі результати досліджень [2, 4–6]. Ними підтверджено криптографічну стійкість вибраних ФГ до усіх відомих класичних атак.

2.4 Результати досліджень

В ході аналізу були отримані такі результати.

1. Усі ФГ успішно пройшли статистичні тести [10] (рис.5) з такими значеннями випадковості для NIST STS 800-22(2009 р.).

Отримано наступні показники складності (швидкодії) гешування (рис.6).

Геш-функція	Кількість тестів, що успішно пройдені на рівні α =0,99	Кількість тестів, що успішно пройдені на рівні α =0,96015
ГОСТ 34.11	126	188
SHA-2 256	130	188
SHA-2 512	135	187
SHA-3 256	133	187
SHA-3 512	126	186
Whirlpool	132	187
Kupyna 256	139	187
Kupyna 512	136	187

Рис. 5. Підсумкові середні дані статистичного тестування властивостей ФГ

На рис. 6 по осі Y відображена швидкість роботи у Мегабайтах/секунду. По осі X розташовані ФГ.

Отримані результати показують, що найкращі показники були отримані у ФГ SHA-2 та Kupyna. Найгірші результати швидкості серед цих алгоритмів виявились у SHA-3. Проте навіть ці дані можна покращити у подальших дослідженнях, адже реалізація була мовою програмування Java, яка дещо програє у швидкості мові C та C++ або мові програмування Assembler.

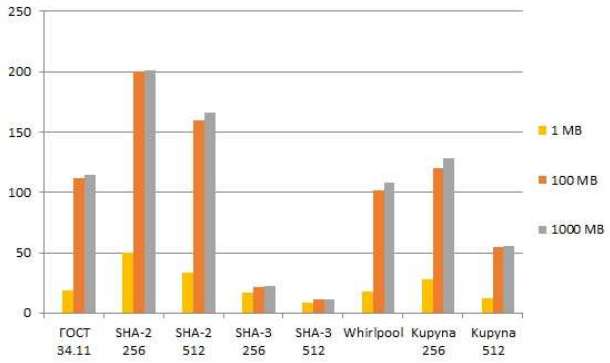

Рис. 6. Результати вимірів швидкості роботи швидкості ФГ

Для технології блокчейн швидкостей гешування за допомогою ФГ SHA-2 та Kupyna буде достатньо, адже середній розмір блоків у блокчейн мережах лише інколи перевищує 1МБ.

2. Результати аналізу стійкості ФГ наведені на рисунку 7.

Алгоритм	Безпека у бітах		
	Пошук колізії	Пошук прообразу	Пошук другого прообразу
ГОСТ 34.11-45	128 (256)	248 (512)	248 (512)
SHA-2 256	128	248	248
SHA-2 512	256	494	494
SHA-3 256	128	256	256
SHA-3 512	256	512	512
WHIRLPOOL	120	512	512
Kupyna-256	128	256	256
Kupyna-512	256	512	512

Рис. 7. Порівняльний аналіз класичних атак на ФГ

У цілому за результатами аналізу можна зробити висновок, що не дивлячись на зростання потужностей класичних комп'ютерів, сучасні функції гешування дозволяють забезпечити необхідний рівень стійкості проти усіх відомих атак.

Додатково була проаналізована загроза щодо створення та застосування квантового комп'ютера для здійснення атак на ФГ. Проведений аналіз показав, що квантовий комп'ютер здатний створити загрозу сучасним алгоритмам ФГ тільки з обмеженими розмірами параметрів, проте не на теперішньому етапі їх розвитку.

Було зроблено певний прогноз [11] щодо збільшення обчислювальної потужності квантового комп'ютера на найближчі 10 років із урахуванням необхідної для атаки потужності.

Припустимо, що кількість кубітів зростатиме експоненційно [12], тобто подвоюватиметься кожні 10 місяців, тоді як менш оптимістичне припущення передбачає подвоєння кожні 20 місяців. Ці дві екстраполяції наведені на рис. 8.

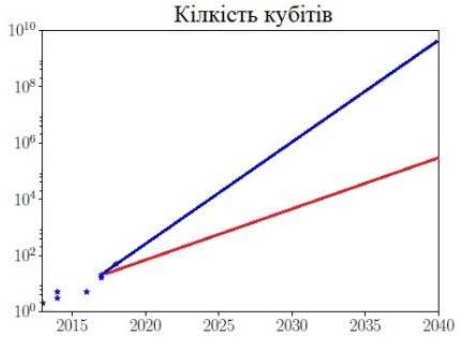

Рис. 8. Перспективи розвитку квантового комп'ютера

Далі, атака за допомогою методу Гровера [2] дозволяє зменшити час, необхідний для пошуку, наприклад, колізії функції гешування, до приблизно кореня із часу виконання класичної атаки:

$$O\left(\sqrt{\frac{N}{l}}\right).$$

Це дуже суттєве поліпшення, проте цього на даний момент не достатньо. На прикладі мережі Біткоїн, що використовує ФГ SHA-2 256 можна побачити, що потужності одного квантового комп'ютера не достатньо для суттєвого впливу на мережу.

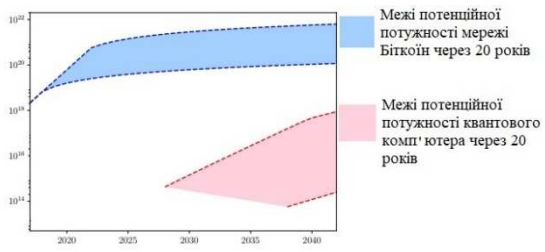

Рис. 9. Порівняння потужності мережі Біткоїн та квантового комп'ютера

Також була обчислена необхідна кількість кубітів для виконання атаки. Йде мова про фіксоване значення у 2402 логічних кубіти, що буде досягнуто згідно з прогнозами не раніше, ніж у 2027 році.

ВИСНОВКИ

На основі отриманих даних можна виділити таке:

1. Всі розглянуті стандартизовані ФГ пройшли статистичне тестування, тестування щодо складності гешування (швидкодії) та перевірку на криптографічну стійкість проти класичних атак.

3. Певну перевагу на сьогодні для використання при побудові блокчейн мереж є SHA-2, SHA-3 та Kupyna. Вказані і ФГ є стійкими проти класичного криптоаналізу, в тому числі: до знаходження прообразу; до знаходження другого прообразу та до виникнення чи створення колізій.

4. Більш конкретно можна відзначити SHA-2 та Kupyna, адже ці ФГ показують також найкращі результати щодо складності (швидкодії) гешування даних.

5. З огляду на потенціальну небезпеку квантового комп'ютера, доцільно використовувати ФГ із виходом у 512 біт: SHA-2 512, SHA-3 512 та Kupyna 512, адже це суттєво збільшує стійкість мережі, а втрати у швидкодії гешування є незначними.

6. Практично та швидще всього і теоретично, якщо розглядати атаку на основі методу Гровера ФГ при довжинах геш значень 512 бітів, а в перехідний період і при довжинах 256 бітів, забезпечуватимуть експоненційну складність навіть найбільш загрозливих атак на основі створення колізій.

7. Аналіз розвитку квантових комп'ютерів показав, що навіть за оптимістичного прогнозу, квантовий комп'ютер та відповідне математичне забезпечення можуть бути створені не раніше 2027 року.

8. На наш погляд, у ході оцінки можливостей та обґрунтування вибору ФГ для застосування в перспективних блокчейн мережам, можна рекомендувати до застосування ФГ із виходом у 512 біт: SHA-2 512, SHA-3 512 та Kupyna 512.

9. Безумовним є той факт, що в подальшому потрібно проводити дослідження властивостей та умов застосування вибраних ФГ, що рекомендуються до застосування.

Література

[1] *Andreas M.* Antonopoulos Mastering Bitcoin: Unlocking Digital Cryptocurrencies / Andreas M. Antonopoulos – K.: NGITS, 2014. – С. 10–150.

[2] За редакцією Горбенко І.Д. Горбенко Ю.І. «Побудування та аналіз систем, протоколів та засобів криптографічного захисту інформації». Монографія. Харків. Форт. 2015. – 902 с.

[3] *Don Tapscott, Alex Tapscott* Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World / Don Tapscott, Alex Tapscott Blockchain – K. : Information Systems, 2016 – С. 65–102.

[4] Криптографические хэш-функции [Електронний ресурс]. – Режим доступу: www/ URL: http://bit.nmu.org.ua/

ua/student/metod/cryptology/D0%BB%D0%B5%D0%BA%D1%86%D0%B8%D1%8F17.pdf – 04.10.2018 р.
[5] Возможные атаки на функции хэширования [Електронний ресурс]. – Режим доступу: www/ URL: https://studfiles.net/preview/2157418/page:2/ – 04.10.2018 р.
[6] Прикладна криптологія. Теорія. Практика. Застосування : монографія / І.Д. Горбенко, Ю.І. Горбенко; Міністерство освіти і науки, молоді та спорту України, ХНУРЕ, ПАТ "ІІТ" – Харків, 2012 – С. 340–347.
[7] Алгоритмы шифрования – основа работы криптовалют [Електронний ресурс]. – Режим доступу: www/ URL: https://tgraph.io/Algoritmy-shifrovaniya--osnova-raboty-kriptovalyut-09-27 – 14.09.2018 р.
[8] Comparison of cryptographic hash functions [Електронний ресурс]. – Режим доступу: www/ URL: https://en.wikipedia.org/wiki/Comparison_of_cryptographic_hash_functions – 17.10.2018 р.
[9] NISTIR 7896. Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition / NIST, 2012. – С. 50–65.
[10] Analysis of the Kupyna-256 Hash Function / Christoph Dobraunig, Maria Eichlseder, and Florian Mendel, Graz University of Technology, – Austria. 2016.
[11] Квантовые компьютеры / Л. Федичкин, ФТИ РАН. НиЖ, 2001, – С. 20–33.
[12] Quantum search using Grover's algorithm [Електронний ресурс]. – Режим доступу: www/ URL: http://savepearlharbor.com/?p=222456 – 18.06.2014 р.

Надійшла до редколегії 26.11.2018

Кравчук Павло Вікторович, студент, ХНУРЕ. Галузь наукових інтересів – аналіз і тестування асиметричних перетворень, блокчейн технології.

Горбенко Іван Дмитрович, докт. техн. наук, професор кафедри безпеки інформаційних технологій (БІТ) Харківського національного університету радіоелектроніки (ХНУРЕ), академік Академії наук прикладної радіоелектроніки. Галузь наукових інтересів – створення, аналіз і реалізація систем і засобів захисту інформації; дослідження і реалізація криптографічних протоколів.

Пушкарьов Андрій Іванович, директор департаменту захисту інформації Адміністрації державної служби спеціального зв'язку та захисту інформації України. Галузь наукових інтересів – теорія захисту інформації, інформаційна та кібербезпека держави.

УДК 621.3.06

Кравчук П. В. **Анализ применения хеш-функций в технологии Blockchain** / П. В. Кравчук, И. Д. Горбенко, А. И. Пушкарев // Прикладная радиоэлектроника: науч. – техн. журнал. – 2018. – Том 17. № 3, 4. – С. 147–151.

В статье представлены результаты анализа выбранных хеш-функций для их применения в системах, использующих технологию Blockchain. Кроме того, был проведен сравнительный анализ их стойкости, скорости и противодействие различным атакам; сформированы правила применения хеш-функций.

Ключевые слова: хеш - значения, электронная подпись, криптографические механизмы, криптографическая стойкость, услуги безопасности, технологии блокчейн, сложность криптопреобразований, функция хеширования.

Ил.: 9. Библиогр.: 12 назв.

UDC 621.3.06

Kravchuk P. V. **Analysis of the application of Hash functions in Blockchain technology** / P. V. Kravchuk, I. D. Gorbenko, A. I. Pushkarev // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17. № 3, 4. – P. 147–151.

The paper presents the results of the analysis of selected hash functions for their application in systems using Blockchain technology. In addition, the resuls of a comparative analysis of their stiffness, speed, their defense against various attacks were made; the rules of application of hash functions were formed.

Keywords: hash-value, electronic signature, cryptographic mechanisms, cryptographic stability, security services, blockchain technology, complexity of cryptotransformations, hashing function.

Fig.9. Ref.: 12 items.

МЕТОДЫ И МОДЕЛИ КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ

УДК621.391.15 : 519.7

2-ИЗОГЕНИИ ПОЛНЫХ И КВАДРАТИЧНЫХ КРИВЫХ ЭДВАРДСА НАД ПРОСТЫМ ПОЛЕМ

А. В. БЕССАЛОВ

Дан анализ условий существования 2-изогений полных и квадратичных кривых Эдвардса над простым полем. Дан обзор свойств трех классов кривых в обобщенной форме Эдвардса: полных, скрученных и квадратичных кривых Эдвардса. Для корректной записи отображающих функций и определения степени изогении предложено использовать модифицированный закон сложения точек. Обсуждаются проблемы нахождения дуальных 2-изогений между классами полных, квадратичных и скрученных кривых Эдвардса.

Ключевые слова: кривая в обобщенной форме Эдвардса, скрученная кривая Эдвардса, квадратичная кривая Эдвардса, порядок кривой, порядок точки, изоморфизм, изогения, квадратичное кручение, квадратичный вычет, квадратичный невычет.

ВВЕДЕНИЕ

Одной из известных перспектив постквантовой криптографии являются изогении суперсингулярных эллиптических кривых с возможно большим числом подгрупп их точек. Проблема дискретного логарифмирования классической эллиптической криптографии заменяется проблемой поиска одной из изогений великого множества подгрупп такой нециклической кривой, достаточно стойкой к атакам виртуального квантового компьютера. На сегодняшний день нарастающий интерес к изогениям связывается с наименьшей длиной ключа в предлагаемых алгоритмах в сравнении с другими известными кандидатами постквантовой криптографии при заданном уровне стойкости.

Свойства изогений для кривых в форме Веерштрасса достаточно хорошо изучены. Значительно меньше нам известны эффективные методы построения и свойства изогений перспективных классов кривых в форме Эдвардса.

Кривые Эдвардса с одним параметром, определенные в работе [1], имеют очень привлекательные для криптографии преимущества: максимальная скорость экспоненцирования точки, полнота и универсальность закона сложения точек, аффинные координаты нейтрального элемента группы точек, повышенная безопасность в отношении атак побочного канала. Программирование групповых операций становится более эффективным и ускоряется в связи с отсутствием особой точки на бесконечности как нуля абелевой группы точек. Введение второго параметра кривой в работе [2] расширило класс кривых в форме Эдвардса и породило кривые с новыми свойствами, интересными для криптографических приложений. В данной статье обсуждаются свойства 2-изогений двух классов этих кривых, в частности, условия их существования над простым полем.

Среди многочисленных работ по этой проблематике выделим статьи [3, 4], в которых впервые получены формулы изогений для двух классов кривых в форме Эдвардса. Наш анализ в данной работе опирается на их результаты.

В разделе 1 статьи приводятся основные определения для изоморфных кривых в форме Монтгомери и Эдвардса, законы сложения и удвоения точек последних с модификацией, адаптированной к горизонтальной симметрии обратных точек. Дан краткий обзор свойств трех классов кривых в обобщенной форме Эдвардса в соответствии с принятой в [5, 6] классификацией. В разделе 2 дается детальный анализ одного из методов получения 2-изогений для двух классов полных и квадратичных кривых Эдвардса, приводятся примеры и обсуждаются условия существования 2-изогений в этих классах над простым полем.

1. ИЗОМОРФИЗМЫ И СВОЙСТВА КРИВЫХ ЭДВАРДСА

Анализ изогений кривых Эдвардса часто опирается на кривые в форме Вейерштрасса и их частные случаи изоморфных кривых в форме Монтгомери или Лежандра. Запишем кривую форме Монтгомери над полем $F_q, q = p^m$, уравнением [2]

$$E_{C,D} : Dv^2 = u^3 + Cu^2 + u, \quad C = 2\frac{a+d}{a-d},$$

$$D = \frac{4}{a-d}, \quad a = \frac{C+2}{D}, \quad d = \frac{C-2}{D}, \quad C^2 \neq 4. \qquad (1)$$

Эта кривая рациональным преобразованием координат

$$y=\frac{u}{v},\quad x=\frac{u-1}{u+1},\quad \Rightarrow\quad u=\frac{1+x}{1-x},\, v=\frac{u}{y}. \qquad (2)$$

отображается в бирационально эквивалентную кривую *в обобщенной форме Эдвардса* [2,6] с уравнением

$$E_{a,d}:\ \begin{array}{c} x^2+ay^2=1+dx^2y^2, \\ a,d\in F_p^*, d\neq 1, a\neq d,\, p\neq 2 \end{array}. \qquad (3)$$

В отличие от уравнения этой кривой в [2] здесь мы параметр a умножаем на y^2 вместо x^2. Если квадратичный характер $\chi(ad)=-1$, кривая (3) изоморфна *полной кривой* Эдвардса [1] с одним параметром d

$$E_d:\ x^2+y^2=1+dx^2y^2, \chi(d)=-1, d\neq 0,1, \qquad (4)$$

В случае $\chi(ad)=1$, и $\chi(a)=\chi(d)=1$ имеет место изоморфизм кривой (3) с *квадратичной кривой Эдвардса* [6]

$$E_d:\ x^2+y^2=1+dx^2y^2, \chi(d)=1, d\neq 0,1, \qquad (5)$$

имеющей, в отличие от (4), параметр d, определенный как квадрат. Это отличие ведет к кардинально различным свойствам кривых (4) и (5) [6], которые резюмируются ниже. Несмотря на это, в мировой литературе эти классы кривых объединены общим термином *кривые Эдвардса* [2].

В работе [7] мы предложили поменять местами x и y координаты в форме кривой Эдвардса. Тогда модифицированный универсальный закон сложения точек имеет вид

$$(x_1,y_1)+(x_2,y_2)=$$

$$=\left(\frac{x_1x_2-ay_1y_2}{(1-dx_1x_2y_1y_2)},\frac{x_1y_2+x_2y_1}{(1+dx_1x_2y_1y_2)}\right). \qquad (6)$$

При совпадении двух точек получим из (2) закон удвоения точек

$$2(x_1,y_1)=\left(\frac{x_1^2-ay_1^2}{\left(1-dx_1^2y_1^2\right)},\frac{2x_1y_1}{\left(1+dx_1^2y_1^2\right)}\right). \qquad (7)$$

Использование модифицированных законов (6), (7) позволяет сохранить общепринятую горизонтальную симметрию (относительно оси x) обратных точек. Определяя теперь обратную точку как $-P=(x_1,\,-y_1)$, получим согласно (6) координаты нейтрального элемента группы точек $O=(x_1,y_1)+(x_1,\,-y_1)=(1,\,0)$. Кроме нейтрального элемента О на оси x также всегда лежит точка $D_0=(-1,\,0)$ второго порядка, для которой в соответствии с (7) $2D_0=(1,\,0)=O$. В зависимости от свойств параметров a и d можно получить еще 2 особые точки 2-го порядка и 2 или 4 точки 4-го порядка. Как следует из (3), на оси y могут лежать точки $\pm F_0=\left(0,\,\pm 1/\sqrt{a}\right)$ 4-го порядка, для которых $\pm 2F_0=D_0=(-1,\,0)$. Эти точки существуют над простым полем F_p, если параметр a является квадратом (квадратичным вычетом).

Из уравнения (3) определим квадраты:

$$x^2=\frac{1-ay^2}{1-dy^2},\ y^2=\frac{1-x^2}{a-dx^2},$$

порождающие особые точки на бесконечности (знак "∞"мы ставим при делении на 0):

$$D_{1,2}=\left(\pm\sqrt{\frac{a}{d}},\infty\right),\pm F_{11}=\left(\infty,\pm\frac{1}{\sqrt{d}}\right). \qquad (8)$$

Они возникают в случаях $\chi(ad)=1$ и $\chi(d)=1$ соответственно. Это, например, всегда выполняется в расширении поля F_{p^2}. По правилам предельного перехода и закона удвоения (7) можно проверить, что $2D_{1,2}=O,\ \pm 2F_1=D_0=(-1,\,0)$. Иными словами, при выполнении условий их существования особые точки $D_{1,2}$ есть точки 2-го порядка, а особые точки $\pm F_1$ – точки 4-го порядка.

Кроме перечисленных, точки 4-го порядка могут существовать как не особые при ненулевых координатах x и y.

Теорема 1. *Не особые точки 4-го порядка*

$$\pm F_2=\left(\sqrt[4]{\frac{a}{d}},\pm\sqrt{\frac{-1}{\sqrt{ad}}}\right),\ \pm F_3=\left(-\sqrt[4]{\frac{a}{d}},\pm\sqrt{\frac{-1}{\sqrt{ad}}}\right)$$

кривой в форме (1) при $x\neq 0$ существуют тогда и только тогда, когда выполняются условия:

(i) *при* $p\equiv -1 mod 4$: $\chi(a)=\chi(d)=-1$;

(ii) *при* $p\equiv 1 mod 4$: $\chi(a)=\chi(d)=1,\ ad=c^4$.

Доказательство теоремы 1 дано в работе [5]. Заметим, что с учетом 4-х корней 4-й степени из элемента поля число точек 4-го порядка для кривой (5), определенных теоремой 1, обычно равно 8 (для каждой точки (x_1,y_1) существует точка (y_1,x_1)).

Точки $\pm F_{2,3}$ можно рассматривать как точки деления на 2 особых точек 2-го порядка $D_{1,2}/2$ [6].

Пример 1. Для кривой $x^2-y^2=(1+3x^2y^2) mod 7$ (здесь $a=-1$, $d=3$ – квадратичные невычеты при $p=7\equiv 3\bmod 4$ и выполняются условия (i) теоремы 1)

точки 4-го порядка согласно теореме 1 имеют координаты $\pm F_{2,3} = (\pm 2, \pm 2)$. При удвоении их согласно (7) получим $2F_2 = (\pm 3, \infty) = D_{1,2}$. Порядок N_E этой кривой, включающей точки $O, \pm F_{2,3}, D_{0,1,2}$, равен 8, группа точек нециклическая с типом $T = (2, 2^2)$.

Пример 2. В условиях (ii) теоремы 1 рассмотрим кривую $x^2 + y^2 = (1 + 3x^2 y^2) mod 13$ (здесь $a = 1$, $d = 3$ – квадратичные вычеты при $p = 13$). Согласно теореме 1 находим точки 4-го порядка $\pm F_{2,3} = (\pm 6, \pm 4)$, $\pm F_{4,5} = (\pm 4, \pm 6)$. Кроме того, кривая имеет две точки 4-го порядка $\pm F_0 = (0, \pm 1)$ и две особые точки (8) 4-го порядка $\pm F_1 = (\infty, \pm 3)$. Удвоение точек $F_{2,3}$ согласно (7) дает точки $2F_2 = \left(\pm\sqrt{\frac{a}{d}}, \infty\right) = (\pm 3, \infty) = D_{1,2}$. Эта кривая, таким образом, содержит 12 точек 4-го порядка, имеет порядок $N_E = 16$ и является нециклической с типом $T = (2^2, 2^2)$. Точки 4-го порядка являются ключевыми при нахождении 2-изогений кривых Эдвардса.

С использованием правил предельного перехода в (6) для особых точек, можно найти координаты сумм:

$$
\begin{aligned}
&(x_1, y_1) + (-1, 0) = (-x_1, -y_1) = (x_1, y_1)^*, \\
&(x_1, y_1) + \left(\sqrt{\frac{a}{d}}, \infty\right) = \left(\sqrt{\frac{a}{d}} x_1^{-1}, \frac{1}{\sqrt{ad}} y_1^{-1}\right), \\
&(x_1, y_1) + \left(-\sqrt{\frac{a}{d}}, \infty\right) = \left(-\sqrt{\frac{a}{d}} x_1^{-1}, -\frac{1}{\sqrt{ad}} y_1^{-1}\right), \\
&(x_1, y_1) + \left(\infty, \frac{1}{\sqrt{d}}\right) = \left(-\frac{1}{\sqrt{d}} y_1^{-1}, \frac{1}{\sqrt{d}} x_1^{-1}\right), \qquad (9) \\
&(x_1, y_1) + \left(\infty, -\frac{1}{\sqrt{d}}\right) = \left(\frac{1}{\sqrt{d}} y_1^{-1}, -\frac{1}{\sqrt{d}} x_1^{-1}\right).
\end{aligned}
$$

Все найденные суммы удовлетворяют уравнению (1) при подстановке, т.е. являются точками кривой.

Подчеркнем, что использование правил предельного перехода сохраняет операцию сложения любых пар точек, включая особые. Это позволяет говорить об изоморфизме кривых в форме Монтгомери и Эдвардса [6,8].

Обоснование новой классификации кривых в обобщенной форме Эдвардса дано в работах [6, 8]. Ниже даны определения 3-х классов этих кривых и перечень фундаментальных свойств кривых разных классов.

В зависимости от свойств параметров a и d кривые в обобщенной форме Эдвардса (1) разбиваются на 3 непересекающиеся класса:

- *полные кривые Эдвардса* (с условием C1: $\chi(ad) = -1$;
- *скрученные кривые Эдвардса* (с условиями C2.1: $\chi(a) = \chi(d) = -1$;
- квадратичные кривые Эдвардса (с условиями C2.2: $\chi(a) = \chi(d) = 1$.

Основные свойства этих классов кривых [6–8]:

1. В отношении точек 2-го порядка первый класс полных кривых Эдвардса над простым полем является классом *циклических* кривых, скрученные же и квадратичные кривые Эдвардса образуют классы *нециклических* кривых. Максимальный порядок точек кривых последних классов не превышает $N_E / 2$.

2. Класс полных кривых Эдвардса не содержит особых точек.

3. Скрученные кривые Эдвардса содержат лишь две особых точки 2-го порядка $D_{1,2} = \left(\pm\sqrt{\frac{a}{d}}, \infty\right)$.

4. Квадратичные кривые Эдвардса содержат две особые точки 2-го порядка $D_{1,2} = \left(\pm\sqrt{\frac{a}{d}}, \infty\right)$ и две особые точки 4-го порядка $\pm F_{11} = \left(\infty, \pm\frac{1}{\sqrt{d}}\right)$.

5. Скрученные и квадратичные кривые Эдвардса образуют пары квадратичного кручения на основе преобразования параметров: $\tilde{a} = ca$, $\tilde{d} = cd$, $\chi(c) = -1$.

6. В классах скрученных и квадратичных кривых Эдвардса замена $a \leftrightarrow d$ дает изоморфизм $E_{a,d} \sim E_{d,a}$.

7. Полные и квадратичные кривые Эдвардса изоморфны кривым с параметром $a = 1$: $E_{a,d} \sim E_{1.,d/a}$. Введение нового параметра a в уравнение кривой (1) оправдано лишь для класса скрученных кривых Эдвардса.

8. Скрученные кривые Эдвардса при $p \equiv 1 \bmod 4$ не имеют точек 4-го порядка.

9. Для точек нечетного порядка закон сложения точек (6) всегда является полным (т.е. сумма любой пары точек не дает особой точки).

2. 2-ИЗОГЕНИИ ПОЛНЫХ И КВАДРАТИЧНЫХ КРИВЫХ ЭДВАРДСА

Изогения эллиптической кривой $E(K)$ над полем K в кривую $E'(K)$ есть гомоморфизм $\phi(E(\bar{K})) \to E'(\bar{K})$, задаваемый рациональными функциями. Это значит, что для всех $P, Q \in E(K)$, $\phi(P + Q) = \phi(P) + \phi(Q)$ и существуют рациональные функции [9]

$$\phi(x,y)=\left(\frac{p(x)}{q(x)},y\frac{f(x)}{g(x)}\right)=(x',y'),$$

отображающие точки кривой E в точки кривой E'. Степенью изогении называется максимальная из степеней $\alpha=\deg\phi(x,y)=\max\{\deg p(x),\deg q(x)\}$, а ее ядром – подгруппа $G\subseteq E$ порядка α, точки которой отображаются функцией $\phi(x,y)$ в нейтральный элемент группы О. Изогения сжимает точки кривой E в α раз и является сюръекцией (α точек кривой E отображаются в одну точку кривой E'). При $G=E$ изогения становится изоморфизмом.

Вычисление изогений обычно осуществляется по формулам Велю [9] для кривых в форме Вейерштрасса. В работах [3, 4] получены формулы изогений второй (2-изогении) и нечетных степеней, адаптированные, в частности, к кривым в форме Эдвардса (4) и (5) с одним параметром d. Проанализируем и расширим некоторые из их результатов с акцентом на анализ условий существования 2-изогений над простым полем.

Построение 2-изогений в [3] производится в 3 этапа:

1. Изоморфное преобразование $\psi_1(x,y)=(u,v)$ кривой Эдвардса в форму Монтгомери.
2. Построение 2-изоогений $\psi_2(u,v)=(U,V)$.
3. Обратная трансформация $\psi_3(U,V)=(x,y)$ изогенной кривой в форме Монтгомери в форму Эдвардса.

В итоге находится композиция $\phi(x,y)=$ $=\psi_1\circ\psi_2\circ\psi_3$ трех отображений между кривой E и изогенной кривой E'.

На первом этапе $E_d\to E_{C,D}$ кривая Эдвардса $x^2+y^2=1+dx^2y^2$ рациональным преобразованием (2)

$$\psi_1(x,y)=((a-d)\frac{1+u}{1-u},(a-d)\frac{2u}{v})$$

трансформируется в бирационально эквивалентную форму Монтгомери

$$v^2=u^3+2(a+d)u^2+(a-d)^2u \qquad (10)$$

кривой, изоморфной (1). Точка (0,0) есть точка 2-го порядка этой кривой, которая вместе с точкой на бесконечности как нейтральным элементом группы образует ядро 2-изогении. Требуется найти параметры $\bar a$ и $\bar d$ изогенной кривой E' с уравнением (10) и рациональную функцию $\psi_2(u,v)=(U,V)$.

Для кривой Монтгомери общего вида

$$v^2=u^3+cu^2+bu \qquad (11)$$

нахождение 2-изогений хорошо известно [9]. На основе формул Велю, использующих законы сложения точек кривой в общей форме Веерштрасса, для кривой (11) можно получить 2-изогению ([9], пример 12.4)

$$\psi_2(u,v)=\left(\frac{u^2+cx+b}{u},\frac{u^2-b}{u^2}v\right)=(U,V) \qquad (12)$$

и уравнение изогенной кривой

$$V^2=U^3-2cU^2+(c^2-4b)U. \qquad (13)$$

Дискриминант квадратного уравнения в правой части (12) равен $\Delta=16b$, и, в зависимости от значения $\chi(b)$, кривая (13) имеет одну или 3 точки 2-го порядка. В первом случае можно построить одну 2-изогению, во втором – 3 (для трех ядер как подгрупп 2-го порядка).

Ключевым в данной работе является вопрос о существовании 2-изогений в 3-х классах кривых Эдвардса над простым полем. Как следует из (10) и (11), к форме Монтгомери (1) или (10) (и, соответственно, к форме Эдвардса) можно привести лишь те кривые (11) общего вида, параметр b которых является квадратом: $\chi(b)=1$. Это связано с существованием на кривой (11) точек 4-го порядка $F=(u_1,v_1)$, таких, что $2F=(0,0)$. Тогда, принимая $b=u_1^2$, уравнение (11) приводится к виду

$$v^2=u^3+cu_1u^2+u_1^2u \qquad (14)$$

или изоморфной (1) (или ее квадратичному кручению) кривой

$$c=2\frac{a+d}{a-d}. \qquad (15)$$

Эта кривая бирационально эквивалентна кривой (3) в обобщенной форме Эдвардса (при $v^2\to Dv^2$). Уравнение (14) эквивалентно (10) при $u_1^2=(a-d)^2$ и $cu_1=2(a+d)$.

Таким образом, 2-изогенная кривая (13) с дискриминантом $\Delta=16u_1^2$ в этом случае имеет 3 точки 2-го порядка, и соответствующие изогении могут находиться лишь в классах квадратичных и скрученных кривых Эдвардса, образующих пары квадратичного кручения. В то же время кривая E, для которой строится изогения, может иметь одну точку 2-го порядка и 2 точки 4-го порядка (класс полных кривых Эдвардса), так и принадлежать другим классам кривых Эдвардса с тремя точками 2-го порядка. Например, при $p\equiv 3\bmod 4$ суперсингулярная кривая $v^2=u^3+u$ (для нее $\chi(c^2-4b)=-1$) имеет одну точку 2-го порядка и 2 точки 4-го порядка и изоморфна полной кривой Эдвардса. Ее 2-изогенная кривая (13)

$V^2 = U^3 - 4U$ имеет 3 точки 2-го порядка и попадает в классы квадратичных и скрученных кривых Эдвардса с одним порядком $p+1$ этих кривых. Однако элемент (–4) есть квадратичный невычет и кривая Эдвардса, изоморфная кривой $V^2 = U^3 - 4U$, не существует. Однако, принимая $U \to U - 2$, получим изоморфную кривую, $V^2 = U^3 + 6U^2 + (2\sqrt{2})^2 U$ для которой изоморфизм с кривой Эдвардса при $p \equiv 3 \bmod 8$ существует. Таким образом, исходная кривая E вида (11) с адаптацией к кривым Эдвардса может иметь одну или 3 точки второго порядка и, следовательно, над простым полем принадлежит одному из классов полных, скрученных или квадратичных кривых Эдвардса. Все эти кривые в расширении F_{p^2}, в котором все элементы подполя F_p становятся квадратами, становятся квадратичными кривыми Эдвардса. В расширениях F_{p^n}, разумеется, также можно строить как полные, так и скрученные кривые Эдвардса.

Далее рассмотрим полные и квадратичные кривые (5) с одним параметром d, $a = 1$. В этом случае уравнения (10) и (11) тождественны при $c = 2(1+d)$, $b = (1-d)^2$ тогда $c^2 - 4b = 16d$ и уравнение изогенной кривой (13) в форме Монтгомери имеет вид

$$M1: \quad V^2 = U^3 - 4(1+d)U^2 + 16dU. \qquad (16)$$

Ее дискриминант $\Delta = 16(1-d)^2$, а соответствующие корни определяются как $2(1+d) \pm 2(1-d) = \{4, 4d\}$. Ее, следовательно, можно записать как

$$V^2 = U(U-4)(U-4d).$$

Кривая (16) с точностью до изоморфизма совпадает с изогенной кривой в форме (10), но с параметром $\bar{d}$

$$V^2 = U^3 + 2(1+\bar{d})U^2 + (1-\bar{d})^2 U. \qquad (17)$$

Из (14) – (17) можно получить равенства

$$2\frac{(1+\bar{d})}{(1-\bar{d})}U_1 = -4(1+d), \quad U_1^2 = 16d.$$

Отсюда после подстановки $U_1 = \pm 4\sqrt{d}$ получим

$$\frac{(1+\bar{d})}{(1-\bar{d})} = \frac{\mp(1+d)}{2\sqrt{d}} \Rightarrow \quad \bar{d}_1^{\pm 1} = \left(\frac{1-\sqrt{d}}{1+\sqrt{d}}\right)^2. \quad (18)$$

Итак, для кривой две изогенные кривые имеют два взаимно-обратных параметра изоморфных квадратичных кривых Эдвардса.

Формула (18) справедлива лишь для одной из 3-х точек 2-го порядка (0,0) кривой (13). Линейное смещение координаты $U \to \{U-4, U-4d\}$ в другие значения корней кубики в (16) приводит к двум альтернативным уравнениям изогенных кривых в форме Монтгомери:

$$M2: \quad V^2 = U^3 - 4(d-2)U^2 + 16(1-d)U.$$

$$M3: \quad V^2 = U^3 + 4(2d-1)U^2 - 16d(1-d)U.$$

На основе (14), (15) и этих уравнений можно получить еще две формулы для параметров $\bar{d}_{2,3}$ изогенных кривых, которые приведены ниже в теореме 1.

Обратное преобразование изогенных кривых в форме Монтгомери (M1, M2 и M3) в форму Эдвардса производится на основе рациональных функций (2) с учетом различных значений координат точек 4-го порядка $\pm U_1 \in \{4\sqrt{d}, 4\sqrt{1-d}, 4\sqrt{d(d-1)}\}$ или $\pm U_1 = 1 - \bar{d}$ с помощью функции

$$\psi_3(U,V) = \left(\frac{U-U_1}{U+U_1}, \frac{2U}{V}\sqrt{\frac{U_1}{1-\bar{d}}}\right) = (x,y).$$

Подстановка этих рациональных функций вида (2) в уравнения кривой в форме Монтгомери дает изогенную кривую Эдвардса $x^2 + y^2 = 1 + \bar{d}x^2y^2$.

Композиция $\phi(x,y) = \psi_1 \circ \psi_2 \circ \psi_3$ трех преобразований дает формулы 2-изогений для кривых в форме Эдвардса, приведенные ниже.

В работе [3] доказана теорема 1, которую мы приводим с учетом модификации законов сложения точек кривых Эдвардса и замены $(x \leftrightarrow y)$.

Теорема 1 [3]. Пусть E_d – кривая Эдвардса и определены элементы (возможно, в расширении) поля K $\delta^2 = d, \gamma^2 = 1-d, i^2 = -1$. Тогда существуют 2-изогении $E_d \to E'_d$, заданные функциями ϕ_1, ϕ_2, ϕ_3:

$$\phi_1(x,y) = \left(\frac{d \mp \delta}{d \pm \delta}\frac{\delta x^2 \pm 1}{\delta x^2 \mp 1}, i(\delta \mp 1)xy\right),$$

отображающей E_d в $E'_d : x^2 + y^2 = 1 + \bar{d}_1 x^2 y^2$ с параметрами

$$\bar{d}_1^{\pm 1} = \left(\frac{1-\delta}{1+\delta}\right)^2;$$

$$\phi_2(x,y) = \left(\frac{(\gamma \mp 1)x^2 \pm 1}{(\gamma \pm 1)x^2 \mp 1}, (\gamma \mp 1)xy\right),$$

отображающей E_d в $E'_d : x^2 + y^2 = 1 + \bar{d}_2 x^2 y^2$ с параметрами

$$\bar{d}_2^{\pm 1} = \left(\frac{1-\gamma}{1+\gamma}\right)^2;$$

$$\phi_3(x,y)=\left(-\frac{\delta x^2 \mp i\gamma-\delta}{\delta x^2 \pm i\gamma-\delta},(i\gamma\pm\delta)\frac{y}{x}\right),$$

отображающей E_d в $E'_d: x^2+y^2=1+\bar{d}_3 x^2y^2$ с параметрами

$$\bar{d}_3^{\pm 1}=\left(\frac{i\gamma\mp\delta}{i\gamma\pm\delta}\right)^2.$$

Здесь следует заметить, что общепринятым определением степени изогении является старшая из степеней $\frac{p(x)}{q(x)}$ первой рациональной функции преобразования $\phi(x,y)$ [9]. Оно справедливо для кривых в форме Вейерштрасса. Если обратиться к оригинальной теореме 1 [3], то приходим к парадоксальному результату: степень 2-изогении равна1. Принятый нами модифицированный закон сложения (6) точек кривой Эдвардса с симметрией обратных точек $\pm(x_1,y_1)=(x_1,\pm y_1)$ снимает этот парадокс ($\alpha=\deg(\phi)=2$).

Из формул теоремы 1 следует, что над простым полем существуют 2-изогении кривых E из классов полных (функция $\phi_2(x,y)$ при $\chi(1-d)=1$) и квадратичных кривых (все функции $\phi_{1,2,3}(x,y)$ при $\chi(d)=1$ и $\chi(-1)=1$, или $\chi(1-d)=1$, или $\chi(d)=1$ и $\chi(d-1)=1$). Изогенная кривая E' во всех случаях лежит в классе квадратичных кривых Эдвардса.

Пример 3. Пусть $p=11$ и задана полная кривая Эдвардса $E=E_7: x^2+y^2=1+7x^2y^2$, где $\chi(d=7)=-1$, $\chi(1-d=4)=1$ и порядком $N_E=16$. Согласно теореме 1 существует лишь пара 2-изогенных квадратичных кривых Эдвардса $E'=E_4$ и $E'=E_3$ с параметрами $d_{1,2}=\bar{d}^{\pm 1}=\{4,3\}$ и отображением $\phi_2(x,y)$. Они имеют тот же порядок $N_E=16$ (что отвечает известной теореме Тейта [9]), изоморфны между собой, но вместо одной имеют уже 3 точки 2-го порядка (кривые являются нециклическими) и 12 точек 4-го порядка. Среди них по две особых точки 2-го и 4-го порядков. Точки исходной полной кривой E обозначим P_i, а точки двух изогенных кривых E' как Q_i Как и при удвоении точек, отображение $\phi_2(x,y)$ сжимает прообраз (кривую E) вдвое, т.е. отображает пару точек кривой E в одну точку кривой E'. В отличие от удвоения, 2-изогения не обязательно уменьшает вдвое порядок точки четного порядка.

На кривой E имеем точки $(\pm1,0)$, $(0,\pm1)$, $(\pm2,\pm4)$, $(\pm3,\pm3)$, $(\pm4,\pm2)$. Пусть $P_1=(2,4)$ – точка 16-го порядка кривой. $P_2=(3,3)=6P_1$ – точка 8-го порядка, $P_3=(4,2)=11P_1$. На изогенной кривой $E'=E_4$, кроме базовых точек $O=(1,0)$ $D_0=(-1,0)$, $\pm F_0=(0,\pm1)$, имеем особые точки $D_{1,2}=(\pm5,\infty)$, $\pm F_1=(\infty,\pm5)$, и точки 4-го порядка $(\pm2,\pm3)$, $(\pm3,\pm2)$. Обозначим $Q_1=(2,3)$, $Q_2=(3,2)$. $P^*=P+D_0$. С помощью первого значения функции $\phi_2(x,y)$ вычисляем:

$$\pm\phi_2(P_1,P_1^*)^{(1)}=(3,\pm2)=\pm Q_2,$$

$$\pm\phi_2(P_3,P_3^*)^{(1)}=(-3,\pm2)=\mp Q_2^*,$$

$$\pm\phi_2(P_2,P_2^*)^{(1)}=(\infty,\pm5)=\pm F_1,$$

$$\phi_2(\pm F_0)^{(1)}=(-1,0)=D_0,$$

$$\phi_2(D_0.O)^{(1)}=(1,0)=O.$$

Вторая изогенная кривая $E'=E_3$ с параметром $d=3$, кроме базовых точек, имеет особые точки $D_{1,2}=(\pm2,\infty)$, $\pm F_1=(\infty,\pm2)$, и точки 4-го порядка $(\pm4,\pm4),(\pm5,\pm5)$. Пусть $R_1=(4,4)$, $R_2=(5,5)$. Согласно второго значения функции $\phi_2(x,y)^{(2)}$ получаем:

$$\pm\phi_2(P_1,P_1^*)^{(2)}=(4,\mp4)=\mp R_1,$$

$$\pm\phi_2(P_3,P_3^*)^{(2)}=(-4,\mp4)=\pm R_1^*,$$

$$\pm\phi_2(P_2,P_2^*)^{(2)}=(0,\pm1)=\pm F_0,$$

$$\phi_2(\pm F_0)^{(2)}=(-1,0)=D_0,$$

$$\phi_2(D_0.O)^{(2)}=(1,0)=O.$$

Итак, функция $\phi_2(x,y)$ отображает пару точек одинаковых порядков кривой в одну точку кривой E' (т.е. функция $\phi_2(x,y)$ – сюръекция), причем одна полная кривая Эдвардса отображается в две изоморфные квадратичные кривые.

Рассмотрим изогении квадратичных кривых. Для отображения $\phi_2(x,y)$ справедливо свойство

$$1-\bar{d}_2^{\pm1}=1-\left(\frac{1\mp\gamma}{1\pm\gamma}\right)^2=\frac{\pm4\gamma}{(1\pm\gamma)^2}.$$

Отсюда следует, что из пары изогенных кривых для одной кривой $\chi(1-\bar{d}_2)=1$ и можно вновь пользоваться функцией $\phi_2(x,y)$.

Пример 4. Построим изогению для квадратичной кривой $E=E_3$ из примера 3 с параметрами $d=3, 1-d=9, \gamma=3$. Одна из изогенных кривых при отображении $\phi_2(x,y)$ имеет тот же параметр $d=3$ и

те же точки $R_1=(4,4)$, $R_2=(5,5)$, $D_{1,2}=(\pm 2,\infty)$, $\pm F_1=(\infty,\pm 2)$, $\pm F_0=(0,\pm 1)$, $D_0, O.$ Отображение $\phi_2(x,y)^{(2)}$ этой кривой дает точки кривой E'

$$\pm\phi_2(R_1,R_1^*)^{(2)}=(\infty,\mp 2)=\pm F_1,$$

$$\pm\phi_2(R_2,R_2^*)^{(2)}=(0,\mp 1)=\mp F_0,$$

$$\phi_2(\pm \mathrm{F}_1,)^{(2)}=(2,\infty)=D_1,$$

$$\phi_2(D_1,D_2\ )^{(2)}=(-2,\infty,)=D_2,$$

$$\phi_2(\pm F_0)^{(2)}=(-1,0)=D_0,$$

$$\phi_2(D_0.O)^{(2)}=(1,0)=O.$$

Если повторно применить функцию $\phi_2(x,y)^{(2)}$ к точкам изогенной кривой E', получим точки кривой E'':

$$\phi_2(D_0.O)^{(2)}=(1,0)=O,$$

$$\phi_2(\pm F_0)^{(2)}=(-1,0)=D_0,$$

$$\phi_2(\pm F_1,)^{(2)}=(2,\infty)=D_1,$$

$$\phi_2(D_1,D_2\ )^{(2)}=(-2,\infty,)=D_2.$$

Таким образом, вторая изогения возвращает нас в точки исходной кривой ($E''=E$), причем за 2 шага отображаемые точки кривой E удваиваются (умножаются на α: точки 4-го порядка отображаются в точки 2-го порядка, а точки 2-го порядка – в точку O). Это пример дуальной 2-изогении $\hat{\phi}_2=\phi_2(x,y)^{(2)}$ для квадратичной кривой над простым полем.

В общем случае для изогении $\phi: E\to E'$ существует дуальная изогения $\hat{\phi}: E'\to E$, такая, что $\phi\circ\hat{\phi}=[\deg(\phi)=\alpha]$, [9]. Формулы теоремы 1 доказывают, что над полем F_p для полных кривых Эдвардса дуальная изогения не существует, но она существует в расширении F_{p^2}. В этом расширении параметр d полной кривой E становится квадратом и она становится квадратичной. Для нахождения дуальной изогении $\hat{\phi}: E'\to E$, например, к функции $\phi_1(x,y)$ со значениями параметра изогенной кривой

$$\overline{d}_1^{\pm 1}=\left(\frac{1-\delta}{1+\delta}\right)^2$$

требуется решить обратную задачу: по известному значению d_1 кривой E надо вычислить одно из подходящих значений параметра $\overline{\delta}$ кривой E', которое определяется похожей формулой

$$\overline{\delta}^{\pm 1}=\left(\frac{1\mp\sqrt{d_1}}{1\pm\sqrt{d_1}}\right).$$

Отсюда видно, что отображение в полную кривую Эдвардса E с квадратичным невычетом d_1 существует лишь в расширении F_{p^2} .

Скрученные кривые Эдвардса лежат за пределами отображений теоремы 1. За исключением суперсингулярных кривых, скрученная кривая как квадратичное кручение квадратичной кривой Эдвардса имеет другой порядок и соответствующие изогении не существуют. Если, однако, кривой E является полная кривая Эдвардса, то обращением ее параметра $d\to d^{-1}$ получаем ее квадратичное кручение, а соответствующая ее изогения (квадратичная кривая) будет иметь порядок соответствующей скрученной кривой. Известным преимуществом скрученной кривой перед квадратичной является наличие лишь двух особых точек вместо четырех [5, 6]. Остается открытым вопрос: как построить прямую 2-изогению из класса полных в класс скрученных кривых над простым полем. Для этого следует модифицировать отображения теоремы 1 с учетом двух параметров a и d. Этот вопрос будет рассмотрен в следующей работе. Пока можно утверждать, что в связи с отсутствием точек 4-го порядка в классе скрученных кривых Эдвардса при $p\equiv 1 \bmod 4$ 2-изогений в этом классе над простым полем не существует.

Литература

[1] *Bernstein D.J., Lange T.* Faster Addition and Doubling on Elliptic Curves // Advances in Cryptology — ASIACRYPT'2007 (Proc. 13th Int. Conf. on the Theory and Application of Cryptology and Information Security. Kuching, Malaysia. December 2–6, 2007). Lect. Notes Comp. Sci. V. 4833. Berlin: Springer, 2007. P. 29–50.

[2] *Bernstein Daniel J. , Birkner Peter , Joye Marc , Lange Tanja, Peters Christiane.* Twisted Edwards Curves.// IST Programme under Contract IST–2002–507932 ECRYPT,and in part by the National Science Foundation under grant ITR–0716498, 2008, PP. 1-17.

[3] *Moody D.,, Shumow D. Analogues of Velus formulas for isogenies on alternate models of elliptic curves.* Cryptology ePrint Archive, Report 2011/430, http://eprint.iacr.org/, 2011.

[4] O. *Ahmadi O.,, and Granger R, On isogeny classes of Edwards curves over finite fields*, J. Number Theory, 132 (6), pp. 1337-1358, (2012).

[5] *Бессалов А.В.* Уникальные криптографические свойства нециклических скрученных кривых Эдвардса. Прикладная радиоэлектроника: научно-техн. журнал. – 2018. – Том 17. – №1, 2. – С. 49–54.

[6] *Бессалов А.В.* Эллиптические кривые в форме Эдвардса и криптография. Монография. «Политехника», Киев, 2017. – 272 с.

[7] *Бессалов А.В., Цыганкова О.В.* Взаимосвязь семейств точек больших порядков кривой Эдвардса над простым полем. Проблемы передачи информации. – Том 51, вып 4, 2015. –С. 92–98.

[8] *Бессалов А.В., Цыганкова О.В.* Число кривых в обобщенной форме Эдвардса с минимальным четным кофактором порядка кривой. Проблемы передачи информации, - Том 53 (1). – 2017. –С.101–111.

[9] *Washington L.C.* Elliptic Curvres. Number Theory and Cryptography. Second Edition. CRC Press, 2008.

Поступила в редколлегию 16.10.2018

Бессалов Анатолий Владимирович, доктор технических наук, профессор, профессор НТУУ «КПИ им. Игоря Сикорского». Область научных интересов – асимметричная криптография.

УДК 621.391.15:519.7

Бессалов А. В. **2-ізогенії повних і квадратичних кривих Едвардса над простим полем** / А. В. Бессалов // Прикладная радиоэлектроника: науч. – техн. журнал. – 2018. – Том 17, № 3, 4. – С. 152–159.

Дано аналіз умов існування 2-ізогеній повних і квадратичних кривих Едвардса над простим полем. Дано огляд властивостей трьох класів кривих в узагальненій формі Едвардса: повних, скручених і квадратичних кривих Едвардса. Для коректного запису відображаючох функцій і визначення степеню 2-ізогеній запропоновано застосовувати модифікований закон складання точок. Обговорюються проблеми знаходження дуальних 2-ізогеній між класами повних квадратичних і скручених кривих Едвардса.

Ключові слова: крива в узагальненій формі Едвардса, скручена крива Едвардса, квадратична крива Едвардса, порядок кривої, порядок точки, ізоморфізм, ізогенія, квадратичне кручення, квадратичний лишок, квадратичний нелишок.

Бібліогр. 09 найм.

UDC 621.391.15:519.7

Bessalov A. V. **2-isogenies of complete and quadratic Edwards curves over prime field** / A. V. Bessalov // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 152–159.

An analysis of the conditions for the existence of 2-isogenies of complete and quadratic Edwards curves over a prime field is given. An overview of the properties of the three classes of curves in the generalized Edwards form (complete, twisted, and quadratic Edwards curves) is considered. To correctly record the mapping functions and determine the degree of isogeny, the use of the modified law of addition of points is proposed. The problems of finding dual 2-isogenies between classes of complete, quadratic and twisted Edwards curves are discussed.

Keywords: curve in a generalized Edward form, twisted Edwards curve, quadratic Edwards curve, curve order, points order, addition of points, isomorphism, quadratic twist, square, non-square.

Ref.: 09 items.

УДК 004.056.55

МОДЕЛЬ БЕЗПЕКИ ПОСТКВАНТОВИХ ПРОТОКОЛІВ ІНКАПСУЛЯЦІЇ КЛЮЧІВ

М. В. ЄСІНА

У роботі розглядається модель безпеки постквантових протоколів інкапсуляції ключів Canetti-Krawczyk (CK). Наводяться основні положення стосовно протоколів. Досліджуються моделі неавтентифікованого та автентифікованого порушника. Досліджуються та наводяться приклади SK-безпечних протоколів.

Ключові слова: інкапсуляція ключів, модель безпеки, протокол.

ВСТУП

У критеріях відбору, які висуваються NIST США до кандидатів на постквантові стандарти криптографічного захисту інформації [12], визначено моделі безпеки, яким мають відповідати кандидати. Відповідно трьом кандидатам – асиметричний шифр (АСШ), цифровий підпис та протокол інкапсуляції ключів (ПІК), визначено три моделі безпеки. Стосовно АСШ – IND-CCA2 (IND-CPA, IND-CCA), для підпису – EUF-CMA модель та для ПІК – CK модель [12].

На наш погляд, проблемними сьогодні є питання, що стосуються узагальненого визначення та дослідження моделі безпеки постквантових ПІК, але з урахуванням основних положень та пропозицій, що викладені у [1–11].

Метою цієї статі є узагальнене визначення та дослідження моделей безпеки, зокрема визначення можливостей та умов застосування постквантових ПІК при протидії зі сторони класичного чи квантового порушника [1–11].

1. ВІДОМОСТІ ПРО ПРОТОКОЛИ

Взагалі під протоколами розумітимемо набори інтерактивних процедур, які одночасно виконуються взаємодіючими сторонами, які вказують на особливу обробку вхідних повідомлень та генерацію вихідних повідомлень. Протоколи можуть ініціювати підпротоколи або інші протоколи, і кілька копій таких протоколів можуть одночасно управлятися кожною стороною.

Протоколи обміну ключами (коротко KE) – це механізми, за допомогою яких дві сторони створюють секретний ключ, що в подальшому використовується для захисту інформації, що передається через мережу, до якої має доступ зловмисник (порушник). Протоколи KE є необхідними та виконуються для того, щоб дозволяти використання криптографії спільного ключа для захисту даних, що передаються через небезпечні мережі. Тому протоколи KE є важливими елементами в ході побудови безпечних зв'язків (наприклад, "захищених каналів"). Вони є одними з криптографічних протоколів, що найчастіше використовуються (наприклад, протоколи SSL, IPSec, SSH тощо [9]).

Згідно з [1, 8] протокол обміну ключами вважається безпечним, якщо зловмисник не може відрізнити значення ключа, що створюється KE протоколом, від незалежного випадкового значення. Протоколи KE, що перевірені та забезпечують доказову безпеку, називаються SK-безпечними протоколами. Їх можна використовувати в стандартних умовах надання "безпечних каналів".

PFS – perfect forward secrecy – досконала пряма безпека (секретність). PFS належить до властивості протоколів обміну ключами (KE), за допомогою якої розкриття довгострокового ключа, що використовується у протоколі для автентифікації та узгодження ключів сеансу, не ставить під загрозу секретність ключів сеансу, встановлених до розкриття. Найбільш поширеним способом досягнення PFS у протоколі обміну ключами є використання обміну Diffie-Hellman. Він виконується з тимчасовими показниками, щоб встановити значення сеансового ключа, обмежуючи використання довгострокових ключів для цілі автентифікації обміну.

2. МОДЕЛЬ БЕЗПЕКИ CANETTI-KRAWCZYK (CK)

Основною проблемою, що має бути вирішена для протидії загрозам в ході застосування класичних та особливо квантових комп'ютерів, є забезпечення безпеки встановлення (узгодження) ключів. У [9] такі криптопротоколи отримали назву протоколів інкапсуляції ключів. Як практична основа та нормативно-правової бази таких протоколів, певною мірою, рекомендується застосовувати стандарти ANSI NIST.

Проведений аналіз показав [1–8], що стосовно протоколів інкапсуляції ключів серед інших є важливим механізм, що грунтується на моделі безпеки Canetti-Krawczyk (CK). Модель CK включає в себе три основні компоненти: модель неавтентифікованого порушника (UM), модель автентифікованого порушника (AM) та механізм автентифікації (автентифікатор) (MT). Модель безпеки CK використовується для автентифікації обміну ключами (AKE) [1]. Розгляд вказаних моделей складає достатньо складну проблему. Вона розглядається нижче.

Сутність моделі CK. Модель безпеки CK стосується безпеки ключа сеансу, що використовується на сеансі зв'язку. В ході її оцінювання використовується формальна модель для протоколів обміну ключами та можливостей криптоаналітика (зловмисника). Понят-

тя безпеки, яке називається безпекою ключа сеансу (або SK-безпека), направлене на забезпечення безпеки окремих ключів сеансу. Її порушення є компрометацією сеансового ключа. У випадку безпечності ключа, зловмисник "нічого не дізнається про значення ключа", коли він перехоплює дані протоколу обміну ключами та здійснює атаки на інші сеанси та сторони, що взаємодіють. Такий підхід є стандартним для моделі семантичної безпеки, коли криптоаналітик не може відрізнити реальне значення ключа від незалежного випадкового значення. У даному випадку говорять про реалізацію криптографічного протоколу «нульових знань». Водночас таке визначення SK-безпеки необхідно використовувати обережно, тому що забезпечення необхідного рівня стійкості при встановленні та використанні протоколів обміну ключами для реалізації захищених каналів, може досягатись без складних вимог.

3. МОДЕЛЬ НЕАВТЕНТИФІКОВАНОГО ПОРУШНИКА (UM)

Для аналізу та оцінки безпеки протоколу, потрібно визначити можливості зловмисника, тобто можливі дії атакуючого. Необхідно, щоб ці можливості були максимально загальними (на відміну від, наприклад, просто подання списку можливих атак), але не має бути нереалістичних вимог. Визначимо формально зловмисника [1], але спеціалізуємо та розширимо його для випадку протоколів KE. Використовуватимемо термінологію [1], тобто модель, що називається моделлю неавтентифікованого порушника (UM).

Основні можливості атакуючого. Розглянемо імовірнісного атакуючого, що виконується за поліноміальний час (PPT), який має повний контроль над каналами зв'язку: він може прослуховувати всю передану інформацію, вирішувати, які повідомлення досягатимуть їх місця призначення, і коли змінювати ці повідомлення за бажанням або вставляти власні згенеровані повідомлення. Формалізм відображає цю здатність зловмисника, дозволяючи йому відповідати за передачу повідомлень від однієї сторони до іншої. Атакуючий також контролює планування всіх етапів протоколу, включаючи ініціювання протоколів та доставку повідомлень.

Отримання секретної інформації. На додаток до цих базових можливостей, зловмиснику дозволяється отримати секретну інформацію, що зберігається в пам'яті сторін шляхом явних атак. Розглядаємо всю секретну інформацію, що зберігається на стороні, як потенційно вразливу до вторгнення або інших видів витоку. Проте, під час визначення безпеки протоколу важливо гарантувати, що витік певної форми секретної інформації має, як мінімум, можливий вплив на безпеку інших секретів. Наприклад, необхідно гарантувати, що витік інформації, визначеної для одного сеансу (наприклад, витік сеансового ключа або інформації про тимчасовий стан), не матиме впливу на безпеку інших сеансів або, що навіть витік критичних довгострокових секретів (наприклад, особистих ключів), які використовуються під час кількох сеансів, не обов'язково скомпрометує секретну інформацію з усіх минулих сеансів. Також, щоб розрізняти різні вразливості та максимально забезпечити безпеку у разі розкриття інформації, класифікуємо атаки на три категорії залежно від типу інформації, яку хоче отримати зловмисник.

Розкриття стану сеансу. Атакуючий надає ім'я сторони та ідентифікатор сеансу ще неповного сеансу на цій стороні та отримує внутрішній стан цього сеансу (оскільки сеанси є процедури, що виконуються всередині сторони, то внутрішній стан сеансу добре визначений). Важливий момент полягає в тому, яка інформація міститься у локальному стані сеансу. Залишимо це для зазначення кожним протоколом KE. Тому визначення безпеки параметризується типом і кількістю інформації, що розкрита у цій атаці. Наприклад, виявлена таким чином інформація може бути показником x, який використовується стороною для обчислення значення g^x у протоколі обміну ключа Діффі-Гелмана або випадкових бітів, які використовуються для шифрування кількості у рамках імовірнісної схеми шифрування під час сеансу. Як правило, виявлена (розкрита) інформація включає в себе весь локальний стан сеансу та його підпрограми, за винятком локального стану підпрограм, які безпосередньо мають доступ до довгострокової секретної інформації, наприклад, локального ключа підпису/розшифрування криптосистеми з відкритим ключем або довгострокового спільного ключа.

Запит ключа сеансу. Атакуючий надає назву учасника (сторони) та сеансовий ідентифікатор завершеного сеансу на цій стороні, та отримує значення ключа, створеного названим сеансом. Ця атака передбачає формальне моделювання витоку інформації про певні сеансові ключі, що може виникнути внаслідок подібних порушень, криптоаналізу, необережного розпорядження ключами тощо. Воно також служитиме, опосередковано, для забезпечення того, що неминуче витікання інформації, виробленої шляхом використання ключів сеансу в додатку безпеки (наприклад, інформація, що витекла по ключу за допомогою його використання як ключа шифрування), не допоможе в отриманні додаткової інформації про цей та інші ключі.

Пошкодження сторони. Зловмисник може вирішити в будь-який момент пошкодити сторону, і в цьому випадку атакуючий вивчає всю внутрішню пам'ять цієї сторони, включаючи довгострокові секрети (наприклад, особисті ключі або спільні майстер-ключі, що використовуються для різних сеансів) та визначену інформацію про сеанси, що міститься в пам'яті сторони (наприклад, внутрішній стан неповних сеансів і ключів сеансу, відповідних завершеним сеансам). Оскільки, вивчаючи свої довгострокові секрети, атакуючий може видати себе за сторону у всіх

його діях, тоді сторона вважається повністю контрольованою зловмисником з моменту пошкодження і може, зокрема, відмовлятися від специфікацій протоколу.

Термінологія: якщо для сеансу застосовується будь-яка із зазначених вище трьох атак (тобто, розкриття стану сеансу, запит ключа сеансу або пошкодження сторони, яка проводить сеанс), то сеанс називається локально розкритим (locally exposed). Якщо сеанс або його відповідний сеанс локально розкритий, він називається розкритим (exposed) сеансом.

Закінчення терміну дії сеансу. Одним з важливих додаткових елементів моделі безпеки є поняття закінчення терміну дії сеансу. Це відбувається у формі дії протоколу, яка в ході активації призводить до стирання названого ключа сеансу (і будь-якого відповідного стану сеансу) з пам'яті цієї сторони. Дозволяється, щоб сеанс закінчився на одній стороні, не обов'язково закінчуючи відповідний сеанс. Ефект цієї дії у моделі безпеки полягає в тому, що значення сеансового ключа закінченого терміну дії неможливо знайти за допомогою будь-якої з наведених вище атак, якщо ці атаки виконуються після закінчення сеансу. Це має два важливі наслідки: це дозволяє моделювати загальну (та добру) практику безпеки обмеження терміну служби окремих ключів сеансу і дозволяє просте моделювання поняття досконалої прямої безпеки (секретності). Відзначається, що для того, щоб сеанс був локально розкритий (як зазначено вище), атака на сеанс має відбутися до закінчення сеансу.

Підтримка безпеки протоколів обміну ключами. Протоколи обміну ключами, як і інші криптографічні додатки, вимагають підтримки безпеки (особливо для автентифікації) за допомогою деяких передбачених засобів захисту. Приклади включають безпечне створення особистих ключів сторін, встановлення відкритих ключів інших сторін або встановлення спільних майстер-ключів. Тут також дотримуємось підходу [1], де підтримка функцій автентифікації абстрагується у функцію ініціалізації, яка виконується до початку будь-якого протоколу обміну ключами і яка забезпечує безпечний спосіб (тобто без участі зловмисника) необхідної (довгострокової) інформації. Абстрагуючись від цієї початкової фази, дозволяється комбінувати різні протоколи з різними функціями ініціалізації: зокрема, це дозволяє нашому аналізу протоколів (наприклад, Діффі-Гелмана) застосовуватись з двома поширеними параметрами автентифікації: симетрична та асиметрична автентифікація. Два зауваження: (1) специфікація функції ініціалізації є частиною визначення кожного протоколу KE; (2) секретна інформація, сформована цією функцією на заданій стороні, може бути виявлена атакуючим лише після порушення тієї сторони. Підкреслюється, що, хоча ця абстракція додає простоту та застосовність методів аналізу, підтримка безпеки в дійсних протоколах є елементом, який необхідно ретельно проаналізувати (наприклад, взаємодія з CA у випадку протоколів на основі відкритих ключів). Інтеграція цих явних елементів у модель може бути зроблена або безпосередньо, як зроблено у [7], або більш модульним способом через відповідний склад протоколу.

4. МОДЕЛЬ АВТЕНТИФІКОВАНОГО ПОРУШНИКА (AM), ЕМУЛЯЦІЯ ПРОТОКОЛУ ТА АВТЕНТИФІКАТОР

Модель порушника, яка називається моделлю автентифікованого порушника (AM), визначається таким чином, що є ідентичним для UM з однією принциповою різницею: атакуючий обмежується лише доставкою повідомлень, насправді породжених сторонами без будь-яких змін або доповнень до них. Потім вводять поняття "емуляція", щоб відобразити еквівалентність функціональних можливостей між протоколами в різних моделях порушника, зокрема між UM та AM. Протокол π' емулює протокол π в UM, якщо для будь-якого зловмисника, який взаємодіє з π' в UM, існує зловмисник, який взаємодіє з π в AM так, що обидві взаємодії "виглядають однаково" для зовнішнього спостерігача. Розробляються спеціальні алгоритми, які називаються автентифікаторами, з властивістю, що під час введення опису протоколу π автентифікатор виводить опис протоколу π' такого, що π' емулює протокол π в UM. Тобто автентифікатори виконують функцію автоматичного "компілятора", який перетворює протоколи в AM на еквівалентні (або "так само безпечні, як") протоколи в UM.

Для спрощення побудови автентифікаторів [1] пропонується наступна методологія. Спочатку розглянемо дуже простий однопотоковий (одношаровий) протокол в AM, що називається MT, єдиним функціоналом якого є передача одного повідомлення від відправника до одержувача. Тепер створимо автентифікатор з обмеженим типом, який називається MT-автентифікатором, для забезпечення емуляції тільки для цього конкретного протоколу MT. Нарешті, для будь-якого такого MT-автентифікатора λ, зіставляється один алгоритм (або компілятор) $C\lambda$, який перетворює будь-який вхідний протокол π на інший протокол π' таким чином: до кожного з повідомлень, визначених у протоколі π, застосовується MT-автентифікатор λ. У [1] доведено, що $C\lambda$ є автентифікатором (тобто, результуючий протокол π' емулює π в UM). Особливі реалізації MT-автентифікаторів подані у [1] на основі криптографічних функцій різних типів (наприклад, ЕП, шифрування з відкритим ключем, MAC та ін.).

5. ВИЗНАЧЕННЯ SK-БЕЗПЕКИ

Спочатку подано визначення для UM. Формалізація в AM є аналогічною. Почнемо з визначення "експерименту", де зловмисник U вибирає сеанс, в якому потрібно "перевіряти" інформацію, яку він дізнався з ключа сеансу; зокрема, попросимо зловмисни-

ка відрізнити реальне значення вибраного ключа сеансу від випадкового значення.

Для цього експерименту розширюємо звичайні можливості зловмисника, *U*, в UM, дозволяючи йому виконувати запит на тестовий сеанс. Тобто, на додаток до звичайних дій *U* проти протоколу обміну ключами π, *U* дозволяється у будь-який час під час його виконання тестувати сеанс серед сеансів, що були завершені, нереалізовані або нерозкриті на той час. Нехай к – значення відповідного ключа сеансу. Кидаємо монету *b*, $b \xleftarrow{R} \{0,1\}$. Якщо *b*=0, ми забезпечуємо *U* значенням к. В іншому випадку надаємо *U* значення *r*, випадковим чином обране з розподілу ймовірності ключів, створених протоколом π. Зловмиснику *U* тепер дозволено продовжувати регулярні дії UM-зловмисника, але не дозволяється розкривати тестовий сеанс (а саме, не дозволено розкривати стан сеансу, запити на ключі сеансу або пошкоджувати партнера при тестуванні сеансу, або його відповідний сеанс.) В кінці його запуску, *U* виводить біт *b*' (як його припущення для *b*).

Посилатимемося на зловмисника, який дозволяє запити тестового сеансу як KE-зловмисник.

Визначення 1. KE-протокол π називається SK-безпечним, якщо для будь-якого KE-зловмисника *U* в UM існують такі властивості:

1) протокол π задовольняє властивість, що, якщо дві непошкоджені сторони виконують відповідні сеанси, то вони обидві виводять однаковий ключ, та

2) імовірність того, що *U* правильно вгадає біт *b* (тобто виходи *b*'=*b*), не перевищує 1/2 плюс незначну частку в параметрі безпеки.

Якщо вищезазначені властивості задовольняються для всіх KE-зловмисників в AM, то π є SK-безпечним в AM.

Перша умова – це "послідовність" вимоги до сеансів, виконаних двома непошкодженими сторонами. Немає жодних вимог щодо значення сеансового ключа сеансу, де один із партнерів був пошкоджений до завершення сеансу – фактично більшість протоколів KE дозволяють пошкодженій стороні сильно вплинути на обмін ключами. Друга умова – "основна властивість" для SK-безпеки. Зазначимо, що термін "незначний", як звичайно, належить до будь-якої функції (в параметрі безпеки), яка асимптотично зменшується швидше, ніж будь-яка частка поліному. (Це формулювання, за бажанням, дозволяє кількісно оцінювати безпеку за допомогою конкретної процедури безпеки. У цьому випадку кількісно визначається потужність атакуючого через певні межі часу обчислення, кількість пошкоджень тощо, тоді як її перевага обмежується через певний параметр ε.)

Виділяється наступні три аспекти Визначення 1:

– атакуючий може продовжувати працювати і атакувати протокол навіть після отримання відповіді (реальної чи випадкової) на його запит тестового сеансу. Ця здатність (яка є суттєвим посиленням безпеки відносно [2, 3]) є важливою для підтвердження основної властивості SK-безпеки;

– зловмисник не може пошкодити учасників тестового сеансу або випускати будь-яку іншу команду викриття проти цього сеансу, поки він не існує. Це відображає той факт, що неможливо гаран-тувати безпечне використання ключа сеансу, який було виставлено шляхом втручання зловмисника (або криптоаналізу). Зокрема, це обмеження має важливе значення для підтвердження безпеки окремих важливих протоколів (наприклад, обміну ключами Diffie-Hellman);

– наведене вище обмеження для зловмисника, за допомогою якого він не може пошкодити партнера для тестового сеансу, скасовується, як тільки закінчується сеанс цього партнера. У цьому випадку зловмисник повинен залишатися нездатним розрізняти дійсне значення ключа та випадкове значення. Це є основою для гарантії досконалої прямої безпеки (секретності).

6. ПРЯМА БЕЗПЕКА (СЕКРЕТНІСТЬ)

Неформально поняття "досконала пряма безпека (секретність)" (PFS) [4, 5] зазначається як властивість, що "компрометування довгострокових ключів не компрометує минулі ключі сеансу".

Коли доводять, що протокол має бути SK-безпечним за допомогою Визначення 1, автоматично отримується доказ того, що цей протокол гарантує PFS.

Визначення 2. Говорять, що протокол KE задовольняє SK-безпеку без PFS, якщо він використовує SK-безпеку відносно будь-якого KE-зловмисника в UM, що не допускає припинення дії ключів. (Аналогічно, якщо вищесказане виконується для будь-яких таких зловмисників у AM, то говорять, що π є SK-безпечним без PFS в AM).

7. SK-БЕЗПЕЧНІ ПРОТОКОЛИ

7.1. Двопрохідний протокол Diffie-Hellman

Протокол 2DH

Загальна інформація: Прості числа *p*, *q*, *q*|*p*–1, та *g* порядку *q* в *Z*p*.

Крок 1: Ініціатор P_i, на вході (P_i, P_j, *s*), вибирає $x \xleftarrow{R} Z_q$ і посилає (P_i, *s*, $\alpha=g^x$) до P_j.

Крок 2: При отриманні (P_i, *s*, α) відповідач P_j вибирає $y \xleftarrow{R} Z_q$, надсилає (P_j, *s*, $\beta=g^y$) до P_i, стирає *y* і виводить ключ сеансу $\gamma'=\alpha^y$ під ідентифікатором (id) сеансу *s*.

Крок 3: При отриманні (P_j, *s*, β) сторона P_i обчислює $\gamma'=\beta^x$, стирає *x* та виводить ключ сеансу γ' під ідентифікатором (id) сеансу *s*.

Рис. 1. Двопрохідний протокол Diffie-Hellman у AM

7.2 SK-безпечний протокол Diffie-Hellman в UM

Зауваження по протоколу SIG-DH. Протокол є результатом застосування автентифікатора на основі підпису [1] до двопрохідного протоколу Diffie-Hellman, наведеного на рис. 1, де значення α та β (експоненти DH) служать викликами, що вимагаються автентифікатором на основі підпису. Це припускає (як зазначено у протоколі 2DH), що ці експоненти вибираються знов для кожного нового обміну (інакше кожна сторона може додати явне виключення до повідомлень, які також включені у підпис). Зауважимо, що ідентифікатор сторони-отримувача, що входить до складу підписів, є частиною специфікації автентифікатора на основі підпису [1] і є основою для забезпечення безпеки протоколу.

8. ПРОТОКОЛ SIG-DH

Опис SIG-DH на рис. 2 передбачає, як формалізовано в моделі, що значення s ідентифікатора (id) сеансу надано сторонам. На практиці, як правило, генерується ідентифікатор сеансу s як пара (s_1, s_2), де s_1 – це значення, обране P_i, та інше (з дуже великою ймовірністю) з усіх інших таких значень, обраних P_i у його інших сеансах з P_j. Аналогічно, s_2 вибирається P_j з аналогічною властивістю єдиності. Ці значення s_1, s_2 можуть бути обмінені сторонами як попередня частина до вищевказаного протоколу (це може бути у випадку протоколів, які реалізують таку попередню частину для обміну деякою іншою системною інформацією та для обговорення пара-метрів обміну [6]). Крім того, s_1 може бути включено P_i у перше повідомлення SIG-DH, а s_2 може бути включено P_j у друге повідомлення. У будь-якому випадку, важливо, щоб ці значення були включені під підписи сторін.

Крок 1: Ініціатор P_i, на вході (P_i, P_j, s), вибирає $x \xleftarrow{R} Z_q$ і посилає (P_i, s, α=g^x) до P_j.

Крок 2: При отриманні (P_i, s, α) відповідач P_j вибирає $y \xleftarrow{R} Z_q$ та надсилає до P_i повідомлення (P_j, s, β=g^y) разом з його підписом $SIG_j(P_j, s, \beta, \alpha, P_i)$; він також обчислює ключ сеансу γ'=α^y та стирає y.

Крок 3: При отриманні (P_j, s, β) та підпису P_j, сторона P_i перевіряє підпис та правильність значень, що входять до підпису (такі як ідентифікатори гравців, ідентифікатор (id) сеансу, значення показників тощо). Якщо перевірка була успішною, тоді P_i надсилає P_j повідомлення (P_i, s, $SIG_j(P_i, s, \alpha, \beta, P_j)$), обчислює γ'=β^x, стирає x та виводить ключ сеансу γ' під ідентифікатором (id) сеансу s.

Крок 4: Після отримання кортежу (P_i, s, sig), P_j перевіряє підпис P_i sig і значення, що він включає. Якщо перевірка буде успішною, вона виведе ключ сеансу γ під ідентифікатором (id) сеансу s.

Рис. 2. Протокол Diffie-Hellman у UM: автентифікація за допомогою підписів

Загальна інформація: Прості числа p, q, $q|p-1$, та g порядку q в Z^*p. Кожен гравець має особистий ключ для алгоритму підпису SIG, і всі мають від-криті ключі перевірки інших гравців.

9. ЗАСТОСУВАННЯ ДЛЯ ЗАХИЩЕНИХ КАНАЛІВ

Шаблонний протокол: Мережний Канал. Шаблонний протокол, який називається NetChan, застосовується до моделі неавтентифікованого порушника UM, а також до моделі автентифікованого порушника AM. Далі розглядаються конкретні реалізації цього шаблонного протоколу, де загальні примітиви 'відправити' та 'прийняти', визначені там, є екземплярами з дійсними функціями (наприклад, для забезпечення автентифікації та/або шифрування). Також визначається, що означає, що така реалізація буде "безпечною".

Протокол мережного каналу (канал сеансу) NetChan(π, snd, rcv), визначається на основі KE-протоколу π, а також двох загальних функцій snd та rcv. Обидві snd та rcv – це імовірнісні функції, які в якості аргументів використовують ключ сеансу (позначається цей ключ як індексний символ до функції) та повідомлення m. Ці функції також можуть залежати від інших даних сеансу, таких як ідентифікатор сеансу та ідентифікатори партнерів (сторін). Вихідні дані snd є єдиним значенням m', тоді як виходом rcv є пара (v; ok), де ok – біт, і v – довільне значення. (Біт ok буде використано, щоб повернути значення перевірки, наприклад, результат перевірки тегу автентифікації.) На основі таких функцій визначаємо NetChan(π, snd, rcv) на рис. 3 [2–5].

Мережна автентифікація. На підставі вищезгаданого формального визначення розглядається випадок мережних каналів, що забезпечують автентифікацію інформації через канали, що контролюються зловмисником. А саме, ми зацікавлені в протоколі NetChan, який працює в моделі неавтентифікованого порушника UM, та забезпечує автентичність переданих повідомлень. Ця реалізація NetChan (яка називається NetAut) буде спрямована на захоплення практики, за допомогою якої взаємодіючі сторони використовують протокол обміну ключами, щоб створити загальний ключ сеансу, і використовувати цей ключ для автентифікації (за допомогою функції автентифікації повідомлень, MAC) інформації, якою обмінюються під час цього сеансу. А саме, якщо P_i та P_j поділяють відповідний сеанс s і P_i хоче відправити повідомлення m до P_j протягом цього сеансу, то P_i передає m разом з MACк(m), де к – відповідний ключ сеансу. Таким чином, в цьому випадку ілюструватимемо прикладами (підтверджувати) snd та rcv функції NetChan за допомогою функції MAC, як показано нижче [5].

10. ПРОТОКОЛ NETAUT

Нехай π є протоколом KE і нехай f – функція MAC. Протокол NetAut(π, f) – це протокол NetChan(π, snd, rcv), як визначено на рис. 3, де функції snd та rcv визначаються як [2–5]:

– на вході m, $\text{snd}_\kappa(m)$ виробляє вихід $m'=(m, t)=(m, f_k(m))$.

– на вході m', $\text{rcv}(m')$ виводить (v, ok) наступним чином. Якщо m' має форму (m, t), та пара (m, t) проходить функцію перевірки f за ключем к, то ok=1 та $v=m$. Інакше ok=0 та v=null.

Протокол NetChan(π, snd, rcv)

NetChan(π, snd, rcv) ініціалізується з тією ж функцією ініціалізації I протоколу KE π. Потім він може бути викликаний в межах сторони P_i за наступними діями:

1. establish-session(P_i, P_j, s, role): запускає (викликає) KE-сеанс з π всередині P_i з партнером P_j, ідентифікатором сеансу s та role ∈ {initiator; responder} ({ініціатор; відповідач}). Якщо KE-сеанс завершує записи P_i у своєму локальному виводі "встановлено сеанс s з P_j " і зберігає створений сеансовий ключ.

2. expire-session(P_i, P_j, s): P_i позначає сеанс (P_i, P_j, s) (якщо він існує у P_i), як такий, що закінчився, і ключ сеансу стирається. P_i записи в локальному виводі "сеанс s з P_j закінчився".

3. send(P_i, P_j, s, m): P_i перевіряє, що сеанс (P_i, P_j, s) був завершений, і не закінчився, якщо так, він обчислює $m'=\text{snd}_\kappa(m)$, використовуючи відповідний ключ сеансу к, відправляє (P_i, s, m') у P_j, і записує "відправлено повідомлення m до P_j протягом сеансу s" у локальному виводі.

4. На вхідному повідомленні (P_j, s, m') P_i перевіряє, чи сеанс (P_i, P_j, s) був завершений і не закінчився, якщо так, він обчислює $(m, \text{ok})=\text{rcv}_\kappa(m')$ з відповідним сеансовим ключем к. Якщо ok=1, то P_i записує "отримане повідомлення m від P_j протягом сеансу s". Якщо ok=0, то подальших дій не виконується.

Рис. 3. Загальний протокол мережних каналів

11. ПРОТОКОЛ SMT

Розширюється протокол MT з [1], щоб відповідати налаштуванню на основі сеансу, в якому передані повідомлення групуються у різні сеанси. Розширений протокол називається протоколом передачі повідомлень на основі сеансу (SMT) та визначається на рис. 4. Зверніть увагу на структурну схожість між SMT і NetChan – відмінності полягають у тому, що в SMT фактичний обмін ключами не виконується, а функції snd та rcv є екземплярами простих "ідентифікаційних функцій".

Протокол SMT забезпечує цілком автентичний обмін повідомленнями. Реалізація протоколу NetChan є безпечним протоколом мережної автентифікації, якщо він емулює протокол SMT в UM.

Протокол SMT може бути викликаний у стороні P_i за такими діями:

1. establish-session(P_i, P_j, s): у цьому випадку P_i записує у своєму локальному виводі "встановлені сеанси s з P_j".

2. expire-session(P_i, P_j, s): в цьому випадку P_i записує у своєму локальному виводі "сеанс s з P_j закінчився".

3. send(P_i, P_j, s, m): у цьому випадку P_i перевіряє, чи сеанс (P_i, P_j, s) був встановлений і не закінчився, якщо так, то він посилає повідомлення m до P_j разом з ідентифікатором сеансу (тобто значення m та s надсилаються по ідеально-автентифікованому зв'язку між P_i і P_j); P_i записує у своєму локальному виводі "відправлено повідомлення m до P_j протягом сеансу s".

4. На вхідному повідомленні (m, s), отриманому за його посиланням від P_j, P_i перевіряє, чи сеанс (P_i, P_j, s) встановлений, і не закінчився, якщо так, він записує у локальному виводі "отримано повідомлення m від P_j протягом сеансу s".

Рис. 4. SMT: Протокол MT на основі сеансу в AM

ВИСНОВКИ

1. Згідно з аналізом визначено, що стосовно ПІК постквантового періоду, перспективною є модель безпеки Canetti-Krawczyk (CK). Модель CK включає в себе три основні складові компоненти: модель неавтентифікованого порушника (UM), модель автентифікованого порушника (AM) та механізм автентифікації (автентифікатор) (MT). Як правило модель безпеки CK використовується для автентифікації обміну ключами (AKE).

2. Модель безпеки CK стосується безпеки ключа сеансу, що використовується на сеансі зв'язку. В ході оцінки протоколів обміну ключами та можливостей криптоаналітика (зловмисника) використовується формальна модель. Поняття безпеки, яке називається безпекою ключа сеансу (або SK-безпека), направлене на забезпечення безпеки окремих ключів сеансу. Її порушення може призвести до компрометації ключа сеансу. У випадку безпечності ключа зловмисник "нічого не дізнається про значення ключа", коли він перехвачує дані протоколу обміну ключами та здійснює атаки на інші сеанси та сторони, що взаємодіють. Такий підхід є стандартним для моделі семантичної безпеки, коли криптоаналітик не може відрізнити реальне значення ключа від незалежного випадкового значення.

3. Протоколи обміну ключами (KE) – це механізми, за допомогою яких дві сторони створюють

секретний ключ, що в подальшому використовується для захисту інформації, що передається через мережу, до якої має доступ зловмисник (порушник). Протоколи KE, що перевірені та забезпечують доказову безпеку, називаються SK-безпечними протоколами. Їх можна використовувати в стандартних умовах надання "безпечних каналів".

4. Протокол KE задовольняє SK-безпеку без PFS, якщо він використовує SK-безпеку відносно будь-якого KE-зловмисника в UM, що не допускає припинення дії ключів. KE-протокол π називається SK-безпечним, якщо для будь-якого KE-зловмисника U в UM існують такі властивості:

– протокол π задовольняє властивість, що, якщо дві непошкоджені сторони виконують відповідні сеанси, то вони обидві виводять однаковий ключ;

– імовірність того, що U правильно вгадає біт b (тобто виходи $b'=b$), не перевищує 1/2 плюс незначну частку в параметрі безпеки.

5. SK-безпечним протоколом є двопрохідний Diffie-Hellman, протокол SIG-DH та протокол NetAut. Протокол SMT забезпечує цілком автентичний обмін повідомленнями. Реалізація протоколу NetChan є безпечним протоколом мережної автентифікації, якщо він емулює протокол SMT в UM.

6. PFS – це досконала пряма безпека (секретність). PFS належить до властивості протоколів обміну ключами (KE), за допомогою якої розкриття довгострокових ключових даних, що використовується у протоколі для автентифікації та узгодження ключів сеансу, не ставить під загрозу секретність ключів сеансу, встановлених до розкриття.

7. Модель порушника, яка називається моделлю автентифікованого порушника (AM), визначається так, що є ідентичним для UM з однією принциповою різницею: атакуючий обмежується лише доставкою повідомлень, насправді породжених сторонами без будь-яких змін або доповнень до них. Потім вводять поняття "емуляція", щоб відобразити еквівалентність функціональних можливостей між протоколами в різних моделях порушника, зокрема між UM та AM. Протокол π' емулює протокол π в UM, якщо для будь-якого зловмисника, який взаємодіє з π' в UM, існує зловмисник, який взаємодіє з π в AM так, що обидві взаємодії "виглядають однаково" для зовнішнього спостерігача.

8. В моделі UM можливості противника суттєво розширюються, тобто U противник в UM моделі може виконувати запит на тестовий сеанс. На додаток до звичайних дій U у будь-який час під час його виконання може за вибором тестувати певний сеанс, який чи які були завершені, нереалізовані або нерозкриті на той час. Нехай k – значення відповідного ключа сеансу. За цих умов, по суті, підкидається монета b, $b \xleftarrow{R} \{0,1\}$.

9. Окремою особливістю щодо ключів є нерозрізнюваність ключів, модель безпеки, щодо якої безпека визначена як IK-CPA/CCA2. Сутність її в тому, що конфіденційність ключа має забезпечуватися при атаках на основі підібраного(вибраного) відкритого тексту та підібраного(вибраного) шифр-тексту. При таких атаках зловмисник протидіє в два етапи. На етапі find він приймає два відкриті ключі pk_0 і pk_1 (що відповідають секретним ключам sk_0 та sk_1, відповідно) і виводить повідомлення x разом з деякою інформацією про стан s секретних ключів. На етапі guess він викликає шифртекст y, який утворюється шляхом випадкового зашифрування повідомлень з одним із двох ключів, і повинен визначити, який ключ був вибраний.

Література

[1] *Ran Canetti*, *Hugo Krawczyk* Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. – Режим доступу: http://iacr.org/archive/eurocrypt2001/20450451.pdf.

[2] *V. Shoup* On Formal Models for Secure Key Exchange, Theory of Cryptography Library, 1999. – Режим доступу: http://philby.ucsd.edu/cryptolib/1999/9912.html.

[3] *M. Bellare, R. Canetti, H. Krawczyk* A modular approach to the design and analysis of authentication and key-exchange protocols. – 30th STOC. – 1998.

[4] *M. Bellare, E. Petrank, C. Rackoff, P. Rogaway* Authenticated key exchange in the public key model, manuscript. – 1995-96.

[5] *M. Bellare, P. Rogaway* Entity authentication and key distribution, Advances in Cryptology, – CRYPTO'93, Lecture Notes in Computer Science Vol. 773, D. Stinson ed, Springer-Verlag, 1994. – pp. 232-249.

[6] *W. Diffie, P. van Oorschot, M. Wiener* Authentication and authenticated key exchanges, Designs, Codes and Cryptography, 2, 1992. – pp. 107-125.

[7] *C.G. Gunther* An identity-based key-exchange protocol, Advances in Cryptology – EUROCRYPT'89, Lecture Notes in Computer Science Vol. 434, Springer-Verlag, 1990. – pp. 29-37.

[8] *D. Harkins, D. Carrel, ed.* The Internet Key Exchange (IKE), RFC 2409, November 1998.

[9] *Горбенко І.Д., Горбенко Ю.І.* Прикладна криптологія. Підручник. Х. : «Форт», 2013. – 878 с.

[10] *Yoshida Y., Morozov K., Tanaka K.* CCA2 Key-Privacy for Code-Based Encryption in the Standard Model. In: Lange T., Takagi T. (eds) Post-Quantum Cryptography. PQCrypto 2017. Lecture Notes in Computer Science, vol 10346. Springer, Cham.

[11] *M. Bellare, A. Boldyreva, A. Desai, D. Pointcheval* Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248. – pp. 566–582. Springer, Heidelberg (2001). doi:10.1007/3-540-45682-1 33.

[12] Post-Quantum Cryptography. – Electronic resource. – Access mode: https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

Надійшла до редколегії 25.12.2018

Єсіна Марина Віталіївна, канд. техн. наук, старший викладач кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Харківського національного університету імені В. Н. Каразіна. Галузь наукових інтересів – захист інформації, постквантова криптографія.

УДК 004.056.55

Есина М. В. **Модель безопасности постквантовых протоколов инкапсуляции ключей** / М.В. Есина // Прикладная радиоэлектроника: науч.-техн. журнал. – 2018. – Том 17. № 3,4. – С. – 160–167.

В работе рассматривается модель безопасности постквантовых протоколов инкапсуляции ключей Canetti-Krawczyk (CK). Приводятся основные положения относительно протоколов. Исследуются модели неаутентифицированного и аутентифицированного нарушителя. Исследуются и приводятся примеры SK-безопасных протоколов.

Ключевые слова: инкапсуляция ключей, модель безопасности, протокол.

Ил.: 04. Библиогр.: 12 назв.

UDC 004.056.55

Yesina M. V. **Security model of post-quantum key incapsulation protocols** / M. V. Yesina // Applied Radio Electronics: Sci. Joum. – 2018. – Vol. 17. № 3, 4. – P. 160–167.

The paper considers the security model of the post-quantum Canetti-Krawczyk (CK) key encapsulation protocols. The main positions of the protocols are given. Unauthenticated and authenticated attacker models are explored. Examples of SK-secure protocols are investigated and provided.

Key words: key encapsulation, security model, protocol.

Fig. 04. Ref.: 12 items.

УДК 621.391:519.2

МЕТОД РОЗПІЗНАВАННЯ К-ВИМІРНОСТІ БУЛЕВИХ ФУНКЦІЙ, ЗАДАНИХ ЗА ДОПОМОГОЮ ОРАКУЛІВ

С. М. КОНЮШОК

Запропоновано ймовірнісний метод розпізнавання k-вимірности булевих функцій, заданих за допомогою оракулів, який має меншу трудомісткість та характеризується меншою ймовірністю помилки першого роду (при такій самій верхній межі ймовірності помилки другого роду) порівняно з аналогічним раніше відомим методом.

Ключові слова: перевірка властивостей булевих функцій, ймовірнісний метод, k-вимірна функція, перетворення Уолша-Адамара.

ВСТУП

Булеві функції (далі – БФ) є необхідними криптографічними примітивами [1]; що нерідко відіграють ключову роль у створенні багатьох потокових та блокових шифрів. У таких випадках, важливою складовою дослідження криптографічної стійкості шифрів є аналіз криптографічних властивостей відповідних булевих функцій [2]. Тому дослідження криптографічних властивостей БФ не тільки входить до переліку важливих інструментів криптоаналітика, але також має суттєве значення для оцінки стійкості криптографічних алгоритмів в процесі їх синтезу.

Слід зауважити, що криптографічні властивості БФ не є незалежними одна від одної, вони мають певні зв'язки та накладають деякі обмеження одна до одної, і це означає, що неможливо знайти функцію, яка дозволяє досягати найкращих значень за кожною з цих властивостей. Як наслідок, побудова булевої функції, яка забезпечує прийнятні з точки зору практичного використання криптографічні властивості, фактично полягає в досягненні певного стану умовного оптимуму шляхом компромісного зниження вимог до окремих криптографічних властивостей задля досягнення прийнятних значень іншими криптографічними властивостями даної БФ.

Пошук такої булевої функції нерідко може бути пов'язаний зі значним обсягом досліджень великої кількості функцій-претендентів на роль такого важливого елементу криптографічного алгоритму в процесі його розробки.

Таким чином, актуалною є задача зменшення трудомісткості визначення або оцінки значень показників, що характеризують криптографічні властивості булевих функцій шляхом пошуку новітніх або удосконалення наявних підходів.

Розглянемо більш детально одну з важливих криптографічних властивостей булевих функцій.

Так, один із підходів до застосування БФ у потокових шифрах – діяти як функція ускладнення. Задля високої криптографічної стійкості шифру, така функція повинна мати високу нелінійність. Однак, з іншої точки зору, нелінійна функція може допускати представлення себе як суперпозиції лінійних функцій та більш простої нелінійної функції. Формально, така властивість виглядає наступним чином.

Булеву функцію $g:\{0,1\}^n \to \{0,1\}$ називають k-вимірною [3, 4], $0 \le k \le n$, якщо існують функція $\phi:\{0,1\}^k \to \{0,1\}$ та $n \times k$-матриця A над полем з двох елементів такі, що для будь-якого $x \in \{0,1\}^n$ справедлива рівність $g(x) = \phi(xA)$. Функцію g називають алгебраїчно виродженою, якщо вона є k-вимірною для деякого $k < n$ та невиродженою – в іншому випадку [1, 5 – 7].

Перші результати про кореляційні властивості алгебраїчно вироджених булевих функцій належать до 70-х років минулого століття [5]. Дослідження кореляційних властивостей булевих функцій обумовлене задачами криптографічного аналізу та теорії кодування. Відзначимо роботи [8 – 10], де викладений ряд атак на генератори гами потокових шифрів, функції ускладнення яких є алгебраїчно виродженими або близькими до таких.

Наразі, задача побудови ефективних підходів до розпізнавання властивості k-вимірності булевих функцій, є актуальною, як для оцінки стійкості заданого шифру, так і з метою реалізації криптоаналітичної атаки на шифратор, як на "чорну скриньку" (тобто шифратор виступає в ролі оракула).

В [3] був запропонований ймовірнісний алгоритм розпізнавання властивостей k-вимірності. Для будь-якої функції $f:\{0,1\}^n \to \{0,1\}$, що задана за допомогою оракула, та чисел $k \in \{0,1,...,n-1\}$, $\varepsilon \in (0,1)$ цей алгоритм дозволяє перевірити гіпотезу H_0: f – k-вимірна БФ проти альтернативи H_1, яка полягає в тому, що f знаходиться на відстані (Гемінга) не меньше $2^n\varepsilon$ від множини k-вимірних функцій.

Зазначений алгоритм полягає в генерації незалежних випадкових рівноймовірних векторів $h_1, ..., h_l \in V_n$ та переверці рівностей

$$f(h_j \oplus Z_{ij}) = f(Z_{ij}),\ i \in \overline{1,m} \qquad (1)$$

для кожного $j \in \overline{1,l}$, де Z_{ij} – незалежні в сукупності випадкові рівноймовірні вектори з V_n, що не залежать від $h_1, ..., h_l$. Позначимо ν_l число значень $j \in \overline{1,l}$, для яких виконуються рівності (1). Тоді гіпотеза H_0 приймається, якщо $\nu_l \cdot l^{-1} \geq 0{,}9 \cdot 2^{-k}$ та відхиляється у протилежному випадку. В [3] пропонується вибрати $l = 2^k C$, $m = 2^k k\, \varepsilon^{-1} C'$, де $C, C' = const$, що зводить до оцінки трудомісткості алгоритму $O(2^{2k} k\, \varepsilon^{-1})$ запитів до оракула f (або $O(n 2^{2k} k\, \varepsilon^{-1})$ двійкових операцій).

Для оцінювання ймовірності помилки першого роду (тобто ймовірності того, що тест "не визнає" такою k-вимірну функцію) в [3] використовується нерівність Чернова:

$$\mathrm{P}\left(\frac{\nu_l}{l} < 0{,}9 \cdot 2^{-k} \mid H_0\right) \leq$$

$$\leq \mathrm{P}\left(\frac{\nu_l}{l} - \mathrm{E}\frac{\nu_l}{l} < -0{,}1 \cdot 2^{-k} \mid H_0\right) \leq$$

$$\leq \exp\left\{-0{,}02 \cdot \frac{C}{2^k}\right\}. \qquad (2)$$

Зауважимо, що вираз у правій частині (2) залежить від k та не прямує до нуля, якщо k є (як завгодно повільно) зростаючою функцією від n, наприклад, $k = \lceil \log n \rceil$, $n \to \infty$.

У роботі [11] запропонований більш ефективний імовірнісний тест k-вимірності, трудомісткість якого складає $O(2^k k^2\, \varepsilon^{-1})$ запитів до оракула (або $O(n\, 2^k k^2\, \varepsilon^{-1})$ двійкових операцій). При цьому верхня межа ймовірності помилки першого роду запропонованого тесту не залежить від k, а верхня межа ймовірності помилки другого роду є по суті така ж сама, що й для тесту з [3]. Показано також, що при певному природному змінюванні альтернативи H_1 можна побудувати однобічний (з нульовою ймовірністю помилки першого роду) тест k-вимірності, трудомісткість якого складає $O(n(2^k + k\varepsilon^{-2})\log(2^k + k\varepsilon^{-2}))$ двійкових операцій.

Вказаний тест покладено в основу методу розпізнавання k-вимірності булевих функцій, заданих за допомогою оракулів формальний опис якого запропоновано в даній статті.

1. НАУКОВІ ОСНОВИ МЕТОДУ, ЩО ПРОПОНУЄТЬСЯ

Основна ідея, покладена в основу методу, що пропонується, полягає в тому, щоб не вибирати вектори $h_1, ..., h_l$ наугад, а сформувати їх з використанням допоміжної процедури таким чином, щоб множина зазначених векторів з високою ймовірністю містилася у множині I_f, якщо f є k-вимірною функцією. Для цього пропонується розглянути звуження функції f на випадково вибраний підпростір векторного простору V_n. Зазначимо, що ідея застосування таких звужень під час перевірки різноманітних властивостей булевих функцій, імовірно, бере початок з роботи [12] та лежить в основі ймовірнісних алгоритмів тестування степеня поліномів від декількох змінних над полем з двох елементів [13, 14]. У даному випадку ця ідея реалізується наступним чином.

Позначимо $F_{m \times n}$ множину матриць розміру $m \times n$ над полем $F = \mathrm{GF}(2)$. Для будь-якої матриці $X \in F_{t \times n}$, де $k < t < n$, позначимо $f_X(u) = f(uX)$, $u \in V_t$ звуження функції f на підпростір, що породжується рядками матриці X.

Теорема 1. Якщо $f : V_n \to \{0,1\}$ – k-вимірна функція, то функція f_X також є k-вимірною. При цьому ймовірність події, яка полягає в тому, що при випадковому рівноймовірному виборі $t \times n$-матриці X множина $\{aX : a \in I_{f_X}\}$ міститься у множині I_f, є не менше за $1 - 2^{k-t}$.

Доведення. Встановимо ряд допоміжних властивостей k-вимірних булевих функцій. Наступна лема по суті співпадає з твердженням 2 у статті [7].

Лема 1. Функція $f : V_n \to \{0,1\}$ є k-вимірною у тому і тільки тому випадку, коли існують число $l \in \overline{0,k}$, матриця $A \in F_{n \times l}$ та функція $g : V_l \to \{0,1\}$ такі, що

$$f(x) = g(xA),\ x \in V_n. \qquad (3)$$

Якщо при цьому l є найменшим числом із зазначеною властивістю, то $I_f = \{\alpha \in V_n : \alpha A = 0\}$ і $\dim I_f = n - l$.

Назвемо представлення k-вимірної функції f у вигляді (3), яке відповідає найменшому можливому значенню $l \in \overline{0,k}$, незвідним представленням цієї функції.

Наслідок 1. Представлення (3) є незвідним тоді й тільки тоді, коли $rank\, A = l$ та $I_g = \{0\}$.

Лема 2. Нехай (3) є незвідним представленням k-вимірної функції f, де $g : V_l \to \{0,1\}$, $l \in \overline{0,k}$. Тоді

для будь-якої матриці $X \in F_{t\times n}$, де $k < t < n$, функція f_X є k-вимірною. Більш того, якщо $rank\, XA = l$, то

$$\{aX : a \in I_{f_X}\} \subseteq I_f . \quad (4)$$

Доведення. З рівності (3) випливає, що $f_X(u) = f(uX) = g(u(XA))$, $u \in V_t$. Отже, на підставі леми 3.1 f_X є k-вимірною функцією.

Нехай зараз $rank\, XA = l$. Оскільки представлення (3) є незвідним, то, згідно з наслідком 3.1, $I_g = \{0\}$. Отже, $f_X(u) = g(u(XA))$, $u \in V_t$ є незвідним представленням функції f_X і на підставі леми 3.1 $I_{f_X} = \{a \in V_t : aXA = 0\}$. Таким чином, якщо $a \in I_{f_X}$, то для будь-якого $z \in V_n$.

$$f(aX \oplus z) = g(aXA \oplus zA) = g(zA) = f(z),$$

тобто $aX \in I_f$, що й треба було довести.

Лема 3. Нехай $\alpha_1,...,\alpha_l \in V_n$ – лінійно незалежні вектори і $l \le t < n$. Тоді ймовірність того, що при випадковому рівноймовірному виборі матриці $X \in F_{t\times n}$ вектори $X\alpha_1,...,X\alpha_l$ є лінійно залежними, не перевищує 2^{l-t}.

Доведення. Якщо вектори $X\alpha_1,...,X\alpha_l$ лінійно залежні, існує ненульовий вектор $\alpha = c_1\alpha_1 \oplus \cdots \oplus c_l\alpha_l$ ($c_i \in F$, $i \in \overline{1,l}$) такий, що $X\alpha = 0$. Ймовірність останньої події дорівнює 2^{-t}. Отже, ймовірність того, що вектори $X\alpha_1,...,X\alpha_l$ є лінійно залежними не перевищує $(2^l - 1)2^{-t}$.

Лему доведено.

Виходячи з останніх двох лем, неважко переконатися в справедливості теореми 1. Дійсно, розглянемо незвідне представлення k-вимірної функції f у вигляді (3), де $g : V_l \to \{0,1\}$, $l \in \overline{0,k}$. Згідно з лемою 2, при випадковому рівноймовірному виборі матриці $X \in F_{t\times n}$ ймовірність події (4) є не менше ймовірності події $\{rank\, XA = l\}$, яка більше або дорівнює $1 - 2^{l-t} \ge 1 - 2^{k-t}$ на підставі леми 3.

Отже, теорему доведено.

2. ФОРМАЛЬНИЙ ОПИС МЕТОДУ РОЗПІЗНАВАННЯ *K*-ВИМІРНОСТІ БУЛЕВИХ ФУНКЦІЙ, ЗАДАНИХ ЗА ДОПОМОГОЮ ОРАКУЛІВ

Метод призначений для оцінки та обгрунтування стійкості блокових та потокових шифрів.

Основним показником ефективності є трудомісткість виражена в числі запитів до оракулу при заданих верхніх межах ймовірностей помилки першого та другого роду.

Додатковим показником ефективності є трудомісткість, виражена в числі двійкових операцій при заданих верхніх межах ймовірностей помилки першого та другого роду.

Сутність методу полягає в тому, щоб не вибирати вектори $h_1,...,h_l$ наугад, а сформувати їх з використанням допоміжної процедури таким чином, щоб множина зазначених векторів з високою ймовірністю містилася у множині I_f, якщо f є k-вимірною функцією. Це відрізняє запропонований метод від раніше відомого [3], який полягає в генерації незалежних випадкових рівноймовірних векторів $h_1,...,h_l \in V_n$.

Вхідними даними для застосування методу є такі параметри:

$$f : V_n \to \{0,1\};\ k \in \overline{0,n-1};\ \varepsilon \in (0,1);$$

$$t = k + c;\ m = 2^{t+4} t\, \varepsilon^{-1}\delta^{-1},$$

де $c \in \mathrm{N}$, $\delta \in (0, 1/2)$, $c, \delta = const$.

Припущення та обмеження: вважається, оракул, яким задана булева функція має нехтувано малий час оброблення запиту.

Алгоритм реалізації методу складається з двох кроків.

Крок 1. Згенерувати випадкову рівноймовірну $t \times n$-матрицю X, побудувати множину $Sp(f_X)$, за якою знайти базис $a_1,...,a_l$ векторного простору I_{f_X} (дуального до підпростору, що породжується множиною $Sp(f_X)$). Перевірити умову

$$l \ge t - k, \quad (5)$$

за виконанням якої перейти до кроку 2. У протилежному випадку – прийняти гіпотезу H_1 (f знаходиться на відстані не менше $2^n\varepsilon$ від множини k-вимірних функцій).

Крок 2. Для кожного $j \in \overline{1,l}$ покласти $h_j = a_j X$, згенерувати незалежні випадкові рівноймовірні вектори $Z_{1j},...,Z_{mj}$ та перевірити рівності (3.1). За виконанням зазначених рівностей для всіх $j \in \overline{1,l}$ прийняти гіпотезу H_0 (f – k-вимірна функція), у протилежному випадку – прийняти гіпотезу H_1.

Оцінка ефективності методу.

Теорема 3.2. Наведений алгоритм виконує $O(2^k k^2\, \varepsilon^{-1})$ запитів до оракула f та має трудомісткість $O(n\, 2^k k^2\, \varepsilon^{-1})$ двійкових операцій. При цьому ймовірність помилки першого роду (відхилити вірну гіпотезу H_0) не перевищує 2^{-c}, а ймовірність помилки другого роду (відхилити вірну гіпотезу H_1) не перевищує $\max\{5 \cdot 2^{-c-1}, \delta + \exp\{-7c2^c\}\}$.

Доведення. Почнемо з формулювань трьох допоміжних тверджень. Перше з них доведено в [15] та використовується в [3] як "факт 9". Друге твердження являє собою "факт 11" з [3], а третє – варіант нерівності Чебишова (див. твердження 4 в [3]).

Нагадаємо, що для функції $f:V_n\to\{0,1\}$ множина $Sp(f)$ визначається як сукупність усіх векторів $\alpha\in V_n$ таких, що $\hat f(\alpha)\neq 0$.

Лема 4. Для будь-якої функції $f:V_n\to\{0,1\}$ виконується рівність $I_f=Sp(f)^\perp$; іншими словами, простір I_f складається з векторів $y\in V_n$, що задовольняють умову: $y\alpha=0$ для будь-якого $\alpha\in Sp(f)$.

Лема 5. Нехай Z – випадковий рівноймовірний вектор на множині V_n. Тоді для будь-яких $f:V_n\to\{0,1\}$, $y\in V_n$ виконується рівність

$$\mathrm{P}_Z\{f(y\oplus Z)\neq f(Z)\}=\sum_{\substack{\alpha\in V_n:\\ y\alpha=1}}|\hat f(\alpha)|^2.$$

Лема 6. Нехай $\xi=\sum_{i=1}^N\xi_i$, де $\xi_1,\dots,\xi_N$ – попарно незалежні випадкові величини такі, що $0\le\xi_i\le\tau$, $i\in\overline{1,N}$. Тоді, якщо $\mathrm{E}\xi>0$, то для будь-якого $\delta>0$ справедлива нерівність $\mathrm{P}\{\xi\le(1-\delta)\mathrm{E}\xi\}\le\dfrac{\tau}{\delta^2\mathrm{E}\xi}$.

Лема 7. Алгоритм реалізації методу характеризується ймовірністю помилки першого роду не більше за 2^{-c}, виконує $O(2^k k^2\,\varepsilon^{-1})$ запитів до оракула f та має трудомісткість $O(n\,2^k k^2\,\varepsilon^{-1})$ двійкових операцій.

Доведення. Перше твердження леми випливає з теореми 1 та леми 4. Дійсно, якщо f є k-вимірною функцією, то такою ж є функція f_X. Отже, рівність (5) напевно виконується, і тест може здійснити помилку тільки в тому випадку, коли на кроці 2 порушується хоча б одна з рівностей (1). Проте на підставі теореми 1 ймовірність останньої події є не більше за $2^{k-t}=2^{-c}$, що й треба було довести.

Оцінимо трудомісткість алгоритму. На кроці 1 для обчислення значень функції f_X потрібно здійснити 2^t запитів до оракула f, кожен з яких вимагає порядка nt двійкових операцій. Далі, для знаходження коефіцієнтів Уолша-Адамара функції f_X треба виконати $O(2^t t)$ додавань або віднимань не більш ніж t-розрядних цілих чисел, що складає $O(2^t t^2)$ двійкових операцій. Такий саме час знадобиться для побудови базису векторного простору I_{f_X} за допомогою методу Гауса. На кроці 2 перевірка рівностей (1) для кожного з отриманих $l\le t$ базисних векторів вимагатиме не більше за $2mt$ запитів до оракула f, що складає $O(nmt)$ двійкових операцій.

Таким чином, з урахуванням значень параметрів m і t, загальне число запитів до оракула дорівнює $O(2^t+mt)=O(2^k k^2\,\varepsilon^{-1})$, а підсумкова трудомісткість алгоритму – $O(n2^t t+2^t t^2+nmt)=O(n2^k k^2\,\varepsilon^{-1})$ двійкових операцій.

Лему доведено.

Для оцінки ймовірності помилки другого роду скористаємося методом, що запропоновано в [3]. Зафіксуємо число $\theta\in(0,1)$ та розглянемо множини

$$B(\theta)=\{\alpha\in V_n\setminus\{0\}:|\hat f(\alpha)|\ge\theta\},$$

$$S(\theta)=\{\alpha\in V_n\setminus\{0\}:|\hat f(\alpha)|<\theta\}.$$

Покажемо спочатку, що якщо множина $B(\theta)$ породжує простір вимірності більше за k (і, отже, f напевно не є k-вимірною функцією), то Алгоритм реалізації методу здійснить помилку з нехтовно малою ймовірністю.

Лема 8. Нехай функція f є такою, що множина $B(\theta)$ містить щонайменше $k+1$ лінійно незалежних векторів $\alpha_1,\dots,\alpha_{k+1}$. Тоді ймовірність того, що Алгоритм реалізації методу здійснить помилку (тобто прийме f за k-вимірну функцію), не перевищує

$$p_1=2^{1-c}+(k+1)(1-\theta^2)^m 2^{k+c-1}. \qquad (6)$$

Доведення. Нехай алгоритм здійснює помилку. Тоді або вектори $X\alpha_1,\dots,X\alpha_{k+1}$ лінійно залежні (згідно з лемою 3, ймовірність цієї події становить не більше за $2^{k+1-t}=2^{1-c}$), або вони є лінійно незалежними, і тоді щонайменше один з них, скажимо, α_i, $i\in\overline{1,k+1}$, не належить множині $Sp(f_X)$. Оскільки на підставі леми 4 $I_{f_X}=Sp(f_X)^\perp$, то щонайменше один з базисних векторів $a_1,\dots,a_l$ простору I_{f_X} не є ортогональним вектору α_i. Отже, існує ненульовий вектор $a\in V_t$ такий, що $aX\alpha_i=1$ і для вектора $h_j=aX$ виконуються рівності (1). Таким чином, ймовірність помилки алгоритму не перевищує

$$2^{1-c}+\mathrm{P}_{X,Z_1,\dots,Z_m}\left(\bigcup_{i=1}^{k+1}\bigcup_{a\in V_t\setminus\{0\}}M_{i,a}\right)\le$$

$$\le 2^{1-c}+(k+1)2^t\max_{\substack{i\in\overline{1,k+1},\\ a\in V_t\setminus\{0\}}}\mathrm{P}_{X,Z_1,\dots,Z_m}\left(M_{i,a}\right),$$

де $M_{i,a}=\left\{aX\alpha_i=1, f(aX\oplus Z_s)=f(Z_s), s\in\overline{1,m}\right\}$.

Далі, на підставі незалежності та рівноймовірності випадкової матриці X та векторів $Z_1, \dots, Z_m$ для будь-яких $i \in \overline{1,k+1}$, $a \in V_t \setminus \{0\}$ виконується рівність

$$\mathrm{P}_{X,Z_1,\dots,Z_m}\left\{aX\alpha_i = 1, f(aX \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\} =$$

$$= \sum_{\substack{y \in V_n: \\ y\alpha_i = 1}} 2^{-nt} \sum_{\substack{X \in F_{t\times n}: \\ aX = y}} \left(\mathrm{P}_Z\{f(y \oplus Z) = f(Z)\}\right)^m .$$

При цьому, якщо $y\alpha_i = 1$, то на підставі леми 5 та умови $\alpha_i \in B(\theta)$ справедливі такі співвідношення:

$$\mathrm{P}_Z\{f(y \oplus Z) \neq f(Z)\} = \sum_{\substack{\alpha \in V_n: \\ y\alpha = 1}} |\hat{f}(\alpha)|^2 \geq |\hat{f}(\alpha_i)| \geq \theta^2 .$$

Отже,

$$\mathrm{P}_{X,Z_1,\dots,Z_m}\left\{aX\alpha_i = 1, f(aX \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\} \leq$$

$$\leq \sum_{\substack{y \in V_n: \\ y\alpha_i = 1}} 2^{-nt} \sum_{\substack{X \in F_{t\times n}: \\ aX = y}} (1-\theta^2)^m = \frac{1}{2}(1-\theta^2)^m .$$

З отриманих нерівностей випливає, що ймовірність помилки алгоритму не перевищує значення (6). Лему доведено.

Зауважимо, що у наведеному доведенні не використовується припущення про те, що функція f знаходиться на відстані не менше за $2^n\varepsilon$ від множини k-вимірних функцій. Залишається розглянути другий випадок, коли множина $B(\theta)$ породжує простір вимірності не більше за k. Міркування у цьому випадку значною мірою близькі до таких, що проводяться в [3]. Зокрема, наступна лема по суті співпадає з лемою 8 в [3].

Лема 9. Нехай функція f знаходиться на відстані не менше за $2^n\varepsilon$ від множини k-вимірних функцій n змінних, а множина $B(\theta)$ породжує підпростір вимірності не більше за k. Тоді

$$\sum_{\alpha \in S(\theta)} |\hat{f}(\alpha)|^2 \geq \varepsilon . \qquad (7)$$

Отже, для завершення доведення залишається оцінити ймовірність помилки алгоритму в припущенні справедливості нерівності (7).

Лема 10. Нехай виконується нерівність (7). Тоді ймовірність того, що Алгоритм реалізації методу здійснить помилку (тобто прийме f за k-вимірну функцію), не перевищує

$$p_2 = 2^{k+c}(8\varepsilon^{-1}\theta^2 + (1-\varepsilon/4)^m) . \qquad (8)$$

Доведення. Якщо алгоритм здійснює помилку, то існує, принаймні, один вектор $a \in V_t \setminus \{0\}$, для якого випадковий вектор $h_j = aX$ задовольняє рівності (1). Оскільки цей вектор має рівномірний розподіл на множині V_n, то ймовірність помилки алгоритму не перевищує

$$2^t \mathrm{P}_{Y,Z_1,\dots,Z_m}\left\{f(Y \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\}, \quad (9)$$

де $Y, Z_1, \dots, Z_m$ – незалежні в сукупності випадкові рівноймовірні вектори на V_n.

Для знаходження оцінки параметра (9) скористаємося міркуваннями, аналогічними таким, що використовуються у доведенні леми 7 в [3]. Розглянемо випадкову величину

$$\xi(Y) = \sum_{\alpha \in S(\theta)} |f(\alpha)|^2 I_\alpha(Y) ,$$

де $I_\alpha(Y)$ – індикатор події $Y\alpha = 1$, $\alpha \in S(\theta)$. Оскільки за означенням вектори $\alpha \in S(\theta)$ є ненульовими, то в силу нерівності (7)

$$\mathrm{E}\xi(Y) = \frac{1}{2} \sum_{\alpha \in S(\theta)} |f(\alpha)|^2 \geq \frac{\varepsilon}{2} . \qquad (10)$$

Крім того, випадкові величини $I_\alpha(Y)$, $\alpha \in S(\theta)$, є попарно незалежними. Отже, на підставі леми 6, означення множини $S(\theta)$ та нерівності (10)

$$\mathrm{P}_Y\left\{\xi(Y) \leq \frac{1}{2}\mathrm{E}\xi(Y)\right\} \leq \frac{\max\limits_{\alpha \in S(\theta)} |\hat{f}(\alpha)|}{1/4 \cdot \mathrm{E}\xi(Y)} \leq 8\theta^2\varepsilon^{-1} .$$

Помітимо зараз, що

$$\mathrm{P}_{Y,Z_1,\dots,Z_m}\left\{f(Y \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\} \leq$$

$$\leq \mathrm{P}_Y\left\{\xi(Y) \leq \frac{1}{2}\mathrm{E}\xi(Y)\right\} + \mathrm{P}_{Y,Z_1,\dots,Z_m}(M_Y) \leq$$

$$\leq 8\theta^2\varepsilon^{-1} + 2^{-n} \sum_{\substack{y \in V_n: \\ \xi(y) > \frac{1}{2}\mathrm{E}\xi(Y)}} \left(P_Z\{f(y \oplus Z) = f(Z)\}\right)^m ,$$

де $M_Y = \left\{f(Y \oplus Z_s) = f(Z_s), s \in \overline{1,m}, \xi(Y) > \frac{1}{2}\mathrm{E}\xi(Y)\right\}$.

При цьому на підставі леми 5 та нерівності (10) для будь-якого $y \in V_n$ такого, що $\xi(y) > \frac{1}{2}\mathrm{E}\xi(Y)$, справедливі такі нерівності:

$$\mathrm{P}_Z\{f(y \oplus Z) \neq f(Z)\} = \sum_{\substack{\alpha \in V_n: \\ y\alpha = 1}} |\hat{f}(\alpha)|^2 \geq \xi(y) > \frac{\varepsilon}{4} .$$

З останніх двох співвідношень отримаємо кінцеву оцінку параметра (9):

$$2^t \mathrm{P}_{Y,Z_1,\ldots,Z_m}\left\{f(Y\oplus Z_s)=f(Z_s),\, s\in\overline{1,m}\right\}\le$$

$$\le 2^t(8\varepsilon^{-1}\theta^2+(1-\varepsilon/4)^m).$$

Таким чином, ймовірність помилки алгоритму не перевищує значення (8), що й треба було довести.

Для завершення доведення теореми залишається зауважити, що на підставі лем 8 та 10 ймовірність помилки другого роду Алгоритму реалізації методу не перевищує $\max\{p_1, p_2\}$, де

$$\begin{aligned} p_1 &= 2^{1-c}+(k+1)(1-\theta^2)^m 2^{k+c-1},\\ p_2 &= 2^{k+c}(8\varepsilon^{-1}\theta^2+(1-\varepsilon/4)^m). \end{aligned} \qquad (11)$$

Вважаючи у формулах (11) $\theta^2=2^{-t-3}\,\varepsilon\,\delta$, $m=2^{t+4}t\,\varepsilon^{-1}\delta^{-1}$, де $\delta\in(0,1/2)$, отримаємо, що

$$p_1 \le 2^{1-c}+1/2\cdot(k+1)\exp\{k+c-\theta^2 m\}=$$

$$=2^{1-c}+1/2\cdot e^{-c}(k+1)e^{-k}<$$

$$<2^{1-c}+1/2\cdot 2^{-c}=5\cdot 2^{-c-1},$$

$$p_2=2^{k+c}(8\varepsilon^{-1}\theta^2+(1-\varepsilon/4)^m)=$$

$$=\delta+2^{k+c}(1-\varepsilon/4)^m<\delta+2^{k+c}\exp\{-m\varepsilon/4\}=$$

$$=\delta+2^{k+c}\exp\{-4t2^t\delta^{-1}\}<\delta+2^{k+c}\exp\{-8t2^t\}<$$

$$<\delta+\exp\{t-8t2^t\}<\delta+\exp\{-7t2^t\}<\delta+\exp\{-7c2^c\}.$$

Отже, ймовірність помилки другого роду Алгоритму реалізації методу не перевищує $\max\{5\cdot 2^{-c-1},\ \delta+\exp\{-7c2^c\}\}$.

Теорему повністю доведено.

Результати моделювання запропонованого методу. Для оцінки ефективності методу на практиці, наведений вище алгоритм реалізації методу був реалізовані програмно та застосовані для розпізнавання k-вимірності фільтрувальної функції f шифру Decim^{v2} [16].

Decim^{v2} – це синхронний потоковий шифр з довжиною ключа, яка дорівнює 80 бітів та вектором ініціалізації довжиною 64 біт. Decim^{v2} на сьогодні стандартизований [17]. В структурі Decimv2 можливо виділити (див. рис. 1): регістр зсуву з лінійним зворотним зв'язком довжиною 192 біти, нелінійну функцію f для генерації двійкової послідовності, функцію вибірки ABSG та буфер, що призначений для забезпечення безперервної видачі бітів гами (оскільки функція ABSG не забезпечує систематичну видачу бітів).

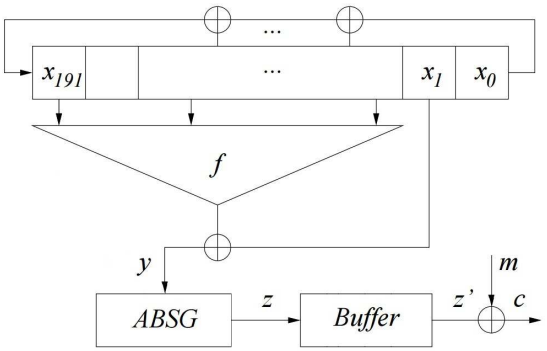

Рис. 1. Структурна схема шифру Decim^{v2}

Як зазначено в [17], функція f може бути означена таким чином

$$f(x_0,\ldots,x_{191})=\begin{cases}0, \textit{ якщо } S \bmod 4<2,\\ 1, \textit{ в іншому випадку}\end{cases},$$

де $S=1+x_{13}+x_{28}+x_{45}+x_{54}+x_{65}+x_{104}+x_{111}+$ $+x_{144}+x_{162}+x_{172}+x_{178}+x_{186}+x_{191}$.

Як видно з опису шифру Decim^{v2}, функція f має 192 змінних серед яких 13 є суттєвими, а, отже, k не перевищує 13.

Експерименти проведені з використанням пакета прикладних програм Maple на ПЕОМ типу Intel(R) Core(TM) i7-3770K 3,5 GHz, 8 Gb RAM у середовищі операційної системи Windows 7 та наступним чином.

Фіксувалися значення вхідних даних k, ε та δ (від якого безпосередньо залежить ймовірність помилки другого роду та значення p_0 ймовірності помилки першого роду тесту, яке дозволяє обчислити значення параметрів $c=-\log_2(p_0)$ та $t=k+c$), що задає значення $m=2^{t+4}t\,\varepsilon^{-1}\delta^{-1}$. Під час проведення обчислювального експерименту значення ε, δ та p_0 були обрані рівними 0,125.

Слід зауважити, що для моделювання випадкових рівноймовірних матриці X та векторів $Z_{1j},\ldots,Z_{mj}$ застосовувалась випадкова послідовність достатньо високої якості, що була сформована та протестована заздалегідь.

Кількість запусків обрана рівною 150 для кожного значення k від 1 до 13, щоб кількість відхилених вірних гіпотез H_0 (або H_1) не перевищувала 12,5% з надійністю не меншою 0,9973 (див., наприклад, [18], с. 99 – 100).

Як видно з таблиці 1, тест жодного разу не припустився помилки 1 роду, а ймовірність помилки 2 роду не перевищила 8%, що відповідає вихідним параметрам тесту, крім того, середній час перевірки гіпотези H_0 значно перевищує відповідний показник

для H_1, що пов'язано з необхідністю виконання кроку 2 вдосконаленого тесту k-вимірності в повному обсязі для кожного $j \in \overline{1,l}$ для перевірки гіпотези H_0.

Таблиця 1.

Результати дослідження функції f

k	$2^k k^2 \varepsilon^{-1}$	Кількість прийнятих гіпотез		Середній час перевірки гіпотези, сек.	
		H_0	H_1	H_0	H_1
1	16	0	150	–	0,04
2	128	0	150	–	0,07
3	576	0	150	–	0,11
4	2048	0	150	–	0,18
5	6400	0	150	–	0,30
6	18432	0	150	–	0,58
7	50176	0	150	–	1,06
8	131072	0	150	–	2,09
9	331776	0	150	–	4,21
10	819200	0	150	–	8,41
11	1982464	6	144	7071,48	17,41
12	4718592	12	138	12948,79	35,32
13	11075584	150	0	24687,61	–

Таким чином, вдосконалений тест може бути ефективно застосований на практиці до розпізнавання k-вимірності булевих функцій (зокрема, від десятків чи сотен змінних), які використовуються в сучасних симетричних криптосистемах.

Література

[1] *Chuan-Kun Wu* Boolean Functions and Their Applications in Cryptography / Chuan-Kun Wu, Dengguo Feng. Springer-Verlag Berlin Heidelberg, 2016, p. 267.

[2] *Логачев, О. А.* Булевы функции в теории кодирования и криптологии / О. А. Логачев, А. А. Сальников, В. В. Ященко. М. : МЦНМО, 2004. 470 с.

[3] *Gopalan, P.* Testing Fourier dimensionality and sparsity / P. Gopalan, R. O'Donnel, A. Servedio, A. Shpilka, K. Wimmer // SIAM J. on Computing. 2011. V. 40(4). P. 1075–1100.

[4] *Gopalan, P.* A Fourier-analytic approach to Reed-Muller decoding / P. Gopalan // Annual IEEE Symp. on Foundation in Computer Science. – FOCS 2010, Proceedings. Berlin. Springer-Verlag. 2010. P. 685–694.

[5] *Lechner, R. L.* Harmonic analysis of switching functions / R. L. Lechner // Recent Developments in Switching Theory. New-York. Academic Press. 1971. P. 122–228.

[6] *Dawson, E.* Construction of correlation immune Boolean functions / E. Dawson, C.K. Wu // Information and Communication Security, Proceedings. Berlin. Springer-Verlag. 1997. P. 170–180.

[7] *Алексеев, Е. К.* О некоторых мерах нелинейности булевых функций / Е. К. Алексеев // Прикладная дискретная математика. 2011. № 2(12). С. 5–16.

[8] *Daemen, J.* Resynchronization weaknesses in synchronous stream ciphers / J. Daemen, R. Govaerts, J. Vandewalle // Advances in Cryptology – EUROCRYPT'93, Proceedings. Berlin. Springer-Verlag. 1993. P. 159–167.

[9] *Golić, J.* On the resynchronization attack / J. Golić, G. Morgari // Fast Software Encryption. – FSE'03, Proceedings. Berlin. Springer-Verlag. 2003. P. 100–110.

[10] *Алексеев, Е. К.* Об атаке на фильтрующий генератор с функцией усложнения, близкой к алгебраически вырожденной / Е. К. Алексеев // Сборник статей молодых ученых факультета МВК МГУ, 2011. В. 8. С. 114–123.

[11] *Алексейчук, А. Н.* Усовершенствованный тест k-мерности для булевых функций / А. Н. Алексейчук, С. Н. Конюшок // Кибернетика и системный анализ. 2013. Т. 49. № 2. С. 27–35.

[12] *Levin L.A.* Randomness and non-determinism / Levin L.A. // J. of Symbolic Logic. – 1993. – Vol. 58. – № 3. – P. 1102 – 1103.

[13] *Alon N.* Testing Reed-Muller codes / Alon N., Kaufman T., Krivelevich M., Litsyn S., Ron D. // IEEE Trans. on Inform. Theory. – 2005. – Vol. 51(11). – P. 4032 – 4039.

[14] *Bhattacharyya A.* Optimal testing of Reed-Muller codes / Bhattacharyya A., Kopparty S., Schoenebeck G., Sudan M., Zuckerman D. // Proc. of the 51st Annual IEEE Symposium on Fundations of Computer Sci. – Las Vegas, Nevada, Oct. 23 – 26, 2010. – P. 488 – 497.

[15] *Ященко В.В.* О критерии распространения для булевых функций и бент-функциях / Ященко В.В // Проблемы передачи информации. – 1997. – Т. 33. – № 1. – С. 75 – 86.

[16] *Berbain C.* Decim^{v2} / Berbain C., Billet O., Canteaut A., Courtois N., Debraize B., Gilbert H., Goubin L., Gouget A., Granboulan L., Lauradoux C., Minier M., Pornin T., Sibert H. // URL: https://www.rocq.inria.fr/secret/Anne.Canteaut/Publications/BBC08b.pdf (last access: 25.05.18).

[17] ISO/IEC 18033-4: 2011(E). Information technology – Security techniques – Encryption algorithm – Part 4: Stream ciphers, 2011. – 92 p.

[18] *Ширяев А.Н.* Вероятность / А.Н. Ширяев. В 2-х кн. – 3-е изд., перераб. и доп. – М.: МЦНМО, 2004. – Кн. 1. – 520 с.

Надійшла до редколегії 25.12.2018

Конюшок Сергій Миколайович, кандидат технічних наук, доцент, заступник начальника інституту (з наукової роботи) Інституту спеціального зв'язку та захисту інформації Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Галузь наукових інтересів – криптографічні властивості булевих функцій.

УДК 621.391:519.2

Конюшок С. Н. **Метод распознавания k-мерности булевых функций, заданных с помощью оракулов** / С. Н. Конюшок // Прикладная радиоэлектроника: научный журнал. – 2018. – Том 17, №. 3, 4. – С. 168–175.

Предложен вероятностный метод распознавания k-мерности булевых функций, заданных с помощью оракулов, имеющий меньшую трудоемкость и характеризующийся меньшей вероятностью ошибки первого рода (при той же верхней границе вероятности ошибки второго рода) по сравнению с аналогичным ранее известным методом.

Ключевые слова: проверка свойств булевых функций, вероятностный метод, k-мерная функция, преобразование Уолша-Адамара.

Табл. 01. Іл. 01. Библиогр.: 18 назв.

UDC 621.391:519.2

Koniushok S. M. **The method for recognizing the k-dimensionality of Boolean functions given by the oracles** / S. M. Koniushok // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 168–175.

A probabilistic method of recognizing k-dimensionality of Boolean functions given by means of oracles is proposed. The proposed method has less time complexity and less likely to be characterized by a less first kind error probability (at the same upper bound of the second kind error probability) compared to the previously known method.

Keywords: testing properties of Boolean functions, probabilistic method, k-dimensional function, Walsh–Hadamard transform.

Tab. 01. Fig. 01. Ref.: 18 items.

УДК 621.391:519.2

МЕТОД РОЗПІЗНАВАННЯ К-ВИМІРНОСТІ БУЛЕВИХ ФУНКЦІЙ, ЗАДАНИХ ЗА ДОПОМОГОЮ ОРАКУЛІВ

С. М. КОНЮШОК

Запропоновано ймовірнісний метод розпізнавання k-вимірности булевих функцій, заданих за допомогою оракулів, який має меншу трудомісткість та характеризується меншою ймовірністю помилки першого роду (при такій самій верхній межі ймовірності помилки другого роду) порівняно з аналогічним раніше відомим методом.

Ключові слова: перевірка властивостей булевих функцій, ймовірнісний метод, k-вимірна функція, перетворення Уолша-Адамара.

ВСТУП

Булеві функції (далі – БФ) є необхідними криптографічними примітивами [1]; що нерідко відіграють ключову роль у створенні багатьох потокових та блокових шифрів. У таких випадках, важливою складовою дослідження криптографічної стійкості шифрів є аналіз криптографічних властивостей відповідних булевих функцій [2]. Тому дослідження криптографічних властивостей БФ не тільки входить до переліку важливих інструментів криптоаналітика, але також має суттєве значення для оцінки стійкості криптографічних алгоритмів в процесі їх синтезу.

Слід зауважити, що криптографічні властивості БФ не є незалежними одна від одної, вони мають певні зв'язки та накладають деякі обмеження одна до одної, і це означає, що неможливо знайти функцію, яка дозволяє досягати найкращих значень за кожною з цих властивостей. Як наслідок, побудова булевої функції, яка забезпечує прийнятні з точки зору практичного використання криптографічні властивості, фактично полягає в досягненні певного стану умовного оптимуму шляхом компромісного зниження вимог до окремих криптографічних властивостей задля досягнення прийнятних значень іншими криптографічними властивостями даної БФ.

Пошук такої булевої функції нерідко може бути пов'язаний зі значним обсягом досліджень великої кількості функцій-претендентів на роль такого важливого елементу криптографічного алгоритму в процесі його розробки.

Таким чином, актуалною є задача зменшення трудомісткості визначення або оцінки значень показників, що характеризують криптографічні властивості булевих функцій шляхом пошуку новітніх або удосконалення наявних підходів.

Розглянемо більш детально одну з важливих криптографічних властивостей булевих функцій.

Так, один із підходів до застосування БФ у потокових шифрах – діяти як функція ускладнення. Задля високої криптографічної стійкості шифру, така функція повинна мати високу нелінійність. Однак, з іншої точки зору, нелінійна функція може допускати представлення себе як суперпозиції лінійних функцій та більш простої нелінійної функції. Формально, така властивість виглядає наступним чином.

Булеву функцію $g:\{0,1\}^n \to \{0,1\}$ називають k-вимірною [3, 4], $0 \le k \le n$, якщо існують функція $\phi:\{0,1\}^k \to \{0,1\}$ та $n \times k$-матриця A над полем з двох елементів такі, що для будь-якого $x \in \{0,1\}^n$ справедлива рівність $g(x) = \phi(xA)$. Функцію g називають алгебраїчно виродженою, якщо вона є k-вимірною для деякого $k < n$ та невиродженою – в іншому випадку [1, 5 – 7].

Перші результати про кореляційні властивості алгебраїчно вироджених булевих функцій належать до 70-х років минулого століття [5]. Дослідження кореляційних властивостей булевих функцій обумовлене задачами криптографічного аналізу та теорії кодування. Відзначимо роботи [8 – 10], де викладений ряд атак на генератори гами потокових шифрів, функції ускладнення яких є алгебраїчно виродженими або близькими до таких.

Наразі, задача побудови ефективних підходів до розпізнавання властивості k-вимірності булевих функцій, є актуальною, як для оцінки стійкості заданого шифру, так і з метою реалізації криптоаналітичної атаки на шифратор, як на "чорну скриньку" (тобто шифратор виступає в ролі оракула).

В [3] був запропонований ймовірнісний алгоритм розпізнавання властивостей k-вимірності. Для будь-якої функції $f:\{0,1\}^n \to \{0,1\}$, що задана за допомогою оракула, та чисел $k \in \{0,1,...,n-1\}$, $\varepsilon \in (0,1)$ цей алгоритм дозволяє перевірити гіпотезу H_0: f – k-вимірна БФ проти альтернативи H_1, яка полягає в тому, що f знаходиться на відстані (Гемінга) не меньше $2^n\varepsilon$ від множини k-вимірних функцій.

Зазначений алгоритм полягає в генерації незалежних випадкових рівноймовірних векторів $h_1, ..., h_l \in V_n$ та переверці рівностей

$$f(h_j \oplus Z_{ij}) = f(Z_{ij}),\ i \in \overline{1,m} \qquad (1)$$

для кожного $j \in \overline{1,l}$, де Z_{ij} – незалежні в сукупності випадкові рівноймовірні вектори з V_n, що не залежать від $h_1, ..., h_l$. Позначимо ν_l число значень $j \in \overline{1,l}$, для яких виконуються рівності (1). Тоді гіпотеза H_0 приймається, якщо $\nu_l \cdot l^{-1} \geq 0{,}9 \cdot 2^{-k}$ та відхиляється у протилежному випадку. В [3] пропонується вибрати $l = 2^k C$, $m = 2^k k\, \varepsilon^{-1} C'$, де $C, C' = const$, що зводить до оцінки трудомісткості алгоритму $O(2^{2k} k\, \varepsilon^{-1})$ запитів до оракула f (або $O(n 2^{2k} k\, \varepsilon^{-1})$ двійкових операцій).

Для оцінювання ймовірності помилки першого роду (тобто ймовірності того, що тест "не визнає" такою k-вимірну функцію) в [3] використовується нерівність Чернова:

$$\mathrm{P}\left(\frac{\nu_l}{l} < 0{,}9 \cdot 2^{-k} \mid H_0\right) \leq$$

$$\leq \mathrm{P}\left(\frac{\nu_l}{l} - \mathrm{E}\frac{\nu_l}{l} < -0{,}1 \cdot 2^{-k} \mid H_0\right) \leq$$

$$\leq \exp\left\{-0{,}02 \cdot \frac{C}{2^k}\right\}. \qquad (2)$$

Зауважимо, що вираз у правій частині (2) залежить від k та не прямує до нуля, якщо k є (як завгодно повільно) зростаючою функцією від n, наприклад, $k = \lceil \log n \rceil$, $n \to \infty$.

У роботі [11] запропонований більш ефективний імовірнісний тест k-вимірності, трудомісткість якого складає $O(2^k k^2\, \varepsilon^{-1})$ запитів до оракула (або $O(n\, 2^k k^2\, \varepsilon^{-1})$ двійкових операцій). При цьому верхня межа ймовірності помилки першого роду запропонованого тесту не залежить від k, а верхня межа ймовірності помилки другого роду є по суті така ж сама, що й для тесту з [3]. Показано також, що при певному природному змінюванні альтернативи H_1 можна побудувати однобічний (з нульовою ймовірністю помилки першого роду) тест k-вимірності, трудомісткість якого складає $O(n(2^k + k\varepsilon^{-2})\log(2^k + k\varepsilon^{-2}))$ двійкових операцій.

Вказаний тест покладено в основу методу розпізнавання k-вимірності булевих функцій, заданих за допомогою оракулів формальний опис якого запропоновано в даній статті.

1. НАУКОВІ ОСНОВИ МЕТОДУ, ЩО ПРОПОНУЄТЬСЯ

Основна ідея, покладена в основу методу, що пропонується, полягає в тому, щоб не вибирати вектори $h_1, ..., h_l$ наугад, а сформувати їх з використанням допоміжної процедури таким чином, щоб множина зазначених векторів з високою ймовірністю містилася у множині I_f, якщо f є k-вимірною функцією. Для цього пропонується розглянути звуження функції f на випадково вибраний підпростір векторного простору V_n. Зазначимо, що ідея застосування таких звужень під час перевірки різноманітних властивостей булевих функцій, імовірно, бере початок з роботи [12] та лежить в основі ймовірнісних алгоритмів тестування степеня поліномів від декількох змінних над полем з двох елементів [13, 14]. У даному випадку ця ідея реалізується наступним чином.

Позначимо $F_{m \times n}$ множину матриць розміру $m \times n$ над полем $F = \mathrm{GF}(2)$. Для будь-якої матриці $X \in F_{t \times n}$, де $k < t < n$, позначимо $f_X(u) = f(uX)$, $u \in V_t$ звуження функції f на підпростір, що породжується рядками матриці X.

Теорема 1. Якщо $f : V_n \to \{0,1\}$ – k-вимірна функція, то функція f_X також є k-вимірною. При цьому ймовірність події, яка полягає в тому, що при випадковому рівноймовірному виборі $t \times n$-матриці X множина $\{aX : a \in I_{f_X}\}$ міститься у множині I_f, є не менше за $1 - 2^{k-t}$.

Доведення. Встановимо ряд допоміжних властивостей k-вимірних булевих функцій. Наступна лема по суті співпадає з твердженням 2 у статті [7].

Лема 1. Функція $f : V_n \to \{0,1\}$ є k-вимірною у тому і тільки тому випадку, коли існують число $l \in \overline{0,k}$, матриця $A \in F_{n \times l}$ та функція $g : V_l \to \{0,1\}$ такі, що

$$f(x) = g(xA),\ x \in V_n. \qquad (3)$$

Якщо при цьому l є найменшим числом із зазначеною властивістю, то $I_f = \{\alpha \in V_n : \alpha A = 0\}$ і $\dim I_f = n - l$.

Назвемо представлення k-вимірної функції f у вигляді (3), яке відповідає найменшому можливому значенню $l \in \overline{0,k}$, незвідним представленням цієї функції.

Наслідок 1. Представлення (3) є незвідним тоді й тільки тоді, коли $rank\, A = l$ та $I_g = \{0\}$.

Лема 2. Нехай (3) є незвідним представленням k-вимірної функції f, де $g : V_l \to \{0,1\}$, $l \in \overline{0,k}$. Тоді

для будь-якої матриці $X \in F_{t\times n}$, де $k < t < n$, функція f_X є k-вимірною. Більш того, якщо $rank\, XA = l$, то

$$\{aX : a \in I_{f_X}\} \subseteq I_f . \quad (4)$$

Доведення. З рівності (3) випливає, що $f_X(u) = f(uX) = g(u(XA))$, $u \in V_t$. Отже, на підставі леми 3.1 f_X є k-вимірною функцією.

Нехай зараз $rank\, XA = l$. Оскільки представлення (3) є незвідним, то, згідно з наслідком 3.1, $I_g = \{0\}$. Отже, $f_X(u) = g(u(XA))$, $u \in V_t$ є незвідним представленням функції f_X і на підставі леми 3.1 $I_{f_X} = \{a \in V_t : aXA = 0\}$. Таким чином, якщо $a \in I_{f_X}$, то для будь-якого $z \in V_n$.

$$f(aX \oplus z) = g(aXA \oplus zA) = g(zA) = f(z),$$

тобто $aX \in I_f$, що й треба було довести.

Лема 3. Нехай $\alpha_1,...,\alpha_l \in V_n$ – лінійно незалежні вектори і $l \le t < n$. Тоді ймовірність того, що при випадковому рівноймовірному виборі матриці $X \in F_{t\times n}$ вектори $X\alpha_1,...,X\alpha_l$ є лінійно залежними, не перевищує 2^{l-t}.

Доведення. Якщо вектори $X\alpha_1,...,X\alpha_l$ лінійно залежні, існує ненульовий вектор $\alpha = c_1\alpha_1 \oplus \cdots \oplus c_l\alpha_l$ ($c_i \in F$, $i \in \overline{1,l}$) такий, що $X\alpha = 0$. Ймовірність останньої події дорівнює 2^{-t}. Отже, ймовірність того, що вектори $X\alpha_1,...,X\alpha_l$ є лінійно залежними не перевищує $(2^l - 1)2^{-t}$.

Лему доведено.

Виходячи з останніх двох лем, неважко переконатися в справедливості теореми 1. Дійсно, розглянемо незвідне представлення k-вимірної функції f у вигляді (3), де $g : V_l \to \{0,1\}$, $l \in \overline{0,k}$. Згідно з лемою 2, при випадковому рівноймовірному виборі матриці $X \in F_{t\times n}$ ймовірність події (4) є не менше ймовірності події $\{rank\, XA = l\}$, яка більше або дорівнює $1 - 2^{l-t} \ge 1 - 2^{k-t}$ на підставі леми 3.

Отже, теорему доведено.

2. ФОРМАЛЬНИЙ ОПИС МЕТОДУ РОЗПІЗНАВАННЯ *K*-ВИМІРНОСТІ БУЛЕВИХ ФУНКЦІЙ, ЗАДАНИХ ЗА ДОПОМОГОЮ ОРАКУЛІВ

Метод призначений для оцінки та обгрунтування стійкості блокових та потокових шифрів.

Основним показником ефективності є трудомісткість виражена в числі запитів до оракулу при заданих верхніх межах ймовірностей помилки першого та другого роду.

Додатковим показником ефективності є трудомісткість, виражена в числі двійкових операцій при заданих верхніх межах ймовірностей помилки першого та другого роду.

Сутність методу полягає в тому, щоб не вибирати вектори $h_1,...,h_l$ наугад, а сформувати їх з використанням допоміжної процедури таким чином, щоб множина зазначених векторів з високою ймовірністю містилася у множині I_f, якщо f є k-вимірною функцією. Це відрізняє запропонований метод від раніше відомого [3], який полягає в генерації незалежних випадкових рівноймовірних векторів $h_1,...,h_l \in V_n$.

Вхідними даними для застосування методу є такі параметри:

$$f : V_n \to \{0,1\};\ k \in \overline{0,n-1};\ \varepsilon \in (0,1);$$

$$t = k + c;\ m = 2^{t+4} t\, \varepsilon^{-1}\delta^{-1},$$

де $c \in \mathrm{N}$, $\delta \in (0, 1/2)$, $c, \delta = const$.

Припущення та обмеження: вважається, оракул, яким задана булева функція має нехтувано малий час оброблення запиту.

Алгоритм реалізації методу складається з двох кроків.

Крок 1. Згенерувати випадкову рівноймовірну $t \times n$-матрицю X, побудувати множину $Sp(f_X)$, за якою знайти базис $a_1,...,a_l$ векторного простору I_{f_X} (дуального до підпростору, що породжується множиною $Sp(f_X)$). Перевірити умову

$$l \ge t - k, \quad (5)$$

за виконанням якої перейти до кроку 2. У протилежному випадку – прийняти гіпотезу H_1 (f знаходиться на відстані не менше $2^n\varepsilon$ від множини k-вимірних функцій).

Крок 2. Для кожного $j \in \overline{1,l}$ покласти $h_j = a_j X$, згенерувати незалежні випадкові рівноймовірні вектори $Z_{1j},...,Z_{mj}$ та перевірити рівності (3.1). За виконанням зазначених рівностей для всіх $j \in \overline{1,l}$ прийняти гіпотезу H_0 (f – k-вимірна функція), у протилежному випадку – прийняти гіпотезу H_1.

Оцінка ефективності методу.

Теорема 3.2. Наведений алгоритм виконує $O(2^k k^2\, \varepsilon^{-1})$ запитів до оракула f та має трудомісткість $O(n\, 2^k k^2\, \varepsilon^{-1})$ двійкових операцій. При цьому ймовірність помилки першого роду (відхилити вірну гіпотезу H_0) не перевищує 2^{-c}, а ймовірність помилки другого роду (відхилити вірну гіпотезу H_1) не перевищує $\max\{5 \cdot 2^{-c-1}, \delta + \exp\{-7c2^c\}\}$.

Доведення. Почнемо з формулювань трьох допоміжних тверджень. Перше з них доведено в [15] та використовується в [3] як "факт 9". Друге твердження являє собою "факт 11" з [3], а третє – варіант нерівності Чебишова (див. твердження 4 в [3]).

Нагадаємо, що для функції $f:V_n\to\{0,1\}$ множина $Sp(f)$ визначається як сукупність усіх векторів $\alpha\in V_n$ таких, що $\hat f(\alpha)\neq 0$.

Лема 4. Для будь-якої функції $f:V_n\to\{0,1\}$ виконується рівність $I_f=Sp(f)^\perp$; іншими словами, простір I_f складається з векторів $y\in V_n$, що задовольняють умову: $y\alpha=0$ для будь-якого $\alpha\in Sp(f)$.

Лема 5. Нехай Z – випадковий рівноймовірний вектор на множині V_n. Тоді для будь-яких $f:V_n\to\{0,1\}$, $y\in V_n$ виконується рівність

$$\mathrm{P}_Z\{f(y\oplus Z)\neq f(Z)\}=\sum_{\substack{\alpha\in V_n:\\ y\alpha=1}}|\hat f(\alpha)|^2.$$

Лема 6. Нехай $\xi=\sum_{i=1}^N\xi_i$, де $\xi_1,\dots,\xi_N$ – попарно незалежні випадкові величини такі, що $0\le\xi_i\le\tau$, $i\in\overline{1,N}$. Тоді, якщо $\mathrm{E}\xi>0$, то для будь-якого $\delta>0$ справедлива нерівність $\mathrm{P}\{\xi\le(1-\delta)\mathrm{E}\xi\}\le\dfrac{\tau}{\delta^2\mathrm{E}\xi}$.

Лема 7. Алгоритм реалізації методу характеризується ймовірністю помилки першого роду не більше за 2^{-c}, виконує $O(2^k k^2\,\varepsilon^{-1})$ запитів до оракула f та має трудомісткість $O(n\,2^k k^2\,\varepsilon^{-1})$ двійкових операцій.

Доведення. Перше твердження леми випливає з теореми 1 та леми 4. Дійсно, якщо f є k-вимірною функцією, то такою ж є функція f_X. Отже, рівність (5) напевно виконується, і тест може здійснити помилку тільки в тому випадку, коли на кроці 2 порушується хоча б одна з рівностей (1). Проте на підставі теореми 1 ймовірність останньої події є не більше за $2^{k-t}=2^{-c}$, що й треба було довести.

Оцінимо трудомісткість алгоритму. На кроці 1 для обчислення значень функції f_X потрібно здійснити 2^t запитів до оракула f, кожен з яких вимагає порядка nt двійкових операцій. Далі, для знаходження коефіцієнтів Уолша-Адамара функції f_X треба виконати $O(2^t t)$ додавань або віднимань не більш ніж t-розрядних цілих чисел, що складає $O(2^t t^2)$ двійкових операцій. Такий саме час знадобиться для побудови базису векторного простору I_{f_X} за допомогою методу Гауса. На кроці 2 перевірка рівностей (1) для кожного з отриманих $l\le t$ базисних векторів вимагатиме не більше за $2mt$ запитів до оракула f, що складає $O(nmt)$ двійкових операцій.

Таким чином, з урахуванням значень параметрів m і t, загальне число запитів до оракула дорівнює $O(2^t+mt)=O(2^k k^2\,\varepsilon^{-1})$, а підсумкова трудомісткість алгоритму – $O(n2^t t+2^t t^2+nmt)=O(n2^k k^2\,\varepsilon^{-1})$ двійкових операцій.

Лему доведено.

Для оцінки ймовірності помилки другого роду скористаємося методом, що запропоновано в [3]. Зафіксуємо число $\theta\in(0,1)$ та розглянемо множини

$$B(\theta)=\{\alpha\in V_n\setminus\{0\}:|\hat f(\alpha)|\ge\theta\},$$

$$S(\theta)=\{\alpha\in V_n\setminus\{0\}:|\hat f(\alpha)|<\theta\}.$$

Покажемо спочатку, що якщо множина $B(\theta)$ породжує простір вимірності більше за k (і, отже, f напевно не є k-вимірною функцією), то Алгоритм реалізації методу здійснить помилку з нехтовно малою ймовірністю.

Лема 8. Нехай функція f є такою, що множина $B(\theta)$ містить щонайменше $k+1$ лінійно незалежних векторів $\alpha_1,\dots,\alpha_{k+1}$. Тоді ймовірність того, що Алгоритм реалізації методу здійснить помилку (тобто прийме f за k-вимірну функцію), не перевищує

$$p_1=2^{1-c}+(k+1)(1-\theta^2)^m 2^{k+c-1}. \qquad (6)$$

Доведення. Нехай алгоритм здійснює помилку. Тоді або вектори $X\alpha_1,\dots,X\alpha_{k+1}$ лінійно залежні (згідно з лемою 3, ймовірність цієї події становить не більше за $2^{k+1-t}=2^{1-c}$), або вони є лінійно незалежними, і тоді щонайменше один з них, скажимо, α_i, $i\in\overline{1,k+1}$, не належить множині $Sp(f_X)$. Оскільки на підставі леми 4 $I_{f_X}=Sp(f_X)^\perp$, то щонайменше один з базисних векторів $a_1,\dots,a_l$ простору I_{f_X} не є ортогональним вектору α_i. Отже, існує ненульовий вектор $a\in V_t$ такий, що $aX\alpha_i=1$ і для вектора $h_j=aX$ виконуються рівності (1). Таким чином, ймовірність помилки алгоритму не перевищує

$$2^{1-c}+\mathrm{P}_{X,Z_1,\dots,Z_m}\left(\bigcup_{i=1}^{k+1}\bigcup_{a\in V_t\setminus\{0\}}M_{i,a}\right)\le$$

$$\le 2^{1-c}+(k+1)2^t\max_{\substack{i\in\overline{1,k+1},\\ a\in V_t\setminus\{0\}}}\mathrm{P}_{X,Z_1,\dots,Z_m}\left(M_{i,a}\right),$$

де $M_{i,a}=\left\{aX\alpha_i=1, f(aX\oplus Z_s)=f(Z_s), s\in\overline{1,m}\right\}$.

Далі, на підставі незалежності та рівноймовірності випадкової матриці X та векторів $Z_1, \dots, Z_m$ для будь-яких $i \in \overline{1,k+1}$, $a \in V_t \setminus \{0\}$ виконується рівність

$$\mathrm{P}_{X,Z_1,\dots,Z_m}\left\{aX\alpha_i = 1, f(aX \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\} =$$

$$= \sum_{\substack{y \in V_n: \\ y\alpha_i = 1}} 2^{-nt} \sum_{\substack{X \in F_{t\times n}: \\ aX = y}} \left(\mathrm{P}_Z\{f(y \oplus Z) = f(Z)\}\right)^m .$$

При цьому, якщо $y\alpha_i = 1$, то на підставі леми 5 та умови $\alpha_i \in B(\theta)$ справедливі такі співвідношення:

$$\mathrm{P}_Z\{f(y \oplus Z) \neq f(Z)\} = \sum_{\substack{\alpha \in V_n: \\ y\alpha = 1}} |\hat{f}(\alpha)|^2 \geq |\hat{f}(\alpha_i)| \geq \theta^2 .$$

Отже,

$$\mathrm{P}_{X,Z_1,\dots,Z_m}\left\{aX\alpha_i = 1, f(aX \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\} \leq$$

$$\leq \sum_{\substack{y \in V_n: \\ y\alpha_i = 1}} 2^{-nt} \sum_{\substack{X \in F_{t\times n}: \\ aX = y}} (1-\theta^2)^m = \frac{1}{2}(1-\theta^2)^m .$$

З отриманих нерівностей випливає, що ймовірність помилки алгоритму не перевищує значення (6). Лему доведено.

Зауважимо, що у наведеному доведенні не використовується припущення про те, що функція f знаходиться на відстані не менше за $2^n\varepsilon$ від множини k-вимірних функцій. Залишається розглянути другий випадок, коли множина $B(\theta)$ породжує простір вимірності не більше за k. Міркування у цьому випадку значною мірою близькі до таких, що проводяться в [3]. Зокрема, наступна лема по суті співпадає з лемою 8 в [3].

Лема 9. Нехай функція f знаходиться на відстані не менше за $2^n\varepsilon$ від множини k-вимірних функцій n змінних, а множина $B(\theta)$ породжує підпростір вимірності не більше за k. Тоді

$$\sum_{\alpha \in S(\theta)} |\hat{f}(\alpha)|^2 \geq \varepsilon . \qquad (7)$$

Отже, для завершення доведення залишається оцінити ймовірність помилки алгоритму в припущенні справедливості нерівності (7).

Лема 10. Нехай виконується нерівність (7). Тоді ймовірність того, що Алгоритм реалізації методу здійснить помилку (тобто прийме f за k-вимірну функцію), не перевищує

$$p_2 = 2^{k+c}(8\varepsilon^{-1}\theta^2 + (1-\varepsilon/4)^m) . \qquad (8)$$

Доведення. Якщо алгоритм здійснює помилку, то існує, принаймні, один вектор $a \in V_t \setminus \{0\}$, для якого випадковий вектор $h_j = aX$ задовольняє рівності (1). Оскільки цей вектор має рівномірний розподіл на множині V_n, то ймовірність помилки алгоритму не перевищує

$$2^t \mathrm{P}_{Y,Z_1,\dots,Z_m}\left\{f(Y \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\}, \quad (9)$$

де $Y, Z_1, \dots, Z_m$ – незалежні в сукупності випадкові рівноймовірні вектори на V_n.

Для знаходження оцінки параметра (9) скористаємося міркуваннями, аналогічними таким, що використовуються у доведенні леми 7 в [3]. Розглянемо випадкову величину

$$\xi(Y) = \sum_{\alpha \in S(\theta)} |f(\alpha)|^2 I_\alpha(Y) ,$$

де $I_\alpha(Y)$ – індикатор події $Y\alpha = 1$, $\alpha \in S(\theta)$. Оскільки за означенням вектори $\alpha \in S(\theta)$ є ненульовими, то в силу нерівності (7)

$$\mathrm{E}\xi(Y) = \frac{1}{2} \sum_{\alpha \in S(\theta)} |f(\alpha)|^2 \geq \frac{\varepsilon}{2} . \qquad (10)$$

Крім того, випадкові величини $I_\alpha(Y)$, $\alpha \in S(\theta)$, є попарно незалежними. Отже, на підставі леми 6, означення множини $S(\theta)$ та нерівності (10)

$$\mathrm{P}_Y\left\{\xi(Y) \leq \frac{1}{2}\mathrm{E}\xi(Y)\right\} \leq \frac{\max\limits_{\alpha \in S(\theta)} |\hat{f}(\alpha)|}{1/4 \cdot \mathrm{E}\xi(Y)} \leq 8\theta^2\varepsilon^{-1} .$$

Помітимо зараз, що

$$\mathrm{P}_{Y,Z_1,\dots,Z_m}\left\{f(Y \oplus Z_s) = f(Z_s), s \in \overline{1,m}\right\} \leq$$

$$\leq \mathrm{P}_Y\left\{\xi(Y) \leq \frac{1}{2}\mathrm{E}\xi(Y)\right\} + \mathrm{P}_{Y,Z_1,\dots,Z_m}(M_Y) \leq$$

$$\leq 8\theta^2\varepsilon^{-1} + 2^{-n} \sum_{\substack{y \in V_n: \\ \xi(y) > \frac{1}{2}\mathrm{E}\xi(Y)}} \left(P_Z\{f(y \oplus Z) = f(Z)\}\right)^m ,$$

де $M_Y = \left\{f(Y \oplus Z_s) = f(Z_s), s \in \overline{1,m}, \xi(Y) > \frac{1}{2}\mathrm{E}\xi(Y)\right\}$.

При цьому на підставі леми 5 та нерівності (10) для будь-якого $y \in V_n$ такого, що $\xi(y) > \frac{1}{2}\mathrm{E}\xi(Y)$, справедливі такі нерівності:

$$\mathrm{P}_Z\{f(y \oplus Z) \neq f(Z)\} = \sum_{\substack{\alpha \in V_n: \\ y\alpha = 1}} |\hat{f}(\alpha)|^2 \geq \xi(y) > \frac{\varepsilon}{4} .$$

З останніх двох співвідношень отримаємо кінцеву оцінку параметра (9):

$$2^t \mathrm{P}_{Y,Z_1,\ldots,Z_m}\left\{f(Y\oplus Z_s)=f(Z_s),\, s\in\overline{1,m}\right\}\le$$

$$\le 2^t(8\varepsilon^{-1}\theta^2+(1-\varepsilon/4)^m).$$

Таким чином, ймовірність помилки алгоритму не перевищує значення (8), що й треба було довести.

Для завершення доведення теореми залишається зауважити, що на підставі лем 8 та 10 ймовірність помилки другого роду Алгоритму реалізації методу не перевищує $\max\{p_1, p_2\}$, де

$$\begin{aligned} p_1 &= 2^{1-c}+(k+1)(1-\theta^2)^m 2^{k+c-1},\\ p_2 &= 2^{k+c}(8\varepsilon^{-1}\theta^2+(1-\varepsilon/4)^m). \end{aligned} \qquad (11)$$

Вважаючи у формулах (11) $\theta^2=2^{-t-3}\,\varepsilon\,\delta$, $m=2^{t+4}t\,\varepsilon^{-1}\delta^{-1}$, де $\delta\in(0,1/2)$, отримаємо, що

$$p_1 \le 2^{1-c}+1/2\cdot(k+1)\exp\{k+c-\theta^2 m\}=$$

$$=2^{1-c}+1/2\cdot e^{-c}(k+1)e^{-k}<$$

$$<2^{1-c}+1/2\cdot 2^{-c}=5\cdot 2^{-c-1},$$

$$p_2=2^{k+c}(8\varepsilon^{-1}\theta^2+(1-\varepsilon/4)^m)=$$

$$=\delta+2^{k+c}(1-\varepsilon/4)^m<\delta+2^{k+c}\exp\{-m\varepsilon/4\}=$$

$$=\delta+2^{k+c}\exp\{-4t2^t\delta^{-1}\}<\delta+2^{k+c}\exp\{-8t2^t\}<$$

$$<\delta+\exp\{t-8t2^t\}<\delta+\exp\{-7t2^t\}<\delta+\exp\{-7c2^c\}.$$

Отже, ймовірність помилки другого роду Алгоритму реалізації методу не перевищує $\max\{5\cdot 2^{-c-1},\ \delta+\exp\{-7c2^c\}\}$.

Теорему повністю доведено.

Результати моделювання запропонованого методу. Для оцінки ефективності методу на практиці, наведений вище алгоритм реалізації методу був реалізовані програмно та застосовані для розпізнавання k-вимірності фільтрувальної функції f шифру Decim^{v2} [16].

Decim^{v2} – це синхронний потоковий шифр з довжиною ключа, яка дорівнює 80 бітів та вектором ініціалізації довжиною 64 біт. Decim^{v2} на сьогодні стандартизований [17]. В структурі Decimv2 можливо виділити (див. рис. 1): регістр зсуву з лінійним зворотним зв'язком довжиною 192 біти, нелінійну функцію f для генерації двійкової послідовності, функцію вибірки ABSG та буфер, що призначений для забезпечення безперервної видачі бітів гами (оскільки функція ABSG не забезпечує систематичну видачу бітів).

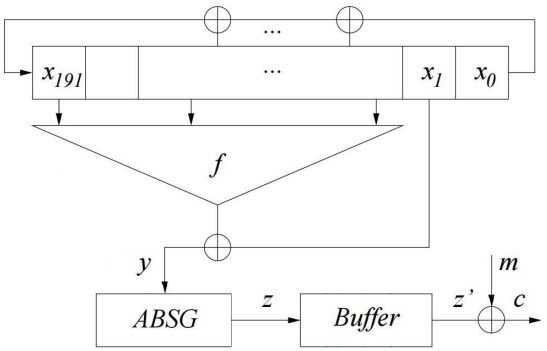

Рис. 1. Структурна схема шифру Decim^{v2}

Як зазначено в [17], функція f може бути означена таким чином

$$f(x_0,\ldots,x_{191})=\begin{cases}0, \textit{ якщо } S \bmod 4<2,\\ 1, \textit{ в іншому випадку}\end{cases},$$

де $S=1+x_{13}+x_{28}+x_{45}+x_{54}+x_{65}+x_{104}+x_{111}+$ $+x_{144}+x_{162}+x_{172}+x_{178}+x_{186}+x_{191}$.

Як видно з опису шифру Decim^{v2}, функція f має 192 змінних серед яких 13 є суттєвими, а, отже, k не перевищує 13.

Експерименти проведені з використанням пакета прикладних програм Maple на ПЕОМ типу Intel(R) Core(TM) i7-3770K 3,5 GHz, 8 Gb RAM у середовищі операційної системи Windows 7 та наступним чином.

Фіксувалися значення вхідних даних k, ε та δ (від якого безпосередньо залежить ймовірність помилки другого роду та значення p_0 ймовірності помилки першого роду тесту, яке дозволяє обчислити значення параметрів $c=-\log_2(p_0)$ та $t=k+c$), що задає значення $m=2^{t+4}t\,\varepsilon^{-1}\delta^{-1}$. Під час проведення обчислювального експерименту значення ε, δ та p_0 були обрані рівними 0,125.

Слід зауважити, що для моделювання випадкових рівноймовірних матриці X та векторів $Z_{1j},\ldots,Z_{mj}$ застосовувалась випадкова послідовність достатньо високої якості, що була сформована та протестована заздалегідь.

Кількість запусків обрана рівною 150 для кожного значення k від 1 до 13, щоб кількість відхилених вірних гіпотез H_0 (або H_1) не перевищувала 12,5% з надійністю не меншою 0,9973 (див., наприклад, [18], с. 99 – 100).

Як видно з таблиці 1, тест жодного разу не припустився помилки 1 роду, а ймовірність помилки 2 роду не перевищила 8%, що відповідає вихідним параметрам тесту, крім того, середній час перевірки гіпотези H_0 значно перевищує відповідний показник

для H_1, що пов'язано з необхідністю виконання кроку 2 вдосконаленого тесту k-вимірності в повному обсязі для кожного $j \in \overline{1,l}$ для перевірки гіпотези H_0.

Таблиця 1.

Результати дослідження функції f

k	$2^k k^2 \varepsilon^{-1}$	Кількість прийнятих гіпотез		Середній час перевірки гіпотези, сек.	
		H_0	H_1	H_0	H_1
1	16	0	150	–	0,04
2	128	0	150	–	0,07
3	576	0	150	–	0,11
4	2048	0	150	–	0,18
5	6400	0	150	–	0,30
6	18432	0	150	–	0,58
7	50176	0	150	–	1,06
8	131072	0	150	–	2,09
9	331776	0	150	–	4,21
10	819200	0	150	–	8,41
11	1982464	6	144	7071,48	17,41
12	4718592	12	138	12948,79	35,32
13	11075584	150	0	24687,61	–

Таким чином, вдосконалений тест може бути ефективно застосований на практиці до розпізнавання k-вимірності булевих функцій (зокрема, від десятків чи сотен змінних), які використовуються в сучасних симетричних криптосистемах.

Література

[1] *Chuan-Kun Wu* Boolean Functions and Their Applications in Cryptography / Chuan-Kun Wu, Dengguo Feng. Springer-Verlag Berlin Heidelberg, 2016, p. 267.

[2] *Логачев, О. А.* Булевы функции в теории кодирования и криптологии / О. А. Логачев, А. А. Сальников, В. В. Ященко. М. : МЦНМО, 2004. 470 с.

[3] *Gopalan, P.* Testing Fourier dimensionality and sparsity / P. Gopalan, R. O'Donnel, A. Servedio, A. Shpilka, K. Wimmer // SIAM J. on Computing. 2011. V. 40(4). P. 1075–1100.

[4] *Gopalan, P.* A Fourier-analytic approach to Reed-Muller decoding / P. Gopalan // Annual IEEE Symp. on Foundation in Computer Science. – FOCS 2010, Proceedings. Berlin. Springer-Verlag. 2010. P. 685–694.

[5] *Lechner, R. L.* Harmonic analysis of switching functions / R. L. Lechner // Recent Developments in Switching Theory. New-York. Academic Press. 1971. P. 122–228.

[6] *Dawson, E.* Construction of correlation immune Boolean functions / E. Dawson, C.K. Wu // Information and Communication Security, Proceedings. Berlin. Springer-Verlag. 1997. P. 170–180.

[7] *Алексеев, Е. К.* О некоторых мерах нелинейности булевых функций / Е. К. Алексеев // Прикладная дискретная математика. 2011. № 2(12). С. 5–16.

[8] *Daemen, J.* Resynchronization weaknesses in synchronous stream ciphers / J. Daemen, R. Govaerts, J. Vandewalle // Advances in Cryptology – EUROCRYPT'93, Proceedings. Berlin. Springer-Verlag. 1993. P. 159–167.

[9] *Golić, J.* On the resynchronization attack / J. Golić, G. Morgari // Fast Software Encryption. – FSE'03, Proceedings. Berlin. Springer-Verlag. 2003. P. 100–110.

[10] *Алексеев, Е. К.* Об атаке на фильтрующий генератор с функцией усложнения, близкой к алгебраически вырожденной / Е. К. Алексеев // Сборник статей молодых ученых факультета МВК МГУ, 2011. В. 8. С. 114–123.

[11] *Алексейчук, А. Н.* Усовершенствованный тест k-мерности для булевых функций / А. Н. Алексейчук, С. Н. Конюшок // Кибернетика и системный анализ. 2013. Т. 49. № 2. С. 27–35.

[12] *Levin L.A.* Randomness and non-determinism / Levin L.A. // J. of Symbolic Logic. – 1993. – Vol. 58. – № 3. – P. 1102 – 1103.

[13] *Alon N.* Testing Reed-Muller codes / Alon N., Kaufman T., Krivelevich M., Litsyn S., Ron D. // IEEE Trans. on Inform. Theory. – 2005. – Vol. 51(11). – P. 4032 – 4039.

[14] *Bhattacharyya A.* Optimal testing of Reed-Muller codes / Bhattacharyya A., Kopparty S., Schoenebeck G., Sudan M., Zuckerman D. // Proc. of the 51st Annual IEEE Symposium on Fundations of Computer Sci. – Las Vegas, Nevada, Oct. 23 – 26, 2010. – P. 488 – 497.

[15] *Ященко В.В.* О критерии распространения для булевых функций и бент-функциях / Ященко В.В // Проблемы передачи информации. – 1997. – Т. 33. – № 1. – С. 75 – 86.

[16] *Berbain C.* Decim^{v2} / Berbain C., Billet O., Canteaut A., Courtois N., Debraize B., Gilbert H., Goubin L., Gouget A., Granboulan L., Lauradoux C., Minier M., Pornin T., Sibert H. // URL: https://www.rocq.inria.fr/secret/Anne.Canteaut/Publications/BBC08b.pdf (last access: 25.05.18).

[17] ISO/IEC 18033-4: 2011(E). Information technology – Security techniques – Encryption algorithm – Part 4: Stream ciphers, 2011. – 92 p.

[18] *Ширяев А.Н.* Вероятность / А.Н. Ширяев. В 2-х кн. – 3-е изд., перераб. и доп. – М.: МЦНМО, 2004. – Кн. 1. – 520 с.

Надійшла до редколегії 25.12.2018

Конюшок Сергій Миколайович, кандидат технічних наук, доцент, заступник начальника інституту (з наукової роботи) Інституту спеціального зв'язку та захисту інформації Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Галузь наукових інтересів – криптографічні властивості булевих функцій.

УДК 621.391:519.2

Конюшок С. Н. **Метод распознавания k-мерности булевых функций, заданных с помощью оракулов** / С. Н. Конюшок // Прикладная радиоэлектроника: научный журнал. – 2018. – Том 17, №. 3, 4. – С. 168–175.

Предложен вероятностный метод распознавания k-мерности булевых функций, заданных с помощью оракулов, имеющий меньшую трудоемкость и характеризующийся меньшей вероятностью ошибки первого рода (при той же верхней границе вероятности ошибки второго рода) по сравнению с аналогичным ранее известным методом.

Ключевые слова: проверка свойств булевых функций, вероятностный метод, k-мерная функция, преобразование Уолша-Адамара.

Табл. 01. Іл. 01. Библиогр.: 18 назв.

UDC 621.391:519.2

Koniushok S. M. **The method for recognizing the k-dimensionality of Boolean functions given by the oracles** / S. M. Koniushok // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 168–175.

A probabilistic method of recognizing k-dimensionality of Boolean functions given by means of oracles is proposed. The proposed method has less time complexity and less likely to be characterized by a less first kind error probability (at the same upper bound of the second kind error probability) compared to the previously known method.

Keywords: testing properties of Boolean functions, probabilistic method, k-dimensional function, Walsh–Hadamard transform.

Tab. 01. Fig. 01. Ref.: 18 items.

УДК 004.652 : 004.658.3

МЕТОД ПРЕОБРАЗОВАНИЯ УНАСЛЕДОВАННЫХ БАЗ ДАННЫХ В БАЗУ ДАННЫХ С УНИВЕРСАЛЬНЫМ БАЗИСОМ ОТНОШЕНИЙ

В. И. ЕСИН, В. В. ВИЛИГУРА

Предлагается метод преобразования унаследованных баз данных в базу данных с универсальным базисом отношений, позволяющий расширить функциональность унаследованных БД, улучшить определенные их характеристики качества, в том числе, способность к адаптации в условиях динамичных изменений предметных областей, способствующий снижению стоимости сопровождения баз данных.

Ключевые слова: база данных, унаследованная база данных, база данных с универсальным базисом отношений.

ПОСТАНОВКА ЗАДАЧИ

Устаревают платформы систем управления базами данных (СУБД), растет количество разрозненных источников информации, данные которых продолжают иметь значимую практическую ценность, но их изолированность, несогласованность, различное представление мешает связывать и анализировать содержащуюся в них информацию. Все это в целом на фоне возрастающих требований к поддерживаемой функциональности, качеству действующих баз данных (БД) информационных систем (ИС) приводит к необходимости адаптации ранее созданных и функционирующих БД и программ работы с ними к новым языкам, операционным средам и платформам. То есть ведет к росту востребованности проектов модернизации, интеграции, замены существующих систем – реинжинирингу действующих систем и их основного компонента – БД.

При этом условия, в которых приходиться осуществлять реинжиниринг БД, характеризуются таким особенностями как: а) априорной неопределенностью в периодичности потребности данных различных типов, структур; б) жесткостью требований к значениям обеспечиваемых характеристик надежности, безопасности хранения и доступности использования актуальных, согласованных больших объемов структурированных и неструктурированных данных из существенно отличающихся ПрО; в) ограниченностью отводимых временных и финансовых ресурсов. Организации обычно расходуют (20-40) % своего IT-бюджета на эволюционирование данных путем миграции (изменение местоположения), очистки или преобразования (изменения формы или структуры данных) [1].

В сложившейся ситуации продолжает иметь место объективная потребность в исследовании и переосмыслении существующих подходов, методологий и технологий реинжиниринга ИС, актуализирующая задачи: разработки целостных методологий реинжиниринга баз данных ИС, механизмов адаптации методологий реинжиниринга БД в реальных проектах; инструментальных средств, обеспечивающих комплексное решение задач по реинжинирингу БД; оценки качества создаваемых в процессе реинжиниринга БД ИС.

Опираясь на результаты проведенного анализа: различных информационных технологий (ИТ) управления данными, интеграции данных, реинжиниринга информационных систем, а также тенденций их развития; классических методов проектирования баз данных и, в первую очередь, реляционных; различных моделей данных, используемых при моделировании предметных областей, учитывая требования, предъявляемые к корпоративным БД ИС, на основании созданных моделей [2–7] и схемы БД, инвариантной к предметным областям [8], была разработана информационная технология, обеспечивающая механизм адаптируемости БД к изменениям условий функционирования. Она, как совокупность методов, в основу которых были положены обоснованные и разработанные модель «объект-событие» [2–5], модель данных с универсальным базисом отношений [6, 7], инвариантная к предметным областям (ПрО) схема БД [8], а также специально созданного программного обеспечения, не привязана к конкретным аппаратным платформам, хотя некоторые ее реализации связаны с определенными программными системами.

Один из методов данной технологии – метод преобразования унаследованных баз данных, ведущий к замещению одной БД другой, построенной на основе инвариантной к ПрО схемы [8], предложен ниже.

МЕТОД ПРЕОБРАЗОВАНИЯ УНАСЛЕДОВАННЫХ БАЗ ДАННЫХ

Основные технологические операции данного метода, связываемые с соответствующими этапами преобразования унаследованных баз данных, представлены на рис. 1 в виде функциональных блоков методологии моделирования IDEF0.

Рассмотрим их для каждого этапа более подробно.

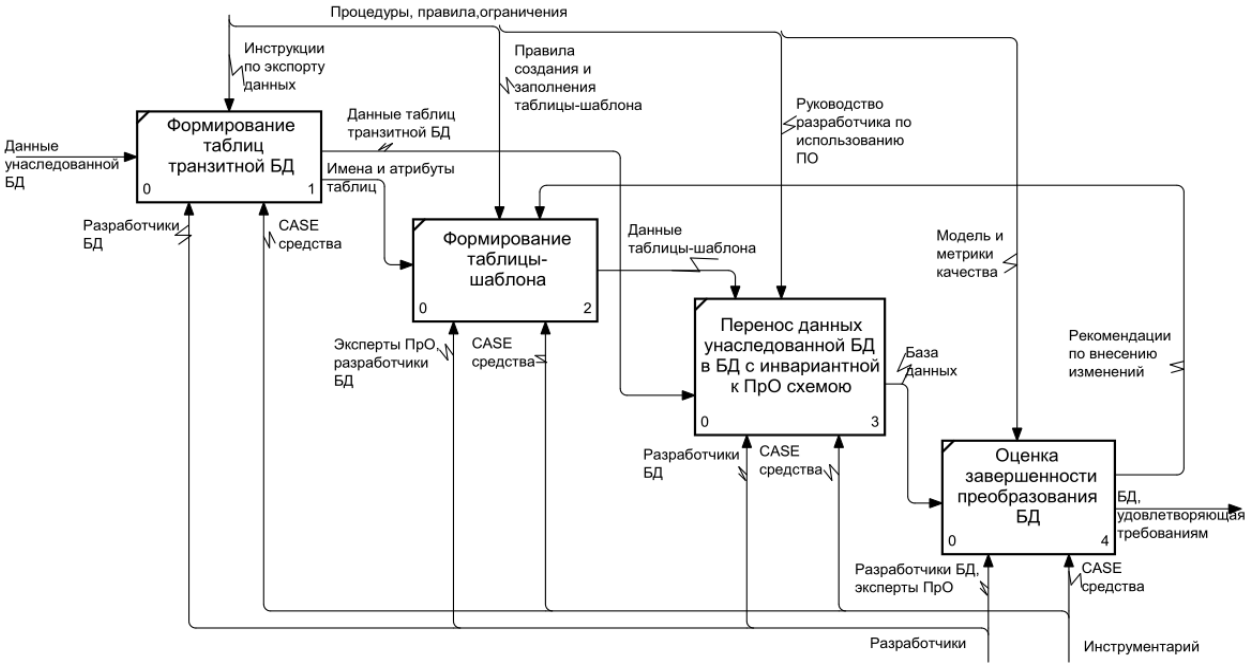

Рис. 1. Представление основных технологических операций метода

Этап 1. Формирование таблиц транзитной БД.

Данные унаследованной БД экспортируются во временные таблицы, так называемой, транзитной (временной) БД, реализованной на платформе реляционной СУБД, такой как: Oracle, PostgreSQL, Access или другой, совместимой с ODBC. Перенос данных между различными СУБД можно осуществить, например, как описано в работе [9], используя либо специальный драйвер (интерфейс), в частности ODBC, OLE DB (в этом случае две СУБД соединяются друг с другом и непосредственно передают данные), либо транзитные файлы (как правило, формата dbf, т. к. большинство СУБД, формат хранения данных которых отличается от формата dbf, снабжены утилитами или драйверами, которые позволяют перенести данные в этот формат), в которые копируются данные из унаследованной БД для последующего переноса в транзитную БД. При этом если унаследованная СУБД не является реляционной, то данные транзитных файлов должны быть приведены к табличному виду и первой нормальной форме.

Этап 2. Формирование таблицы-шаблона.

В транзитной БД также создается, так называемая, таблица-шаблон, заголовок которой формально можно представить следующим образом:

$$H_p = \langle A_{id}, A_{tt}, A_{op}, A_v \rangle \qquad (1)$$

Основными компонентами кортежа (1) являются:

– A_{id} – первичный ключ временной таблицы;

– $A_{tt} = \{a_{tt}^1, a_{tt}^2, a_{tt}^3\}$ – множество имен атрибутов, значениями которых являются: имена временных таблиц транзитной БД (a_{tt}^1 ="преобразуемая таблица"), в которые были перенесены данные из унаследованной БД; имена атрибутов (a_{tt}^2 ="*атрибут*") в соответствующих временных таблицах; определенные конструкции языка модели данных (ЯМД) [10] (a_{tt}^3 ="*конструкция ЯМД*"), заканчивающиеся знаком равно, которым может быть присвоено значение атрибута a_{tt}^2 соответствующей строки временной таблицы;

– $A_{op} = \{a_{op}^1, a_{op}^2, a_{op}^3, a_{op}^4, a_{op}^5\}$ – множество имен атрибутов, значениями которых являются конструкции ЯМД: для a_{op}^1 ="оператор РазделВлад": конструкция ЯМД – "<Раздел>=" или последовательность операторов ЯМД (в случае иерархии разделов в моделируемой ПрО) обязательно заканчивающаяся конструкцией – "<Раздел>="; для a_{op}^2 = ="*оператор ЭкзОВлад*": конструкция ЯМД "<ЭкзО>=" или последовательность операторов ЯМД (в случае иерархии экземпляров объектов), обязательно заканчивающаяся конструкцией – "<ЭкзО>="; для a_{op}^3 ="*оператор КлассОЭкзОВл*": конструкция ЯМД – "<КлассО>=" или последовательность операторов ЯМД (в случае иерархии классов объектов), обязательно заканчивающаяся конструкцией – "<КлассО>="; для a_{op}^4 ="*оператор ТипОЭкзОВл*": конструкция ЯМД – "<ТипО>=" или последовательность операторов ЯМД, обязательно заканчивающаяся конструкцией – "<ТипО>="; конкретный (из числа допустимых) оператор ЯМД (без указания значения после знака равенства, в соответствии с синтаксисом ЯМД) или после-

довательность операторов ЯМД (для $a^5_{op} = \varnothing \cup \{a^{51}_{op},...\}$, если $a^5_{op} \neq \varnothing$, то в зависимости от необходимости, каждому элементу множества a^5_{op} будет присвоено имя следующего формата: «оператор n», где n=1,…, например, $a^{51}_{op} = "оператор\ 1"$ и т. д.);

– $A_v = \{a^1_v, a^2_v, a^3_v, a^4_v, a^5_v\}$ – множество имен атрибутов, значениями которых являются: для $a^1_v = "Раздел\ Влад"$: имя раздела, присваиваемое значению атрибута a^1_{op} соответствующей строки таблицы-шаблона, которое может задаваться константой или выбираться как из строк временной таблицы транзитной БД в соответствии с указанным атрибутом, так и собственно из формируемой БД (в которую переносятся данные); для $a^2_v = "ЭкзОВлад"$: имя экземпляра объекта, присваиваемое значению атрибута a^2_{op} соответствующей строки таблицы-шаблона, которое также может задаваться константой или выбираться как из строк временной таблицы транзитной БД в соответствии с указанным атрибутом, так и собственно из формируемой БД; для $a^3_v = "КлассОЭкзОВлад"$: имя класса объекта-владельца, присваиваемое значению атрибута a^3_{op} соответствующей строки таблицы-шаблона, которое задается или выбирается аналогично рассмотренным выше для a^1_v, a^2_v именам; для $a^4_v = "ТипОЭкзОВл"$: имя типа объекта-владельца, присваиваемое значению атрибута a^4_{op} соответствующей строки таблицы-шаблона, которое также задается или выбирается аналогично рассмотренным выше для a^1_v, a^2_v, a^3_v именам; имя, присваиваемое последнему оператору конструкции ЯМД a^5_{op} соответствующей строки таблицы-шаблона (для $a^5_v = \varnothing \cup \{a^{51}_v,...\}$, $(a^5_{op} = \varnothing) \Rightarrow (a^5_v = \varnothing)$, если $a^5_{op} \neq \varnothing$, то в зависимости от количества элементов a^5_{op} аналогичную мощность будет иметь множество a^5_v ($|a^5_v| = |a^5_v|$), и в этом случае каждому элементу множества a^5_v будет присвоено имя следующего формата: «Значение n», где n=1,…, например, $a^{51}_v = "Значение\ 1"$ и т. д.).

Заполнение таблицы-шаблона осуществляется в соответствии с синтаксисом ЯМД и следующими правилами.

При составлении конструкций, помещаемых в таблицу-шаблон, следует учитывать особенности их формирования, а именно те, которые связаны с добавлением определенных служебных символов в стандартные конструкции операторов строк метаописаний ЯМД. Рассмотрим их более подробно.

Символ $, указанный первым в строке атрибутов, начинающихся словом «оператор» таблицы-шаблона ($A_{op} = \{a^1_{op}, a^2_{op}, a^3_{op}, a^4_{op}, a^5_{op}\}$), свидетельствует о том, что информация, обрабатываемой строки, используется для формирования строки метаописания. При этом за символом $ обязательно должен следовать необходимый оператор ЯМД (любые другие символы будут игнорироваться) и последним ключевым (служебным) словом в этой синтаксической конструкции должно быть одно из следующих: <КлассО>, <КлассС>, <ЭкзО>, <Папка>, <Документ>.

Символ $, указанный первым в строке атрибутов, начинающихся словом «значение» таблицы-шаблона (a^5_v), свидетельствует о том, что информация о значении, обрабатываемой строки, будет выбираться из формируемой БД, построенной на основе инвариантной к ПрО схемы. При этом следует учитывать, что исполнение такой синтаксической конструкции в некоторых случаях может вызвать исключительную ситуацию – выбор нескольких значений. Поэтому использовать эту возможность необходимо в обоснованных случаях, в которых заранее известно, что они не вызовут такую ситуацию.

Символ #, указанный первым в строке атрибутов $A_v = \{a^1_v, a^2_v, a^3_v, a^4_v, a^5_v\}$, или указанный внутри строки метаописания, определенного символом ***$*** в строке атрибутов $A_{op} = \{a^1_{op}, a^2_{op}, a^3_{op}, a^4_{op}, a^5_{op}\}$ таблицы-шаблона, свидетельствует о том, что данные выбираются из соответствующего атрибута (имя атрибута определяется идентификатором, следующим за символом #) указанной временной таблицы транзитной БД.

Пример сформированной таблицы-шаблона приведен в таблице 1.

Этап 3. Перенос данных унаследованных БД в БД с инвариантной к ПрО схемою.

На основании содержания таблицы-шаблона (данных ее строк и столбцов) и временных таблиц транзитной БД, специально разработанное приложение в соответствии с алгоритмом преобразования, блок-схема которого представлена на рис. 2, формирует строки метаописания на ЯМД, которые затем и исполняет.

В результате таких действий данные, ранее хранящихся в исходной БД и перемещенные во временные таблицы транзитной БД, помещаются в БД, построенную на основе инвариантной к ПрО схемы, в виде определенных метаданных и данных моделируемой ПрО. При этом следует отметить, что возникающие при традиционном способе миграции унаследованных систем осложнения, вызванные различием логических структур данных, систем кодирования

Таблица 1

Пример таблицы-шаблона

ID	Преобразуемая таблица	Атрибут	Конструкция ЯМД	Оператор Раздел Влад	Раздел Влад	оператор ЭкзОВлад	ЭкзОВлад	Оператор КлассОЭкзОВл	КлассО ЭкзОВлад	Оператор ТипО ЭкзОВл	ТипО ЭкзОВл	Оператор 1	Значение 1
1	Vehicle classifier	Код типа техники	<ЗначХО>=	<Раздел>=	UGES	<ЭкзО>=	#Наименование_типа_техники	<КлассО>=	классификация техники	<ТипО>=	-	<ТипХО>[] (строковая, любая, фактическая)=	Код типа техники
2	Vehicle classifier	Код подтипа техники	<ЗначХО>=	<Раздел>=	UGES	<ЭкзО>=	#Наименование_типа_техники	<КлассО>=	классификация техники	<ТипО>=	-	<КлассО>=	Наименование подтипа техники
3	Vehicle classifier	Код наименования техники	<ЗначХО>=	<Раздел>=	UGES	<ЭкзО>=	#Наименование_подтипа_техники	$<КлассО>= классификация техники; <ТипО>=-; <ЭкзО>=# Наименование_типа_техники ; / <КлассО>=	Наименование подтипа техники	<ТипО>=	-		
4	Motor	Марка двигателя		<Раздел>=	UGES			<КлассО>=	Двигатель	<ТипО>=	#Марка двигателя		
5	Driver	Номер водительских прав	<ЗначХС>=	<Раздел>=	UGES	<ЭкзО>=	#ФИО водителя	<КлассО>=	Водитель	<ТипО>=	-	$<ВремяНС>=# Дата получения водительских прав; <ТипХС>[] (строковая, любая, 1)=Номер водительских прав; <КлассС>=	Получение водительских прав
6	Driver	Категория прав	<ЗначХС>=	<Раздел>=	UGES	<ЭкзО>=	#ФИО водителя	<КлассО>=	Водитель	<ТипО>=	-	$<ВремяНС>= # Дата получения водительских прав;<ТипХС>[] (строковая, списочная, 1)= Категория водительских прав;<КлассС>=	Получение водительских прав
…	…	…	…	…	…	…	…	…	…	…	…	…	…

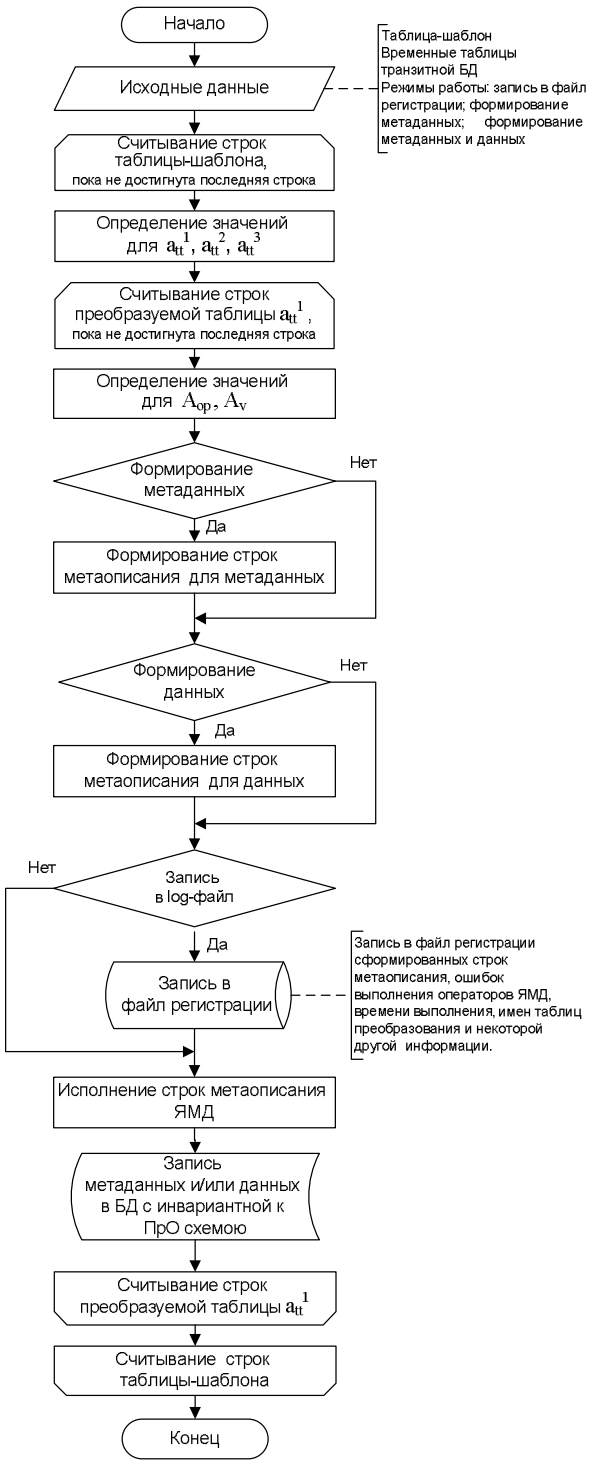

Рис. 2. Блок-схема алгоритма преобразования данных унаследованной БД в БД с инвариантной к ПрО схемою

информации, типов СУБД, сложностью автоматического выполнения переноса таких объектов и конструкций, как функции, триггеры, хранимые процедуры и т. д., в рассматриваемом методе отсутствуют благодаря использованию схемы БД с универсальным базисом отношений.

Этап 4. Оценка завершенности преобразования БД.

Решение о завершении процесса преобразования унаследованной БД, в БД, построенную на основе инвариантной к ПрО схемы, принимается на основе сравнительного анализа значений атрибутов качества созданной БД, полученных с использованием метрик модели качества [11]:

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

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), $At_{jl}^{DB(i)} \in Z$, и требований, предъявляемых к этим атрибутам со стороны потребителя информационного продукта. В случае неудовлетворительных значений атрибутов качества БД формулируется набор рекомендаций по внесению соответствующих изменений, и этапы 2–4 повторяются.

В результате, предлагаемый метод обеспечивает комплексное решение задачи преобразования унаследованных БД. Ценность такого решения заключается в том, что оно способствует расширению функциональности унаследованных БД, снижению стоимости их сопровождения, улучшению определенных характеристик качества, в том числе способности к адаптации в условиях динамичных изменений ПрО, как результат использования в качестве новой БД базы данных, построенной на основе схемы, инвариантной к предметным областям.

Данный метод был апробирован на нескольких БД для газовой промышленности, систем бухгалтерского учета.

ВЫВОДЫ

В данной научной статье предложен метод преобразования унаследованных баз данных в базу данных с универсальным базисом отношений, обеспечивающий комплексное решение задачи модернизации унаследованных БД. Разработанный метод позволяет расширить функциональность унаследованных БД, улучшить определенные их характеристики качества, в том числе, способность к адаптации в условиях динамичных изменений предметных областей, способствует снижению стоимости сопровождения БД ИС. Благодаря разработанному методу становится возможным использование имеющих значимую практическую ценность данных унаследованных БД в системе, основанной на новой технологии.

Литература

[1] Measuring data management practice maturity: a community's self-assessment / [P. Aiken, M. D. Allen, B. Parker, A. Mattia] // IEEE Computer Society. – 2007. – V. 40. – №. 4. – С. 42–50.

[2] *Есин В. И.* Семантическая модель данных «объект-событие» / В. И. Есин // Вісник Харківського національного університету імені В. Н. Каразіна. Сер.: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління. – 2010. – № 925. – С. 65–73.

[3] Security and noise immunity of telecommunication systems: new solutions to the codes and signals design problem. Collective monograph. – Minden, Nevada, USA : ASC Academic Publishing. – 2017. – 198 p. (Yesin V. I., Yesina M. V. Chapter 8, Means for conceptual modeling of information system databases, P. 160–196).

[4] *Есин В. И.* Модель данных «объект-событие»: требования и синтез модели // Computer science and cyber security – International electronic scientific journal. – 2017. – Issue. 3 (7). – P. 33–44, https://periodicals.karazin.ua/cscs/article/view/10003, last accessed 2018/07/28.

[5] *Есин В. И.* Выразительные средства модели данных «объект-событие» / В. И. Есин // Радиотехника: Всеукр. межвед. науч.-техн. сб. – 2017. Вып. 191. – С. 99–112.

[6] *Есин В. И.* Универсальная модель данных и ее математические основы / В. И. Есин // Системи обробки інформації. – 2011. – № 2(92). – С. 21–24.

[7] *Есин В. И.* Модель данных с универсальной фиксированной структурой / В. И. Есин // Теоретичні та прикладні аспекти побудови програмних систем : матеріали міжнародної наукової конференції, м. Київ, 15–17 грудня 2014 р. – Кіровоград : ФО-П Александрова М. В., 2014. – С. 112–116.

[8] *Есин В. И.* Инвариантная к предметным областям схема базы данных и ее отличительные особенности / В. И. Есин // Радиотехника: науч.-техн. журнал. – 2018. Вып. 193. – С. 133–142.

[9] *Лаврищева Е. М.* Методы и средства инженерии программного обеспечения // Е. М. Лаврищева, В. А. Петрухин. – М. : МОН РФ, 2007. – 415 с.

[10] *Есин В. И.* Язык для универсальной модели данных / В. И. Есин, М. В. Есина // Системи обробки інформації. – 2011. – № 5(95). – С. 193–197.

[11] *Yesin V. I.* A cybernetic approach to solving the problem of database reengineering // Telecommunications and Radio Engineering. – 2018. Volume 77, Issue 5. – P. 399–409. doi: 10.1615/TelecomRadEng.v77.i5.40.

Надійшла до редколегії 25.12.2018

Есин Виталий Иванович, доктор технических наук, доцент, профессор кафедры безопасности информационных систем и технологий ХНУ имени В. Н. Каразина. Область научных интересов – модели и методы разработки баз данных информационных систем и обеспечение их безопасности.

Вилигура Владислав Викторович, аспирант факультета компьютерных наук ХНУ имени В. Н. Каразина. Область научных интересов – базы данных и обеспечение их безопасности.

УДК 004.652 : 004.658.3

Єсін В. І. **Метод перетворення успадкованих баз даних у базу даних з універсальним базисом відношень** / В. І. Єсін, В. В. Вілігура // Прикладна радіоелектроніка: наук. – техн. журнал. – 2018. – Том 17, №. 3, 4. – С. 176–181.

Пропонується метод перетворення успадкованих баз даних в базу даних з універсальним базисом відношень, що дозволяє розширити функціональність успадкованих БД, поліпшити певні їхні характеристики якості, в тому числі, здатність до адаптації в умовах динамічних змін предметних областей, що сприяє зниженню вартості супроводу баз даних.

Ключові слова: база даних, успадкована база даних, база даних з універсальним базисом відношень.

Табл.: 01. Іл.: 02. Бібліогр.: 11 найм.

UDC 004.652 : 004.658.3

Yesin V. I. **The method of converting legacy databases into a database with an universal basis of relations** / V.I. Yesin, V.V. Vilihura // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 176–181.

The method of converting legacy databases into a database with an universal basis of relations is proposed, which enables to extend the functionality of legacy databases, improve their certain quality characteristics, including the ability to adapt in the conditions of dynamic changes in subject domains, which contributes to reducing the cost of maintaining databases.

Keywords: database, legacy database, the database with an universal basis of relations.

Tab. 01. Fig. 02. Ref.: 11 items.

УДК 004.056.55

ПОРІВНЯЛЬНИЙ АНАЛІЗ БІОМЕТРИЧНИХ КРИПТОСИСТЕМ

М. С. ЛУЦЕНКО, О. О. КУЗНЕЦОВ, Д. І. ПРОКОПОВИЧ-ТКАЧЕНКО, В. П. ЗВЄРЄВ, А. О. УВАРОВА

Досліджуються існуючі біометричні криптографічні системи, які призначено, зокрема для формування надійних та безпечних псевдовипадкових послідовностей (т. з. криптографічних ключів, паролів, кодів доступу тощо). Проводиться порівняльний аналіз різних видів біометричних криптосистем (зі звільненням ключа; зі зв'язуванням ключа; з генерацією ключа), визначаються їх переваги та недоліки. Наводяться конкретні схеми та обчислювальні алгоритми використання біометричних образів для формування криптографічних ключів, обґрунтовуються перспективні напрямки подальших досліджень.

Ключові слова: біометричні образи, кодові криптосистеми, генерація криптографічних ключів.

ВСТУП

З розповсюдженням інформації через Інтернет та необхідністю збереження конфіденційних даних у відкритих мережах створення надійної інформаційної системи постає важливою та актуальною науковою задачею [1–3].

Наразі існує досить багато доказово стійких криптографічних алгоритмів, які можуть бути використані в різних інформаційних системах [3]. Однак, слід зауважити, що зазвичай досить велику роль в безпеці системи відіграють криптографічні ключі (паролі, коди доступу, тощо) які використовуються для ініціації інших криптопримітивів та для організації первинного доступу до приватної інформації. Запам’ятати криптографічно сильний ключ користувачеві фізично неможливо або вкрай складно, саме тому надійність системи часто залежить від простого для запам’ятовування користувачем секретного слова (пароля). Очевидно, що такий підхід має потенційні ризики для безпеки інформаційної системи [1–3].

Біометрична автентифікація дозволяє реалізувати механізм захисту ключових даних, використовуючи унікальні біометричні ознаки користувача [4–12]. Наразі вже розроблено кілька варіантів побудови таких систем. Наприклад, коли користувач хоче отримати доступ до захищеного ключа, йому може бути запропоновано надати відповідний біометричний зразок. Якщо цей зразок пройде верифікацію, тоді криптографічний ключ буде можливо використати [4].

Таким чином, біометрична автентифікація може замінити або вдосконалити використання кодів доступу, секретних ключів, паролів, тощо. Це забезпечує зручність, оскільки більше не потрібно запам'ятовувати чи зберігати в надійному середовищі криптостійку псевдовипадкову послідовність. До того ж, біометричні дані можуть стати повною заміною криптографічним ключам шляхом формування стійких та надійних псевдовипадкових послідовностей [9, 10, 12]. Ці ключі можуть бути використані в різноманітних програмах, включаючи доступ до віртуальних приватних мереж, шифрування файлів та автентифікації користувачів.

Слід відмітити, що біометричні системи також мають певні недоліки з практичного застосування [5, 6]. Зокрема, за результатом проведеного аналізу встановлено, що біометрична криптографія потенційно вразлива до таких поширених атак [1–4]:

– атака підміною (англ. Spoofing attack). Було продемонстровано, що іноді біометричну систему можна обманювати, застосовуючи підроблені біометричні образи;

– атака заміни (англ. Substitution Attack): біометричний шаблон повинен зберігатися для підтвердження користувача. Якщо зловмисник отримує доступ до сховища, локально або віддалено, він може перезаписати шаблон легітимного користувача;

– атака маскарад (англ. Masquerade Attack). Цифровий образ може бути створений з шаблону біометричного образу. Ця атака створює реальну загрозу для систем віддаленої автентифікації;

– атака найближчого самозванця (англ. Nearest Impostor Attack). Ця атака стосується систем, у яких використовуються шаблони. У цій атаці використовується велика множина різних біометричних шаблонів для викриття секретних даних з високою ймовірністю;

– недостатня точність багатьох біометричних систем, як з точки зору фальшивого відхилення (англ. False Reject Rate), так за оцінкою частоти помилок (англ. False Accept Rate). Високий рівень FRR створює незручності для законних користувачів і спонукає знизити поріг перевірки. Це неминуче призводить до FAR, що, в свою чергу, знижує рівень безпеки системи.

Залежно від мети застосування біометрії в криптографії виділяються кілька видів біометричних криптографічних систем [4–6]. Їхню загальну класифікацію за результатами проведеного аналізу зображено на рис. 1:

– системи зі звільненням ключа (англ. Key Release Cryptosystems);

– системи зі зв'язуванням ключа (англ. Key Binding Cryptosystems);

– системи з генерацією ключа (англ. Key Generation Cryptosystems).

Метою цієї статті є аналіз та порівняльні дослідження біометричних криптографічних систем, об-

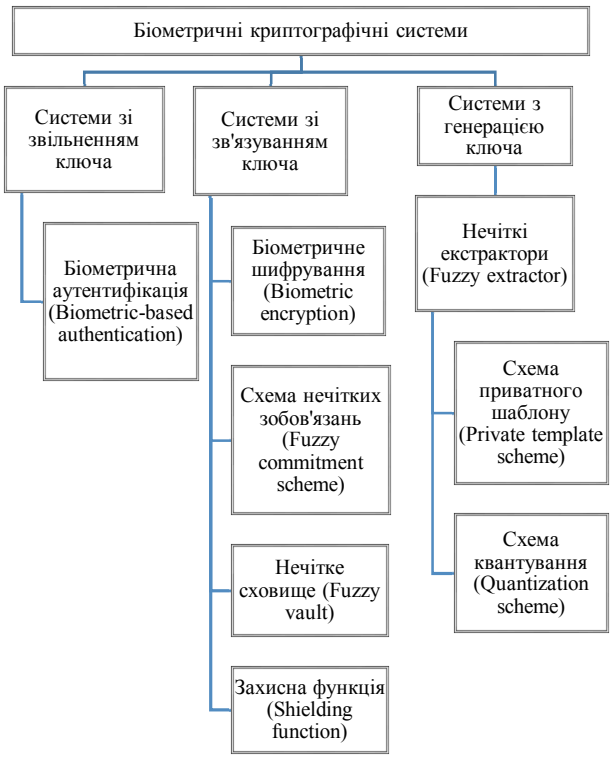

Рис. 1. Види та підвиди біометричних криптографічних систем

ґрунтування перспективних напрямків їхнього розвитку та можливого застосування.

2. БІОМЕТРИЧНІ КРИПТОСИСТЕМИ ЗІ ЗВІЛЬНЕННЯМ КЛЮЧА

У режимі звільнення ключа біометрична автентифікація здійснюється незалежно від механізму звільнення ключа, біометричний еталон і ключ зберігаються окремо один від одного, сам ключ звільняється після успішної біометричної автентифікації [4–6]. Схематичне зображення такої біометричної криптографічної системи наведено на рис. 2.

Рис. 2. Схематичне зображення біометричної криптографічної системи зі звільненням ключа

Отже, біометрична автентифікація відокремлена від криптографічної частини системи. В такому випадку, як пароль від ключа використовуються результати порівняння отриманого біометричного зображення з шаблоном. Даний вид біометричних криптосистем непридатний у більшості випадків, оскільки існує можливість заміни модуля порівняння під час виконання автентифікації. Цей метод буде добре працювати в додатках фізичного доступу, де біометричні шаблони та ключі можуть зберігатися в безпечному місці, фізично відокремленому від пристрою захоплення біометричних зображень.

3. БІОМЕТРИЧНІ КРИПТОСИСТЕМИ ЗІ ЗВ'ЯЗУВАННЯМ КЛЮЧА

У криптографічних системах такого типу ключ і біометричний еталон криптографічно пов'язані між собою [4–7]. Ключ за певним алгоритмом пов’язується з біометричним еталоном користувача і зберігається в такому вигляді в базі даних, відповідно розкрити ключ представляється можливим тільки власникові біометричних параметрів. У таких системах передбачається, проте не є необхідним, використання допоміжних даних (англ. Helper Data), для демаскування зашумлених біометричних даних. Схематично приклад біометричних систем зі зв’язуванням ключа подано на рис. 3.

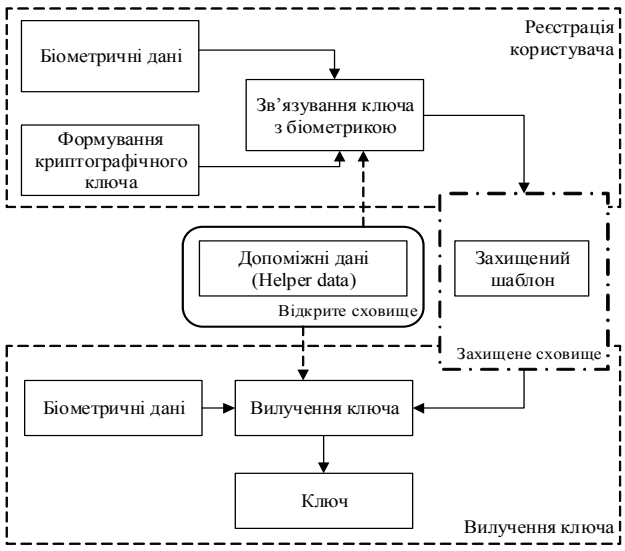

Рис. 3. Схематичне зображення біометричної криптографічної системи зі зв'язуванням ключа

Цей спосіб включає приховування криптографічного ключа в самому біометричному шаблоні реєстрації за допомогою надійного (секретного) алгоритму бітової заміни. Після успішної автентифікації користувачем цей довірений алгоритм просто витягає біти ключа, після цього ключ придатний для користування. На жаль, це означає, що криптографічний ключ буде вилучено з того ж розташування в шаблоні щоразу, коли інший користувач автентифікується системою. Таким чином, якщо зловмисник міг визначити бітові розташування, які вказують ключ, то зловмисник може

відновити вбудований ключ із будь-якого шаблону інших користувачів. Якщо зловмисник мав доступ до системи, то він міг визначити місця розташування ключа, наприклад, додаючи кількох людей у систему, використовуючи однакові ключі для кожної реєстрації. В такому випадку, злочинцю потрібно лише знайти ці місця розташування біт з загальною інформацією в шаблонах.

3.1 Біометричне шифрування

Біометричне шифрування є процесом, який надійно пов'язує криптографічний ключ з біометричним, тому що ні ключ, ні біометричні дані не можуть бути вилучені зі збереженого шаблону [7]. Криптографічний ключ генерується випадковим чином при реєстрації так, що ніхто, включаючи користувача, не знає його. Сам ключ повністю незалежний від біометрії і, отже, завжди може бути змінений або оновлений. Після отримання біометричного зразка створюється захищений шаблон, так званий «приватний шаблон» (англ. private template). По суті, криптографічний ключ шифрується за допомогою біометричного. Під час перевірки користувач представляє свій біометричний зразок, який в ході застосування до легітимного шаблоном дозволить отримати той самий ключ. Ключ відтворюється тільки в тому випадку, якщо під час перевірки представлений правильний біометричний зразок. Таким чином, біометрія служить ключем дешифрування Алгоритм розроблений так, що незначна розбіжність отриманих біометричних образів від однієї особи не впливає на коректну роботу алгоритму.

Підходи, засновані на біометричному шифруванні, чутливі до атаки змішаної заміни, атаки сходження і атаки найближчого самозванця. Більш того, зловмисник може використовувати відновлений секрет для вилучення оригінальних біометричних даних з захищених шаблонів.

Mytec1 був першою реалізацією такої біокріптографічної схеми. Пізніше Mytec2 був розроблений для забезпечення більш складного захисту шаблону відбитка пальця, що зберігається. Схематичне зображення біометричної криптографічної системи наведено на рис. 4.

3.2 Схема нечітких зобов’язань

Схема нечітких зобов'язань [8] є криптографічним алгоритмом, який забезпечує збереження біометричних даних за допомогою методів криптографії та кодування з виправленням помилок. Алгоритм пов'язує секретну інформацію з даними, щоб приховати дані і не дозволити власникові даних розкрити її більш ніж одним способом. Нечіткі схеми зобов'язань використовувалися для збереження біометричних шаблонів. Ця схема, що застосовується до біометричних шаблонів, розглядає сам шаблон без будь-якої модифікації як пошкоджене кодове слово, яке підлягає декодуванню. У таких системах для формування шаблону та вилучення даних використовується так званий свідок (або ключ шифрування). Також вважається, що для коректного функціонування алгоритму свідок може мати схожі, проте не ідентичні метрики.

Схематичне зображення біометричної криптографічної системи наведено на рис. 5.

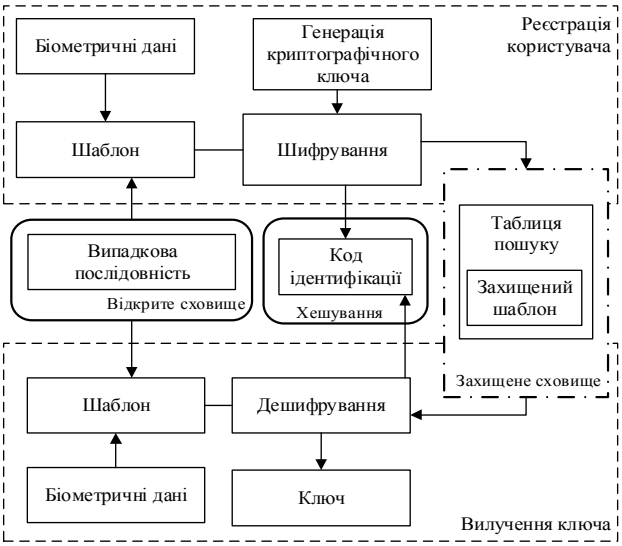

Рис. 4. Схематичне зображення біометричної криптографічної системи зі зв'язуванням ключа: біометричне шифрування

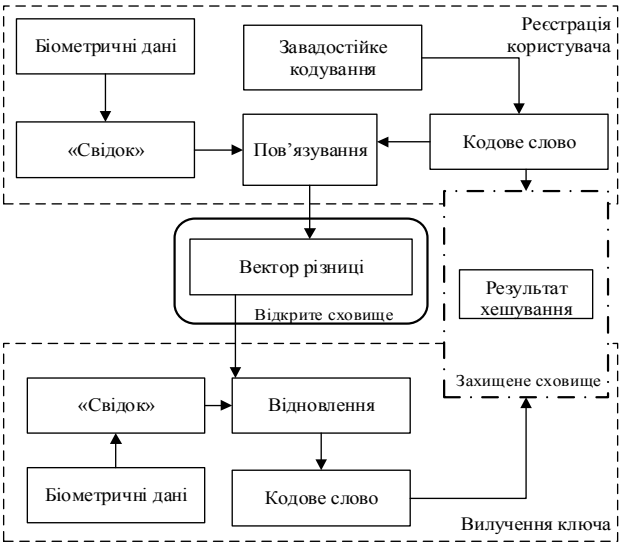

Рис. 5. Схематичне зображення біометричної криптографічної системи зі зв'язуванням ключа: схема нечітких зобов’язань

Основною слабкістю схеми нечітких зобов'язань є сприйнятливість до атак у середовищах, які залучають довірену третю сторону. Схеми нечітких зобов'язань з максимальним розміром ключів забезпечують оптимальну продуктивність для абсолютно безсистемного випадку. Також було виявлено, що нечіткі схеми зобов'язань забезпечують обмежену безпеку секретного ключа та біометричних даних у загальних випадках без пам'яті та у стаціонарних ергодичних випадках. Нечіткі схеми зобов'язань також сприйнятливі до атаки грубою силою. Атаки через множинність записів також

можуть бути використані для декодування захищених біометричних даних.

3.3 Нечіткі сховища

У своїй роботі Арі Джуэлс і Мадху Судан в 2002 році описали нову конструкцію, яку вони назвали нечітким сховищем (англ. Fuzzy vault) [9]. Узагальнено основну ідею, можна описати так. Нехай Аліса помістить деяке секретне значення k у нечітке сховище та «заблокує» його, використовуючи набір (підмножину) елементів з деякої відкритої універсальної множини A. Якщо Боб спробує «розблокувати» таке сховище, використовуючи набір B елементів (потужності підмножини A та B співпадають), то він удачно отримає секретне значення лише у випадку, якщо B подібно A. Тобто тільки, якщо A та B значною мірою є множинами, що перетинаються. Відмінною особливістю цього алгоритму за думкою авторів є те, що ця схема володіє інваріантністю порядку значень в наборах, вважається, що упорядкування підмножин A та B не впливає на роботу схеми. У такому випадку схема має доказану криптографічну стійкість до обчислювальних атак типу груба сила. Схематичне зображення біометричної криптографічної системи наведено на рис. 6.

Рис. 6. Схематичне зображення біометричної криптографічної системи зі зв'язуванням ключа: схема нечітких сховищ

Розглянемо основні принципи схеми нечіткого сховища на прикладі. Нехай Аліса – любитель кіно. Вона шукає того, хто розділяє її інтереси, але не хоче розкривати інформацію про свої уподобання всім людям. Один з підходів, який вона може зробити, полягає в тому, щоб створити набір її улюблених фільмів і опублікувати його в прихованій формі. Наприклад, Аліса може опублікувати зашифрований текст C_A, який представляє її зашифрований на наборі (в даному випадку, ключі) A телефонний номер tel_A. В цьому випадку, якщо інша людина, скажімо, Боб, склав свій список улюблених фільмів B, і якщо вони ідентичні A, він зможе розшифрувати C_A і отримати телефонний номер tel_A Аліси. Якщо ж Боб спробує розшифрувати C_A за допомогою множини даних, що відрізняються від множини інтересів Аліси, він не зможе отримати її номер телефону. Недоліком цього підходу є його точність у визначенні подібності двох множин або відсутність допущення помилок. Якщо інтереси Боба дуже схожі на інтереси Аліси, наприклад, якщо йому подобаються кілька фільмів, які Аліса не любить, то він не дізнається її телефон. Цілком ймовірно, що в цьому випадку Аліса все одно хотіла б, щоб Боб отримав її номер телефону, оскільки їх смаки дуже схожі. Автори ж пропонують поняття нечіткого сховища. Це криптографічна конструкція, відповідно до якої Аліса може заблокувати свій номер телефону tel_A, використовуючи набір даних A, помістивши його в сховище, позначене V_A. Якщо Боб намагатиметься розблокувати сховище V_A, використовуючи свій власний набір, йому це вдасться, якщо множини A та B значно перетинаються. З іншого боку, будь-який, хто намагатиметься розблокувати сховище V_A за множиною даних, що істотно відрізняються від набору Аліси, зазнає невдачі. Таким чином, нечітке сховище можна розглядати як схему перевірки помилок, у якій ключі складаються з наборів.

На думку авторів, їхня система може знайти застосування системах, в яких безпека залежить від людських чинників. Наприклад, нечіткі сховища можуть знайти застосування:

Для захисту конфіденційних даних. У такому випадку, схема може використовуватися для кола людей (наприклад, сім'ї), що володіють деякими схожими параметрами, щоб зберегти деякі дані в таємниці.

Для відновлення пароля або інших даних. Зараз поширена практика, що для відновлення пароля при реєстрації в системі користувач вигадує відповіді на деякі стандартні запитання, потім в разі відновлення пароля йому необхідно дати відповіді на ці запитання, тобто сформувати деякий набір даних, які ідентифікують системі цього користувача. Оскільки дана схема може бути застосована до набору даних, то можливо її використання для відновлення пароля користувача навіть з урахуванням того, що користувач міг дати неправильні відповіді на кілька запитань.

Біометрія. Нечіткі сховища можна застосовувати щодо біометричних зразків. Наприклад, як ключ шифрування. Аліса могла б зберігати пароль, заблокований в нечіткому сховище і зашифрований на її наборі даних отриманих з її біометричного зразка, наприклад, відбит-ка пальця, тим самим забезпечуючи стійкість системи до помилок і конфіденційність, даних що зберігаються в системі.

3.4 Захисна функція

Захисна функція або схема допоміжних даних була розроблена для забезпечення безпеки збережених

біометричних даних і гарантування конфіденційності законних користувачів [4–7]. Цей підхід дозволяє системі автентифікації перевіряти ідентичність користувача без будь-яких знань про біометричні дані користувача. Дельта-договірні і епсілон-виявляючі функції забезпечують основу цієї схеми. Функція дельта-контрактування пов'язує таємницю з біометричними даними, а функція епсилон-виявлення гарантує, що захищений шаблон відкриває лише невелику кількість інформації про випадкові та біометричні дані.

Захисна функція не стійка до атак типу спуфінг та атак грубою силою. Попередні реалізації схем такого типу виробляють ключі, які менше 50 біт. Це не відповідає мінімальним вимогам для біометричних ключів. Також такі системи сприйнятливі до атак з використанням кратності записів і перехресного нападу. Більш того, зловмисник може використовувати реконструйований секрет для отримання оригінальних біометричних даних з компрометованих допоміжних даних.

Схематичне зображення біометричної криптографічної системи наведено на рис. 7.

Рис. 7. Схематичне зображення біометричної криптографічної системи зі зв'язуванням ключа: схема захисної функції

3.5 Порівняльний аналіз біометричних криптографічних систем зі зв'язуванням ключа

Порівняння переваг та недоліків біометричних криптографічних систем зі зв'язуванням ключа наведено у Таблиці 1.

4. БІОМЕТРИЧНІ КРИПТОСИСТЕМИ З ГЕНЕРАЦІЄЮ КЛЮЧА

У такій біометричній криптосистемі ключ формується безпосередньо з біометричних даних користувача і не зберігається в базі даних [4–7, 10]. Дослідник Бодо запропонував такий метод у німецькому патенті. Цей патент передбачає, що дані, отримані з біометричних образів (по суті, біометричного шаблону), використовуються безпосередньо як криптографічний ключ.

Таблиця 1

Порівняльний аналіз біометричних криптографічних систем зі зв'язуванням ключа

	Переваги	Недоліки
Схема біометричного шифрування	1) Застосовує стандартний криптографічний алгоритм для створення безпечного біометричного шаблону 2) Зловмисник не має можливості розшифрувати захищені шаблони без знання алгоритму і криптографічного ключа	1) Існує можливість використання реконструйованого секрету для отримання оригінальних біометричних даних із захищеного шаблону 2) Не стійкі до атак змішаної заміни, атак сходження (атака маскарад) і атак найближчого самозванця, атак множинного запису
Схема нечітких зобов'язань	"Зобов'язання", отримані з біометричних даних і секретного ключа, захищають біометричний шаблон. Також секретний ключ захищений завдяки тому, що зберігається лише його хеш-значення	1) Вразливі перед усіма відомими атаками на завадостійке кодування (залежить від обраного алгоритму кодування) 2) Не стійкі до атак сходження, атак грубою силою, атак декодування та перехресних атак
Схема нечіткого сховища	Сховище не може бути декодовано без біометричних даних, які мають майже ідентичні характеристики з первинними	1) Сприйнятливі до атак грубою силою та колізій 2) Вразливі до атак вторгнень, атак зв'язків, комбінованих та ін'єкційних атак
Схема захисної функції	1) Допоміжні дані та хеш-функції захищають від відтворення біометричні дані та випадкові секрети відповідно 2) Оригінальні біометричні дані не можуть бути відновлені з захищеного шаблону без знання секретного ключа	1) Коротка довжина ключів, що отримуються 2) Можливість використання реконструйованого секрету для отримання оригінальних біометричних даних із компрометованих допоміжних даних 3) Не стійкі до атак грубою силою та перехресних атак, до атак множинного запису

Можливість не зберігати ключ, отриманий з біометричних даних, є незаперечною перевагою методу генерації криптографічних ключів з біометричних даних користувача порівняно з іншими існуючими методами. Таким чином, головною відмінністю двох останніх видів біометричних криптосистем є те, що в одному з них криптографічний ключ тільки закривається за допомогою біометричного зразка, а в іншому ключ генерується безпосередньо з біометричних даних користувача. Схематичне подане такої системи наведено на рис. 8.

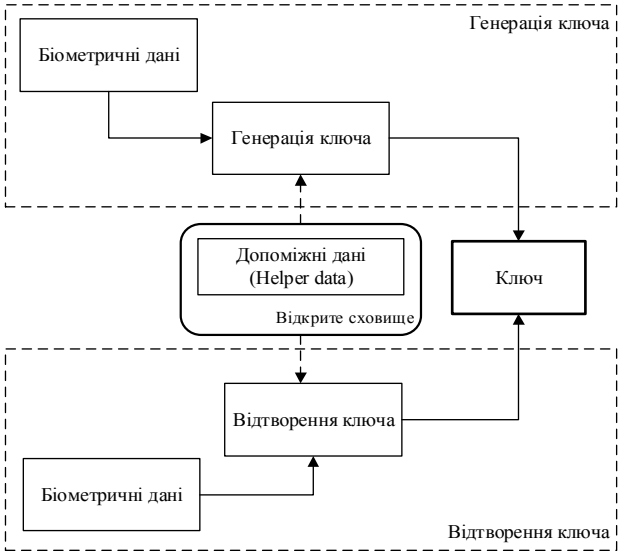

Рис. 8. Схематичне зображення біометричної криптографічної системи з генерацією ключа

Такі системи є більш безпечними, але їх важко застосовувати через навіть незначну мінливість біометричних характеристиках, оскільки необхідно з приблизно схожих даних згенеруючи той самий ключ знову і знову. Також недоліком таких систем є неможливість (або обмеженість кількості можливостей) сформувати новий ключ. Отже, якщо криптографічний ключ коли-небудь буде скомпрометований, то використання цього конкретного біометричного образу та конкретного алгоритму генерації ключа буде неможливе. У системі, де потрібне періодичне оновлення криптографічного ключа, це неприйнятно.

4.1 Нечіткі екстрактори

Найпоширенішою технологією, на якій базуються біометричні криптографічні системи з генерацією ключа, – це нечіткі екстрактори [10–12]. Базова логіка використання нечітких екстракторів схожа з логікою нечітких сховищ. Даний спосіб дозволяє однозначно відновлювати секретний ключ з неточно відтворюваних (зашумленних) біометричних даних. Метод нечітких екстракторів передбачає формування випадкової рівномірно розподіленої послідовності з початкових даних і подальше коректне її відновлення з будь-яких даних, досить схожих з початковими. Такі методи базуються на теорії інформації та завадостійкого кодування. Використовування методу нечітких екстракторів здатне компенсувати помилки, що виникають внаслідок технічної неможливості отримання однакових значень біометричних характеристик під час їхнього повторного введення користувачем. Метод нечітких екстракторів дозволяє отримувати ключ, який задовольняє всі критерії якості криптографічних ключів. У деяких реалізаціях передбачається формування ключа шляхом об'єднання допоміжних даних (отриманих з даного біометричного шаблону) та самого біометричного зразка. Допоміжні дані можуть бути отримані з заданого біометричного еталону і зберігатися у вигляді поновлюваного ключа або хеш-значення.

Загальна концепція побудови такого генератора криптографічних ключів полягає в наступному. Спочатку випадковим чином генерується бітова послідовність, яка кодується завадостійким кодом. Як завадостійкий код, що виправляє помилки, можуть використовуватися коди Хеммінга, Адамара, Боуза – Чоудхурі - Хоквінгема (БЧХ-коди), Ріда - Соломона (є окремим випадком БЧХ). Згенерована бітова послідовність може бути призначена для ідентифікації, автентифікації або генерації криптографічних ключів шифрування. Дана послідовність об'єднується з еталонними характеристиками біометричних ознак суб'єкта (біометричних еталоном).

Способи об'єднання можуть бути різними: від додавання за модулем 2 до використання алгоритмів, що більше орієнтовані на оброблювані дані. Результатом об'єднання є відкритий рядок, який може зберігатися на загальнодоступному сервері. Щоб отримати згенеровану раніше послідовність (криптографічний ключ) користувач вводить власні біометричні ознаки, які обробляються відповідним чином і «віднімаються» з відкритого рядка. Після вилучення біометричних даних отримана бітова послідовність буде змінена внаслідок нечіткості введених біометричних даних відносно еталонних. Після застосування коду, що виправляє помилки до отриманого рядка, в разі високого ступеня «схожості» висунотого біометричного образу і еталонного (тобто, якщо кількість розбіжних біт еталонних і висунутих даних не перевищує виправлячу здатність коду), буде відновлена послідовність бітів, яка і є криптографічним ключем. Описаний метод схематично зображено на рис. 9.

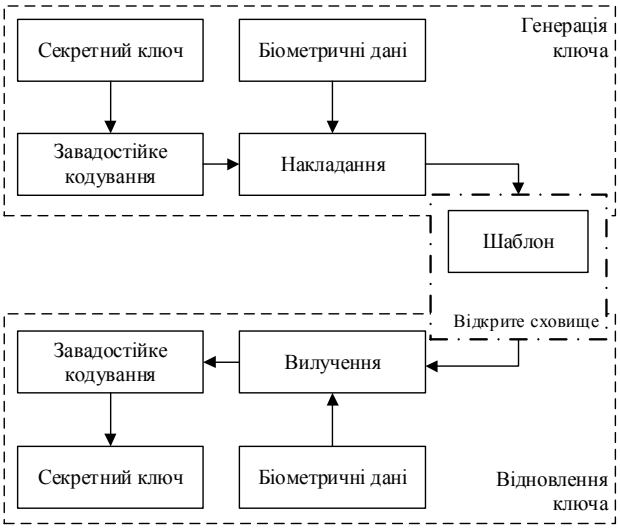

Рис. 9. Схематичне зображення методу нечітких екстракторів

Два основних підходи, що використовуються для генерації біометричних ключів – схема приватного шаблону та схема квантування.

4.2 Схема приватного шаблону

У схемах приватного шаблону (англ. Private template scheme) [10–12] використовуються допоміжні дані – послідовності бітів перевірки для виправлення помилок завадостійким кодом. Сам ключ формується безпосередньо з біометричного образу або з хешу цього біометричного образу. Наведемо як приклад алгоритм схеми приватного шаблону, в якій ключ формується з хешу.

Алгоритм генерації ключа:

1. До біометричного образу довжиною M біт, застосовується завадостійке декодування. У результаті формується вектор $Vec(V)=(V_1,V_2,...,V_n)$ для n-бітного коду, визначений як $Vec(v_i)=(v_{i,1},v_{i,2},...,v_{i,n})$, де $V_j = majority(v_{1,j},v_{2,j},...,v_{M,j})$

2. Отриманий після декодування характеристичний вектор $Vec(T)$ поєднується з вектором контрольної суми $Vec(C)$: $Vec(T)\|Vec(C)$, де вектор $Vec(C)$ є частиною коду, виправляючого помилки.

3. Для формування ключа виконується хешування $Vec(T)\|Vec(C)$ та, залежно від реалізації, з допоміжними даними

$$Key = Hash\left[Vec(T)\|Vec(C), helper\_data\right].$$

Алгоритм відтворення ключа:

1. До біометричного образу довжиною M біт, застосовується мажоритарне декодування. Формується вектор $Vec(V')=(V'_1,V'_2,...,V'_n)$ для n-бітного коду, визначений як $Vec(v'_i)=(v'_{i,1},v'_{i,2},...,v'_{i,n})$, де $V'_j = majority(v'_{1,j},v'_{2,j},...,v'_{M,j})$.

2. Отриманий після декодування характеристичний вектор $Vec(T')$ поєднується з вектором контрольної суми $Vec(C)$

$$Vec(T')\|Vec(C),$$

де вектор $Vec(C)$ є частиною коду, виправляючого помилки.

3. Для формування ключа виконується хешування $Vec(T)\|Vec(C)$ та, залежно від реалізації, з допоміжними даними

$$Key = Hash\left[Vec(T)\|Vec(C), helper\_data\right].$$

Відтворений ключ можна використовувати за призначенням.

Схематичне зображення такого алгоритму наведено на рис. 10. Експериментальні результати та аналіз безпеки показують, що схема приватних шаблонів є простою та ефективною, також слід зазначити, що під час використання такої схеми відновлення початкових біометричних даних із захищених шаблонів неможливе.

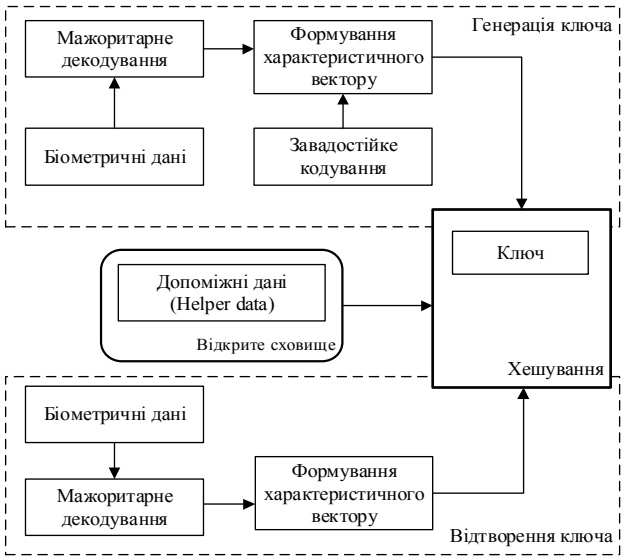

Рис. 10. Схематичне зображення біометричної криптографічної системи з генерацією ключа: схема приватного шаблону

4.3 Схема квантування

Схеми квантування (англ. Quantization scheme) [10]12] створюють біометричні ключі, використовуючи допоміжні дані та бінаризовані (або квантовані) біометричні характеристики. Унікальною особливістю схем квантування є можливість отримувати одні і ті самі ключі з досить різноманітних біометричних образів особи, навіть якщо вони отримані за допомогою різних сканерів. Результати експериментів з використанням як біометричні образи відбитків пальців показують, що до 40% відбитків створюють один і той самий криптографічний ключ, навіть коли різні сканери використовувались для захоплення відбитків пальців.

Схематично функціонування схеми квантування подано на рис. 11.

Рис. 11. Схематичне зображення біометричної криптографічної системи з генерацією ключа: схема квантування

Схеми квантування можуть бути реалізовані як мультимодальні системи; тобто унікальний ключ може бути отриманий з об'єднання двох або більше біометричних метрик.

Слабкістю схем квантування на основі допоміжних даних є можливість відновлення оригінальних біометричних зображень із захищених шаблонів. Зловмисник може отримати характеристичні вектори з захищених шаблонів, а потім відновити реальний біометричний образ. Схеми квантування, які використовують допоміжні дані, вразливі для атаки за допомогою множинності записів. Отже, існує потреба в мінімізації обсягу корисної інформації, яка може бути доступна зловмиснику, якщо система буде скомпрометована.

Для функціонування схеми квантування необхідні характеристичні вектори декількох біометричних зразків для обчислення відповідних інтервалів для кожного елемента характеристики (необхідні характеристичні вектори з реальними значеннями). Ці інтервали кодуються та зберігаються у вигляді допоміжних даних. Під час ідентифікації знову фіксуються біометричні характеристики суб'єкта та відображаються у попередньо визначені інтервали, генеруючи хеш-ключ. Для того, щоб забезпечити оновлюванні ключі або хеші, більшість схем забезпечують параметризоване кодування інтервалів. Схеми квантування дуже пов'язані з захисними функціями (англ. shielding function), оскільки обидва методи виконують квантування біометричних характеристик шляхом побудови відповідних інтервалів функцій. На відміну від захисних функцій, загальні схеми квантування визначають інтервали для кожної окремої біометричної характеристики, виходячи з її дисперсії. Це дає змогу покращувати коректування збережених допоміжних даних до характеру застосованої біометрії.

4.4 Порівняльний аналіз біометричних криптографічних систем з генерацією ключа

Порівняння переваг та недоліків криптографічних методів наведено у таблиці 2.

Таблиця 2

Порівняльний аналіз біометричних криптографічних систем з генерацією ключа

	Переваги	Недоліки
Схема приватного шаблону	1) Формує конкретні ключі безпосередньо з біометричних даних 2) Забезпечує захист шаблонів та конфіденційність користувачів, оскільки біометричні дані відсутні в системі автентифікації. 3) Використання сильних хеш-алгоритмів для генерації випадкових ключів з біометричних даних забезпечує криптографічну стійкість до атак грубою силою	Ключі не підлягають оновленню в разі компрометації
Схема квантування	1) Генерує біометричні ключі, використовуючи допоміжні дані та квантовані біометричні характеристики. 2) Можливо відновлювати одні і ті самі ключі з декількох екземплярів біометричних даних, отриманих навіть від різних джерел (сканерів, датчиків) 3) Забезпечує безпеку та конфіденційність, оскільки не зберігає інформацію про користувача, таку як біометричні дані.	Вразлива для атаки через множинність запису

5. ПОРІВНЯЛЬНИЙ АНАЛІЗ БІОМЕТРИЧНИХ КРИПТОСИСТЕМ

Проведемо аналіз переваг та недоліків зазначених вище біометричних криптографічних систем. Результати наведено у таблиці 3.

Таблиця 3

Порівняльний аналіз біометричних криптосистем

	Переваги	Недоліки
Біометричні криптографічні системи з звільненням ключа	1) Простота практичної реалізації 2) Більш надійна заміна паролів від ключів 3) Генерація ключа відбувається надійним криптографічним генератором	1) Шаблон потрібно зберігати в базі даних, що означає, що його можна викрасти 2) Криптографічний ключ має зберігатися як частина шаблону 3) Не гарантує високого рівня безпеки даних
Біометричні криптографічні системи зі зв'язуванням ключа	1) Ключ зберігається в замаскованому вигляді 2) Генерація ключа відбувається надійним криптографічним генератором 3) Допоміжні дані не містять ніякої інформації про біометричний образ та ключ, отже можуть зберігатися у відкритому доступі	1) Шаблон потрібно зберігати в базі даних, що означає, що його можна викрасти 2) Криптографічний ключ має зберігатися як частина шаблону
Біометричні криптографічні системи з генерацією ключа	1) Генерування ключів безпосередньо з біометричних шаблонів 2) Відсутність вразливостей, які існують під час зберігання ключа в сховищі 3) Допоміжні дані не містять ніякої інформації про біометричний образ та ключ, отже можуть зберігатися у відкритому доступі 4) Згенерований ключ не містить інформації власника біометричного образу	1) Біометричні характеристики не дають достатньої інформації для отримання надійного, поновлюваного ключа без використання будь-яких допоміжних даних 2) Складність створення нового ключа, при компрометації попереднього (обмеженість людини як носія біометричних образів) (у випадку, коли допоміжні дані не використовуються) 3) Вразливість перед атаками грубою силою та маскарад, атак помилкової ідентифікації

ВИСНОВКИ

Методи вилучення біометричних даних відомі досить давно. Проте саме на етапі створення постквантової криптографії біометричні криптографічні системи можуть знайти своє широке застосування та стати більш надійною заміною не лише паролів, формованих від особистих ключів, але й самих ключів.

Проте застосування біометричних даних як джерело ключової інформації має ряд складнощів:

– біометричні дані складно багаторазово чітко відтворити;

– біометричні дані можуть змінюватися з часом і залежать від фізичного та емоційного стану їх власника;

– проблема зміни ключів – самі по собі біометричні дані у разі компрометації більше не можуть бути використані;

– при сучасному розвитку апаратних та програмних ресурсів, біометричні дані можуть бути перехоплені та відтворені зловмисником, наприклад, зображення райдужної оболонки ока може бути просто зафіксовано камерою зловмисника.

У даній роботі розглянуто три види біометричних криптосистем. Визначено, що найбільш перспективним напрямом досліджень, з точки зору криптографічної стійкості, є біометричні криптосистеми з генерацією ключа.

Література

[1] *Горбенко І.Д., Горбенко Ю.І.* Прикладна криптологія. Теорія. Практика. Застосування: монографія. –Харків: «Форт», 2012. – 870 с.

[2] *Есин В.І., Кузнецов О.О., Сорока Л.С.* Безпека інформаційних систем і технологій. – Харків: ХНУ ім. В.Н. Каразіна, 2013. – 632 с.

[3] *Горбенко Ю.І.* Побудування та аналіз систем, протоколів і засобів криптографічного захисту інформації: монографія. – Харків: «Форт», 2015. – 960 с.

[4] *A. K. Jain, R. Bolle, S. Pankanti.* Biometrics: personal identification in networked society, 1999, Vol. 1, 434 p.

[5] *U. Uludag, S. Pankanti, S. Prabhakar, A. Jain.* Biometric Cryptosystems: Issues and Challenges, Proceedings of the IEEE, June 2004, Vol. 92, NO. 6.

[6] *Anil K. Jain, Arun Ross.* An Introduction to Biometric Recognition. IEEE Transactions on circuits and systems for video technology, 2004, Vol. 14, NO. 1. pp. 4–20.

[7] *A. Juels, M. Sudan.* A fuzzy vault scheme. Des. Codes Cryptography, 2006, Vol. 38, NO. 2, pp. 237–257

[8] *Juels A.* fuzzy commitment scheme / A. Juels, M. Wattenberg. 6th ACM Conference on Computer Communications and Security, Singapore, 1999, pp. 28-36

[9] *X. Boyen* Reusable cryptographic fuzzy extractors / X. Boyen. 11th ACM Conference on Computer and Communications Security, USA, 2004, pp. 82–91.

[10] *Y. Dodis, R. Ostrovsky, L. Reyzin, A. Smith.* Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM Journal on Computing, 2008, Vol. 38, No. 1, pp. 97–139

[11] *G. Davida, Y. Frankel, B. Matt.* On the relation of error correction and cryptography to an off-line biometric identification scheme. Proceedings of Workshop on Coding and Cryptography, Paris, France, 1999, pp. 129–138.

[12] *R. Sashank Singhvi, S. P. Venkatachalam, P. M. Kannan, V. Palanisamy.* Cryptography key generation using biometrics. International Conference on Control, Automation, Communication and Energy Conservation (INCACEC), 2009, pp. 1–6.

Надійшла до редколегії 15.12.2018

Луценко Марія Сергіївна, науковий співробітник ПАТ «ІІТ», здобувач кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – біометрична криптографія, блокові симетричні шифри.

Кузнецов Олександр Олександрович, доктор технічних наук, професор, заступник головного конструктора ПАТ «ІІТ», професор кафедри безпеки інформаційних систем і технологій Харківського національного університету ім. В.Н. Каразіна. Галузь наукових інтересів – криптографія та автентифікація, алгебраїчна теорія кодів, обробка, передача та захист інформації.

Прокопович-Ткаченко Дмитро Ігоревич, кандидат технічних наук, завідувач кафедри кібербезпеки Університету митної справи та фінансів. Заступник Голови Державної служби спеціального зв'язку та захисту інформації України (2013 – 2014 р.). Галузь наукових інтересів – інформаційна та кібербезпека держави, біометрична криптографія, автентифікація та безпека безпроводових мереж.

Звєрєв Володимир Павлович, кандидат технічних наук, старший науковий співробітник, Помічник Голови Національної поліції України, Голова Державної служби спеціального зв'язку та захисту інформації України (2014 – 2015 р.). Галузь наукових інтересів – інформаційна та кібернетична безпека держави.

Уварова Анна Олександрівна, провідний інженер Конструкторського бюро «Південне» ім. М. К. Янгеля», здобувач кафедри безпеки інформаційних систем і технологій Харківського національного університету імені В.Н. Каразіна. Галузь наукових інтересів – біометрична криптографія, блокові симетричні шифри.

УДК 004.056.55

Луценко М. С. **Сравнительный анализ биометрических криптосистем** / М. С. Луценко, А. А. Кузнецов, Д. И. Прокопович-Ткаченко, В. П. Зверев, А. А. Уварова // Прикладная радиоэлектроника: научно-техн. журнал. – 2018. – Том 17, №. 3, 4. – С. 182–191.

Исследуются существующие биометрические криптографические системы, предназначенные, в частности для формирования надежных и безопасных псевдослучайных последовательностей (т.н. криптографических ключей, паролей, кодов доступа и т.д.). Проводится сравнительный анализ различных видов биометрических криптосистем (с освобождением ключа; со связыванием ключа, с генерацией ключа), определяются их преимущества и недостатки. Приводятся конкретные схемы и вычислительные алгоритмы использования биометрических образов для формирования криптографических ключей, обосновываются перспективные направления дальнейших исследований.

Ключевые слова: биометрические образы, кодовые криптосистемы, генерация криптографических ключей

Табл.: 03. Ил.: 11. Библиогр.: 12 наим.

UDC 004.056.55

Lutsenko M. **Comparative analysis of biometric cryptosystems** / M. Lutsenko, A. Kuznetsov, D. Prokopovych-Tkachenko, V. Zvieriev, A. Uvarova // Applied Radio Electronics: Sci. Journ. – 2018. – Vol. 17, № 3, 4. – P. 182–191.

Existing biometric cryptographic systems, intended, in particular, for formation of reliable and secure pseudo-random sequences (so-called cryptographic keys, passwords, access codes, etc.) are investigated. A comparative analysis of various types of biometric cryptosystems (with the release of the key; with the binding of the key, with the generation of the key) is carried out; their advantages and disadvantages are determined. Specific schemes and computational algorithms for the use of biometric images for forming cryptographic keys are given, promising areas for further research are justified.

Keywords: biometric images, code cryptosystems, generation of cryptographic keys.

Tab. 3. Fig. 11. Ref.: 12 items.