УДК 004.056.55

МОДЕЛИРОВАНИЕ ПЕРСПЕКТИВНОГО БЛОЧНОГО ШИФРА «КАЛИНА»

А.А. КУЗНЕЦОВ, Д.В. ИВАНЕНКО, Е.П. КОЛОВАНОВА

Рассматривается структура и основные базовые преобразования перспективного блочного шифра «Калина». Разрабатывается уменьшенная модель шифра посредством масштабирования основных криптопреобразований с сохранением их алгебраической структуры. Разработанная модель предназначена для проведения исследований основных характеристик шифра и прогнозирования уровня криптографической стойкости полной версии преобразований «Калина».

Ключевые слова: ключ, криптоалгоритм, уменьшенная версия шифра, схема разворачивания ключа.

ВВЕДЕНИЕ

В Украине последнее десятилетие в области криптографии стало особенным. Необходимость принятия нового стандарта блочного шифрования повлекло за собой проведение внутреннего конкурса по выдвижению кандидатов на национальный стандарт блочного симметричного шифрования (БСШ), где было представлено пять предложений [1–6]. Задача состояла не только в поиске убедительных теоретических обоснований принимаемых решений, но и в получении реальных практических результатов, что позволит накопить объективные данные для сравнительного анализа претендентов, сформулировать требования к проекту национального стандарта блочного симметричного шифрования на современном этапе развития криптографии [7].

Для преодоления трудностей анализа полномасштабных моделей (алгоритмов) шифрования в работах [8–10] было предложено пойти путем разработки и исследования уменьшенных моделей прототипов. Построение уменьшенных моделей, которые сохраняют все свои свойства, позволяет решить многие задачи анализа и сравнения по показателям стойкости соответствующих больших версий [11–13].

Задачей этой статьи является анализ одного из претендентов на национальный стандарт БСШ, а именно алгоритм «Калина» [14, 15], как дальнейшее развитие конкурсного предложения «Калина» [2], построение уменьшенной модели шифра посредством масштабирования основных криптопреобразований с сохранением их алгебраической структуры. Разрабатываемая модель предназначается для проведения исследований основных характеристик шифра и прогнозирования уровня криптографической стойкости полной версии преобразований «Калина» [14, 15].

1. БСШ «КАЛИНА»

Шифр «Калина» является дальнейшим развитием БСШ «Калина», который, в свою очередь, разрабатывался на основе известной структуры «квадрат» БСШ Rijndael [16].

В шифре «Калина» сохранены все базовые операции шифра «Rijndael», основные отличия состоят в использовании разных S-блоков (используются 4 различных S-блока), сгенерированных случайным образом, вместо одинаковых S-блоков, и в применении попеременного сложения с цикловыми подключами по модулю 2 и по модулю 2^{64}. Кроме того, линейное преобразование выполняется с помощью матрицы МДР кода размера 8×8 над полем $GF(2^8)$, что также потенциально улучшает криптографические свойства шифра. Параметры шифра «Калина» представлены в табл. 1.

Таблица 1

Характеристики шифра «Калина»

Key length / Block size	128 ($N_k = 2$)	256 ($N_k = 4$)	512 ($N_k = 8$)
128 ($N_b = 2$)	10	14	–
256 ($N_b = 4$)	–	14	18
512 ($N_b = 8$)	–	–	18

1.1 Алгоритм шифрования. Схема шифрования (см. рис. 1) БСШ «Калина» обладает следующими особенностями:

– процедура забеливания выполняется с помощью сложения с нулевым подключом по модулю 2^{64};

– в циклах от первого до N_r-1 цикловые подключи вводятся с помощью операции побитового сложения по модулю 2;

– на последнем цикле шифрования выполняется очередное цикловое преобразование SubBytes, ShiftRows, MixColumns и сложение с цикловым подключем по модулю 2^{64};

– в шифре «Калина» применяется четыре разных S-блока.

1.2 Схема разворачивания ключей. В шифре «Калина» предложена новая, более эффективная схема разворачивания ключей. При шифровании используется N_r+1 цикловых ключей $k_i (i=0,1..,N_r)$, каждый длиной $64\times N_b$ бит (размер циклового ключа совпадает с размером открытого текста/шифртекста и текущего состояния шифра). Алгоритм разворачивания ключа можно представить в виде последовательного выполнения трех этапов (процедур):

1) из ключа шифрования формируется промежуточный ключ K_t длиной, равной размеру блока ($64\times N_b$ бит), с применением трех циклов шифрования;

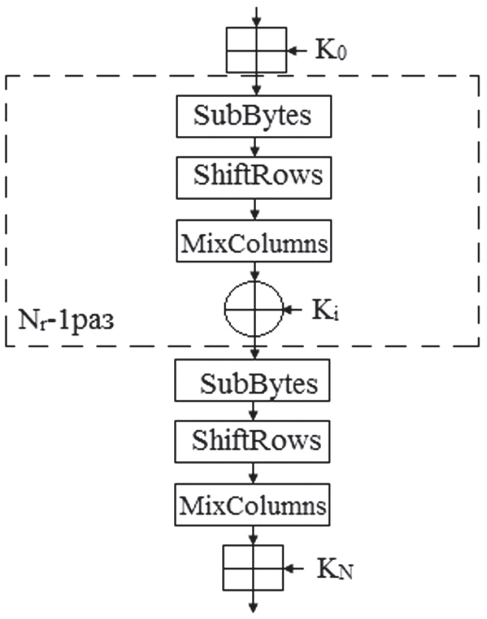

Рис. 1. Схема шифрования шифра «Калина»

2) на основе ключа шифрования K и промежуточного ключа K_t формируются цикловые ключи k_{2i} (с четными индексами) с длинами, равными размеру блока ($64\times N_b$ бит), с применением двух циклов шифрования для каждого циклового ключа;

3) из цикловых ключей k_{2i} с четными индексами формируются цикловые ключи k_{2i+1} (с нечетными индексами) путем циклического сдвига предыдущего ключа с четным индексом влево на $(2N_b + 3)$ байта.

Схема разворачивания ключей представлена на рис. 2.

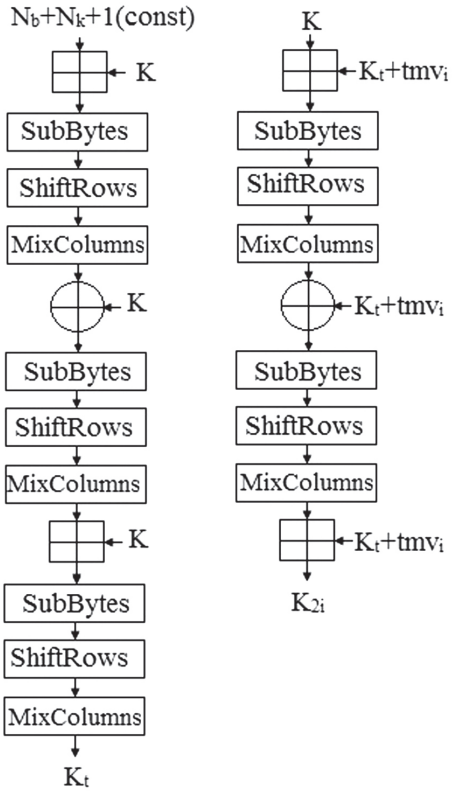

Рис. 2. Схема разворачивания ключей

1.2.1 Формирование промежуточного ключа K_t. Промежуточный ключ K_t имеет длину $64\times N_b$ бит (совпадает по размеру с длиной блока открытого текста) и формируется на основе ключа шифрования K длиной $64\times N_k$ бит с использованием преобразований, использованных при построении цикловых функций.

Для формирования K_t выполняются три цикла шифрования. На вход трёхциклового преобразования (с начальным забеливанием) подается двоичное число (в нулевое состояние заносится арифметическая сумма N_b+N_k+1, остальные байты заполняются нулями), определяемое текущим размером блока и длиной ключа, а в качестве цикловых ключей используется ключ шифрования (или его младшая и старшая половины, если ключ длиннее блока). Значение, полученное на выходе преобразования, является промежуточным ключом K_t.

1.2.2 Формирование цикловых ключей с четными индексами. При генерации цикловых ключей с четными индексами ключ шифрования K или его циклический сдвиг подаются на двухцикловое преобразование (с начальным забеливанием и операцией *Add64RoundKey* во втором цикле) как открытый текст (один или два блока, в зависимости от длины формируемого ключа). В качестве циклового ключа используется результат обработки промежуточного ключа K_t функцией *Add64RoundKey* (сложение K_t по модулю 2^{64} со вторым аргументом), где вторым аргументом является значение (двоичное число), зависящее от индекса формируемого циклового ключа (переменная tmv_i).

Размер переменной tmv_i равен длине блока. Константа формируется повторением байтов 0x01, 0x00 (в шестнадцатеричном представлении) до заполнения состояния.

При переходе к формированию очередного циклового ключа с чётными индексами производится модификация переменой tmv_i с помощью операции *ShiftLeft*, которая обрабатывает состояние шифра (значение переменой tmv_i) как последовательность 64-битных слов. Каждое слово состояния $(w_0, w_1, ..., w_{N_b-1})$ с каждым циклом преобразования логически сдвигается влево на 1 бит, т.е. $w_i = 2\ w_i (\mathrm{mod}\ 2^{64})$.

Результирующие цикловые ключи с чётными индексами формируются из состояний на выходе двухцикловых преобразований с помощью операции Rotate, которая обрабатывает состояние как последовательность 64-битных слов $(w_0, w_1, ..., w_{N_k-1})$, выполняя циклический сдвиг и возвращая состояние $(w_1, ..., w_{N_k-1}, w_0)$.

1.2.3 Формирование цикловых ключей с нечетными индексами. Каждый из цикловых ключей с четным индексом, полученный на предыдущем этапе процедуры генерации, представляется в виде байтовой строки. Эта строка циклически сдвигается влево на $2N_b + 3$ байта, и затем снова представляется в виде состояния, которое ис-

пользуется как цикловой ключ с нечетным индексом, т.е. $k_{2i+1} = k_{2i} <<< (2N_k + 3)$.

Таким образом, байтовая строка b_0, b_1,..., b_{8N_b-1} после преобразования приобретает вид b_{2N_b+3}, b_{2N_b+4}, ..., b_{8N_b-1}, b_0, b_1, ... , b_{2N_b+2}.

Зависимость константы, определяющей значение сдвига влево циклового ключа с четным индексом, от размера блока, приведена в табл. 2.

Таблица 2

Константа, определяющая значение сдвига влево циклового ключа с четным индексом, в зависимости от размера блока

Размер блока, бит (байт)	Сдвиг влево (байт)
128 (16)	7
256 (32)	11
512 (64)	19

2. РАЗРАБОТКА УМЕНЬШЕННОЙ МОДЕЛИ БСШ «КАЛИНА»

Алгоритм шифрования мини-Калина повторяет принципиальные решения, использованные при построении полной версии шифра «Калина» и практически является результатом масштабирования оригинальной разработки к размеру входного блока и ключа равному 16 битам (см. табл. 3, а) и 3, б)).

Таблица 3.а)

Представление 16-битного слова, $P=36=\{0010,0100,0000,0000\}_2$

P_0				P_1				P_2				P_3			
bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8	bit 9	bit 10	bit 11	bit 12	bit 13	bit 14	bit 15
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Таблица 3.б)

Представление 16-битного слова

P_0	P_2
P_1	P_3

Миниверсия «Калина» практически полностью повторяет решения по построению функций ShiftRows, MixColumn шифра miniAES [9, 10], слой нелинейного преобразования реализован с помощью двух S-блоков. Рассмотрим особенности построения отдельных слоев преобразования миниверсии БСШ «Калина».

2.1 Nibblesub. NibbleSub – это простая операция подстановки Nibble в Nibble, где Nibble – 4-битное слово. Блок представляется в виде матрицы 2×2 Nibble. Главное отличие разрабатываемой миниверсии от miniAES [9, 10] состоит в использовании двух таблиц (см. табл. 4) подстановок: первая таблица для $\{P_0,P_1\}$, вторая – для $\{P_2,P_3\}$.

На вход блока NibbleSub подается множество P ($P=\{P_0,P_1,P_2,P_3\}$), на выходе блока получаем множество A ($A=\{a_0,a_1,a_2,a_3\}$) (см. рис. 3).

Пример 1: на вход подано множество P={1101,0111,1010,0110}, на выходе получаем множество A={0010,1110,0111,1100}={2,14,7,12}.

Таблица 4

Sbox для NibbleSub

		P_0,P_1	P_2,P_3
0	0000	10	10
1	0001	13	15
2	0010	1	14
3	0011	3	9
4	0100	12	2
5	0101	4	1
6	0110	6	11
7	0111	14	5
8	1000	9	0
9	1001	5	12
10	1010	11	7
11	1011	0	3
12	1100	7	6
13	1101	2	4
14	1110	8	13
15	1111	15	8

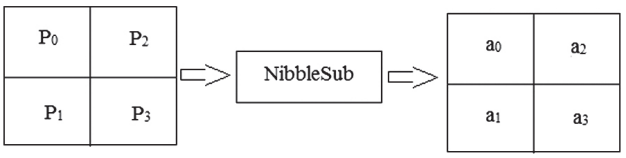

Рис. 3. Пример подстановки

2.2 ShiftRow. Блок ShiftRow меняет местами P_3 и P_1 входного блока, оставляя их без изменения. P_2 и P_0 остаются без изменения. Иллюстрацию примера можно посмотреть на рис. 4.

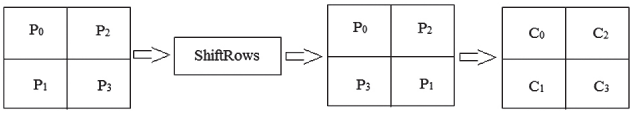

Рис. 4. Операция ShiftRow

Пример 2: на вход блока подано $P=\{P_0,P_1,P_2,P_3\}$ на выходе блока получим $C=\{C_0,C_1,C_2,C_3\}=\{P_0,P_3,P_2,P_1\}$.

2.3 MixColumn. Блок MixColumn – это перемножение входной матрицы на матрицу-константу, получая матрицу на выходе (рис. 5), где $P=\{P_0,P_1,P_2,P_3\}$ подается на вход, $C=\{C_0,C_1,C_2,C_3\}$ получается на выходе, соответственно. Умножение элементов происходит в поле $GF(2^4)$, которое построено на полиноме x^4+x+1.

Рис. 5. Операция MixColumn

Матрица-константа, определяющая преобразование MixColumn, имеет вид: $\begin{bmatrix} 10 & 7 \\ 7 & 15 \end{bmatrix}$.

Пример 3: Пусть $P=\{P_0,P_1,P_2,P_3\}$ входной блок и $C=\{C_0,C_1,C_2,C_3\}$ выходной блок. Если

входной блок P, матрица 2×2, тогда первый столбец умножается на матрицу-константу. На выходе получим столбец:

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 10 & 7 \\ 7 & 15 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \end{bmatrix},$$

где

$$C_0 = (1010 \otimes P_0) \oplus (0111 \otimes P_1)$$

$$\text{и } C_1 = (0111 \otimes P_0) \oplus (1111 \otimes P_1).$$

Следующим шагом является нахождение второго столбца C, для этого второй столбец умножается на константу-матрицу:

$$\begin{bmatrix} c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 10 & 7 \\ 7 & 15 \end{bmatrix} \begin{bmatrix} P_2 \\ P_3 \end{bmatrix},$$

где

$$C_2 = (1010 \otimes P_2) \oplus (0111 \otimes P_3)$$

$$\text{и } C_3 = (0111 \otimes P_2) \oplus (1111 \otimes P_3).$$

2.4 KeyAddition. В блоке KeyAddition входное сообщение складывается с ключом по модулю 2^8. 16-битное входное сообщение делится на два байта и побайтно складывается с 16-битным ключом, результат берется по модулю 2^8, затем результат конкатенируется.

Пример 4: Пусть $P = \{P_0, P_1, P_2, P_3\} = \{1111, 0011, 0011, 1011\}$ входное сообщение и $K = \{K_0, K_1, K_2, K_3\} = \{1010, 1101, 1111, 1001\}$ ключ. Представим входное сообщение и ключ в виде двух байтов:

P=(11110011,00111011) и K={10101101,11111001)

Тогда выходное сообщение C:

C=({11110011+10101101}‖{00111011+11111001})=
=(1010,0000,0011,0100).

2.5 XorRoundKey. В блоке XorRoundKey 16-битное входное сообщение складывается с 16-битным ключом, соответствующего раунда, по модулю 2. Полученное сообщение подается на выход.

Пример 5: Пусть $P = \{P_0, P_1, P_2, P_3\} = \{1111, 0011, 0011, 1011\}$ входное сообщение и $K = \{K_0, K_1, K_2, K_3\} = \{1010, 1101, 1111, 1001\}$ ключ. Тогда выходное сообщение C будет иметь вид:

$$C = P \oplus K = (0101, 1110, 1100, 0010).$$

2.6 Схема разворачивания ключей. При шифровании (см. рис. 1) используется $N_r + 1$ цикловых ключей k_i ($i = 0,1,..., N_r$), каждый длиной $8 \times N_b$ бит (размер циклового ключа совпадает с размером открытого текста/шифртекста и текущего состояния шифра).

Алгоритм разворачивания ключа можно представить в виде последовательного выполнения трёх этапов (трёх процедур):

1) из ключа шифрования формируется промежуточный ключ K_t длиной, равной размеру блока ($8 \times N_b$ бит), с применением трех циклов шифрования;

2) на основе ключа шифрования K и промежуточного ключа K_t формируются цикловые ключи k_{2i} (с четными индексами) с длинами, равными размеру блока ($8 \times N_b$ бит), с применением двух циклов шифрования для каждого циклового ключа;

3) из цикловых ключей k_{2i} с четными индексами формируются цикловые ключи k_{2i+1} (с нечетными индексами) путем циклического сдвига предыдущего ключа с четным индексом влево на $(2N_b + 3)$ бита.

2.6.1. Формирование промежуточного ключа K_t. Промежуточный ключ K_t имеет длину $8 \times N_b$ бит (совпадает по размеру с длиной блока открытого текста) и формируется на основе ключа шифрования K длиной $8 \times N_k$ бит с использованием преобразований, использованных при построении цикловых функций.

Для формирования K_t выполняются три цикла шифрования. На вход трехциклового преобразования (с начальным забеливанием) подается двоичное число, определяемое текущим размером блока и длиной ключа (арифметическая сумма $(N_b + N_k + 1)$, в остальные биты заполняются 0), а в качестве цикловых ключей используется ключ шифрования (или его младшая и старшая половины, если ключ длиннее блока). Значение, полученное на выходе преобразования, является промежуточным ключом K_t.

2.6.2. Формирование цикловых ключей с четными индексами. При генерации цикловых ключей с четными индексами ключ шифрования K или его циклический сдвиг подаются на двухцикловое преобразование (с начальным забеливанием и операцией *AddRoundKey* во втором цикле) как открытый текст (один или два блока, в зависимости от длины формируемого ключа). В качестве циклового ключа используется результат обработки промежуточного ключа K_t функцией *AddRoundKey* (сложение K_t по модулю 2^8 со вторым аргументом), где вторым аргументом является значение (двоичное число), зависящее от индекса формируемого циклового ключа (переменная tmv_i).

Размер переменной tmv_i равен длине блока. Константа формируется повторением битов {01}, {00} до заполнения состояния.

При переходе к формированию очередного циклового ключа с чётными индексами производится модификация переменой tmv_i с помощью операции *ShiftLeft*, слово состояния $(w_0, w_1, ..., w_{N_b-1})$ с каждым циклом преобразования логически сдвигается влево на 1 бит, т.е. $w_i = 2\, w_i \pmod{2^{16}}$.

2.6.3 Формирование цикловых ключей с нечетными индексами. Каждый из цикловых ключей с четным индексом, полученный на предыдущем этапе процедуры генерации, представляется в виде 16-битной строки. Эта строка циклически сдвигается влево на $(2N_b + 3)$ бита, и затем снова представляется в виде состояния, которое используется как цикловой ключ с нечетным индексом, т.е. $k_{2i+1} = k_{2i} <<< (2N_k + 3)$.

Для данного случая $N_b = 2$, следовательно, константа, определяющая значение сдвига влево циклового ключа с четным индексом, равна 7.

ВЫВОДЫ

Проведенный анализ структуры и основных базовых преобразований БСШ «Калина» показал, что данный криптоалгоритм имеет ярко выраженную алгебраическую структуру, которая унаследована от алгоритмов-прототипов «Калина» и «Rijndael» [2, 16]. В частности, алгоритм «Калина» имеет структуру «квадрат», при этом значение состояния шифра представляется матрицей из восьми строк и $N_b \in \{2, 4\ 8\}$ столбцов (в зависимости от размера блока (128, 256, 512 бит) шифрования). Основным отличием от БСШ «Rijndael» является увеличенный размер состояния (восемь строк вместо четырех) с использованием увеличенной матрицы рассеивания (на основе матрицы МДР кода размера 8x8 вместо 4x4 для БСШ «Rijndael»), а также использованием нескольких нелинейных узлов замен, сформированных случайным образом (вместо алгебраической структуры S-блоков шифра «Rijndael»). По сравнению с БСШ «Калина» новый алгоритм шифрования «Калина» отличается усовершенствованной схемой разворачивания ключей, а также новыми S-блоками с улучшенными криптографическими свойствами.

Разработанная уменьшенная модель шифра «Калина» сохраняет алгебраическую структуру алгоритма-прототипа. Посредством масштабирования основных криптографических преобразований удается моделировать базовые преобразования и исследовать их характеристики. В частности, переход к 16-битной версии шифра позволяет исследовать дифференциальные и линейные свойства шифрующего преобразования, производить соответствующую оценку стойкости к криптоаналитическим атакам, посредством переноса и интерпретации полученных результатов на полную версию алгоритма «Калина» прогнозировать поведение БСШ в случае применения различных методов криптоанализа, т.е. давать оценку уровня криптографической стойкости полной версии БСШ «Калина». В приложении А представлены примеры расчетов с различными входными данными, согласно разработанной миниверсии «Калина».

Перспективным направлением дальнейших исследований является разработка программной реализации мини-версии БСШ «Калина», проверка адекватности уменьшенной модели, исследование дифференциальных и линейных свойств криптоалгоритма с прогнозированием уровня стойкости к криптоаналитическим атакам.

Литература

[1] Положення про проведення відкритого конкурсу криптографічних алгоритмів. http://dstszi.gov.ua/dstszi/control/uk/publish/, 2006.

[2] Горбенко І.Д. Перспективний блоковий симетричний шифр «Калина» / І.Д. Горбенко, В.І. Долгов, Р.В. Олійников та ін. // Прикладна радіоелектроніка. – 2007. – Т.6. – № 2. – С. 195–208.

[3] Головашич С.А. Спецификация алгоритма блочного симметричного шифрования «Лабиринт» // Прикладная радиоэлектроника. – Харьков: ХТУРЭ. – 2007. – Том. 6, №2. – С. 230–240.

[4] Горбенко І.Д. Перспективний блоковий симетричний шифр «Мухомор» – основні положення та специфікація / І.Д. Горбенко, М.Ф. Бондаренко, В.І. Долгов та ін. // Прикладная радиоэлектроника – Харьков: ХТУРЭ. – 2007. – Том. 6, №2. – С. 147–157.

[5] Кузнецов А.А. Симметричный криптографический алгоритм ADE (Algorithm of Dynamic Encryption). / А.А. Кузнецов, Р.В. Сергиенко, А.А. Наумко // Прикладная радиоэлектроника. – Харьков: ХТУРЭ. – 2007. – Том 6, №2 – С. 241–249.

[6] Белецкий А.Я. Семейство симметричных блочних RSB криптографических алгоритмов с динамически управляемыми параметрами шифрования / А.Я. Белецкий, А.А. Белецкий, А.А. Кузнецов // Електроніка та системи управління. – 2007. – № 1 (11). – С. 5–16.

[7] Горбенко И.Д. Стандартизация алгоритмов шифрования. Требования к проекту национального стандарта блочного симметричного шифрования на современном этапе развития криптографии. / И.В. Горбенко, И.В. Лисицкая // Радиотехника. Всеукр. межвед. научн.-техн. сб. – 2011. – Вып. 166. – С. 5–10.

[8] A Description of Baby Rijndael // ISU CprE/Math 533; NTU ST765-U. – 2003.

[9] Raphael Chung-Wei Phan. Mini Advanced Encryption Standard (Mini-AES): A testbed for Cryptanalysis Students / Raphael Chung-Wei Phan // Cryptologia. – October 2002. – XXVI(4). – P. 283–306.

[10] Raphael Chung-Wei Phan. Impossible Differential Cryptanalysis of Mini-AES / Raphael Chung-Wei Phan // Cryptologia. – October 2003. – XXVII(4). – P. 361–374.

[11] Долгов В.И. Дифференциальные свойства блочных симметричных шифров, представленных на украинский конкурс. / В.И. Долгов, А.А. Кузнецов, С.А. Исаев. // Электронное моделирование. – 2011. – Т. 33, № 6. – С. 81–99.

[12] Долгов В.И. Исследование дифференциальных свойств мини-шифров Baby-ADE и Baby-AES / В.И. Долгов, А.А. Кузнецов, Р.В. Сергиенко, О.И. Олешко // Прикладная радиоэлектроника. – Харьков: ХНУРЭ. – 2009. – Т. 8, № 3. – С. 252–257.

[13] Кузнецов А.А. Линейные свойства блочных симметричных шифров, представленных на украинский конкурс. / А.А. Кузнецов, И.В. Лисицкая, С.А. Исаев // Прикладная радиоэлектроника. – 2011. – Т. 10, № 2 – С. 135–140.

[14] Проект стандарту ДСТУ "Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення", друга (остаточна) редакція, Харків: АТ «ІІТ», 238 с.

[15] Розробка нового блокового симетричного шифру: звіт за перший етап НДР «Алгоритм» (проміжний) / АТ «ІІТ»; кер. Горбенко І.Д.; викон.: Олейніков Р.В. [та ін.]. – Харків, 2014. – Т. 4. – 304 с.

[16] J. Daemen and V. Ridjmen. The Design of Rijndael: AES — the Advanced Encryption Standard, Springer-Verlag, Berlin, 2002.

Поступила в редколлегию 18.06.2014

ПРИЛОЖЕНИЕ А

Таблица А.1

Расчеты для схемы разворачивания ключей для четырехраундовой мини-версии «Калина» для разных ключей, при const=5 и Tmv_0=8738

	Key=5	Key=10	key=6405	key=58723
1	2	3	4	5
Add(const,key)	0101000000000000	1111000000000000	0101000010011000	0001011010100111
NibbleBox	0100101010101010	1111101010101010	0100101011000000	1101011001110101
ShiftRow	0100101010101010	1111101010101010	0100000011001010	1101010101110110
MixColumn	1101001110111111	1111011110111111	1110111100101110	0011001100100010
Xor(state,key)	0111001110111111	1010011110111111	0100111110110110	1111010110000101
NibbleBox	1110001100111000	1011111000111000	1100111100111011	1111010000000001
ShiftRow	1110100000110011	1011100000111110	1100101100111111	1111000100000100
MixColumn	1011110101001011	1111010100011100	0101000101100011	1011010011111001
Add(state,key)	0100001101001011	1001110100011100	1111000111111011	0000100100100001
NibbleBox	1100001100100011	0101001011110110	1111110110000011	1010010111101111
ShiftRow	1100001100100011	0101011011110010	1111001110001101	1010111111100101
MixColumn	1000000011101100	0101110000100110	0101100110101010	0011100111101010
KS	1000000011101100	0101110000100110	0101100110101010	0011100111101010
1ая пара ключей	1round	1round	1round	1round
Add(KS,Tmv_0)	1100010010011010	0011101001100001	0011110111101110	0111110110011110
Add(KS_Tmv_0,key)	0001010010011010	0110011001100001	1000001100001001	1000010001111010
NibbleBox	1101110011000111	0110011010111111	1001001110101100	1001110001010111
ShiftRow	1101011111001100	0110111110110110	1001110010100011	1001011101011100
MixColumn	1101111000111010	0010101100110000	0111001000010001	0011000101100000
	2round	2round	2round	2round
Xor(state,KS_Tmv_0)	0001101010100000	0001000101010001	0100111111111111	0100110011111110
NibbleBox	1101101101111010	1101110100011111	1100111110001000	1100011110001101
ShiftRow	1101101001111011	1101111100011101	1100100010001111	1100110110000111
MixColumn	1000100101110101	0000111111110000	1100001101000111	0100010110010110
Key0=				
Add(state,KS_Tmv_0)	0010110111100000	0011001010101001	1111111010011010	0000011001000111
Key1=Key0<<<7	1111000000010110	0101010010011001	0100110101111111	0010001110000011
Tmv_1=Tmv_0<<1	1000100010001000	1000100010001000	1000100010001000	1000100010001000
2ая пара ключей	1round	1round	1round	1round
Add(KS,Tmv_1)	0100100000010010	1101001010101110	1101010101100110	1011010100010110
Add(KS_Tmv_1,key)	1110100000010010	1010101010101110	0000110111111110	0000100010110010
NibbleBox	1000100111111110	1011101101111101	1010001010001101	1010100100111110
ShiftRow	1000111011111001	1011110101111011	1010110110000010	1010111000111001
MixColumn	0011100001100101	0111001101110101	1101010000010000	0100011001110111
	2round	2round	2round	2round
Xor(state,KS_Tmv_1)	0111000001110111	1010000111011011	0000000101110110	1111001101100001
NibbleBox	1110101001010101	1011110101000011	1010110101011011	1111001110111111
ShiftRow	1110010101011010	1011001101001101	1010101101011101	1111111110110011
MixColumn	1110101001110100	1011011010111000	1100000000011111	0111000110110110
Key2=				
Add(state,KS_Tmv_1)	1001011001101110	0001110101001001	0111010101111010	1101110010101011
Key3= Key2<<<7	0011011101001011	1010010010001110	1011110100111010	0101010111101110
Tmv_2=Tmv_1<<1	0001000100010000	0001000100010000	0001000100010000	0001000100010000
3ая пара ключей	1round	1round	1round	1round
Add(KS,Tmv_2)	1001000111111100	0100001100110110	0100010010111010	0010010011111010
Add(KS_Tmv_2,key)	0111000111111100	0011001100110110	1110010001101110	1110000100100010
NibbleBox	1110110110000110	0011001110011011	1000110010111101	1000110111101110
ShiftRow	1110011010001101	0011101110010011	1000110110111100	1000111011101101
MixColumn	0111100010101010	1001101011001000	1010101000001100	0011100000111011
	2round	2round	2round	2round
Xor(state,KS_Tmv_1)	1110100101010110	1101100111111110	1110111010110110	0001110011000001
NibbleBox	1000010100011011	0010010110001101	1000100000111011	1101011101101111
ShiftRow	1000101100010101	0010110110000101	1000101100111000	1101111101100111
MixColumn	1011111000100001	0010100101111011	1011111000001000	0000111111111010
Key4=				
Add(state,KS_Tmv_2)	0110000011000011	0110101001010010	1111100110110110	0010100001111101
Key5= Key4<<<7				
	0110000110110000	0010100100110101	1101101101111100	0011111010010100

Таблица А.1

Шифрование четырехраундовой мини-версии «Калина» разных входных сообщений, при const=5, Tmv_0=8738 и key=5

	plaintext=10241	plaintext=15966	plaintext=56343	plaintext=2204
	round 0	round 0	round 0	round 0
Add(plaintext,key0)	1010110111110100	0100100010100010	1101001111000111	0000101011110000
	round 1	round 1	round 1	round 1
NibbleBox	1011001010000010	1100100101111110	0010001101100101	1010101110001010
ShiftRow	1011001010000010	1100111001111001	0010010101100011	1010101010001011
MixColumn	1100100100010000	1101011110011000	1111100000000011	1011111110111110
Xor(state,roundKey1)	0011100100000110	0010011110001110	0000100000010101	0100111110101000
	round 2	round 2	round 2	round 2
NibbleBox	0011010110101011	0001111000001101	1010100111110001	1100111101110000
ShiftRow	0011101110100101	0001110100001110	1010000111111001	1100000001111111
MixColumn	1001101000000101	1111000011000101	1111110001100101	0001001010001100
Xor(state,roundKey2)	0000110001101011	0110011010101011	0110101000001011	1000010011100010
	round 3	round 3	round 3	round 3
NibbleBox	1010011110110011	0110011001110011	0110101110100011	1001110011011110
ShiftRow	1010001110110111	0110001101110110	0110001110101011	1001111011011100
MixColumn	0001000101001111	0000001100100010	0000001111000000	1001111110011101
Xor(state,roundKey3)	0010011000000100	0011010001101001	0011010010001011	1010100011010110
	round 4	round 4	round 4	round 4
NibbleBox	0001011010100010	0011110010111100	0011110000000011	1011100101001011
ShiftRow	0001001010100110	0011110010111100	0011001100001100	1011101101001001
MixColumn	0100101010010111	1111000100001100	0100101100101000	0110011101000001
Add(state,roundKey4)	0001101000110101	1010100111001111	0001101111101011	0011011110100010

Кузнецов Александр Александрович, доктор технических наук, профессор кафедры БИСТ ХНУ им. В.Н. Каразина. Научные интересы: криптография, теория обработки и передачи данных, стеганографические методы защиты информации.

Иваненко Дмитрий Викторович, кандидат технических наук, старший преподаватель кафедры БИТ ХНУРЭ. Научные интересы: криптография, атаки по сторонним каналам, анализ стойкости блочных симметричных шифров.

Колованова Евгения Павловна, ведущий инженер кафедры БИТ ХНУРЭ. Научные интересы: криптография и аутентификация, анализ стойкости блочных симметричных шифров.

УДК 004.056.55

Моделювання перспективного блочного шифру «Калина»/ О.О. Кузнецов, Д.В. Іваненко, Є.П. Колованова // Прикладна радіоелектроніка: наук.-техн. журнал. — 2014. — Том 13. — № 3. — С. 201–207.

Розглядається структура та загальні базові перетворення перспективного блокового шифру «Калина». Розробляється зменшена модель шифру шляхом масштабування загальних криптоперетворень зі збереженням їх алгебраїчної структури. Розроблена модель призначена для проведення дослідження загальних характеристик шифру та прогнозування рівня криптографічної стійкості повної версії перетворень «Калина».

Ключові слова: ключ, криптоалгоритм, зменшена версія шифру, схема розгортання.

Табл.: 5. Іл.: 5. Бібліогр.: 16 найм.

UDC 004.056.55

Perspective block cipher «Kalyna» modelling / A.A. Kuznetsov, D.V. Ivanenko, E.P. Kolovanova // Applied Radio Electronics: Sci. Journ. — 2014. — Vol. 13. — № 3. — P. 201–207.

The structure and common basic transformations of the perspective block cipher «Kalyna» are considered in this paper. A cipher mini-model is developed by scaling common cipher cryptotransformations with preservation of their algebraic structure. The developed mini-model is designed for studying the general cipher characteristics and forecasting the level of cryptographic strength for the full version of «Kalyna» transformations.

Keywords: key, cryptoalgorithm, minicipher, key schedule.

Tab.: 5. Fig.: 5. Ref.: 16 items.