УДК 681.3.07

КРИПТОГРАФИЧЕСКИЕ ПРИМИТИВЫ, ОСНОВАННЫЕ НА МЕТОДЕ СКОЛЬЗЯЩЕГО КОДИРОВАНИЯ

А.Я. Белецкий, проф.; А.А. Белецкий, мл. науч. сотрудник
Национальный авиационный университет, г. Киев

Предлагаются оригинальные криптографические примитивы, в основу которых положен так называемый метод «преобразований Грея наоборот». Рассматриваются варианты однородных и смешанных арифметико-логических операций над многоразрядными двоичными кодовыми комбинациями шифруемого текста.

ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ

Современные методы защиты информации в компьютерных сетях (шифрование) представляют собой математические преобразования (алгоритмы), в которых сообщения рассматриваются как числа или алгебраические элементы [1]. Криптографические алгоритмы отображают область «осмысленных сообщений» (входной или открытый текст) в область «бессмысленных сообщений» (выходной или шифротекст, шифрограмма). С позиций теории сигналов и процессов зашифрование исходного (коррелированного, избыточного, сжимаемого) текста состоит в его «отбеливании», т.е. обращении в некоррелированную последовательность символов (элементов) шифрограммы (практически несжимаемой) с плотностью распределения элементов выходного алфавита, максимально близкой к равномерной.

Криптостойкие системы могут быть построены путем многократного применения относительно простых криптографических преобразований (примитивов), в качестве которых К.Шенон предложил использовать подстановки (substitution) и перестановки (permutation). Схемы, реализующие эти преобразования, называются *SP-сетями*. Часто используемыми криптографическими примитивами являются также преобразования типа циклический сдвиг (круговая прокрутка блоков), гаммирование и ряд других.

В данной работе вводится новый подкласс обратимых примитивов, названных авторами операциями *скользящего кодирования*. Схема построения криптопримитивов типа «скользящего кодирования» подобна схемам лево- и правостороннего преобразований Грея [2] «наоборот» в том смысле, что прямому скользящему кодированию отвечают схемы обратных преобразований Грея, а обратному кодированию – схемы прямых преобразований Грея. Криптопримитивы скользящего кодирования можно строить на основе арифметических, логических или смешанных (арифметико-логических) операций преобразования элементов (совокупности n-битных комбинаций) шифруемого текста.

Предлагаемые алгоритмы скользящего кодирования удачно развивают ***RSB***-технологию построения блочных симметричных криптографических систем защиты информации, первое описание которой приведено в [3]. Аббревиатура ***RSB*** происходит от ключевых слов ***R****ound,* ***S****tep,* ***B****lock* – подчеркивая тем самым, что основными для криптоалгоритма являются раундовые преобразования (***R***), разбитые на определенное число шагов (***S***), а действие алгоритма осуществляется над блоками (***B***) открытого или закрытого текста.

Перед изложением основного материала кратко поясним алгоритм выработки раундовых ключей в ***RSB*** -шифрах, схема которого приведена на рис. 1.

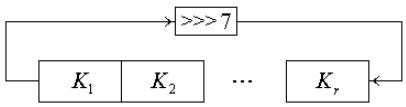

Рисунок 1

В соответствии с рис. 1 общий ключ ***CK*** (***C****ommon* ***K****ey*) ***RSB*** -шифра образуется конкатенацией r 32–битных раундовых ключей K_i ($i = \overline{1, r}$). ***RSB*** -технология предполагает, что криптопреобразование текста осуществляется за $s \geq 1$ шагов, и на каждом шаге шифрования осуществляется частичное обновление (выработка) раундовых ключей за счет круговой прокрутки (влево или вправо в зависимости от режима шифрования) общего ключа.

АРИФМЕТИЧЕСКОЕ СКОЛЬЗЯЩЕЕ КОДИРОВАНИЕ

Выберем 32-битный размер элементов шифруемого текста, совпадающий с размером раундового ключа в ***RSB*** - алгоритмах. Тем самым мы ориентируемся на реализацию алгоритма шифрования на компьютерных платформах с 32-разрядными шинами. С равным успехом можно было принять, например, 64-битным размер элементов и шин, что не влечет за собой каких-либо принципиальных затруднений. Будем также полагать, что шифруемый текст разбит на блоки, каждый из которых содержит четное число 32-битных элементов. Последнее условие, в частности, означает, что размер блока кратен 64 битам, т.е. может быть ориентирован на обработку процессорами с 64-разрядной шиной, к которой склоняется в последнее время международная практика.

По аналогии с лево- и правосторонним преобразованием Грея [4] введем лево- и правостороннее скользящее кодирование с арифметическими преобразованиями элементов шифрования. Для простоты будем называть их *арифметическим скользящим кодированием* (АСК). Структурная схема четырехэлементного прямого левостороннего АСК приведена на рис. 2.

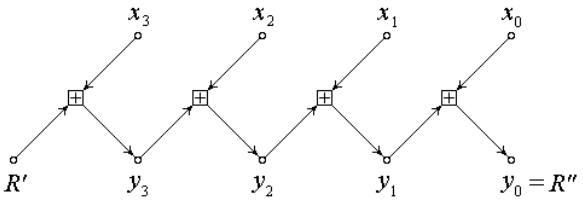

Рисунок 2

На данном рисунке приняты такие обозначения: x_i (y_i), $i = \overline{0, 3}$, – входные (выходные) 32-битные операнды преобразования; $\boxed{+}$ – оператор арифметического суммирования по $\bmod 2^{32}$; R' – 32-битный входной раундовый ключ; R'' – 32-битный выходной раундовый ключ, используемый в качестве входного для последующего преобразуемого блока.

Алгоритму прямого левостороннего АСК (т.е. развивающегося по направлению преобразования слева направо) отвечает система линейных модульных алгебраических уравнений:

$$\begin{aligned} y_3 &= (x_3 + R') \bmod m ; \\ y_2 &= (x_2 + y_3) \bmod m ; \\ y_1 &= (x_1 + y_2) \bmod m ; \\ y_0 &= (x_0 + y_1) \bmod m , \end{aligned} \qquad (1)$$

где $m = 2^{32}$.

Решая формально систему уравнений (1) относительно входных операндов x_i, получим систему:

$$\begin{aligned} x_3 &= (y_3 - R') \bmod m ; \\ x_2 &= (y_2 - y_3) \bmod m ; \\ x_1 &= (y_1 - y_2) \bmod m ; \\ x_0 &= (y_0 - y_1) \bmod m , \end{aligned} \qquad (2)$$

которой отвечает (рис. 3) структурная схема четырехэлементного обратного левостороннего АСК.

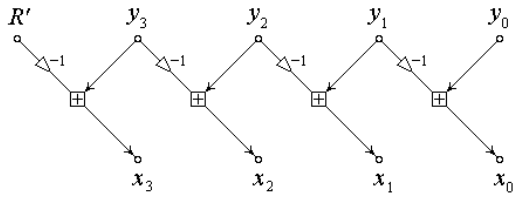

Рисунок 3

RSB -технология строится таким образом, что на нечетных шагах шифрования используется процедура левостороннего скользящего кодирования (в соответствии с рис. 2 – на этапах зашифрования и рис. 3 – на этапах расшифрования), тогда как на четных шагах шифрования привлекаются алгоритмы правостороннего АСК. Структурная схема прямого правостороннего АСК для четырехэлементного блока показана на рис. 4.

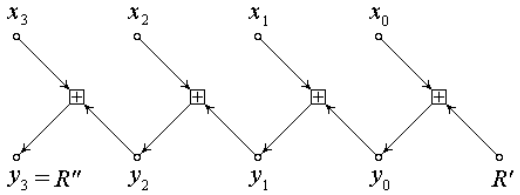

Рисунок 4

Алгоритм прямого правостороннего АСК (рис. 4) можно записать в виде следующей системы линейных модульных преобразований:

$$\begin{aligned} y_0 &= (x_0 + R') \bmod m ; \\ y_1 &= (x_1 + y_0) \bmod m ; \\ y_2 &= (x_2 + y_1) \bmod m ; \\ y_3 &= (x_3 + y_2) \bmod m , \end{aligned} \qquad (3)$$

решая которую относительно входных операндов приходим к системе модульных уравнений, отвечающей алгоритму правостороннего обратного АСК:

$$\begin{aligned} x_0 &= (y_0 - R') \bmod m ; \\ x_1 &= (y_1 - y_0) \bmod m ; \\ x_2 &= (y_2 - y_1) \bmod m ; \\ x_3 &= (y_3 - y_2) \bmod m . \end{aligned} \qquad (4)$$

Структурная схема обратного правостороннего АСК показана на рис. 5.

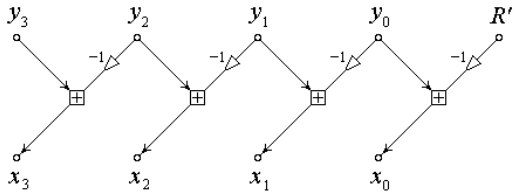

Рисунок 5

Из сопоставления структурных схем левостороннего (рис. 2 и 3) и правостороннего (рис. 4 и 5) АСК следует, что к схемам правосторонних АСК мы приходим в результате поворота на 180° относительно центральной вертикальной оси схем левосторонних АСК при сохранении неизменными позиций операндов преобразования $\boldsymbol{x}$ и $\boldsymbol{y}$.

ЛОГИЧЕСКОЕ СКОЛЬЗЯЩЕЕ КОДИРОВАНИЕ

К такому виду кодирования (ЛСК) мы приходим в результате логических преобразований элементов блоков, в качестве которых выбрана операция поразрядного суммирования по $\bmod 2$, эквивалентная логической операции XOR. Структурные схемы ЛСК легко получим заменой оператора арифметического сложения по модулю $m = 2^{32}$, обозначаемого как $\boxed{+}$, на оператор $\oplus$ поразрядного сложения по $\bmod 2$.

В результате указанных замен на рис. 2 приходим к такому виду структурной схемы прямого левостороннего ЛСК (рис. 6).

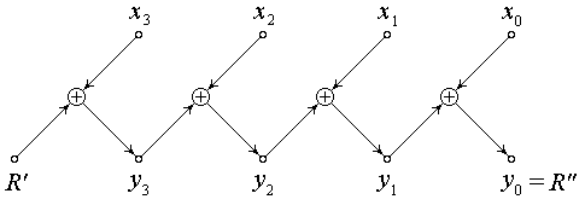

Рисунок 6

Схеме преобразования, приведенной на рис. 6, отвечает система линейных модульных уравнений:

$$\begin{aligned} y_3 &= x_3 \oplus R' ; \\ y_2 &= x_2 \oplus y_3 ; \\ y_1 &= x_1 \oplus y_2 ; \\ y_0 &= x_0 \oplus y_1 , \end{aligned} \qquad (5)$$

причем не следует забывать, что в данном случае операции выполняются поразрядно над каждой парой операндов.

Примем во внимание, что в двоичной арифметике

$$a \oplus b \equiv a \diamond b \ , \qquad (6)$$

где a и b – двоичные одноразрядные операнды (биты), а $\diamond$ – оператор поразрядного вычитания по $\mathrm{mod}\,2$. Тождество (6) дает возможность исключить в схемах обратного ЛСК (рис. 3 и 5) множительные элементы $-\triangleright-$ с коэффициентом передачи -1, что дает возможность, например, следующим образом отобразить структурную схему обратного левостороннего ЛСК (рис. 7).

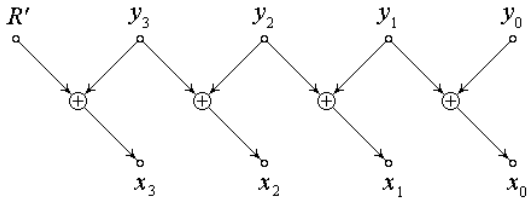

Рисунок 7

Система линейных алгебраических уравнений, соответствующая схеме преобразований, представленной на рис. 7, имеет вид:

$$\begin{aligned} x_3 &= y_3 \oplus R' \ ; \\ x_2 &= y_2 \oplus y_3 \ ; \\ x_1 &= y_1 \oplus y_2 \ ; \\ x_0 &= y_0 \oplus y_1 \ . \end{aligned} \qquad (7)$$

Аналогичным образом легко могут быть построены структурные схемы и системы линейных модульных алгебраических уравнений для прямого и обратного правостороннего ЛСК. Обратим внимание на то, что предлагаемые схемы лево- и правостороннего логического скользящего кодирования и отвечающие им системы уравнений являются ничем иным, как соответствующие (по направлениям) *обратные преобразования Грея* над двоичными кодовыми комбинациями, тогда как обратные ЛСК представляют собой *прямые преобразования Грея*. Тем самым мы вправе определить ЛСК как преобразования Грея «наоборот».

СМЕШАННОЕ СКОЛЬЗЯЩЕЕ КОДИРОВАНИЕ

Данный тип кодирования (ССК) предполагает, что криптографический примитив содержит чередующиеся операции логического и арифметического преобразований так, как это для четырехэлементных блоков шифруемого текста показано на рис. 8 для левостороннего смешанного зашифрования.

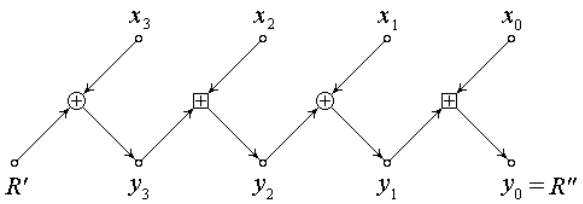

Рисунок 8

Система линейных модульных уравнений, отвечающая прямому левостороннему ССК (рис. 8), имеет вид:

$$\begin{aligned} y_3 &= x_3 \oplus R' ; \\ y_2 &= (x_2 + y_3) \bmod m ; \\ y_1 &= x_1 \oplus y_2 ; \\ y_0 &= (x_0 + y_1) \bmod m . \end{aligned} \qquad (8)$$

Решая систему уравнений (8) относительно операндов x_i, получим

$$\begin{aligned} x_3 &= y_3 \oplus R' ; \\ x_2 &= (y_2 - y_3) \bmod m ; \\ x_1 &= y_1 \oplus y_2 ; \\ x_0 &= (y_0 - y_1) \bmod m . \end{aligned} \qquad (9)$$

Структурная схема обратного левостороннего ССК, соответствующая системе линейных модульных уравнений (9), представлена на рис. 9.

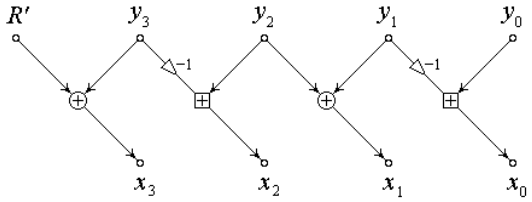

Рисунок 9

Аналогичным образом могут быть построены структурные схемы и составлены системы линейных модульных уравнений для правосторонних ССК.

АЛГОРИТМЫ ШИФРОВАНИЯ ДАННЫХ ПРИМИТИВАМИ СКОЛЬЗЯЩЕГО КОДИРОВАНИЯ

В рамках данного раздела статьи с целью упрощения как записи алгебраических преобразований, так и структурных схем шифрования мы ограничимся минимальными размерами текста и минимальными значениями параметров шифрования.

Будем считать, что открытый текст состоит из двух блоков, каждый из которых содержит по два 32-битных элемента. Обозначим через x_3, x_2, x_1 и x_0 элементы входного текста, причем x_3 и x_2 относятся к первому (расположенному слева), а x_1 и x_0 относятся ко второму блоку. Выберем схему смешанного скользящего кодирования, полагая, что на этапе зашифрования первым выполняется логическое скользящее кодирование (СК), а вторым – арифметическое СК как для левостороннего, так и правостороннего преобразования. И, наконец, будем считать, что общий ключ образуется конкатенацией двух 32-битных раундовых ключей R_1 и R_2, а процесс зашифрования, как и расшифрования, завершается за два шага.

Структурная схема алгоритма зашифрования на первом шаге преобразования (напомним, что на нечетных шагах применяется прямое левостороннее скользящее кодирование) показана на рис. 10.

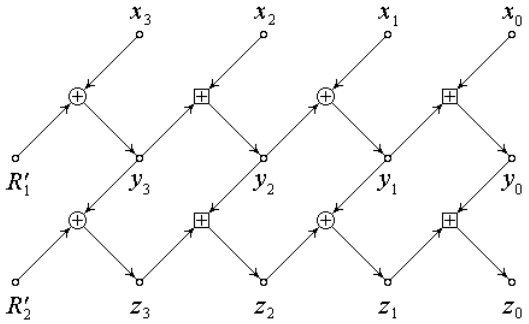

Рисунок 10

Согласно рис. 10 первый шаг зашифрования начинается преобразованием (прямым левосторонним ССК) операндов x под управлением раундового ключа $R_1' = R_1$. Результатом преобразования является четырехэлементный вектор y. Завершается первый шаг зашифрования преобразованием (также прямым левосторонним ССК), но теперь уже операндов y под управлением раундового ключа $R_2' = R_2$. Финальным результатом первого шага зашифрования является вектор z.

Перед вторым шагом зашифрования раундовые ключи модифицируются по схеме, показанной на рис. 11.

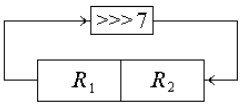

Рисунок 11

Обозначим через R_1'' и R_2'' раундовые ключи, образованные циклическим сдвигом общего ключа (конкатенации R_1 и R_2) на семь разрядов влево, т.е.

$$R_1 \circ R_2 <<< 7 \Rightarrow R_1'' \circ R_2'' \ , \qquad (10)$$

где $\circ$ – знак конкатенации.

На втором (четном) шаге зашифрования продолжаются преобразования (но теперь уже в режиме прямого правостороннего ССК) компонент вектора z сначала под управлением раундового ключа R_1'', а затем ключа R_2''. Структурная схема алгоритма зашифрования на втором шаге представлена на рис. 12.

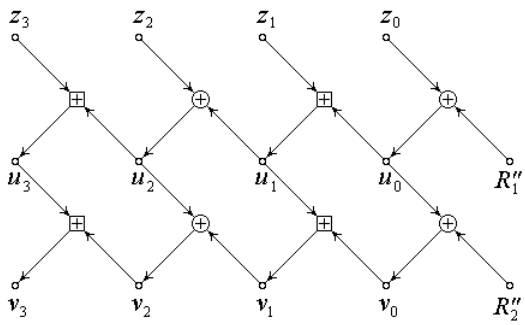

Рисунок 12

Таким образом, в результате зашифрования входного вектора $\boldsymbol{x} = \{x_3, x_2, x_1, x_0\}$ за два шага преобразования по схеме ССК под управлением двух раундовых ключей R_1 и R_2 сформирован шифротекст – четырехэлементный вектор $\boldsymbol{v} = \{v_3, v_2, v_1, v_0\}$.

Перейдем к конструированию алгоритма (структурной схемы) расшифрования криптограммы с целью восстановления исходного вектора $\boldsymbol{x}$. Процесс расшифрования является обратным зашифрованию. Вспомним, что для рассматриваемого примера зашифрование мы закончили на четном шаге (при этом привлекалась схема прямого правостороннего СК) под управлением раундового ключа R_2''. Поэтому процесс расшифрования мы должны начинать с преобразования типа обратного правостороннего ССК также под управлением ключа R_2''. Раундовый ключ R_2'' может быть сформирован по схеме, показанной на рис. 11, которой отвечает соотношение (10); т.е. общий ключ, образуемый конкатенацией R_1 и R_2, следует подвергнуть циклическому сдвигу на семь разрядов влево и взять правую половину модифицированного общего ключа.

Воспользовавшись обозначениями, принятыми на рис. 12, запишем систему линейных модульных уравнений формирования элементов вектора $\boldsymbol{v}$ из элементов вектора $\boldsymbol{u}$. Имеем:

$$\begin{aligned} v_0 &= u_0 \oplus R_2'' ; \\ v_1 &= (u_1 + v_0) \bmod m ; \\ v_2 &= u_2 \oplus v_1 ; \\ v_3 &= (u_3 + v_2) \bmod m . \end{aligned} \qquad (11)$$

Система (11) приводит к такому алгоритму восстановления компонент вектора $\boldsymbol{u}$ по заданным компонентам вектора $\boldsymbol{v}$:

$$\begin{aligned} u_0 &= v_0 \oplus R_2'' ; \\ u_1 &= (v_1 - v_0) \bmod m ; \\ u_2 &= v_2 \oplus v_1 ; \\ u_3 &= (v_3 - v_2) \bmod m . \end{aligned} \qquad (12)$$

Проверим корректность обратных преобразований (12). Для начала подставим в правую часть первого уравнения системы (12) вместо переменной v_0 ее значение из первого уравнения системы (11). Имеем

$$u_0 = (v_0) \oplus R_2'' = (u_0 \oplus R_2'') \oplus R_2'' \equiv u_0 \ .$$

Далее подставим вместо переменной v_1 во втором уравнении системы (12) ее значение из второго уравнения системы (1). Получим

$$u_1 = ((v_1) - v_0) \bmod m = ((u_1 + v_0) - v_0) \bmod m \equiv u_1 \ .$$

Аналогичным образом подтверждаются тождества для переменных u_2 и u_3. Впрочем, группа обратимых преобразований (11) и (12) и последующие доказательства тождественности являются тривиальными и приведены здесь с единственной целью - подтвердить единственность и корректность операций расшифрования криптограммы, образованной зашифрованием открытого текста с помощью простейших примитивов типа смешанного скользящего кодирования.

Система модульных уравнений (12), будучи продолжена на восстановление вектора z, приводит к следующей схеме (рис. 13) алгоритма первого шага расшифрования (напомним, что он является преобразованием, обратным зашифрованию на четном шаге, т.е. обратным правосторонним ССК).

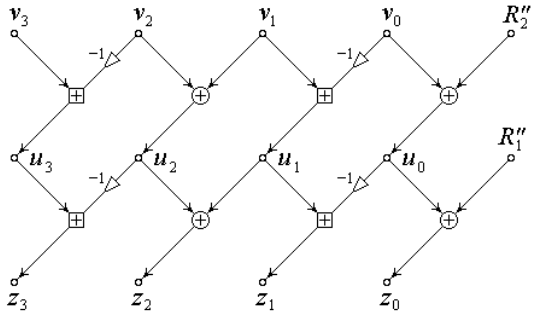

Рисунок 13

Перед выполнением второго шага расшифрования нам необходимо восстановить состояние раундовых ключей R'_1 и R'_2, под управлением которых проводилось зашифрование открытого текста x на первом шаге (т.е. с использованием прямого левостороннего ССК). С этой целью достаточно подвергнуть конкатенацию ключей K''_1 и K''_2 циклическому сдвигу на семь разрядов, но теперь уже вправо, т.е.

$$R''_1 \circ R''_2 >>> 7 \Rightarrow R'_1 \circ R'_2 = R_1 \circ R_2 \ .$$

Поскольку на первом шаге зашифрования последним раундовым ключом, под управлением которого выполнялись криптопреобразования, являлся ключ $R'_2 = R_2$, то с этого ключа и следует начинать процедуру расшифрования на втором шаге. Приходим к такой схеме финального шага алгоритма расшифрования (рис. 14).

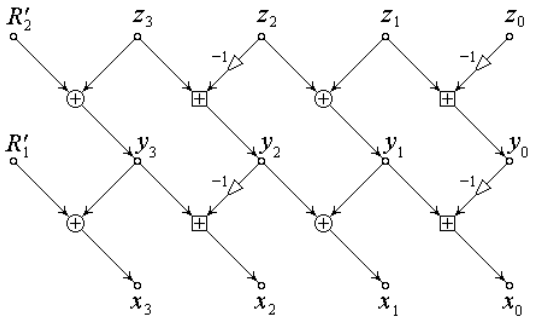

Рисунок 14

Опираясь на изложенную выше методику, легко можно составить как структурные схемы, так и системы линейных модульных уравнений, отвечающих алгоритмам криптопреобразования с заданными параметрами (числом раундов и шагов) шифрования. Легко убедиться в том, что предложенные примитивы типа скользящего кодирования размывают четкую структуру блоков шифруемого текста, характерную для классических алгоритмов симметричного блочного шифрования. Из этого, в частности, следует, что после завершения скользящего кодирования одинаковым блокам открытого текста будут соответствовать

различные (отличающиеся друг от друга) блоки закрытого текста. Следствием отмеченной особенности примитивов скользящего кодирования является гипотеза, согласно которой может оказаться излишним классический режим шифрования типа *сцепления блоков шифротекста* – ***CBC*** (*Ciphertext Block Chaining*), поскольку его функции реализуются СК.

ВЫВОДЫ

Предлагаемые примитивы типа скользящего кодирования могут быть вставлены в алгоритмы криптографической защиты информации только в сочетании с другими примитивами, например, замены, перестановки, циклического сдвига, гаммирования и т.д. Наиболее перспективным направлением использования СК примитивов является их применение для построения так называемых *управляемых операций преобразования* [4], для которых функции криптографического преобразования становятся зависимыми не только от секретного ключа, но и шифруемого текста. Один из примеров построения симметричных блочных криптоалгоритмов с управляемыми операциями шифрования на основе примитива типа логического скользящего кодирования рассмотрен в [5].

SUMMARY

Original cryptographic primitives which are based on the so-called method «Gray transformations on the contrary» are offered. Variants of the homogeneous and mixed arithmetic-logic operations above multidigit binary code combinations of the ciphered text are considered.

СПИСОК ЛИТЕРАТУРЫ

1. Шнайер Б. Прикладная криптография. – М.: ТРИУМФ, 2003. – 816 с.
2. Белецкий А. Я. Комбинаторика кодов Грея. – Киев: КВІЦ, 2003. – 506 с.
3. Белецкий А. Я., Белецкий А. А. Симметричный блочный RSB-криптоалгоритм // Захист інформації, 2006. - № 2. – С. 38–51.
4. Молдовян А. А., Молдовян Н. А., Гуц Н. Д., Изотов Б. В. Криптография: скоростные шифры. – СПб.: БХВ-Петербург, 2002. – 406 с.
5. Белецкий А. Я., Белецкий А. А., Кузнецов А. А. Семейство симметричных блочных криптографических алгоритмов защиты информации с динамически управляемыми параметрами шифрования // Тезисы 3-й МНК «Современные методы кодирования в электронных системах». – Сумы: СГУ, 2006. – С. 30-31.

Поступила в редакцию14 декабря 2006 г.