ТЕОРІЯ ТА МЕТОДИ ОБРОБЛЕННЯ СИГНАЛІВ

УДК 004.056.5(045)

А. А. Белецкий

ГИБКОЕ РАСПИСАНИЕ ФОРМИРОВАНИЯ БАЗОВЫХ ПОДКЛЮЧЕЙ В СИММЕТРИЧНОМ БЛОЧНОМ *RSB* КРИПТОАЛГОРИТМЕ

Інститут електроніки та систем управління НАУ, e-mail: abel@nau.edu.ua

Приведен алгоритм квазистохастической модификации базовых подключей RSB шифра, являющихся компонентами основного ключа криптографических преобразований.

Введение и постановка задачи. В известных блочных шифрах ключ шифрования используется по детерминированному расписанию, т. е. на заданных шагах шифрования используются одни и те же подключи для всех преобразуемых входных блоков данных [1]. Однако тот факт, что криптоаналитику известна последовательность использования подключей, требует использования достаточно большого количества раундов шифрующего преобразования. Последнее ведет к проблематичности получения высоких скоростей шифрования. Для построения скоростных программных шифров была предложена идея использования псевдослучайных выборок из ключа шифрования [2]. Криптосхемы с подобной недетерминированной выборкой подключей характеризуются тем, что элементы криптографического ключа, используемые при преобразовании входного блока данных, выбираются в зависимости от структуры ключа шифрования. Для шифров, основанных на подобной криптосхеме, расписание выборки подключей не является предопределенным, что является существенным фактором повышения стойкости шифрования. За счет этого количество раундов шифрования может быть уменьшено и тем самым повышена скорость шифрования.

Шифры, в которых алгоритм шифрования формируется в зависимости от секретного ключа, называются гибкими или недетерминированными.

RSB – это итерационный блочный шифр, который доставляет уникальную возможность изменения как размеров блоков шифруемых данных (текстов), так и секретных ключей, а также количества шагов (и, соответственно, раундов) шифрования [3; 4]. 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 количество шагов (***S***), а действие алгоритма осуществляется над блоками (***B***) открытого или закрытого текстов, причем размер раундового ключа (как элемента общего ключа) составляет 32 бит.

Общий ключ в ***RSB*** шифре образуется конкатенацией (объединением) ***r*** 32-разрядных базовых раундовых ключей. При условии, что количество шагов шифрования ***s*** больше единицы, за счет циклического (кругового) сдвига на семь разрядов влево (рис. 1) происходит частичное обновление общего ключа и процедура криптографического преобразования шифруемого текста на очередном шаге зашифрования осуществляется под управлением модифицированного общего ключа.

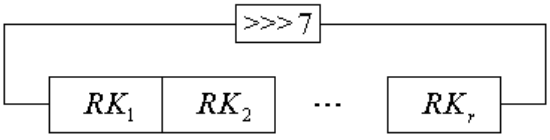


Рис. 1

Очевидно, что схема модификации общего ключа шифрования, который будем называть также мастер-ключом (***M****aster* ***K****ey*), представленная на рис. 1, сохраняет

детерминированным алгоритм изменения этого ключа. Основная задача, достижение которой ставится в данной работе, состоит в разработке достаточно простого в программной реализации алгоритма квазистохастического (псевдослучайного) перемешивания битов мастер-ключа. В результате такого перемешивания образуются «шаговые» мастер-ключи ($\boldsymbol{MK}_i$), индивидуальные и различные для каждого i-го шага шифрования.

Изложение основного материалы. Итак, пусть $\boldsymbol{MK}_0$ – исходный $32\boldsymbol{r}$-битный общий ключ шифрования, причем параметр $\boldsymbol{r}$ в ***RSB*** алгоритме должен быть четным количеством, таким, что $\boldsymbol{r} \geq 4$. Мастер-ключ $\boldsymbol{MK}_i$ для i-го шага зашифрования образуется из мастер-ключа $\boldsymbol{MK}_{i-1}$ предыдущего $(i-1)$-го шага в результате последовательно выполняемых двух линейных псевдослучайных операций: циклического сдвига ($\boldsymbol{S}$) и перестановки ($\boldsymbol{P}$) элементов общего ключа $\boldsymbol{K}$, а также одной нелинейной операции ($\mathcal{N}$), т. е.

$$\boldsymbol{MK}_i = \mathcal{N}\left\{\boldsymbol{P}_{i-1}\left[\vec{\boldsymbol{S}}_{i-1}(\boldsymbol{MK}_{i-1})\right]\right\}, \tag{1}$$

где стрелка над $\boldsymbol{S}$ означает направление кругового сдвига, в данном случае (на этапе зашифрования) – левостороннее.

Нелинейное преобразование $\mathcal{N}$ осуществляется над каждым байтом $\boldsymbol{x}$ ключа $\boldsymbol{MK}$ и наследует основные черты аналогичного преобразования в стандарте ***AES*** [5]

$$\boldsymbol{y} = (\boldsymbol{x}^{-1}\boldsymbol{M})_2, \tag{2}$$

в котором $(\boldsymbol{a})_2$ означает приведение $\boldsymbol{a}$ к остатку по $\bmod 2$; $\boldsymbol{x}^{-1}$ – мультипликативно обратное значение двоичного байта $\boldsymbol{x}$ над неприводимым полиномом $\varphi(x)$ восьмого порядка, в качестве которого принят полином

$$\varphi(x) = x^8 + x^4 + x^3 + x + 1,$$

и, наконец,

$$\boldsymbol{M} = \begin{bmatrix} 0&0&0&0&1&1&1&0 \\ 0&0&0&0&1&1&0&1 \\ 0&0&0&0&1&0&1&1 \\ 0&0&0&0&0&1&1&1 \\ 1&1&1&0&0&0&0&0 \\ 1&1&0&1&0&0&0&0 \\ 1&0&1&1&0&0&0&0 \\ 0&1&1&1&0&0&0&0 \end{bmatrix}$$

– инволютивная матрица преобразования.

Инволютивной двоичной матрицей является такая квадратная матрица, обратная матрица которой над кольцом вычетов по $\bmod 2$ совпадает с исходной. Это означает, в частности, что операция (2), но совершаемая над вектором $\boldsymbol{y}$, восстанавливает вектор $\boldsymbol{x}$.

Параметры преобразования $\boldsymbol{S}$ и $\boldsymbol{P}$ определяются состоянием байта

$$B_i = \{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}, \tag{3}$$

который образуется в результате поразрядного суммирования по $\bmod 2$ всех байтов ключа $\boldsymbol{K}_{i-1}$.

Операцией псевдослучайного циклического сдвига осуществляется стохастический круговой сдвиг влево (для операции зашифрования) общего ключа $\boldsymbol{K}$ на случайное нечетное количество, которое задается пятиразрядным двоичным словом $\boldsymbol{S}$. Четыре старших разрядов этого слова считываются из старшего полубайта (3), а в младший разряд слова принудительно записывается единица. Тем самым код, которым определяется порядок циклического сдвига ключа $\boldsymbol{K}$, будет содержать нечетное количество в интервале от 1 до 33.

Операция псевдослучайной перестановки элементов ключа $\boldsymbol{K}$ реализуется следующим образом. $32\boldsymbol{r}$-битный ключ $\boldsymbol{K}$ представляется в виде совокупности 16 элементов, каждый из которых включает $2\boldsymbol{r}$ битов. Элемент ключа, расположенный в $2\boldsymbol{r}$-битной ячейке, номер которой равен $\boldsymbol{x}$, перемещается в ячейку под номером $\boldsymbol{y}$, причем

$$\boldsymbol{y} = (\boldsymbol{x}\boldsymbol{M}_p)_2, \tag{4}$$

где $\boldsymbol{M}_p$ – матрица перестановки, в качестве которой выбирается одна из 16 инволютивных двоичных матриц четвертого порядка (табл. 1), являющихся индикаторными матрицами систем функций Уолша 16-го порядка [6].

Таблица 1

m	n			
	0	1	2	3
0	1 0 0 0 1 1 1 0 0 0 1 0 0 0 1 1	0 1 1 1 0 0 0 1 1 1 0 1 0 1 0 0	0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0	0 0 1 1 1 1 0 1 0 0 1 0 1 0 1 0
4	0 1 1 1 1 0 1 1 0 0 0 1 0 0 1 0	0 1 1 0 1 1 1 1 0 1 1 1 1 0 1 0	0 0 1 0 1 0 1 1 1 0 0 0 1 1 1 0	0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 0
8	0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0	1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1	1 0 0 0 0 1 1 0 0 0 1 0 1 0 0 1	1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1
12	1 1 1 1 0 0 1 1 1 0 0 1 1 1 0 1	1 0 1 1 1 0 0 1 1 1 0 0 1 1 1 1	1 0 1 0 1 1 0 1 0 0 1 0 0 0 1 1	1 1 0 0 0 1 0 0 1 1 1 1 0 1 0 1

Номер p матрицы перестановки в уравнении (4) определяется параметрами m и n (табл. 1) по формуле

$$p = m + n.$$

Перестановки, отвечающие матрицам $\boldsymbol{M}_p$ (табл. 1), сведены в табл. 2.

Таблица 2

p	Десятичный номер ячейки $\boldsymbol{x}$															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	3	2	1	14	13	12	15	8	11	10	9	6	5	4	7
1	0	4	13	9	1	5	12	8	7	3	10	14	6	2	11	15
2	0	12	15	3	7	11	8	4	6	10	9	5	1	13	14	2
3	0	10	2	8	13	7	15	5	3	9	1	11	14	4	12	6
4	0	2	1	3	11	9	10	8	7	5	6	4	12	14	13	15
5	0	10	7	13	15	5	8	2	6	12	1	11	9	3	14	4
6	0	14	8	6	11	5	3	13	2	12	10	4	9	7	1	15
7	0	6	15	9	14	8	1	7	5	3	10	12	11	13	4	2
8	0	14	13	3	11	5	6	8	7	9	10	4	12	2	1	15
9	0	7	10	13	12	11	6	1	8	15	2	5	4	3	14	9
10	0	9	2	11	6	15	4	13	8	1	10	3	14	7	12	5
11	0	7	11	12	13	11	6	1	14	9	5	2	3	4	8	15
12	0	13	9	4	3	14	10	7	15	2	6	11	12	1	5	8
13	0	15	12	3	9	6	5	10	11	4	7	8	2	13	14	1
14	0	3	2	1	13	14	15	12	10	9	8	11	7	4	5	6
15	0	5	15	10	4	1	11	14	12	9	3	6	8	13	7	2

Структурная схема алгоритма управления общими ключами в четырехшаговом ***RSB*** шифре показана на рис. 2. На схеме приняты такие обозначения: B_i – байт (4), образованный поразрядным суммированием по $\mathrm{mod}\,2$ всех байтов мастер-ключа $\boldsymbol{MK}_{i-1}$; $\boldsymbol{Cr}_i$ – криптограмма, сформированная из входного текста (***In Text***) на i-м шаге зашифрования.

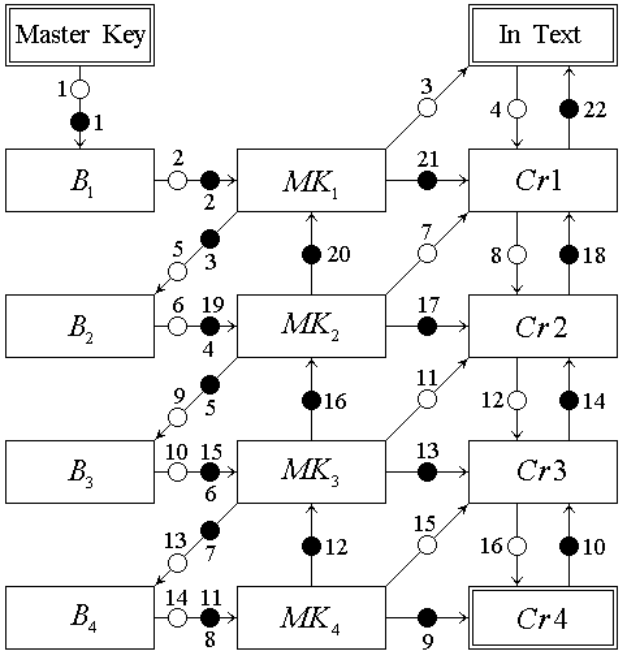


Рис. 2

Элементы (блоки) структурной схемы объединены направленными ребрами, указывающими направление передачи управления преобразованиями. На ребрах размещены светлые кружочки, символизирующие этапы зашифрования, и темные кружочки, символизирующие этапы расшифрования. Номер этапа указан в окрестности кружочка.

Кратко поясним схему модификации общих ключей для режима зашифрования. В соответствии с рис. 2 на этапе 1 вычисляется байт B_1, с помощью которого (этап 2) из ***Master Key*** преобразованием (1) формируется общий ключ $\boldsymbol{MK}_1$ для первого шага зашифрования (этапы 3 и 4). После этого из ключа $\boldsymbol{MK}_1$ образуется байт B_2 (этап 5), с помощью которого на этапе 6 сначала формируется ключ $\boldsymbol{MK}_2$, а затем на этапах 7 и 8 вычисляется криптограмма $\boldsymbol{Cr}2$, завершающая второй шаг зашифрования. Аналогично продолжается процесс зашифрования входного текста на третьем и четвертом шагах зашифрования.

Возможны два варианта построения алгоритма формирования гибкого расписания для общих ключей в процессе расшифрования. Оба этих варианта предполагают обязательное прохождение этапов с первого по восьмой (для схемы, представленной на рис. 2), причем в первом варианте предусматривается запоминание всех байтов $B_1 - B_4$ и последнего мастер-ключа $\boldsymbol{MK}_4$. На первой ступени расшифрования (этапы 9 и 10) из криптограммы $\boldsymbol{Cr}4$ под управлением общего ключа $\boldsymbol{MK}_4$ восстанавливается криптограмма $\boldsymbol{Cr}3$, а этапами 11 и 12 ключ $\boldsymbol{MK}_4$ преобразуется в ключ $\boldsymbol{MK}_3$. Данное преобразование является обратным преобразованию (1), т. е. в общем случае

$$\boldsymbol{MK}_{i-1} = \vec{\boldsymbol{S}}_i\{\boldsymbol{P}_i[\mathcal{N}(\boldsymbol{MK}_i)]\}. \qquad (5)$$

Уточним преобразование (5) для $i = 4$, в ходе которого на основании B_4 из $\boldsymbol{MK}_4$ формируется ключ $\boldsymbol{MK}_3$. Напомним, что общий ключ $\boldsymbol{MK}_4$ образуется из ключа $\boldsymbol{MK}_3$ в результате выполнения таких действий. Сначала ключ $\boldsymbol{MK}_3$ сворачивается (по схеме поразрядного суммирования) в байт B_4. Младшие четыре разряда байта (3) переписываются в старшие разряды пятибитного регистра $\boldsymbol{S}$, в младшем разряде которого постоянно удерживается единица. Общий ключ $\boldsymbol{MK}_3$ подвергается циклическому сдвигу влево на нечетное количество разрядов, содержащееся в регистре $\boldsymbol{S}$. После этого $2\boldsymbol{r}$-битные элементы ключа подвергаются псевдослучайной перестановке (4), матрица $\boldsymbol{M}_p$ которой выбирается по адресу, содержащемуся в четырех старших разрядах байта B_4. И, наконец, завершающей операцией в (1) является нелинейное преобразование $\mathcal{N}$, реализуемое соотношением (2).

Отметим следующие особенности обратного преобразования (5), восстанавливающего ключ $\boldsymbol{MK}_3$. Во-первых, поскольку матрицы преобразований $\boldsymbol{M}$ и $\boldsymbol{M}_p$ являются инволютивными, т. е. такими, что

$$\boldsymbol{M}^{-1} \equiv \boldsymbol{M}; \qquad \boldsymbol{M}_p^{-1} \equiv \boldsymbol{M}_p, \tag{6}$$

то операции $\mathcal{N}$ и $\boldsymbol{P}_i$ в соотношении (5) выполняются по схемам (2) и (4), как и в преобразовании (1). В самом деле, согласно тождества (6) обратные преобразования

$$(2):(\boldsymbol{y}^{-1}\boldsymbol{M})_2 = ((\boldsymbol{x}^{-1}\boldsymbol{M})^{-1}\boldsymbol{M})_2 = ((\boldsymbol{x}\boldsymbol{M})^{-1}\boldsymbol{M})_2 = \boldsymbol{x}$$

и

$$(4):(\boldsymbol{y}\boldsymbol{M}_p)_2 = ((\boldsymbol{x}\boldsymbol{M}_p)\boldsymbol{M}_p)_2 = (\boldsymbol{x}\boldsymbol{M}_p^2)_2 = \boldsymbol{x}.$$

Тем самым мы подтвердили, что повторные преобразования (2) и (4) восстанавливают исходное состояние кодовых комбинаций. И, во-вторых, левостороннему круговому сдвигу ($\overleftarrow{\boldsymbol{S}}$) в преобразовании (1) отвечает правосторонний сдвиг ($\vec{\boldsymbol{S}}$) в преобразовании (6), что естественно.

Второй вариант гибкого расписания предполагает запоминание мастер-ключей $\boldsymbol{MK}_i$ для всех шагов расшифрования. Такая версия алгоритма доставляет большую скорость расшифрования, но ее реализация сопряжена с увеличением востребуемого объема памяти.

Заключение. Недетерминированные (псевдослучайные) шифры, предполагающие гибкое расписание мастер-ключей для различных шагов шифрования, ставят перед криптоаналитиками принципиально новую проблему – алгоритмическую неопределенность, в качестве количественной характеристики которой выступает количество различных потенциально реализуемых модификаций алгоритма шифрования. Особый интерес в криптографии представляет шифрующий механизм, в котором операции, зависящие от преобразуемых данных, выполняются над подключами. Такие операции называют «управляемыми». Смысл выполнения управляемых операций над подключами (в ***RSB*** алгоритме – это раундовые подключи) состоит в том, чтобы осуществить модифицирование подключей при шифровании каждого входного блока. В этом случае различные входные тексты будут шифроваться с помощью различных наборов модифицированных значений подключей, что затрудняет вычисление последних.

Список литературы

1. *Молдавян А. А., Молдавян Н. А., Советов Б. Я.* Криптография. – СПб.: Изд-во «Лань», 2001. – 220 с.
2. *Молдавян А. А., Молдавян Н. А*. Псевдовероятностные скоростные блочные шифры для программной реализации // Кибернетика и системный анализ. – № 4. – 1997. – С. 133 – 141.
3. *Белецкий А. Я., Белецкий А. А.* Симметричный блочный RSB-32 криптоалгоритм // Захист інформації. – 2006. – № 2. – С. 38 – 51.
4. *Белецкий А. Я., Белецкий А. А., Кузнецов А. А.* Семейство симметричных блочных RSB криптографических алгоритмов с динамически управляемыми параметрами шифрования // Електроніка та системи управління, – 2007. – № 1(11). – С. 5 – 16.
5. *Зензин О. С., Иванов М. А.* Стандарт криптографической защиты AES. Конечные поля. – М.: КУДИЦ-ОБРАЗ, 2002. – 176 с.
6. *Белецкий А. Я.* Комбинаторика кодов Грея. – К.: КВІЦ, 2003. – 506 с.

О. А. Білецький

Гнучкий розклад формування базових підключів у симетричному блоковому *RSB* криптоалгоритмі

Наведено алгоритм квазістохастичної модифікації базових підключів ***RSB*** шифру, що є компонентами основного ключа криптографічних перетворень.

A. A. Beletskiy

The flexible schedule of base subkeys forming into symmetric block RSB cryptoalgorithm

The algorithm of quasistochastic modifications of RSB cipher subkeys, which are components of main key of cryptographic conversions, is to be discussed.