

ПРИНЦИП ПОСТРОЕНИЯ «ИНВОЛЮТИВНЫХ» ШИФРОВ

Одной из первоочередных задач, решаемых при проектировании современных информационных объектов является обеспечение информационной безопасности. Игнорирование вопросов безопасности информационного объекта может привести не только к потере информации, но и нарушению таких её свойств как конфиденциальность и целостность, что, в свою очередь, может нанести значительный ущерб как владельцам так и пользователям информации. Решение двух указанных подзадач безопасности невозможно без применения современных средств криптографической защиты информации. Неотъемлемым элементом этих средств являются блочные симметричные шифры (БСШ).

Сегодня на Украине отсутствует национальный стандарт БСШ, поэтому особый интерес представляет вопрос определения методов построения алгоритмов БСШ, удовлетворяющих требованиям криптостойкости, производительности и эффективности реализации. Затраты на реализацию алгоритма могут значительно сокращены, если процедуры прямого и обратного преобразований являются идентичными. Далее мы рассмотрим общий принцип построения таких алгоритмов БСШ и несколько примеров его реализации.

1. Конструкция современного блочного шифра

Абсолютное большинство современных блочных симметричных шифров, освещённых в открытой печати, являются *итеративными*. Идея построения итеративных схем шифрования заключается в многократном повторении некоторого сравнительно простого в реализации шифрующего преобразования. Одна итерация такого преобразования обычно называется *циклом*, а само преобразование *цикловым*. При этом, цикловое преобразование может быть «слабым» шифром, однако его многократное повторение должно формировать сильный алгоритм шифрования. Кроме итерируемого циклового преобразования, процедура шифрования может дополнительно включать фиксированные либо ключезависимые начальное ИТ и конечное ФТ преобразования.

Таким образом, процедура r -циклового зашифрования имеет вид последовательности состоящей из параметризованных цикловым ключом k_i , и возможно номером цикла i , биективных отображений (подстановок) F_{i,k_i} , а также дополнительных начальной ИТ и конечной ФТ подстановок:

$$E_K = ИТ \circ F_{1,k_1} \circ F_{2,k_2} \circ \dots \circ F_{r,k_r} \circ ФТ.$$

Процедура обратного преобразования – расшифрования, представляет собой противоположную по направлению последовательность подстановок, обратных итерациям зашифрования:

$$D_K = ФТ^{-1} \circ F_{r,k_r}^{-1} \circ F_{r-1,k_{r-1}}^{-1} \circ \dots \circ F_{1,k_1}^{-1} \circ ИТ^{-1}.$$

Длина ключа, используемого F -функцией, обычно (в несколько раз) меньше длины исходного *пользовательского* ключа шифрования, однако суммарная длина всех цикловых подключей, составляющих *рабочий* ключ шифрования, значительно превышает длину исходного ключа. Для формирования рабочего ключа (k_1, k_2, \dots, k_n) по заданному пользовательскому ключу K используется процедура «разворачивания ключа» KS , выполняемая однократно, при загрузке в шифратор нового пользовательского ключа: $k_i = KS(K, i)$, $i = \overline{1, r}$.

Как видно из приведенных соотношений, последовательность цикловых ключей

расшифрования должна быть противоположной по направлению, последовательности подключей зашифрования. Кроме секретного подключа, в качестве параметра цикловой функции, также может использоваться некоторая константа, идентифицирующая номер цикла. В этом случае, последовательности констант зашифрования и расшифрования, также, должны быть противоположными по направлению. Отметим, что для сокращения аппаратных затрат на реализацию БСШ, цикловые ключи и константы расшифрования могут быть обратными по значению соответствующим ключам и константам зашифрования (в контексте используемых операций):

$$\begin{cases} K^e = (k_1^e, \dots, k_r^e) = KS^e(K) \\ K^d = (k_r^d, \dots, k_1^d) = KS^d(K) \end{cases}.$$

Введём понятие «инволютивного» шифра, под которым будем понимать БСШ с идентичными процедурами прямого E_K и D_K обратного преобразования, т.е. алгоритмы зашифрования и расшифрования отличаются только процедурой разворачивания ключа:

$$\begin{cases} E_K = \Pi \circ F_{1, k_1^e} \circ \dots \circ F_{r, k_r^e} \circ FT, & K^e = KS^e(K) \\ D_K = \Pi \circ F_{r, k_r^d} \circ \dots \circ F_{1, k_1^d} \circ FT, & K^d = KS^d(K) \end{cases}.$$

«Инволютивные» шифры являются частным случаем построения БСШ. Их достоинство заключается в возможности существенного сокращения затрат на реализацию, путём применения общей процедуры шифрования для прямого и обратного преобразований. При этом, желательно, чтобы процедуры разворачивания ключа для прямого и обратного преобразований имели минимальные отличия (с точки зрения сложности реализации), в идеале — только противоположную последовательность цикловых ключей k_i .

Отметим, что в зависимости от вида циклового преобразования, применение фиксированных начального Π и/или конечного FT преобразований может являться необходимым условием построения «инволютивного» шифра.

Наибольшее распространение в этом классе получили шифры на базе цепи Фейстеля [1], позволяющей построить инволютивную цикловую функцию на основе произвольного криптопреобразования. В качестве альтернативной конструкции может рассматриваться схема предложенная в алгоритме IDEA [2], дальше будем называть её IDEA-цепью.

Классическая цепь Фейстеля, а также IDEA-цепь используют свойство инволютивности операции сложения по модулю 2 (XOR). Применение этой операции в цикловой цепи оказывается не желательным при совместном использовании с цикловыми функциями использующими аналогичную операцию для ввода ключа на входе либо выходе циклового преобразования, как в случае алгоритма DES [3]. Более того, определённый интерес представляет изучение возможности замены простейшей операции XOR более сложными управляемыми преобразованиями, позволяющими улучшить некоторые криптографические показатели циклового преобразования.

Анализ конструкций цепи Фейстеля и IDEA-цепи позволяет сформулировать один из возможных принципов построения «инволютивного» итеративного шифра общего вида. Рассмотрим этот принцип и схемы шифраторов на его основе.

2. Общий принцип построения «инволютивного» шифра

В общем случае, «инволютивную» процедуру шифрования можно построить на основе нескольких различных итеративных преобразований. В этом случае используемое множество итеративных преобразований, должно включать пары взаимно-обратных (и возможно самообратных) шифрующих преобразований.

Допустим, что процедура зашифрования может быть представлена в виде q идентичных

последовательностей, состоящих из t различных итеративных преобразований F_{j, k_i} , параметризованных цикловым ключом k_i , а также начального ИТ и конечного ФТ преобразований. Тогда процедура шифрования представляет собой последовательность из $r = q \times t$ итеративных преобразований:

$$E_K = \text{ИТ} \circ (F_0 \circ F_1 \circ \dots \circ F_{t-1})^q \circ \text{ФТ}.$$

В такой схеме последовательность из t различных итеративных преобразований $(F_0 \circ F_1 \circ \dots \circ F_{t-1})$ может рассматриваться в качестве одного цикла, повторяемого q раз. Для того чтобы шифр указанной структуры был «инволютивным» необходимо выполнить следующие два условия:

1) композиция конечного ФТ_k и начального ИТ_k преобразований на одном и том же ключе k является не зависящим от ключа преобразованием, т.е.: $\text{ФТ}_k \circ \text{ИТ}_k = \Phi$, где Φ – некоторое фиксированное инволютивное преобразование ($\Phi = \Phi^{-1}$), откуда

$$\text{ИТ}_k \circ \Phi = \text{ФТ}_k^{-1}; \quad (1)$$

2) если итеративные преобразования упорядочить в порядке их применения, то они должны удовлетворять следующему соотношению:

$$\Phi \circ F_{j, k} \circ \Phi = F_{t-1-j, k}^{-1}, \quad 0 \leq j < t, \quad (2)$$

т.е. итеративные преобразования, расположенные «зеркально», относительно середины $(t/2)$ последовательности циклового преобразования, должны быть «взаимно обратными», с точностью до фиксированного преобразования Φ . Из (2) следует, что

$$F_{t-1-j, k} \circ \Phi \circ F_{j, k} = \Phi, \quad 0 \leq j < t. \quad (3)$$

Так как конечной целью разработки «инволютивного» шифра является минимизация затрат на его реализацию, то нежелательной является ситуация когда каждое итеративное преобразование реализуется независимо. Поэтому, в случае $t > 1$, особый интерес представляет возможность совместного использования на всех итерациях некоторого общего ресурсоёмкого преобразования.

Если представить итеративное преобразование F_j как управляемую цикловым ключом и промежуточными данными биекцию, то его можно разделить на две составляющие: функцию формирования «управляющего сигнала» (f -функцию) и «частичное» биективное преобразование блока данных $\Theta_{j, Z}$, управляемое выходом Z указанной f -функции.

В общем случае, в качестве отображений $\Theta_{j, Z}$ могут использоваться произвольные обратимые преобразования, однако в качестве «управляемой составляющей» (обозначим её $\Psi_{j, Z}$) предпочтительно использовать аффинные преобразования, т.к. в большинстве случаев, они могут быть сравнительно просто реализованы на широком спектре аппаратных средств, включая современные универсальные процессоры. При этом, выбирая управляемое отображение, необходимо учитывать сложность реализации как прямого $\Theta_{j, Z}$, так и обратного $\Theta_{j, Z}^{-1}$ преобразований. Кроме того, применение в составе отображения $\Theta_{j, Z}$ фиксированных линейных преобразований позволяет увеличить эффекты «рассеивания» и «размножения» активизации, а нелинейных преобразований — эффект «смешивания».

Отметим, что преобразование обратное к $\Psi_{j, Z}$ может быть выражено через обратное значение «управляющего сигнала» Z^{-1} (в контексте используемых операций): $\Psi_{j, Z}^{-1} = \Psi_{j, Z^{-1}}$.

В рассмотренной модели шифратора основная «нагрузка» по обеспечению криптостойкости итеративного преобразования обычно возлагается на f-функцию, которая должна быть «сильным» нелинейным криптопреобразованием (возможно однонаправленным). В связи с этим, на f-функцию обычно приходится основная сложность реализации алгоритма, поэтому её целесообразно выполнить общей для всех итераций прямого и обратного преобразований. В этом случае различные итерации шифрования будут отличаться только биективными преобразованиями $\Theta_{j,Z}$.

Для построения «инволютивного» шифра на базе указанной модели итеративного преобразования необходимо обеспечить выполнение условия (3) для отдельных биекций $\Theta_{j,Z}$, т.е.:

$$\Theta_{t-1-j,Z} \circ \Phi \circ \Theta_{j,Z} = \Phi, \quad 0 \leq j < t, \quad (4)$$

а также необходимо, чтобы значения «управляющего сигнала» Z_i (выход f-функции), на соответствующих циклах процедур зашифрования и расшифрования совпадали. Для выполнения последнего условия цикловые ключи и аргументы f-функции на соответствующих итерациях должны быть идентичными. Идентичность цикловых ключей обеспечивается процедурой «разворачивания ключа», а для получения идентичных аргументов f-функции необходимо использовать такие отображения $\Omega_j(X)$ полного блока данных X в некоторый битовый вектор, которые бы удовлетворяли условию:

$$\Theta_{j,Z} \circ \Phi \circ \Omega_j(X) = \Omega_j(X) = \Omega_{t-1-j}(X), \quad 0 \leq j < t. \quad (5)$$

Так в случае классической цепи Фейстеля (рис. 1-а) на каждой итерации ($t = 1$) одна половина блока данных R управляет отображением другой половины L этого блока:

$$\Theta_Z(\langle L, R \rangle) = \langle R, L \oplus Z \rangle, \quad Z = f_{k_i}(\Omega(\langle L, R \rangle)), \quad \Omega(\langle L, R \rangle) = R.$$

В случае IDEA-цепи (рис. 1-б), где также $t = 1$, для выделения «аргумента» f-функции используется свойство инволютивности операции сложения по модулю 2:

$$\Theta_Z(\langle L, R \rangle) = P(\langle L \oplus Z, R \oplus Z \rangle), \quad Z = f_{k_i}(\Omega(\langle L, R \rangle)), \quad \Omega(\langle L, R \rangle) = L \oplus R,$$

где P – фиксированная перестановка «четвертушек» блока данных.

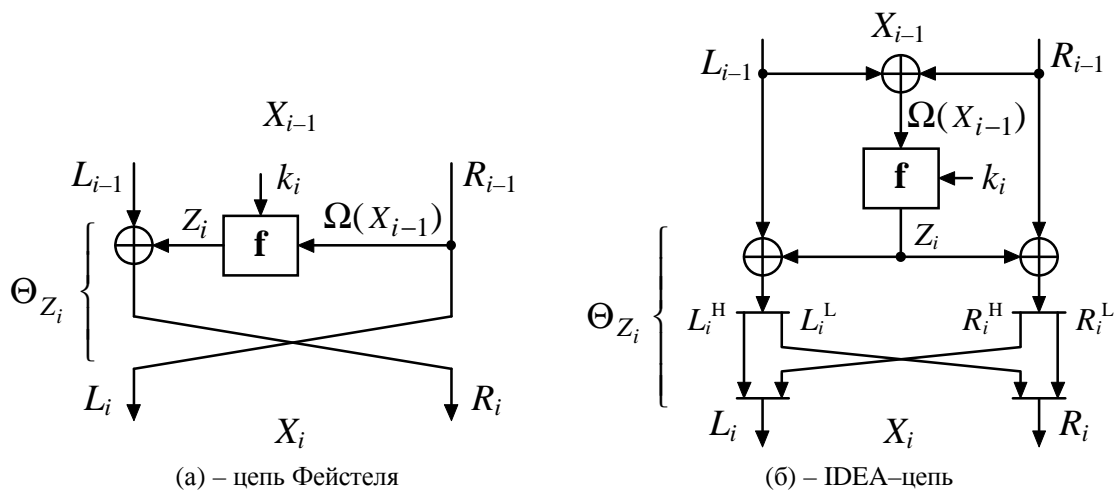


Рис. 1.

В общем виде процедуры зашифрования / расшифрования для «инволютивного» шифра на базе общей f-функции можно представить следующим образом:

$$\begin{aligned}
F_{k_i, \dots, k_{i+t-1}} : \quad & X_{i+t} = \Theta_{0, Z_i} \circ \dots \circ \Theta_{t-1, Z_{i+t-1}}(X_i), \quad i = t \times m, \quad m = \overline{0, q-1}, \\
& B_i = X_i, \quad X_0 = \Pi_{k_{\text{IT}}}(P), \quad C = \text{FT}_{k_{\text{FT}}}(X_r); \\
F_{k_{i+t-1}, \dots, k_i}^{-1} : \quad & Y_i = \Theta_{0, Z_{i+t-1}} \circ \dots \circ \Theta_{t-1, Z_i}(Y_{i+t}), \quad i = r - t \times m, \quad m = \overline{1, q}, \\
& B_i = Y_i, \quad Y_r = \Pi_{k_{\text{FT}}}(C), \quad P = \text{FT}_{k_{\text{IT}}}(Y_0); \\
& X_{i+1} = \Theta_{j, Z_i}(B_i), \quad Z_i = f_{k_i}(\Omega_j(B_i)), \quad j = i \bmod t, \quad \Phi(Y_i) = X_i, \quad 0 \leq i < r,
\end{aligned} \tag{6}$$

где P – блок открытого текста;

C – блок криптограммы;

$\Omega_j(X)$ – функция «выделения» аргумента f -функции из полного блока данных;

$\Theta_{j, Z}(X)$ – биективное отображение блока X , управляемое параметром Z ;

$f_k(X)$ – функция нелинейного «смешивания» вектора X , управляемая ключом k ;

k_i – цикловый ключ, который может содержать не только информацию о секретном пользовательском ключе, но и некоторую вспомогательную (открытую) информацию, например номер цикла i .

Таким образом, для построения «инволютивного» шифра на базе модели (6) необходимо выбрать f -функцию, некоторое множество управляемых биекций $\{\Theta_0, \dots, \Theta_{t-1}\}$, а также Π и FT преобразования, обладающих необходимыми криптографическими показателями и удовлетворяющих условиям (1), (4) и (5). Выбирая множество $\{\Theta_0, \dots, \Theta_{t-1}\}$ следует учитывать, что конечной целью построения «инволютивного» шифра является минимизация затрат на его реализацию, поэтому предпочтение следует отдавать алгоритмам с минимальным числом разнотипных биекций Θ_j . В связи с этим особый интерес представляют следующие два варианта:

- 1) $t = 1$: $\Phi \circ \Theta_Z \circ \Phi = \Theta_Z^{-1}$;
- 2) $t = 2$: $\Phi \circ \Theta_{0, Z} \circ \Phi = \Theta_{1, Z}^{-1}$, $\Omega_0 = \Omega_1$.

Рассмотрим несколько примеров построения «инволютивных» шифров в соответствии с рассмотренными принципами, используя в качестве основы цепь Фейстеля и IDEA-цепь.

3. Цепи Фейстеля и IDEA общего вида

Цепью Фейстеля общего вида будем называть схему циклового преобразования, в соответствии с которой исходный блок представляется в виде вектора $X = (x_{n-1}, \dots, x_1, x_0)$, состоящего из n подблоков, а преобразования $\Theta_{j, Z}$ состоят из управляемого биективного отображения одного из подблоков (например x_{n-1}) и фиксированного циклического сдвига всего блока на один подблок, при этом отображение $\Omega(X)$, «выделяющее» аргумент f -функции, возвращает вектор содержащий все подблоки, кроме модифицируемого на данной итерации, т.е. $\Omega(X) = (x_{n-2}, \dots, x_1, x_0)$.

В случае применения в ветви преобразования подблока x_{n-1} не инволютивных управляемых преобразований, в такой схеме достаточно воспользоваться двумя ($t = 2$) взаимнообратными преобразованиями Ψ_Z и Ψ_Z^{-1} . В этом случае, каждое из преобразований применяется на n соседних итерациях, и общее число итераций шифрования должно быть кратно $2n$.

Шифр, полученный в соответствии с указанной схемой (рис. 2), для $n > 2$ не будет «инволютивным» — процедуры прямого и обратного преобразования будут отличаться

противоположным направлением «обхода» блока (на рис. 2: $[\lll]$ – циклический сдвиг на один подблок влево, $[\ggg]$ – циклический сдвиг на один подблок вправо). Однако, указанное отличие процедур зашифрования и расшифрования, с точки зрения сложности реализации, является несущественным, и учитывая, что конечной целью является минимизация затрат на реализацию, а не «инволютивность» шифратора, решение на основе указанной схемы может быть вполне оправданным.

Применение схемы с $n > 2$ позволяет выбрать на основе данных преобразование Ψ из множества $2^{(n-1)b}$, где b – длина подблока в битах, однако для двух соседних итераций расстояние между указанными множествами будет составлять 2^b . Кроме того, для достижения функциональной зависимости каждого бита выхода от всех битов входа потребуется как минимум n итераций шифрования, если Ψ выполняет «смешивание» всех битов подблока.

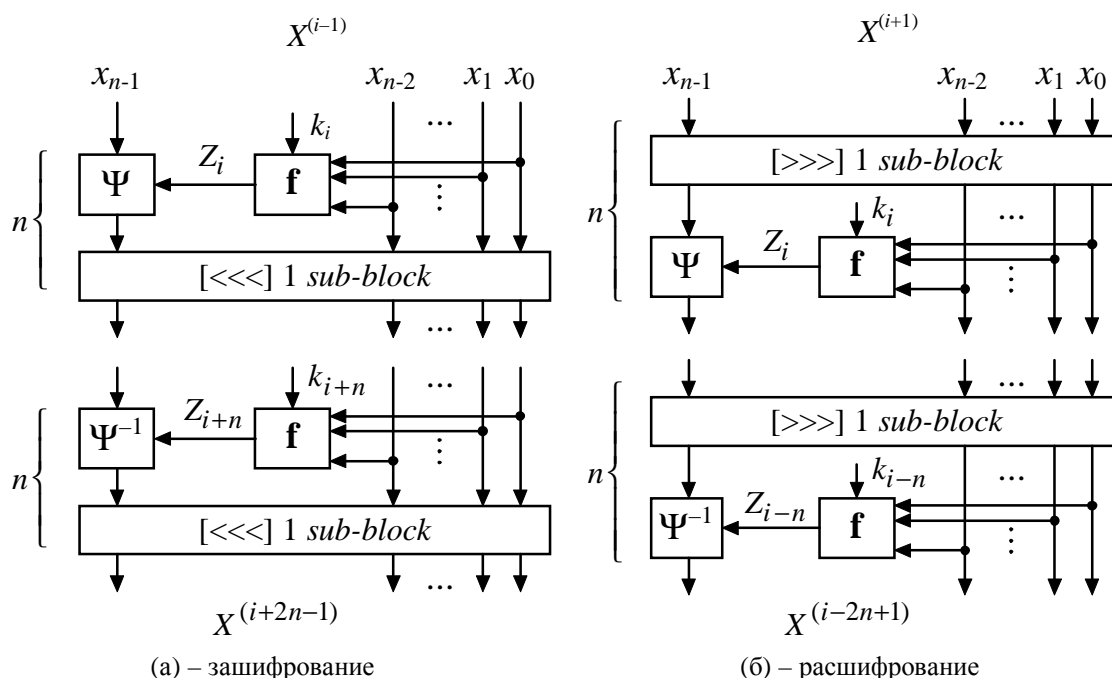


Рис. 2.

В связи с этим, для современных длин блока (128–512 бит) и разрядности массовых процессоров (32–64 бита), большой интерес представляет случай $n = 2$, позволяющий получить действительно «инволютивную» процедуру шифрования, а также позволяющий (при соответствующем выборе f и Ψ) всего 2–3 итерации достигнуть зависимости каждого выхода от всех входов. Рассмотрим пример построения Фейстель-подобной «инволютивной» цепи (случай $n = 2$), эффективно реализуемой на современной аппаратной базе (рис. 3-а).

Для случая $n = 2$, в качестве преобразования Ψ целесообразно использовать некоторую групповую операцию, т.к. размерности пространства значений «управляющего сигнала» и модифицируемого полублока равны. Классическая цепь Фейстеля использует в качестве Ψ простейшую в реализации групповую операцию – XOR. Другой, наиболее эффективно поддерживаемой большинством современных процессоров групповой операцией является сложение (и обратная к ней – вычитание) по модулю 2^w , $w = 8, 16, 32, 64$. Данная операция, в отличие от XOR, обеспечивает зависимость каждого разряда суммы не только от одноимённых разрядов слагаемых, но и от всех младших разрядов, и на $GF(2)^w$ является нелинейной. Кроме того не совместимость операций XOR и ADD [2] позволяет использовать их последовательно (одну для «введения» ключа в f -функцию, а другую для образования цепи).

В качестве фиксированной составляющей преобразования $\Theta_{j,Z}$ могут использоваться S-блоки (например, для нелинейного «смешивания» одноимённых разрядов «слов» суммы) и фиксированная перестановка P (для выполнения «рассеивания» активизации). В качестве простейшей перестановки может использоваться циклический сдвиг «слов» составляющих подблоков. При этом, величины сдвигов должны обеспечивать максимальный период циклического повторения. Для этого величины сдвигов различных «слов» должны быть взаимно простыми нечётными числами (для случая $b = 2^w$).

Как видно из рис. 3-а для получения «инволютивного» шифра обратное преобразование Θ_Z^{-1} должно состоять из противоположной по направлению последовательности преобразований, обратных соответствующим элементарным преобразованиям составляющим Θ_Z . Кроме того, преобразование Φ должно определять перестановку левого L и правого R полублоков (для этого можно использовать ИТ – тождественное, а FT = Φ).

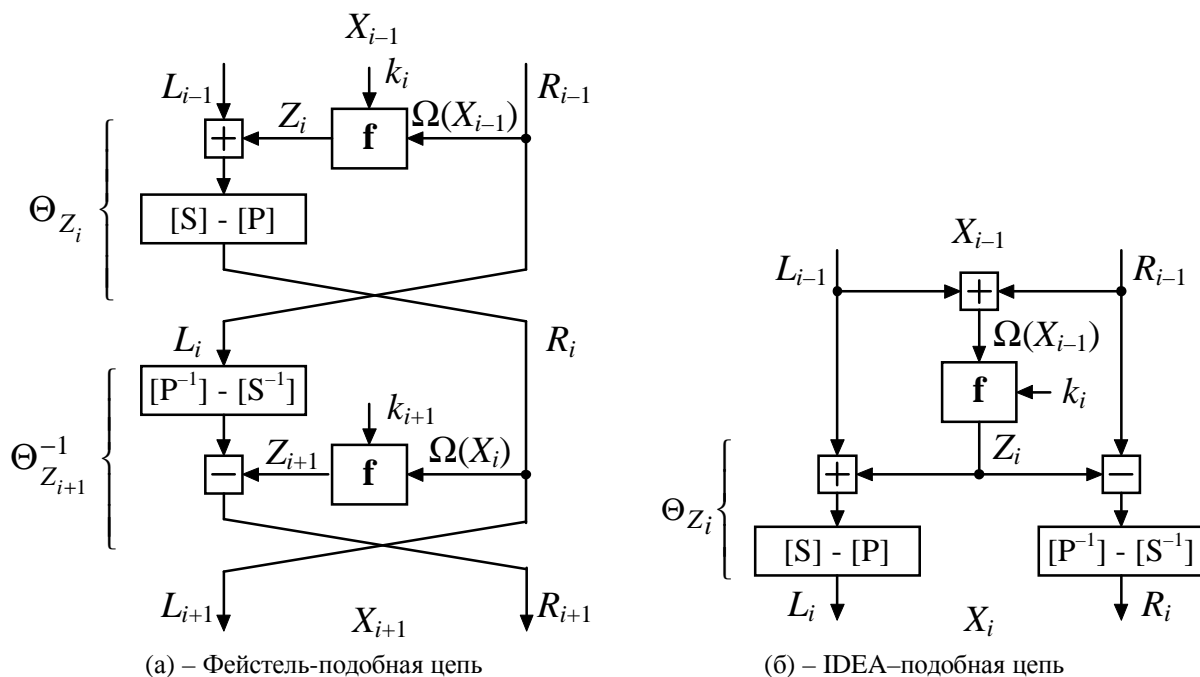


Рис. 3

IDEA-подобные цепи позволяют построить «инволютивный» шифр на основе только одной ($t = 1$) управляемой биекции Θ_Z , так в оригинальном алгоритме базовая цепь является инволюцией. Однако, в отличие от цепи Фейстеля, IDEA-цепь является менее универсальным решением, т.к. в качестве управляемого преобразования Ψ предполагает использование некоторой групповой операции.

Следует отметить, что фиксированные перестановки «четвертушек» блока в оригинальном алгоритме (рис. 1-б) не являются необходимым условием построения «инволютивного» шифра, а введены в алгоритм для повышения его криптостойкости, более того в ветвях полублоков (либо «четвертушек») обязательно должны присутствовать некоторые преобразования несовместимые (некоммутативные) с Ψ , т.к. в противном случае происходит снижение криптостойкости алгоритма. Так, если не выполнять даже перестановку «четвертушек» блока в оригинальном алгоритме, то получим

$$\Omega(X_{i+1}) = \Omega(X_i),$$

т.е. аргумент f-функции всегда будет постоянным (не будет зависеть от выхода f-функции на предыдущих циклах). Если кроме базовой цепи выполнять только перестановку «четвертушек» блока, то для трёх соседних итераций получим:

$$\begin{aligned}
X_i &= \{L_i, R_i\} = \{L_i^H, L_i^L, R_i^H, R_i^L\} \\
X_{i+1} &= \{L_i^H \oplus Z_{i+1}^H, R_i^H \oplus Z_{i+1}^H, L_i^L \oplus Z_{i+1}^L, R_i^L \oplus Z_{i+1}^L\} \\
\Omega(X_{i+1}) &= \{L_i^H \oplus L_i^L \oplus Z_{i+1}^H \oplus Z_{i+1}^L, R_i^H \oplus R_i^L \oplus Z_{i+1}^H \oplus Z_{i+1}^L\} \\
X_{i+2} &= \{L_i^H \oplus Z_{i+1}^H \oplus Z_{i+2}^H, L_i^L \oplus Z_{i+1}^L \oplus Z_{i+2}^H, \\
&\quad R_i^H \oplus Z_{i+1}^H \oplus Z_{i+2}^L, R_i^L \oplus Z_{i+1}^L \oplus Z_{i+2}^L\} \\
\Omega(X_{i+2}) &= \{L_i^H \oplus R_i^H \oplus Z_{i+2}^H \oplus Z_{i+2}^L, L_i^L \oplus R_i^L \oplus Z_{i+2}^H \oplus Z_{i+2}^L\} \\
X_{i+3} &= \{L_i^H \oplus Z_{i+1}^H \oplus Z_{i+2}^H \oplus Z_{i+3}^H, R_i^H \oplus Z_{i+1}^H \oplus Z_{i+2}^L \oplus Z_{i+3}^H, \\
&\quad L_i^L \oplus Z_{i+1}^L \oplus Z_{i+2}^H \oplus Z_{i+3}^L, R_i^L \oplus Z_{i+1}^L \oplus Z_{i+2}^L \oplus Z_{i+3}^L\} \\
\Omega(X_{i+3}) &= \{L_i^H \oplus L_i^L \oplus Z_{i+1}^H \oplus Z_{i+1}^L \oplus Z_{i+3}^H \oplus Z_{i+3}^L, \\
&\quad R_i^H \oplus R_i^L \oplus Z_{i+1}^H \oplus Z_{i+1}^L \oplus Z_{i+3}^H \oplus Z_{i+3}^L\}.
\end{aligned}$$

Воспользовавшись обозначением $Z_i^S = Z_i^H \oplus Z_i^L$, получим следующие соотношения для аргументов f-функции:

$$\begin{aligned}
\Omega(X_i) &= \{L_i^H \oplus R_i^H, L_i^L \oplus R_i^L\} \\
\Omega(X_{i+1}) &= \{L_i^H \oplus L_i^L \oplus Z_{i+1}^S, R_i^H \oplus R_i^L \oplus Z_{i+1}^S\} \\
\Omega(X_{i+2}) &= \{L_i^H \oplus R_i^H \oplus Z_{i+2}^S, L_i^L \oplus R_i^L \oplus Z_{i+2}^S\} \\
\Omega(X_{i+3}) &= \{L_i^H \oplus L_i^L \oplus Z_{i+1}^S \oplus Z_{i+3}^S, R_i^H \oplus R_i^L \oplus Z_{i+1}^S \oplus Z_{i+3}^S\}.
\end{aligned}$$

Из полученных соотношений видно, что если в ветвях цепи используется только перестановка «четвертушек» блока, то вход f-функции на текущем цикле всегда зависит не от полных, а от вдвое «сжатых» Z_i^S выходов f-функции на предыдущих циклах. В связи с этим применение рассмотренной цепи без дополнительных «смешивающих» преобразований в ветвях блока данных невозможно, т.к. приводит к существенному снижению криптостойкости схемы. В тоже время, следует отметить, что чередование пар «четвертушек» блока данных формирующих вход f-функции на соседних итерациях, обеспечивает уникальные значения $\Omega(X_i)$ на двух соседних циклах.

Для построения IDEA-подобного шифратора (рис. 3-б) можно воспользоваться тем же набором элементарных операций, что и в случае цепи Фейстеля (рис. 3-а), т.е. Ψ – операция сложения по модулю 2^w (ADD), фиксированные преобразования в ветвях – нелинейная подстановка (S-блоки) и простейшая перестановка P – циклический сдвиг отдельных «слов» подблока. При этом, для достижения «инволютивности», взаимно противоположные преобразования располагаются в различных ветвях и фиксированные преобразования (S, P, S^{-1} , P^{-1}) на последнем цикле не выполняются (рис. 3-б). Кроме того, как и в случае Фейстель-подобной цепи, преобразование Φ должно определять перестановку левого L и правого R полублоков (т.е. IT – тождественное, FT = Φ).

Вывод

Рассмотренный принцип построения «инволютивных» шифров позволяет синтезировать БСШ с идентичными процедурами прямого и обратного преобразования, на основе единой f-функции и произвольного набора управляемых биективных преобразований. Применение общей f-функции позволяет разработчику сократить затраты на реализацию алгоритма, и при этом, оставляет большую свободу выбора конструкции алгоритма (за счёт возможности использования произвольных управляемых преобразований Θ).

Две приведенные схемы построения циклового преобразования позволяют «быстрее» выполнить нелинейное «смешивание» всех битов блока данных, а также усложняют задачи построения дифференциальных и линейных характеристик по сравнению с соответствующими базовыми схемами (классическая цепь Фейстеля и IDEA-цепь).

Список литературы: 1. Н. Feistel, «Cryptography and computer privacy», Scientific American, 228 (May 1973), pp. 15–23. 2. X. Lai, «On the design and security of block ciphers», ETH Series in Information Processing, J.L. Massey (editor), vol. 1, Hartung-Gorre Verlag Konstanz, Technische Hochschule (Zurich), 1992. 3. FIPS 46, «Data encryption standard», Federal Information Processing Standards Publication 46, U.S. Department of Commerce/National Bureau of Standards, National Technical Information Service, Springfield, Virginia, 1977 (revised as FIPS 46-1:1988; FIPS 46-2:1993).

*Харьковский национальный университет
радиоэлектроники*

Поступила в редколлегию