
МЕТОД КОНСТРУИРОВАНИЯ ЦИКЛОВЫХ ФУНКЦИЙ БСШ

Предлагается метод построения f -функций итеративных алгоритмов блочного симметричного шифрования на базе фиксированных нелинейных преобразований (S -блоков). Предлагаемый метод позволяет, независимо от используемого циклового ключа, получить фиксированные показатели «рассеивания» и «размножения» активизации отдельных S -блоков.

Введение

Стойкость современных алгоритмов блочного симметричного шифрования (БСШ) к классу атак «криптоанализа цикловой функции» определяется свойствами этой функции. Цикловая функция итеративного БСШ должна выполнять «смешивание» и «рассеивание», в соответствии с Шенноном [1]. Свойство «смешивания» обычно реализуется путём применения нелинейных преобразований (S -блоков), а свойство рассеивания определяется структурой циклового преобразования. Для построения современных БСШ (с длиной блока 128 и более бит) на основе S -блоков малой разрядности (4-8 бит), f -функция, также, должна выполнять «размножение активизации» [2,3]. Реализация последнего свойства определяет минимальное число циклов шифрования, необходимых для достижения границы устойчивости к некоторой криптоатаке и возлагается на структуру циклового преобразования. Рассмотрим задачу выбора структуры многоразрядного (64 и более бит) биективного криптопреобразования, обеспечивающего нелинейную зависимость каждого разряда выхода от всех входных разрядов, а также фиксированные показатели «рассеивания» и «размножения» (независимо от используемого циклового ключа).

На сегодняшний день наибольшее распространение получили две конструкции шифрующих преобразований: SPN-структуры (последовательность «слоёв» нелинейной подстановки «слов» и перестановки «букв» между «слоями») и SLTN-структуры (последовательность «слоёв» нелинейной подстановки «слов» и линейного комбинирования различных «слов»). Наряду со «смешиванием», конструкции первого типа реализуют «рассеивание», а второго – «размножение». В качестве основы будем использовать указанные конструкции и будем предполагать применение биективных S -блоков и операции сложения по модулю 2 для введения циклового ключа.

1. Реализация свойства «рассеивания»

«Рассеивание» активизации может достигаться посредством применения SPN-структуры, выполняющей «ветвление» выходов каждого S -блока текущего уровня на различные S -блоки следующего уровня. Поэтому, наиболее оптимальной конструкцией, удовлетворяющей указанному выше свойству, будет частный случай SPN-структуры, когда разрядность преобразуемого блока L равна произведению разрядностей l_i используемых биективных S -блоков в k соседних «слоях», т.е. $L = \prod_{i=1}^k l_i$. Назовём такую конструкцию *пропорциональной* SPN-структурой.

Рассмотрим два «слоя» параллельных нелинейных преобразований, на базе биективных S-блоков S_1 и S_2 разрядностью l_1 и l_2 соответственно, соединённых по принципу «пирамиды», т.е. на входы каждого S-блока второго «слоя» поступает по одному выходному разряду с l_2 различных S-блоков первого «слоя»:

$$\begin{aligned} \{f_{i,0}^{S_k, S_{k+1}}, \dots, f_{i, l_{k+1}-1}^{S_k, S_{k+1}}\} &= S_{k+1}(f_{0,i}^{S_k}, f_{1,i}^{S_k}, \dots, f_{l_{k+1}-1,i}^{S_k}) = \\ &= \left\{ f_{i,0}^{S_{k+1}}(f_{0,i}^{S_k}, \dots, f_{l_{k+1}-1,i}^{S_k}), \dots, f_{i, l_{k+1}-1}^{S_{k+1}}(f_{0,i}^{S_k}, \dots, f_{l_{k+1}-1,i}^{S_k}) \right\}, \quad i = \overline{0, l_k - 1} \end{aligned}$$

где $f_{i,j}^{S_k}$ – булева функция j -го выхода S-блока i «слоя» k ;

$f_{i,j}^{S_k, S_{k+1}}$ – булева функция комбинированного «двухслойного» преобразования на выходе j S-блока i «слоя» $k+1$.

Применение подобной конструкции позволяет решить задачу «рассеивания активизации» между двумя «S-слоями», при этом каждый выходной бит будет нелинейно зависеть от $l_k \times l_{k+1}$ входных битов, а также будет справедлива следующая лемма.

Лемма 1. Если минимальные значения алгебраической степени аргументов булевых функций $f_{i,j}^{S_k}$ и $f_{i,j}^{S_{k+1}}$ равны соответственно d_k и d_{k+1} , то алгебраическая степень любого аргумента «комбинированной» булевой функции $f_{i,j}^{S_k, S_{k+1}}$ будет не меньше чем $d_k \times d_{k+1}$.

Процедура преобразования в соответствии с «пропорциональной» SPN-структурой может быть описана более наглядно, если блок данных представить в виде k -мерного пространства битов «объёмом» L , с размерностями координат равными разрядностям используемых S-блоков l_i .

Тогда процедуру «пропорционального» SPN-преобразования можно рассматривать как последовательное выполнение нелинейного смешивания «вдоль» каждой из координатных осей указанного пространства, т.е. параллельное применение L/l_i однотипных S-блоков разрядностью l_i (один «S-слой»). Полное нелинейное «смешивание» битов блока будет достигнуто за k шагов однослойного нелинейного преобразования, при условии, что на каждом шаге в качестве «направления» нелинейного смешивания (применения S-блока) выбирается уникальная ось координат, как в примере на рис. 1.

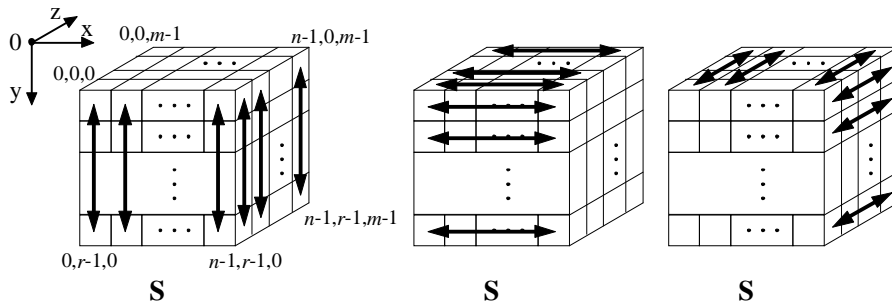


Рис. 1. 3-х мерная «пропорциональная» SPN-структура

При аппаратной реализации SPN-структуры все затраты будут связаны только со сложностью реализации «S-слоя», т.к. перестановка битов заменяется «виртуальной» переадресацией.

При программной реализации SPN-структуры, наоборот, основную сложность составляет реализация перестановки битов. Необходимость в фактической перестановке

битов возникает в случае применения S-блоков одного типа (табличных либо вычисляемых) в направлении различных координат. Наиболее эффективно эта задача может быть реализована, если перестановки имеет регулярную структуру.

При реализации S-блоков в виде булевых соотношений, простейшим решением будет циклический сдвиг содержимого регистров, содержащих одноимённые входные либо выходные разряды S-блока.

При табличной реализации S-блоков, простейшим вариантом регулярной перестановки битов является «перестановка» пары «координат» бита в k -мерном пространстве. Допустим, что размерность двух координат (разрядность S-блока) равна n , тогда «перестановка координат» описывается следующим образом:

$$x \leftrightarrow y: b_{x,y}^t \rightarrow b_{y,x}^{t+1}, \quad x, y \in Z_n,$$

где $b_{x,y}^t$ – значение бита y в n -разрядном «слове» по смещению x в массиве размерностью n , на шаге t .

В качестве другого, простого в реализации, варианта регулярной перестановки битов (для случая табличных S-блоков) может использоваться следующая схема

$$x \leftrightarrow y: b_{x,y}^t \rightarrow b_{(x+y) \bmod n, y}^{t+1}, \quad x, y \in Z_n.$$

Оба приведенных варианта определяют перестановку битов n -разрядного «слова» B_i в n различных «слов» (и наоборот, «слово» B_i формируется конкатенацией одиночных битов, полученных из n различных «слов»).

При реализации на современных универсальных CPU оба варианта регулярной перестановки могут быть реализованы табличным способом. При этом, перестановка может быть реализована для всех «слов» блока на основе одной общей таблицы (в сочетании с дополнительной операцией сдвига), что позволяет сократить затраты «памяти». Для реализации первого варианта перестановки потребуется операция логического сдвига (обозначим её [\ll]), а для второго — циклического сдвига (обозначим её [\lll]). Т.о. перестановка битов между n -разрядными словами $\{B_0, B_1, \dots, B_{n-1}\}$, для первого и второго вариантов соответственно, может быть задана следующим образом:

$$\begin{aligned} 1) \text{ TP}_x(b_{n-1}, \dots, b_1, b_0) &= ((0, \dots, 0, b_{n-1}), \dots, (0, \dots, 0, b_1), (0, \dots, 0, b_0)) \ll x; \\ 2) \text{ TP}_x(b_{n-1}, \dots, b_1, b_0) &= ((b_{n-1}, 0, \dots, 0), \dots, (0, \dots, 0, b_1, 0), (0, \dots, 0, b_0)) \lll x; \\ (B_{n-1}, \dots, B_1, B_0)^{(t+1)} &= \bigvee_{x=0}^{n-1} \text{TP}_x(B_x^{(t)}), \quad B_x = (b_{n-1}, \dots, b_1, b_0)_x, \quad b_i \in \{0,1\}. \end{aligned}$$

Первый вариант регулярной перестановки является более предпочтительным, т.к. логический сдвиг, с точки зрения аппаратной реализации, проще циклического сдвига и на ряде современных процессоров поддерживает большую разрядность операндов.

Разрядность «слова» $n = 8$, является наиболее оптимальным решением, при реализации на современных CPU общего назначения. Для табличной реализации перестановки битов между байтами ($n = 8$), в соответствии с любым из вариантов, потребуется таблица размерностью $2^8 \times 8 = 2 \text{ KB}$, где 256 входов в таблицу соответствуют различным входным байтам, а 8-байтовый выход соответствует записи каждого бита в виде отдельного байта, как было показано выше.

Выше было рассмотрено решение задачи «рассеивания», однако, для построения сильной цикловой функции, необходимо, также, решить задачу «размножения активизации». Для SPN-преобразования максимальная вероятность дифференциальной

[3] либо линейной [4] характеристики, покрывающей некоторое число «S-слоев» будет иметь место в случае когда в каждом «S-слое» активизируется только по одному S-блоку, для «пропорциональной» SPN-структуры это возможно в случае выполнения только однобитных переходов на внутренних «S-слоях».

2. Реализация свойства «размножения»

Основным требованием к схеме «размножения активизации» является преобразование любой одномерной активизации в многомерную, т.е. увеличение числа активных S-блоков при одноблочной активизации. Для конструкций на базе «пропорциональной» SPN-структуры, эта задача может рассматриваться как задача увеличения веса Хемминга для векторов, содержащих только 1 единичный бит.

Для выполнения этого требования может использоваться невырожденное линейное преобразование. Наиболее простым в реализации линейным преобразованием, обеспечивающим одинаковый коэффициент «размножения» по каждому из аргументов, является схема «все без одного», соответствующая умножению вектора на матрицу вида $A_{m \times m} = \{a_{i,j}\}$, $a_{i,i} = 0$, $a_{i,j} = 1, (i \neq j)$. Эта схема позволяет получить коэффициент «размножения» $(m - 1)$ для любой одномерной активизации (где m – число аргументов):

$$X \mapsto L(X): X'_j = \sum_{i=0}^{m-1} X_i - X_j, \quad X_i \in G, \quad |G| = 2^n, \quad X \in G^m, \quad j = \overline{0, m-1}.$$

Данная схема для линейного смешивания m аргументов, требует 1 операцию загрузки, $m-1$ операций «сложения» и m операций «вычитания», при этом операция «сложения» (и обратная к ней – «вычитание») задаются в контексте некоторой группы $(G,+)$. Учитывая, что операция сложения по модулю 2 имеет наиболее простую аппаратную реализацию, она является наиболее оптимальным решением для схем не требующих выполнения линейного «смешивания» внутри слагаемых и комбинирующих малое число аргументов ($m = 4$). В этом случае, линейное преобразование выполняет независимое смешивание только одноимённых разрядов «слов» X_j и может быть представлено в следующем виде:

$$X \mapsto L(X): x_j = \Sigma \oplus x_j, \quad \Sigma = \bigoplus_{i=0}^{m-1} x_i, \quad x_j, \Sigma \in F_2, \quad X \in F_2^m, \quad j = \overline{0, m-1}.$$

При этом, указанное преобразование является инволютивным. Коэффициент «размножения» $W_H(L(X))/W_H(X)$, для данной схемы, зависит только от веса Хемминга входного сигнала, и может быть определён из следующего соотношения:

$$W_H(L(X)) = \begin{cases} W_H(X), & W_H(X) \bmod 2 = 0 \\ m - W_H(X), & W_H(X) \bmod 2 = 1 \end{cases}, \quad X \in F_2^m,$$

т.е. коэффициент «размножения» будет отличен от 1 только для входных сигналов с нечётным весом Хемминга. Аналогично, «количество ветвей» $B = W_H(X) + W_H(L(X))$, также, зависит от четности веса Хемминга входного сигнала:

$$B = \begin{cases} 2 \times W_H(X), & W_H(X) \bmod 2 = 0 \\ m, & W_H(X) \bmod 2 = 1 \end{cases}, \quad X \in F_2^m,$$

Основным недостатком рассмотренной схемы является сохранение низкого «веса» активизации (например 2, 4) при чётном $W_H(X)$. Однако, этот недостаток существенен только при большом числе аргументов ($m > 6$), а для $m = 4$ может быть проигнорирован. Более того из последнего соотношения следует, что для $m = 4$ «количество ветвей» будет фиксированным $B = 4$, если $W_H(X) > 0$.

Рассмотренная схема «все без одного» может быть эффективно применена для линейного комбинирования n -разрядных аргументов (суммирование по модулю 2^n):

$$X \mapsto L(X): X'_j = X_j - C_0 \times \sum_{i=0}^{m-1} X_i, \quad j = \overline{0, m-1},$$

$$C_0, X_i \in \mathbb{Z}_{2^n}, \quad X \in \mathbb{Z}_{2^n}^m, \quad (C_0, 2^n) = 1, \quad (m-1, 2^n) = 1.$$

В этом случае, обратное преобразование будет иметь аналогичный вид:

$$X' \mapsto L^{-1}(X'): X_j = X'_j - C_1 \times \sum_{i=0}^{m-1} X'_i, \quad C_1 = C_0 \times (m \times C_0 - 1)^{-1} \pmod{2^n}.$$

Для последней схемы, при вычислении «числа ветвей» под весом Хемминга понимается количество отличных от нуля n -разрядных «слов» ($W'_H(X)$). Для этой схемы зависимость выходного веса от входного носит вероятностный характер. В частности, вероятность сохранения «веса активизации» 2, в соответствии с парадоксом «дня рождения», составит $2^{-n/2}$. Кроме того, если $2^n \gg m$, то с высокой вероятностью любая отличная от нуля входная активизация будет приводить к активизации большинства выходных ветвей, вплоть до всех m ветвей, т.е. при условии равновероятности распределения значений активизации отдельных «слов», с высокой вероятностью число ветвей составит $B = W'_H(X) + m$. Очевидно, что вероятность этого события будет возрастать с ростом разрядности слов n , а также будет зависеть от значения C_0 . Для достижения максимальной вероятности многомерной выходной активизации (при любой не нулевой входной активизации), необходимо чтобы константа C_0 имела следующий вид $C_0 = 4 \times t + 3, t \geq 0$.

Для обеспечения нижней границы «количества ветвей» равной $(m + 1)$, при любом (не нулевом) входном сигнале активизации, может использоваться более сложная конструкция, рассматривающая каждое «слово» как элемент расширенного поля \mathbb{F}_{2^n} , а сам блок, подлежащий линейному смешиванию, как расширение порядка m указанного поля. Такая схема была использована в алгоритме Rijndael, для линейного смешивания в «столбцах» [2]. Эта конструкция позволяет устранить недостатки присущие предыдущей схеме и связанные с отсутствием мультипликативно обратного элемента в кольце \mathbb{Z}_{2^n} . Данное преобразование может быть представлено в виде умножения на фиксированный полином $c(x)$ по модулю $f(x)$:

$$\mathbb{F}_{2^n}^m \rightarrow \mathbb{F}_{2^n}^m: g(x) = c(x) \times h(x) \pmod{f(x)}, \quad (c(x), f(x)) = 1, \quad c(x), f(x) \in \mathbb{F}_{2^n}^m,$$

где полиномы $h(x)$ и $g(x)$ соответствуют входу и выходу преобразования и представляют блоки данных как элементы $\mathbb{F}_{2^n}^m$.

Рассмотрим требования к полиномам $c(x)$ и $f(x)$. Для эффективной реализации данного линейного преобразования, целесообразно использовать в качестве модуля полином $f(x) = x^m + 1$ (с коэффициентами из \mathbb{F}_{2^n}), что позволит реализовать преобразование на базе одной предвычисленной таблицы [2]. В этом случае, линейное преобразование входного вектора может быть представлено в виде умножения на матрицу вида $A_{m \times m} = \{a_{i,j} = c_{(i-j) \bmod m}\}$, где $c_i \in \mathbb{F}_{2^n}$ – коэффициенты полинома $c(x)$.

Учитывая, что в интересующих нас приложениях $m = 2^w$, получим $f(x) = (x+1)^{2^w}$, и следовательно, для получения биективного преобразования, полином $c(x)$ не должен

делиться на $(x + 1)$. С другой стороны, для обеспечения $(m + 1)$ активной ветви при одномерной и двумерной активизации, необходимо чтобы

$$\frac{a_{i,j}}{a_{i+r,j}} \neq \frac{a_{i,j+t}}{a_{i+r,j+t}}, \text{ т.е. } \frac{c_i}{c_{(i+r) \bmod m}} \neq \frac{c_{(i-t) \bmod m}}{c_{(i+r-t) \bmod m}}, \quad c_i \neq 0, \quad 0 \leq i \leq m-1, \quad 1 \leq r, t \leq m-1.$$

В этом случае, при двумерной входной активизации только одно «слово» результата может быть равно 0. Аналогично можно задать условие сохранения числа ветвей B при большем весе входной активизации.

Для эффективной реализации рассмотренного преобразования потребуется таблица $TL[x]$ размерностью $2^n \times m \times n$ бит, на её основе преобразование может быть выполнено следующим образом.

$$TL[x] = (c_{m-1}x, \dots, c_1x, c_0x), \quad x, c_i \in F_{2^n},$$

$$Y = \bigoplus_{j=0}^{m-1} TL[x_j] \lll (n \times j), \quad X = (x_{m-1}, \dots, x_0), Y \in F_{2^m}^m.$$

Дальше «слой», выполняющий фиксированное линейное преобразование, будем называть «L-слоем». А три рассмотренные схемы будем называть линейными преобразованиями, соответственно, 1–3-го типа.

3. Комбинирование схем «размножения» и «рассеивания»

Для обеспечения f-функцией свойств «рассеивания» и «размножения», она должна содержать оба рассмотренные выше преобразования: битовую перестановку и линейное комбинирование, соответственно. Назовем такую конструкцию SPLN-структурой. Объединить указанные преобразования можно двумя способами: «параллельно» и «последовательно». «Параллельным» объединением будем называть конструкцию, когда схема «рассеивания» и «размножения» применяются вдоль разных «осей координат» k -мерного пространства блока, а «последовательным» — когда вдоль общих «осей». Способ объединения выбирается исходя из необходимой разрядности f-функции, при этом «последовательный» способ рекомендуется применять только если соотношение разрядности f-функции и S-блока не позволяет выполнить «параллельное» объединение. Рассмотрим оба указанных способа.

«Параллельное» объединение можно выполнить на базе как минимум 3-х мерного пространства, при этом на каждом шаге преобразования «вдоль» каждой из координат применяется один из «слоёв» в следующем порядке: P (фиксированная перестановка), L (схема линейного комбинирования 1-го типа), S (нелинейное преобразование) (рис. 2).

Утверждение 1. Булевы функции выходов последовательности «слоёв» S–P–L–S последней конструкции (рис. 2), будут удовлетворять лемме 1 и будут зависеть от $n^2 \times (m - 1)$ переменных, где n – размерность первых двух координат («слои» S и P), m – размерность третьей координаты («слой» L).

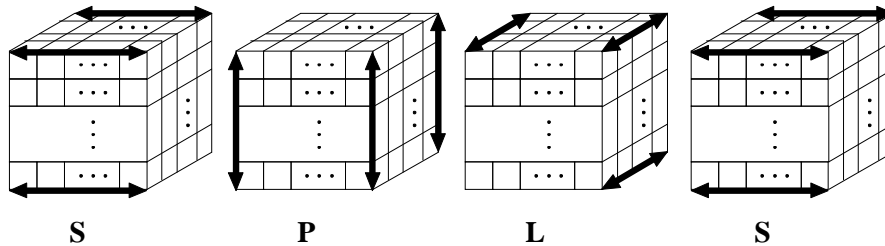


Рис. 2. 3-мерная «параллельная» SPLN-структура

Отметим, что «L-слой» 1-го типа может быть эффективно дополнен управляемой перестановкой, выполняемой вдоль той же «оси», что и линейное комбинирование. В этом случае, битовая перестановка π_k , управляемая ключом k , будет вносить неопределённость относительно «отсутствующего» слагаемого $\pi_k(j)$, по каждому разряду j результата:

$$X \mapsto \text{PL}(X): x'_{i,j} = \left(\bigoplus_{t=0}^{m-1} x_{i,t} \right) \oplus x_{i,\pi_k(j)}, \quad X \in (\mathbb{F}_2^n)^m, \quad i = \overline{0, n-1}, \quad j = \overline{0, m-1}.$$

Добавление управляемой перестановки не окажет влияние на «коэффициент размножения», и полученная последовательность S–P–L–S также будет удовлетворять утверждению 1. Кроме того, подобное управляемое преобразование для малых m может быть эффективно реализовано в виде матричного умножения, при этом соответствующая матрица может быть получена путём инвертирования элементов соответствующей управляющей матрицы перестановки.

«Последовательное» объединение, как было отмечено выше, применяется в случае $l_f = n^2$, где l_f – разрядность f-функции, n – разрядность S-блока, при этом количество аргументов линейного преобразования $m = n$. В этом случае, рекомендуется использовать линейное преобразование 3-го типа, и разместить его между S-P-S цепочками (рис. 3).

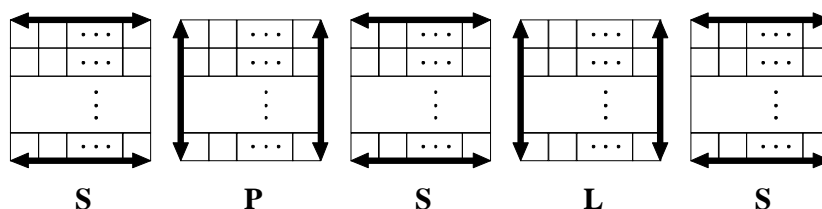


Рис. 3. 2-х мерная «последовательная» SPLN-структура

Рассмотренная конструкция «последовательного» объединения не достигает максимального показателя «размножения», возможного при использовании только «слоёв» S и L, однако, благодаря использованию «P-слоя» она обеспечивает свойство «рассеивания».

Вывод

Рассмотренная общая конструкция многоразрядного нелинейного преобразования на основе SPLN-структуры позволяет строить криптографически «сильные» цикловые функции, обеспечивающие свойства «рассеивания» и «размножения» активизации. При этом показатели указанных свойств не зависят от используемого циклового ключа, а предложенные схемы перестановки, линейного комбинирования и способа их объединения позволяют максимизировать соответствующие показатели.

Список литературы: 1. Шеннон К. «Работы по теории информации и кибернетике». Перевод В.Ф. Писаренко. М., ИЛ, 1963, С. 333–369. 2. J. Daemen, V. Rijmen. AES Proposal: Rijndael, 1999. 3. «Supporting Document on E2», Nippon Telegraph and Telephone Corporation, June 14, 1998. 4. E. Biham, A. Shamir, «Differential Cryptanalysis of the Data Encryption Standard», Springer-Verlag, New York, 1993. 5. M. Matsui, «Linear cryptanalysis method for DES cipher», Advances in Cryptology – EUROCRYPT '93 (LNCS 765), 386–397, 1994.

Поступила в редколлегию 00.00.00

Головашич Сергей Александрович, старший преподаватель кафедры БИТ ХНУРЭ.

Научные интересы: криптография, системы комплексной защиты информации. Конт. тел. 40-94-25.