ЛЕКЦИЯ 6 ПОТОЧНЫЕ ШИФРЫ

В блочном шифре из двух одинаковых блоков открытого текста получаются одинаковые блоки шифрованного текста. Избежать этого позволяют поточные шифры, в которых шифрующее преобразование "элемента" открытого текста меняется от одного элемента к другому. Так в DES в режиме сцепления блоков фактически происходит преобразование блочного шифра в поточный, что облегчает обнаружение искажений блоков и затрудняет попытки имитации и подмены. Специалисты, однако, используют термин ***поточный шифр*** только в том случае, когда "элементы" открытого текста очень малы (одна буква или один бит). Обычно аппаратные реализации поточных шифров быстрее и проще, чем блоковых. Потоковые шифры пригодны для шифрования непрерывных потоков данных, например, в сетях передачи данных.

Самые популярные сейчас поточные шифры можно назвать двоичными аддитивными. В таких шифрах k-битовый секретный ключ $Z^{'}$ используется только для управления генератором ключевого потока, порождающего двоичную последовательность $z_0, z_1, ..., z_{N-1}$, называемую ключевым потоком, где $N >> k$. Шифртекст образуется путем сложения по модулю 2 битов открытого текста и битов ключевого потока:

Шифрование $Y_i = X_i \oplus z_i$, $i = 0,1,..., N-1$

Дешифрование $X_i = Y_i \oplus z_i$, $i = 0,1,..., N-1$

Шифрование и дешифрование выполняются одинаковыми устройствами. Аддитивный поточный шифр похож на двоичный шифр с ключом однократного применения. Если $z_i = Z_i^{'}$, $i = 0,1,..., k$, $k = N$, т. е. секретный ключ используется как ключевая последовательность, то аддитивный поточный шифр есть шифр с ключом однократного применения.

В практических поточных шифрах длина N шифртекста много больше длины k секретного ключа, а ключевая последовательность является *псевдослучайной* и имеет некоторый период. Стойкость системы целиком зависит от внутренней структуры генератора ключевой последовательности. Если генератор выдает последовательность с небольшим периодом, то стойкость системы будет невелика. Если бы генератор выдавал бесконечную последовательность битов, в которой каждый бит порождался независимо и с вероятностью 1/2 принимал значения 0 или 1, то мы получили бы совершенно стойкий шифр.

Генераторы ключевых последовательностей как правило представляют собой ***регистры сдвига с обратной связью***. Простейшим видом генератора на основе регистра сдвига с обратной связью является ***регистр с линейной обратной связью***. Общая схема генератора псевдослучайной ключевой последовательности на основе регистра с линейной обратной связью показана на Рис. 1

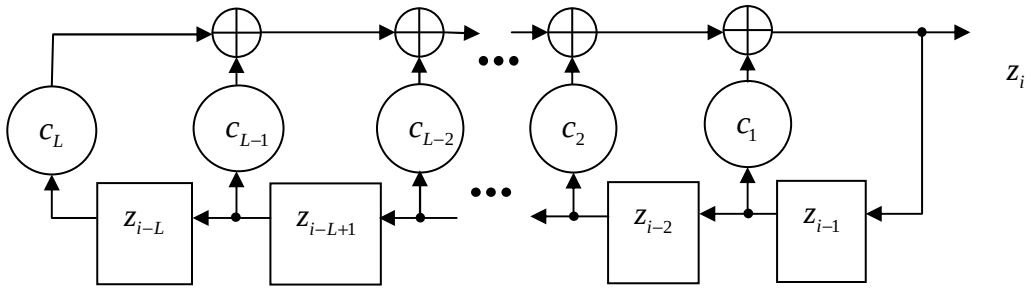

Рис. 1 Генератор ключевой последовательности на основе регистра сдвига с обратной связью

В нашем примере регистр сдвига имеет длину L. Обратные связи соответствуют коэффициентам полинома $c(x) = c_L x^L + c_{L-1} x^{L-1} + ... + c_1 x + 1,\ c_i \in \{0,1\}$, $i = 1,...,L,\ c_0 = 1,\ 0$ соответствует отсутствию связи, а 1 – ее наличию. Все операции выполняются по модулю 2. Говорят, что двоичная последовательность $\mathbf{z}$ порождается полиномом $c(x)$, если

$$\sum_{j=0}^{L} c_j z_{i-j} = 0 \text{ или } z_i = -\sum_{j=1}^{L} c_j z_{i-j} \quad (1).$$

Заметим, что так как все операции выполняются по модулю 2, то (1) эквивалентно $z_i = \sum_{j=1}^{L} c_j z_{i-j}$.

Последовательность $\mathbf{z}$ называется ***линейной рекуррентной последовательностью***, а уравнение (1) называется разностным уравнение. Если вся последовательность $\mathbf{z} = z_0, z_1, z_2, ...$ может быть порождена регистром длины L и не может быть порождена более коротким регистром (разностным уравнением более низкого порядка), то говорят, что ***линейная сложность последовательности*** $LC(\mathbf{z})$ равна L.

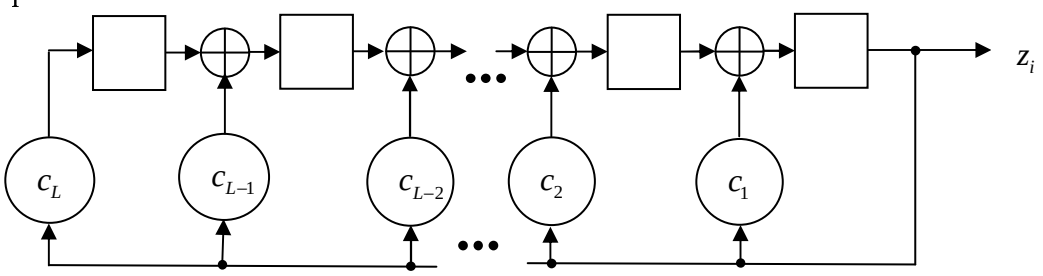

Рис. 2 Эквивалентная форма генератора ключевой последовательности.

Заметим, что уравнение (1) имеет бесконечное множество решений. Для получения единственного решения необходимо задать L начальных условий (начальное состояние регистра сдвига).

Часто используемое в литературе эквивалентное представление регистра с обратной связью, показанного на Рис.1 имеет вид, представленный на Рис. 2. Второе представление генератора следует из рассмотрения производящей функции последовательности $\mathbf{z}$, она определяется как

$$Z(x) = \sum_{k=0}^{\infty} z_k x^k ,$$

где x - действительная переменная.

Тогда в терминах производящих функций уравнение (1) можно записать в виде

$$Z(x) = \sum_{j=1}^{L} c_j Z(x) x^j , \qquad (2)$$

при записи (2) использованы свойство линейности и сдвига производящей функции (см., например, Э. Рейнгольд, Ю. Нивергельт и Н. Део Комбинаторные алгоритмы. Теория и практика, М., Мир, 1980 г.). Уравнение (2) можно переписать

$$Z(x) + (c_1 Z(x))x + (c_2 Z(x))x^2 + \ldots + (Z(x)c_L)x^L = 0$$

или

$$((\ldots(c_L Z(x))x + (c_{L-1} Z(x))x + (Z(x)c_{L-2}))x + \ldots + (Z(x)c_1)x + Z(x) = 0 \quad (3)$$

Принимая во внимание, что в области производящих функций умножение на x^j соответствует задержке исходной последовательности на j символов, (3) может быть реализовано с помощью схемы, представленной на Рис. 2 . Заметим, что уравнения (2) и (3) записаны в предположении нулевого начального состояния регистра сдвига. Для получения нетривиального решения (2) (при ненулевом начальном условии) следует переписать (2) в виде

$$Z(x) = \sum_{j=1}^{L} c_j Z(x) x^j + \Delta(x),$$

где $\Delta(x)$ -некоторая функция, определяемая начальным состоянием регистра. Однако, очевидно, что структура генератора ПСП (Рис. 1 и Рис. 2) не зависит от начального состояния регистра сдвига.

Порождаемая таким генератором последовательность будет, очевидно, периодической, причем величина периода совпадает с числом различных

состояний регистра сдвига. Регистр сдвига длины L может находиться в 2^L-1 ненулевом состоянии, определяемом заполнением L ячеек этого регистра. Это означает, что максимальный возможный период ключевой последовательности, получаемой на выходе такого регистра с обратной связью равен 2^L-1. Однако, если обратные связи заданы произвольным полиномом, то число различных реализующихся состояний, а значит и период выходной последовательности, может оказаться меньше 2^L-1. Для получения ключевой последовательности с максимальным периодом необходимо, чтобы полином $c(x)$, определяющий обратные связи, удовлетворял определенным требованиям, а именно был ***примитивным полиномом*** над полем $GF(2^L)$. Псевдослучайные последовательности с максимальным периодом, получаемые на выходе регистра сдвига с обратными связями, задаваемыми примитивным полиномом, называются ***последовательностями максимальной длины*** или M-последовательностями.

Пример 1. Простейшее разностное уравнение имеет вид

$$z_i + c_1 z_{i-1} = 0 .$$

Оно нетривиально только при $c_1 = 1$. При начальном состоянии $z_1 = 1$ оно порождает последовательность (1,1,…,1). Сложность ее равна 1, ей соответствует полином $c(x) = 1 + x$.

Пример 2. Разностное уравнение

$$z_i = z_{i-n} .$$

Соответствующий полином имеет вид $c(x) = 1 + x^n$. При начальном состоянии $(0,0,...,1)$ это уравнение порождает последовательность, представляющую собой периодическое повторение начального состояния $(0,0,...,1)$. Сложность ее равна n и совпадает с периодом порождаемой последовательности.

Пример 3. При любом ненулевом начальном состоянии уравнение

$$z_i = z_{i-1} + z_{i-3}$$

Порождает последовательность с периодом 7. Соответствующий полином равен $c(x) = x^3 + x + 1$ и представляет собой примитивный полином над $GF(2^3)$. Период 7 – это максимально возможный период для последовательностей задаваемых полиномом третьей степени $(2^3 - 1 = 7)$ и соответствующая последовательность – это M-последовательность. Схема генератора приведена на Рис. 3

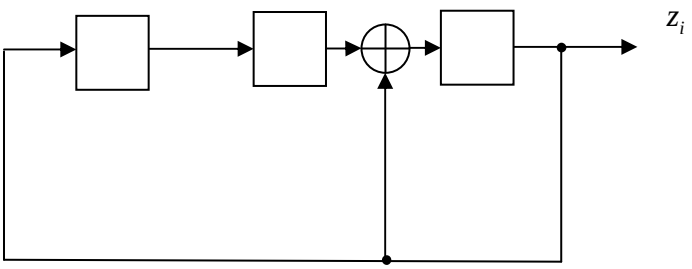

Рис. 3 Генератор M -последовательности

Выберем произвольное начальное состояние регистра 001. Состояния ячеек регистра и соответствующие им элементы М-последовательности приведены в таблице

Номер состояния	Номер ячейки регистра			М-послед.
	1	2	3	
1	0	0	1	1
2	1	0	1	1
3	1	1	1	1
4	1	1	0	0
5	0	1	1	1
6	1	0	0	0
7	0	1	0	0
8	0	0	1	

Последовательности максимальной длины обладают следующими свойствами:

1. Период последовательности равен $2^L - 1$
2. Число 1 в последовательности 2^{L-1} (т.к. состояния регистра – это все возможные последовательности длины n (кроме нулевой), то суммарное число раз когда последним символом будет 1 равно $2^L / 2 = 2^{L-1}$.
3. Каждое возможное состояние или L-разрядная комбинация за время формирования полного периода возникает в некоторый момент времени

только один раз. Исключением является комбинация из всех нулей, в нормальном режиме работы она не возникает и не должна возникать.

4. Множество начальных состояний порождает множество различных последовательностей длины $2^L - 1$ (все они являются $2^L - 1$ различными сдвигами М-последовательности).

Начальное состояние можно использовать в качестве ключа. Если ключ неправильный, то последовательность отличается от правильной в половине символов. Однако, такое шифрование считается плохим. Доказано, что по $2L$ выходным битам генератора можно восстановить примитивный полином, задающий обратные связи. Пусть $L = 56$, период последовательности равен $2^{56} - 1$, но восстановить примитивный полином можно по $2L = 112$ битам выходной последовательности.

Рассмотрим как можно восстановить полином, задающий обратные связи по $2L$ битам М-последовательности.

Докажем следующую теорему:

Теорема. Пусть $LC(\mathbf{z}) = L$. Рассмотрим регистр сдвига длины L с линейной обратной связью, порождающий последовательность $\mathbf{z}$ длины N, где N может быть и бесконечным. Тогда

- L последовательных состояний регистра – линейно независимы
- $L+1$ последовательное состояние регистра – линейно зависимо
- Если $N \geq 2L$ символов последовательности заданы, то полином, задающий обратные связи – однозначно определен.

Доказательство.

Пусть, $c_1, \ldots, c_L$ - коэффициенты полинома, задающего обратные связи. Символы последовательности могут быть записаны в виде

$$z_k = c_1 z_{k-1} + \ldots + c_{L-1} z_{k-L+1} + c_L z_{k-L}. \quad (4)$$

Обозначим через $\mathbf{Z}$ следующую матрицу размера $L \times L$ из символов выходной последовательности $\mathbf{Z} = \begin{pmatrix} z_1 & z_2 & \ldots & \ldots & z_L \\ z_2 & z_3 & \ldots & \ldots & z_{L+1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ z_L & z_{L+1} & \ldots & \ldots & z_{2L-1} \end{pmatrix}$. Тогда (4) может быть переписано в матричной форме

$$\mathbf{Z}\begin{pmatrix} c_L \\ c_{L-1} \\ \dots \\ c_1 \end{pmatrix} = \begin{pmatrix} z_{L+1} \\ z_{L+2} \\ \dots \\ z_{2L} \end{pmatrix} \qquad (5)$$

Из (5) следует, что линейная комбинация из L строк матрицы $\mathbf{Z}$ (последовательных состояний регистра сдвига) равна $L+1$му состоянию регистра, т.е. $L+1$ последовательных состояний – линейно зависимы. Если L последовательных состояний – линейно зависимы, то рекурсия (4) выполняется для $L-1$ элемента последовательности, а значит последовательность может быть получена на выходе регистра сдвига длины меньше, чем L с линейной обратной связью. Если по крайней мере $2L$ элементов последовательности – известны, то матрица $\mathbf{Z}$ - известна. Так как строки матрицы $\mathbf{Z}$ - линейно независимы (в силу линейной независимости L последовательных состояний), то определитель матрицы $\mathbf{Z}$ не равен нулю и матрица обратима, что дает единственное решение $(c_1, c_2, ..., c_L)$ системы (5). Если гипотеза о линейной сложности L верна, то существует единственное решение этой системы уравнений, оно и является решением этой задачи.

Тот факт, что на практике сложность L не известна заранее, не намного усложняет задачу, так как можно по очереди проверять гипотезы $L = 1,2,....$ Решение системы из L уравнений имеет сложность не более L^3, поэтому общая сложность нахождения регистра не превышает L^4, т.е. остается в любом случае полиномиальной.

Пример 4. Рассмотрим последовательность $z_1, z_2, ... =$ 01011110001001101011111…. Найдем ее линейную сложность. Если бы последовательность состояла из одних нулей, то мы бы заключили, что $L = 0$. В случае последовательности из всех единиц, $L = 1$, $c_1 = 1$, а уравнение (1) имеет вид $z_i + c_1 z_{i-1} = 0$. Так как ни та, ни другая ситуация не имеют место, проверяем гипотезу о том, что $L = 2$. Система (3) имеет вид

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} c_2 \\ c_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Ее единственное решение $c_1 = 0$, $c_2 = 1$. Посмотрим, порождает ли этот фильтр всю последовательность

$$c_1 z_2 + c_2 z_1 = c_1 1 + c_2 0 = 0 = z_3$$

$$c_1 z_3 + c_2 z_2 = c_1 0 + c_2 1 = 1 = z_4$$

$$c_1 z_4 + c_2 z_3 = c_1 1 + c_2 0 = 0 \neq z_5 = 1$$

На пятом символе получили несовпадение, следовательно, гипотеза не верна. Положим $L=3$.

$$\begin{pmatrix} c_3 \\ c_2 \\ c_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Решение $(c_1 \quad c_2 \quad c_3) = (0 \quad 1 \quad 1)$. Теперь

$$c_1 z_3 + c_2 z_2 + c_3 z_1 = c_1 0 + c_2 1 + c_3 0 = 1 = z_4$$

$$c_1 z_4 + c_2 z_3 + c_3 z_2 = c_1 1 + c_2 0 + c_3 1 = 1 = z_5$$

$$c_1 z_5 + c_2 z_4 + c_3 z_3 = c_1 1 + c_2 1 + c_3 0 = 1 = z_6$$

$$c_1 z_6 + c_2 z_5 + c_3 z_4 = c_1 1 + c_2 1 + c_3 1 = 0 \neq z_7 = 1$$

Еще одна попытка оказалась неудачной. Попробуем $L=4$.

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} c_4 \\ c_3 \\ c_2 \\ c_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

Решение имеет вид: $(c_1, c_2, c_3, c_4) = (0,0,1,1)$. Предоставляем читателю возможность самостоятельно проверить, что вся последовательность порождается этим фильтром.

Описываемый ниже алгоритм Берлекэмпа-Месси решает совместно обе задачи – определение линейной сложности и нахождение коэффициентов полинома со сложностью порядка L^2. Это упрощение возможно благодаря тому, что матрица коэффициентов имеет специфический вид: строки коэффициентов являются сдвигами предыдущих строк. Такие матрицы называют теплицевыми.

Необходимо найти регистр сдвига с обратной связью наименьшей длины L, генерирующий заданную последовательность $\mathbf{z} = (z_1, z_2, ..., z_n)$ при начальном состоянии $(z_1, z_2, ..., z_L)$, $L < n$. Процедура является рекуррентной. Для каждого

r, начиная с $r=1$, строим регистр, генерирующий первые r элементов последовательности $(z_1, z_2, ..., z_n)$. Обозначим длину построенного регистра через L_r, а сам регистр с обратной связью опишем вектором коэффициентов

$\mathbf{c^{(r)}} = (c_1^{(r)}, c_2^{(r)}, ..., c_{L_r}^{(r)})$. На r-й итерации вычисляем r-й выход предыдущего $(r-1)$-го регистра

$$\hat{z}_r = \sum_{j=1}^{r-1} c_j^{(r-1)} z_{r-j}.$$

На самом деле степень полинома $\mathbf{c}^{(r-1)}$ может быть меньше, чем $r-1$, но, чтобы упростить запись, мы игнорируем тот факт, что многие слагаемые в последней сумме тождественно равны нулю.

Истинный элемент последовательности z_r может не совпадать с $\hat{z}_r$. "Невязку" запишем в виде

$$\Delta_r = z_r + \hat{z}_r = z_r + \sum_{j=1}^{r-1} c_j^{(r-1)} z_{r-j} = \sum_{j=0}^{r-1} c_j^{(r-1)} z_{r-j},$$

где подразумевается, что $c_0^{(r-1)} = 1$. Если невязка - нулевая, то итерация закончена. В противном случае, нужно модифицировать вектор, задающий регистр, чтобы сделать ее нулевой. Новый вектор будем искать в виде ,

$$\mathbf{c^{(r)}} = \mathbf{c^{(r-1)}} + x^l \mathbf{c^{(m-1)}}, \quad (4)$$

где $\mathbf{c^{(m-1)}}$ - это один из уже использовавшихся регистров, а $m < r$ таково, что невязка на m-м шаге была ненулевой. Если к тому же положить $l = r - m$, то после такой модификации

$$\Delta_r' = \sum_{j=0}^{r-1} c_j^{(r-1)} z_{r-j} + \sum_{j=0}^{r-1} c_j^{(m-1)} z_{r-j-l} = \Delta_r + \Delta_m = 1+1 = 0.$$

Осталась некоторая свобода в выборе m. Нужно выбрать его так, чтобы минимизировать длину получаемого регистра. Оказывается, что этому условию удовлетворяет выбор последнего такого m, при котором невязка была равна 1.

Алгоритм Берлекэмпа-Месси.

Вход –последовательность $\mathbf{z}$ длины n.

1. Инициализация: $r = 0$, $c(x) = 1$, $b(x) = 1$.
2. Полагаем $r \leftarrow r+1$. Вычисляем невязку $\Delta = z_r + \sum_{j=1}^{r-1} c_j z_{r-j_r} = \sum_{j=0}^{r-1} c_j z_{r-j_r}$
3. Если $\Delta = 0$, сдвиг: $b(x) = xb(x)$, переходим к 5 в противном случае выполняем 4.
4. Формируем новый полином $t(x) \leftarrow c(x) + xb(x)$. Сохраняем предыдущий полином: $b(x) = c(x)$. Меняем связи регистра $c(x) = t(x)$.

5. Если $r < n$, возвращаемся к 2. В противном случае работа закончена, результаты работы алгоритма – полином ЛРОС $c(x)$ и линейная сложность последовательности $\mathbf{z}$ равная $L = \deg c(x)$.

Пример 5. Применим алгоритм Берлекэмпа-Месси к последовательности из примера 4. Результаты промежуточных вычислений приведены в Таблице 1.

Таблица 1. Вычисление линейной сложности и ЛРОС для последовательности 0101111000100110101111…

r	z_r	Δ	**c**	**b**
0	-	-	1	1
1	0	0	1	x
2	1	1	$1+x^2$	1
3	0	0	$1+x^2$	x
4	1	0	$1+x^2$	x^2
5	1	1	$1+x^2+x^3$	$1+x^2$
6	1	0	$1+x^2+x^3$	$x+x^3$
7	1	1	$1+x^3+x^4$	$1+x^2+x^3$
8	0	0	$1+x^3+x^4$	$(1+x^2+x^3)x$
9	0	0	$1+x^3+x^4$	$(1+x^2+x^3)x^2$
10	…	…	…	…
Результат: $L=4$, $c(x)=1+x^3+x^4$				

Несмотря на то, что шифрование на основе линейных рекуррентных последовательностей считается плохим, поточные шифры на основе линейных рекуррентных последовательностей используются на практике. Для улучшения потокового шифра в схему вводится нелинейный элемент. Генераторы ключевых последовательностей обычно базируются на комбинациях регистров сдвига и нелинейных булевых функциях.

Например, A5 – это поточный шифр, используемый для шифрования в стандарте для цифровых мобильных телефонов GSM (Group Special Mobile). A5 включает три регистра с линейной обратной связью длины 19, 22 и 23, соответственно. Выходом является сумма по модулю 2 выходов трех регистров. Нелинейность алгоритма достигается за счет того, что на каждом такте производится сдвиг некоторых (не менее двух регистров). Сдвиг зависит от значения трех битов – 10-го в первом регистре, 11-го во втором регистре и 12 бита

в третьем регистре. Если все три бита одинаковы, то сдвигаются все регистры, иначе сдвигаются два регистра, у которых управляющие биты совпадают.

Оптимизированный для программной реализации потоковый шифр RC4 основан на другом принципе. Байт ключевой последовательности, который прибавляется по модулю 2 к байту исходного текста, формируется путем выполнения некоторой, зависящей от ключа перестановки байтов состояния (состояние содержит 256 байтов) и выбора соответствующего байта.