ЛАБОРАТОРНАЯ РАБОТА №3
ПОТОКОВОЕ ШИФРОВАНИЕ

Основная идея потоковых криптосистем заключается в шифровании исходного текста M с помощью криптографического ключа K, длина которого равна длине текста. Каждый бит шифротекста C_i является функцией соответствующих битов исходного текста и ключевого потока:

$$C_i = E_{Ki}(M_i) = M_i\, K_i,\ M_i, K_i, C_i \in \{0,1\}.$$

При дешифровании выполняется обратное преобразование D_{Ki}:

$$D_{Ki}(C_i) = C_i \oplus K_i = (M_i \oplus K_i) \oplus K_i = M_i.$$

Символом «⊕» обозначена операция сложения «ИСКЛЮЧАЮЩЕЕ-ИЛИ». Благодаря линейным свойствам этой операции при шифровании и дешифровании используется одинаковый ключевой поток K. Очевидно, что в этом случае длина K должна быть равна длине передаваемого сообщения. Однако обмен ключами большого размера зачастую невозможен. Поэтому на практике для формирования ключевого потока используют генераторы псевдослучайной последовательности (рис.1). Начальные параметры I генераторов на стороне отправителя и получателя должны совпадать, они являются секретным ключом алгоритма. Псевдослучайная последовательность каждого генератора обладает определенным периодом, после которого значения повторяются. Поэтому необходимо выбирать такие генераторы ключа, чтобы этот период превышал длину шифруемой информации.

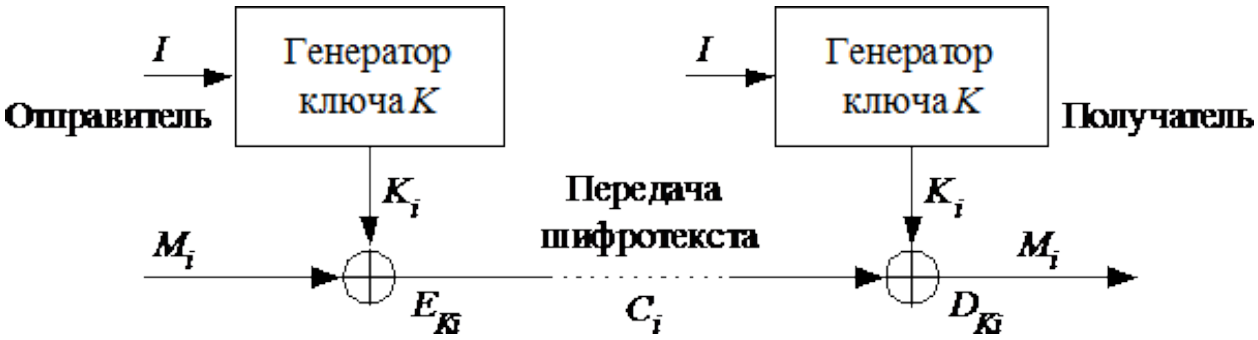

Рисунок 1 – Схема потоковой криптосистемы

Для корректной работы потоковых криптосистем необходимо, чтобы передающая и принимающая сторона имели синхронизированные генераторы ключа K. Искажение отдельных символов не влияет на расшифровку остальных символов шифротекста. Добавление, удаление или дублирование символов шифротекста нарушает синхронизацию ключевой и текстовой последовательностей, и все последующие символы расшифровываются некорректно.

Рассмотрим генераторы ключей на основе сдвиговых регистров с линейной обратной связью LFSR (Linear Feedback Shift Register). Они достаточно просто реализуются в программном и аппаратном виде, обладают высокой скоростью генерации и большим периодом ключа. Регистр LFSR состоит из двух частей: сдвигового регистра, выполняющего сдвиг своих разрядов влево на один разряд, и функции обратной связи, вычисляющей вдвигаемое в первый разряд значение.

Структуру конкретного регистра LFSR принято задавать с помощью характеристического (задающего) многочлена:

$$P(x) = a_m\, x^m + a_{m-1}\, x^{m-1} + \ldots + a_2\, x^2 + a_1\, x + 1, a_k \in \{0,1\}, k = 1 \ldots m.$$

Степень многочлена m задает длину сдвигового регистра. Ненулевые коэффициенты a_k определяют разряды регистра, которые будут участвовать в формировании вдвигаемого в первый разряд значения. Через b_k ($b_k \in \{0,1\}$) обозначены текущие значения разрядов LFSR ($k = 1, \ldots, m$). Новые значения разрядов b^*k ($b^*k \in \{0,1\}$) вычисляются по следующему закону:

$$b^*_k = \begin{cases} \sum_{j=1}^{m} \Theta a_j b_j, & k = 1 \\ b_{k-1}, & k = 2, \ldots, m \end{cases}$$

– функция обратной связи

– сдвиг

Таким образом, новое значение для всех разрядов, кроме первого, берется из ближайшего младшего разряда. Символом $\sum\Theta$ обозначена многоместная операция сложения «ИСКЛЮЧАЮЩЕЕ-ИЛИ». Она используется для сложения значений из разрядов, которые отмечены ненулевыми коэффициентами a_k характеристического многочлена. Полученная сумма вдвигается в первый разряд LFSR. Очередной бит ключа K_i для потоковой криптосистемы равен значению старшего разряда LFSR b_m.

Пример 1.

Построим сдвиговый регистр с линейными обратными связями, задаваемый характеристическим многочленом $P(x) = x^4 + x + 1$.

Схема регистра приведена на рис. 2. Если начальное состояние регистра $I = 1111$, то последовательность генерируемых состояний имеет вид:

1111 → 1110 → 1101 → 1010 → 0101 → 1011 → 0110 → 1100 →
1001 → 0010 → 0100 → 1000 → 0001 → 0011 → 0111 → 1111

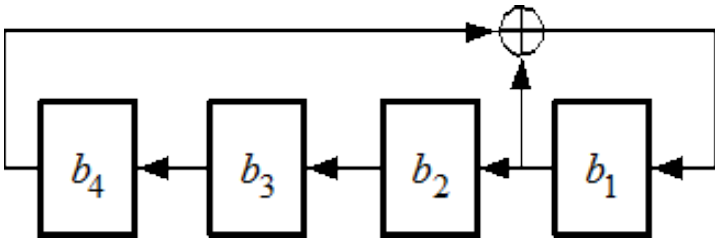

Рисунок 2 – LFSR на основе многочлена $P(x) = x^4 + x + 1$

Последнее состояние совпадает с начальным, после этого указанная последовательность повторяется. Последовательность сгенерированных битов ключа: 1111010110010001.

Свойства генерируемой последовательности определяются постоянными коэффициентами характеристического многочлена a_i. При их соответствующем выборе генерируемая последовательность K будет иметь максимально возможный период, равный $2^m - 1$. Последовательность максимально возможного для данного генератора периода называется М-последовательностью. Основная задача синтеза генератора на основе LFSR – нахождение характеристического многочлена, позволяющего сформировать М-последовательность. Для того чтобы конкретный LFSR имел максимальный период, соответствующий многочлен должен быть примитивным. В общем случае не существует простого способа нахождения примитивных многочленов данной степени, проще выбирать многочлен случайным образом и проверять, является ли он примитивным. Эта задача аналогична задаче проверки на простоту случайно выбранного целого числа и легко решается с помощью вычислительной техники. Табл. 1 содержит некоторые примитивные многочлены.

Таблица 1

Примитивные многочлены

m	$P(x)$	m	$P(x)$	m	$P(x)$
23	$x^{23} + x^5 + 1$	29	$x^{29} + x^2 + 1$	35	$x^{35} + x^2 + 1$
24	$x^{24} + x^4 + x^3 + x + 1$	30	$x^{30} + x^{16} + x^{15} + x + 1$	36	$x^{36} + x^{11} + 1$
25	$x^{25} + x^3 + 1$	31	$x^{31} + x^3 + 1$	37	$x^{37} + x^{12} + x^{10} + x^2 + 1$
26	$x^{26} + x^8 + x^7 + x + 1$	32	$x^{32} + x^{28} + x^{27} + x + 1$	38	$x^{38} + x^6 + x^5 + x + 1$
27	$x^{27} + x^8 + x^7 + x + 1$	33	$x^{33} + x^{13} + 1$	39	$x^{39} + x^4 + 1$
28	$x^{28} + x^3 + 1$	34	$x^{34} + x^{15} + x^{14} + x + 1$	40	$x^{40} + x^{21} + x^{19} + x^2 + 1$

Использование LFSR для создания потоковых криптосистем предполагает уязвимость, связанную с линейным характером генерируемой последовательности. Обладая парой «исходный текст – шифротекст» длиной всего 2*m* бит, взломщик может восстановить характеристический многочлен и расшифровать все сообщение. Для защиты от этой атаки следует увеличивать размер используемого LFSR или использовать более сложные схемы генерации ключа. Рассмотрим, для примера, генератор Геффе (Geffe), представленный на рис. 3.

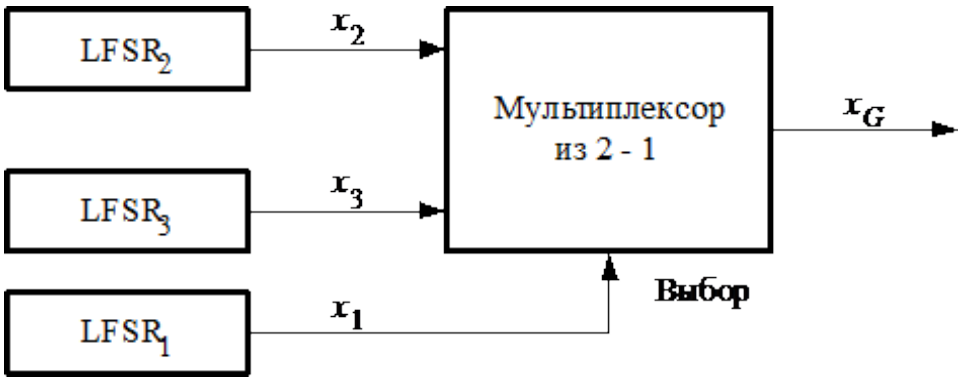

Рисунок 3 – Генератор Геффе

В нем используется три регистра с линейной обратной связью, объединённые нелинейным образом. Два регистра LFSR являются входами мультиплексора, а третий – управляет его выходом. Через x_1, x_2 и x_3 обозначены выходы трёх LFSR, выход генератора Геффе x_G описывается так: $x_G = (x_1 \ and \ x_2) \ or \ (not \ x_1 \ and \ x_3)$. Период данного генератора равен $(2^{m1} - 1)(2^{m2} - 1)(2^{m3} - 1)$, где *m*1, *m*2 и *m*3 – длины первого, второго и третьего LFSR соответственно.

Благодаря простоте реализации, высокой скорости работы и сравнительно высокой криптостойкости, потоковые шифраторы получили широкое распространение для шифрования информации средней степени секретности. Например, алгоритм A5, построенный на основе трех LFSR с прореженными обратными связями, входит в состав стандарта мобильной связи GSM.

Возможны и другие способы генерации ключевой последовательности. Например, генератор ключевого потока *K* в алгоритме RC4 работает на основе подстановочной таблицы *S* (S-бокса) из 256 символов. На первом шаге S-бокс заполняется линейно: *S*[0] = 0, *S*[1] = 1, …, *S*[255] = 255. Затем начальное значение S-бокса меняется на основе пользовательского секретного ключа *U* по следующему алгоритму:

j := 0;
for *i* := 0 **to** 255
j := (*j* + *S*[*i*] + *U*[*i* **mod** length(*U*)]) **mod** 256;
swap(*S*[*i*], *S*[*j*]);

Эта подстановка является функцией от ключа изменяемой длины. Процедура swap меняет местами значения таблицы *S* с заданными индексами.

Полученное значение S-бокса используется для побайтной генерации ключевого потока *K*. Генератор потока имеет два счетчика *i* и *j*, инициализируемых нулевым значением. На каждом шаге генерации выполняются следующие операции:

i = (*i* + 1) **mod** 256
j = (*j* + *S*[*i*]) **mod** 256
swap(*S*[*i*], *S*[*j*])
K = *S*[(*S*[*i*] + *S*[*j*]) **mod** 256]

Байт *K* складывается операцией «ИСКЛЮЧАЮЩЕЕ-ИЛИ» с байтом открытого текста для получения байта шифротекста либо с байтом шифротекста для получения байта исходного текста. Шифрование происходит весьма быстро – примерно в 10 раз быстрее DES-алгоритма. Типичная реализация выполняет 19 машинных команд на каждый байт текста. Алгоритм RC4 может принимать примерно $256! \cdot 2562 = 21700$ возможных состояний. S-бокс медленно изменяется в процессе работы: параметр *i* обеспечивает изменение каждого элемента, а *j* отвечает за то, чтобы эти элементы изменялись случайным

образом. Шифр обладает иммунитетом к методам линейного и дифференциального криптоанализа и до сих пор у него не обнаружены короткие циклы.

Алгоритм RC4, в отличие от алгоритмов на основе сдвиговых регистров LSFR, больше ориентирован на программную реализацию, поскольку работает не с битами, а с целыми байтами исходного текста. Благодаря своей скорости и защищенности, он нашел широкое применение в криптографических системах. Например, он является частью протокола безопасного обмена информацией SSL.