АНАЛИЗ БАЗОВЫХ ПРЕОБРАЗОВАНИЙ СТРУКТУРНЫХ КОМПОНЕНТОВ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ

М. В. Одинцов, А. П. Мартынов

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Для решения задач, связанных с преобразованием данных посредством шифрования и расшифрования, необходимо использовать криптографические алгоритмы, имеющие определенные характеристики и свойства. Для выявления закономерностей преобразований необходимо провести анализ криптоалгоритмов с целью выявления у них недостатков или особенностей в процессе преобразования входной информации. Наибольший интерес из известных криптоалгоритмов представляют криптоалгоритм ЛЮЦИФЕР фирмы IBM, основанный на базе простейших криптографических преобразований, которые сами по себе не обладают достаточной стойкостью к расшифрованию противником. Криптоалгоритм представляет собой сочетание блоков, реализующих функции подстановки (блок подстановки *S*) и перестановки (блок перестановки *P*). Поочередное применение блоков подстановки и перестановки позволяет создавать криптографические системы определенной сложности и стойкости, достаточных для различных областей применения.

Для понимания полученных в ходе анализа результатов напомним основные моменты реализации криптоалгоритма ЛЮЦИФЕР.

Блок подстановки S преобразует *n*-бит входной последовательности в *n*-бит выходной последовательности (для $n = 3$ см. рис. 1). Этот блок является основным для криптоалгоритма Люцифер, так как осуществляет нелинейные преобразования введенной информации.

Входная комбинация, поступающая на вход блока подстановки, представлена в двоичном *n*-битном коде. В блоке подстановки она поступает на вход дешифратора ДШ1, где преобразуется в позиционный $N = 2^n$ - разрядный код. Выходы дешифратора ДШ1 с помощью перемычек соединяются с входами дешифратора ДШ2, который осуществляет преобразование *N*-разрядного входного кода в двоичный *n*-битный выходной код.

Блок перестановки *P* представляет собой устройство, преобразующее *N* битов информации, поступающей на вход блока, в *N* битов зашифрованной информации. Основное отличие от блоков подстановки состоит в том, что блок перестановки только переносит информацию, не осуществляя при этом нелинейные преобразования.

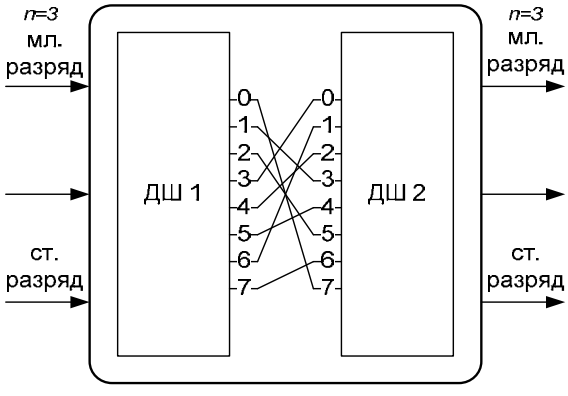
Блок перестановки для входного $N = 9$ битного кода приведен на рис. 2.

Структура алгоритма преобразования данных Люцифер представляет собой сочетание блоков, реализующих функции подстановки (блок подстановки S) и перестановки (блок перестановки *P*).

Структура криптоалгоритма (для 9-разрядной входной последовательности) приведена на рис. 3.

Входные данные при прохождении через чередующиеся слои блоков подстановки и перестановки значительно изменяются, при этом *S*-блоки обеспечивают нелинейную подстановку и выполняют «перемешивание» информации, а *P*-блоки меняют местами цифры, обеспечивая их «рассеивание».

Значительным моментом, влияющим на определение характеристик преобразования, является объем памяти, необходимый для хранения информации, полученной в ходе преобразования.



Преобразование в *S* – блоке

Вход		Выход	
10 с.с	8 с.с	8с.с	10 с.с
0	000	111	7
1	001	011	3
2	010	101	5
3	011	000	0
4	100	010	2
5	101	100	4
6	110	001	1
7	111	110	6

Рис. 1. Блок подстановки S с разрядностью $n = 3$

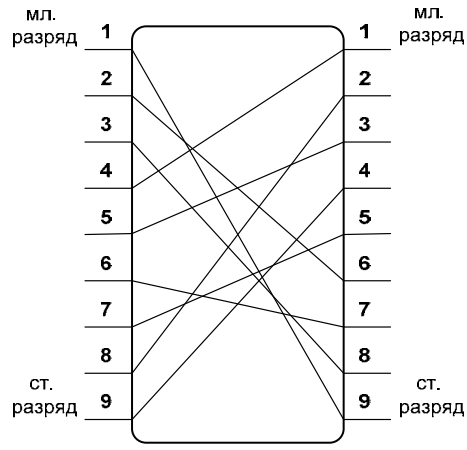


Рис. 2. Блок подстановки *P* с разрядностью *N = 9*

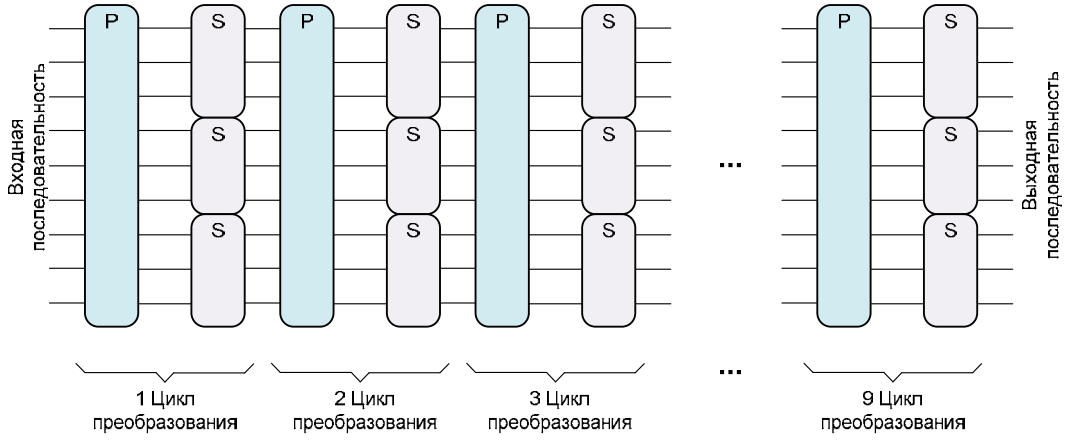


Рис. 3. Структура криптоалгоритма Люцифер с входной 9-разрядной последовательностью

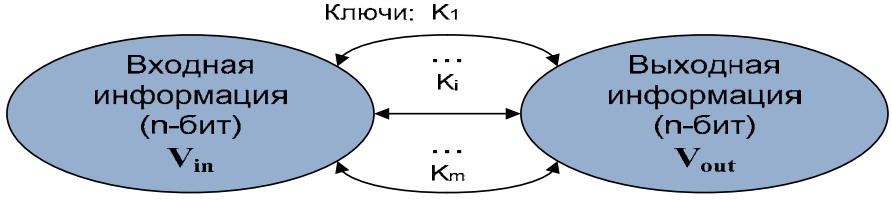


Рис. 4. Преобразование массива входной информации в массив выходной информации

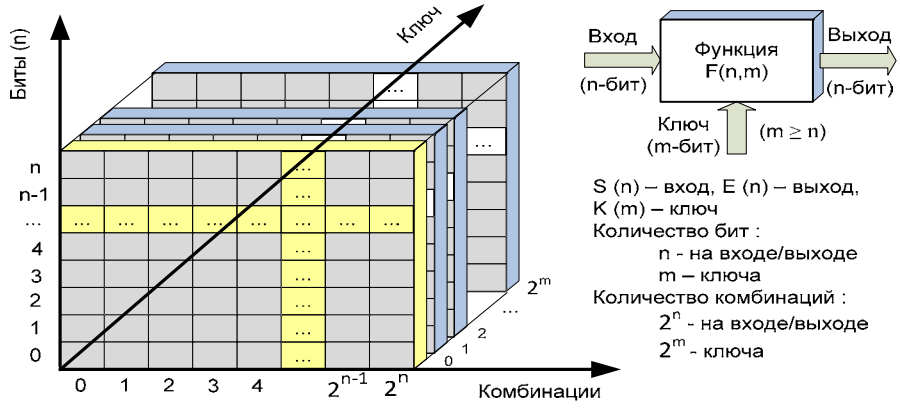


Рис. 5. Графическое представление функции криптографического преобразования и объема памяти, необходимого для хранения входной (выходной) информации

Для определения объема памяти, необходимого для хранения информации, рассмотрим некоторую криптографическую функцию, осуществляющую блочное преобразование информации. Для таких функций преобразование информации осуществляется внутри некоторого массива (рис. 4), в котором объем блока входной информации V_{in} равен объему блока выходной информации V_{out} ($V_{in} = V_{out}$).

Разрядность ключа (m), в соответствии с которым производится преобразование информации, для криптографических функций должен быть не меньше разрядности входной информации (n), т. е. $m \geq n$.

Графическое представление функции криптографического преобразования и объема памяти, необходимого для хранения входной (выходной) информации приведено на рис. 5.

В общем случае объем памяти, необходимый для хранения входной информации в битах, равен

$$V_{in} = N_b \times N_n, \quad (1)$$

где N_b – количество бит входного (выходного) слова, N_n – число возможных комбинаций значений входного (выходного) слова.

На рис. 5 объем памяти входной информации V_{in} соответствует прямоугольнику размером $n \times 2^n$ (выделен желтым цветом). Он равен объему выходной информации V_{out} для одного из значений ключа ($m = 0$)

$$V_{in} = V_{out} \text{ (при } \mathrm{K}_i = \text{const}). \quad (2)$$

Объем памяти, необходимый для хранения выходной информации в битах для всех значений ключа, равен объему памяти входной информации V_{in}, умноженной на объем (количество комбинаций) V_K ключа Km

$$V_{out} = V_{in} \times V_\mathrm{K}, \quad (3)$$

где $V_\mathrm{K} = 2^m$ – объем ключа разрядностью m-бит.

На ЭВМ информацию удобней хранить, обрабатывать и передавать не в битах, а в байтах. Рассмотрим варианты, когда входная (выходная) информация кратна байту. Объем ключа примем равным объему входной (выходной) информации ($n = m$).

На рис. 6 приведено графическое представление объема памяти, необходимого для хранения входной (выходной) информации кратной 1 байту (1, 2, 3 и 4 байта).

В общем случае объем памяти, необходимый для хранения входной информации в байтах, равен

$$V_{in\mathrm{B}} = N_\mathrm{B} \; N_n, \quad (4)$$

где N_B – количество байт входного (выходного) слова, N_n - число возможных комбинаций значений входного (выходного) слова.

Объем памяти, необходимый для хранения входной информации в байтах равен объему выходной информации $V_{out\mathrm{B}}$ для одного из значений ключа

$$V_{in\mathrm{B}} = V_{out\mathrm{B}} \text{ (при } \mathrm{K}_i = \text{const}). \quad (5)$$

Объем памяти, необходимый для хранения выходной информации в байтах для всех значений ключа, равен

$$V_{out\mathrm{B}}(\mathrm{K}) = V_{in\mathrm{B}} \; V_\mathrm{K}, \quad (6)$$

где $V_\mathrm{K} = 2^m$ – объем ключа разрядностью m-бит.

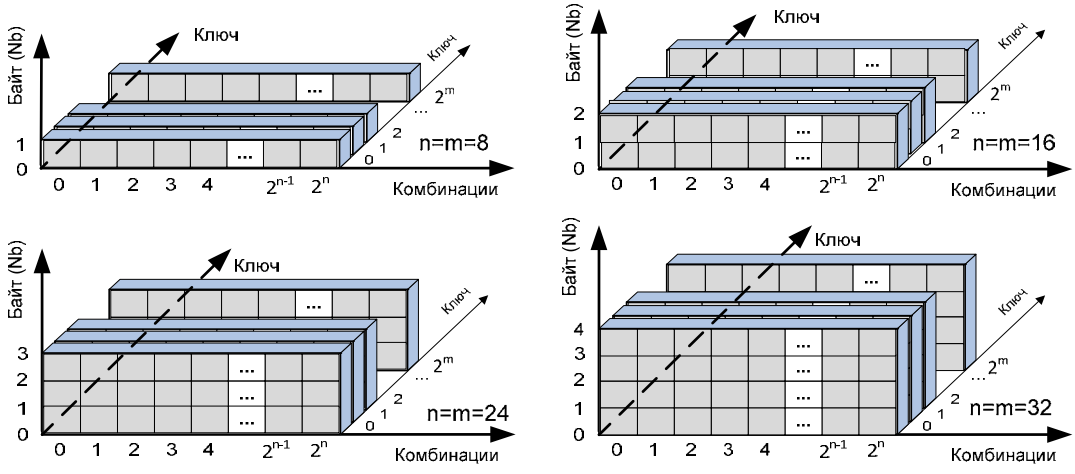


Рис. 6. Графическое представление объема памяти, необходимого для хранения входной (выходной) информации кратной 1 байту (1, 2, 3 и 4 байта)

Выводы

1. Из анализа результатов расчета видно, что количество комбинаций, а следовательно, и объем памяти, необходимый для ее хранения информации, растут очень большими темпами и, как следствие, использование обычных (не специализированных) компьютеров для решения этих задач становится невозможным.

2. Разработку методов криптоанализа блочных шифров необходимо начинать с массивов минимального объема и двигаться в сторону постепенного увеличения разрядности входной информации.

3. Если адрес выходной последовательности будет совпадать со значением входной последовательности, то они будут жестко взаимосвязаны. В этом случае объем памяти, необходимый для хранения информации, можно сократить в 2 раза, так как входную последовательность можно не запоминать (значение входной последовательности равно адресу выходной).

Литература

1. Мартынов А. П., Фомченко В. Н. Криптография и электроника / Под ред. А. И. Астайкина. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2006.

2. Мартынов А. П. Примеры простейших криптографических систем. Саров: СарФТИ, 1998.

3. Рыжиков Ю. И. Программирование на Фортране. М.: Финансы и статистика, 2001.