УДК 004.056.53

# АЛГОРИТМ ШИФРОВАНИЯ НА ОСНОВЕ СТАНДАРТА ГОСТ 28147-89

**Василенко В.А., Старобинская Л.П., Пухов С.Н.**
Севастопольский национальный технический университет
Студгородок, г. Севастополь, Украина, 99053
*E-mail: isd@bios.iuf.net*

*Предлагается один из вариантов модификации криптографической защиты информации на основе стандарта ГОСТ 28147-89.*

Известно три базовых алгоритма (блока) при работе с шифром, соответствующим ГОСТу 28147-89[1]: цикл зашифровывания (32-З); цикл расшифровывания (32-Р); цикл выработки имитовставки (16-З).

*В свою очередь, каждый из базовых циклов представляет собой многократное повторение одной единственной процедуры, называемой далее основным шагом.*

Основные характеристики ключевых структур стандартной системы обработки информации:

1) ***ключ*** является массивом из восьми 32-битных элементов кода, он обозначается символом К, размер ключа составляет $32 \cdot 8 = 256$ битов или 32 байта;

2) ***таблица замен*** является матрицей 8×16, содержащей 4-битовые элементы, которые можно представить в виде целых чисел от 0 до 15; строки таблицы замен называются узлами замен, они должны содержать различные значения, то есть каждый узел замен должен содержать 16 различных чисел от 0 до 15 в произвольном порядке; таблица замен обозначается символом Н; общий объем таблицы замен равен 512 битов или 64 байта.

Был предложен следующий алгоритм зашифровывания.

1. ***Сложение с ключом.*** Младшая половина преобразуемого блока складывается по модулю $2^{32}$ с используемым на шаге элементом ключа, результат передается на следующий шаг.

2. ***Поблочная замена.*** 32-битовое значение, полученное на предыдущем шаге, интерпретируется как массив из восьми 4-битовых блоков кода: $S = (S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7)$.

Далее значение каждого из восьми блоков заменяется на новое, которое выбирается по таблице замен следующим образом: значение блока $S_i$ заменяется на $S_i$-тый по порядку элемент (нумерация с нуля) $i$-того узла

замен (т.е. *i*-той строки таблицы замен, нумерация также с нуля). Другими словами, в качестве замены для значения блока выбирается элемент из таблицы замен с номером строки, равным номеру заменяемого блока, и номером столбца, равным значению заменяемого блока как 4-битового целого неотрицательного числа.

3. ***Циклический сдвиг на 11 бит влево.*** Результат предыдущего шага сдвигается циклически на 11 бит в сторону старших разрядов и передается на следующий шаг.

4. ***Побитовое сложение.*** Значение, полученное на шаге 3, побитно складывается по модулю 2 со старшей половиной преобразуемого блока.

5. ***Сдвиг по цепочке.*** Младшая часть преобразуемого блока сдвигается на место старшей, а на ее место помещается результат выполнения предыдущего шага.

Полученное значение преобразуемого блока возвращается как результат выполнения алгоритма основного шага криптопреобразования.

Алгоритм расшифровывания является обратным процессом алгоритму зашифровывания. Следовательно, чтобы расшифровать блок, подвергнутый процедуре основного шага криптопреобразования, надо выполнить те же процедуры, но в обратном порядке.

Базовые циклы построены из основных шагов криптографического преобразования, рассмотренных выше. В процессе выполнения основного шага используется только один из сегментов ключа, которые выбираются из восьми сегментов, предусмотренных ГОСТом. Базовые циклы заключаются в многократном выполнении основного шага с использованием разных сегментов ключа и отличаются друг от друга только числом повторений шагов и порядком использования ключевых сегментов. Ниже приведен этот порядок для различных циклов.

1. *Цикл зашифровывания 32-З:*

$K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7,K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7,K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7,$
$K_7,K_6,K_5,K_4,K_3,K_2,K_1,K_0.$

2. *Цикл расшифровывания 32-Р:*

$K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7,K_7,K_6,K_5,K_4,K_3,K_2,K_1,K_0,K_7,K_6,K_5,K_4,K_3,K_2,K_1,K_0,$
$K_7,K_6,K_5,K_4,K_3,K_2,K_1,K_0.$

3. *Цикл выработки имитовставки 16-З:*

$K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7,K_0,K_1,K_2,K_3,K_4,K_5,K_6,K_7.$

Каждый из циклов имеет собственное буквенно-цифровое обозначение, соответствующее шаблону «*n*-*X*», где первый элемент обозначения (*n*), задает число повторений основного шага в цикле, а второй элемент обозначения (X), буква, задает порядок зашифровывания («З») или расшифровывания («Р») в использовании ключевых элементов. Цикл расшифровывания должен быть обратным циклу зашифровывания, т.е. последовательное применение этих двух циклов к произвольному

блоку должно дать в итоге исходный блок. Для выполнения этого условия для алгоритмов, подобных упомянутому ранее ГОСТу, необходимо и достаточно, чтобы порядок использования ключевых элементов соответствующими циклами был взаимно обратным.

Цикл выработки имитовставки вдвое короче циклов шифрования, порядок использования ключевых элементов в нем такой же, как в первых 16 шагах цикла зашифровывания. Между циклами шифрования и вычисления имитовставки есть еще одно отличие, не упомянутое выше: в конце базовых циклов шифрования старшая и младшая часть блока результата меняются местами, это необходимо для их взаимной обратимости.

Стандарт предусматривает три основных режима шифрования данных:

простая замена, гаммирование, гаммирование с обратной связью, и один дополнительный

режим выработки имитовставки.

Зашифровывание в *режиме простой замены* заключается в применении цикла 32-З к блокам открытых данных, расшифровывание – цикла 32-Р к блокам зашифрованных данных. Это наиболее простой из режимов, так как 64-битовые блоки данных обрабатываются в нем независимо друг от друга.

Так как блоки данных шифруются независимо друг от друга и от их позиции в массиве, при зашифровывании двух одинаковых блоков открытого текста получаются одинаковые блоки шифротекста и наоборот. Отмеченное свойство позволит криптоаналитику сделать заключение о тождественности блоков исходных данных, если в массиве зашифрованных данных ему встретились идентичные блоки, что является недопустимым для серьезного шифра.

Если длина шифруемого массива данных не кратна 8 байтам или 64 битам, возникает проблема, чем и как дополнять последний неполный блок данных массива до полных 64 битов.

*Гаммирование* – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Предлагается гамму получать следующим образом: с помощью некоторого алгоритмического рекуррентного генератора последовательности чисел (РГПЧ) вырабатываются 64-битные блоки данных, которые далее подвергаются преобразованию по циклу 32-З, т.е. зашифровыванию в режиме простой замены, в результате получаются блоки гаммы. Благодаря тому, что наложение и снятие гаммы осуществляется при помощи одной и той же операции побитового исключающего ИЛИ, алгоритмы зашифровывания и расшифровывания в режиме гаммирования идентичны.

Рекуррентный генератор псевдослучайных чисел, используемый для выработки гаммы, описывается рекуррентной функцией:

$$W_{i+1} = f(W_i),$$

где $\Omega_i$ – элементы рекуррентной последовательности; $f$ – функция преобразования. Следовательно, неизбежно возникает вопрос о его инициализации элемента $\Omega_0$. Этот элемент данных является параметром алгоритма для режимов гаммирования, и называется в криптографии синхропосылкой, а в нашем ГОСТе – начальным заполнением одного из регистров шифрователя. По определенным соображениям разработчики ГОСТа решили использовать для инициализации РГПЧ не непосредственно синхропосылку, а результат ее преобразования по циклу 32-З. С учетом преобразования по алгоритму простой замены добавляется еще и зависимость от ключа.

Таким образом, последовательность элементов гаммы для использования в режиме гаммирования однозначно определяется ключевыми данными и синхропосылкой. Естественно, для обратимости процедуры шифрования в процессах за- и расшифровывания должна использоваться одна и та же синхропосылка.

Режим генерации псевдослучайных чисел имеет следующие характеристики:

1) в 64-битовом блоке старшая и младшая части обрабатываются независимо друг от друга, т.е.:

$$\Omega_i = (\Omega_i^0, \Omega_i^1), |\Omega_i^0| = |\Omega_i^1| = 32, \Omega_{i+1}^0 = f(\Omega_i^0), \Omega_{i+1}^1 = \tilde{f}(\Omega_i^1),$$

фактически, существуют два независимых РГПЧ для старшей и младшей частей блока;

2) рекуррентные соотношения для старшей и младшей частей следующие:

$$\Omega_{i+1}^0 = (\Omega_i^0 + C_1) \bmod 2^{32}, \quad \text{где } C_1 = 1010101_{16};$$

$$\Omega_{i+1}^1 = (\Omega_i^1 + C_2 - 1) \bmod (2^{32} - 1) + 1, \quad \text{где } C_2 = 1010104_{16}.$$

*Нижний индекс в записи числа означает его систему счисления, таким образом, константы, используемые на данном шаге, записаны в 16-ричной системе счисления.*

Период повторения последовательности для младшей части составляет $2^{32}$, для старшей части $2^{32}–1$, для всей последовательности период составляет $2^{32}\cdot(2^{32}–1)$.

Данный режим очень похож на режим гаммирования и отличается от него только способом выработки элементов гаммы – очередной элемент гаммы вырабатывается как результат преобразования по циклу 32-З предыдущего блока зашифрованных данных, а для зашифровывания первого блока массива данных элемент гаммы вырабатывается как результат преобразования по тому же циклу синхропосылки. Этим достигается зацепление блоков – каждый блок шифротекста в этом режиме зависит от соответствующего и всех предыдущих блоков открытого текста. На стойкость шифра факт зацепления блоков не оказывает никакого влияния.

Данный алгоритм шифрования является весьма криптостойким и позволяет создавать системы шифрования в реальном времени при программной реализации.

***Библиографический список***

1. ГОСТ 28147-89. Система обработки информации. Защита криптографическая. Алгоритмы криптографического преобразования. — Взамен ГОСТ 28467-85; Введен 01.01.89. — М.: Изд-во стандартов, 1989. — 9 с.

2. Анин Б.Ю. Защита компьютерной информации / Б.Ю. Анин. — СПб.: Изд-во «БХВ – Санкт-Петербург», 2000. — 384 с.

***Поступила в редакцию 7.05.2001 г.***