АЛГОРИТМ ШИФРОВАНИЯ ПО ГОСТ28147-89 256 бит, реализованный в Key_P1 MULTICLET Цифровой страж

Источник: http://www.enlight.ru/crypto/articles/vinokurov/gost_i.htm

Описание стандарта шифрования Российской Федерации содержится в документе, озаглавленном «Алгоритм криптографического преобразования данных ГОСТ 28147-89». То, что в его названии вместо термина «шифрование» фигурирует более общее понятие «криптографическое преобразование», вовсе не случайно. Помимо нескольких тесно связанных между собой процедур шифрования, в документе описан один построенный на общих принципах с ними алгоритм выработки имитовставки. Последняя является не чем иным, как криптографической контрольной комбинацией, то есть кодом, вырабатываемым из исходных данных с использованием секретного ключа с целью имитозащиты, или защиты данных от внесения в них несанкционированных изменений.

В разработанном варианте программы ГОСТ 28147-89 взят режим гаммирования.

Гаммирование.

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, то есть последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Все элементы гаммы различны для реальных шифруемых массивов и, следовательно, результат зашифрования даже двух одинаковых блоков в одном массиве данных будет различным. Хотя элементы гаммы и вырабатываются одинаковыми порциями в 64 бита, использоваться может и часть такого блока с размером, равным размеру шифруемого блока. Гамма получается следующим образом: с помощью некоторого алгоритмического рекуррентного генератора последовательности чисел (РГПЧ) вырабатываются 64-битовые блоки данных, которые далее подвергаются зашифрованию 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 зашифрования и расшифрования в режиме гаммирования идентичны, их общая схема приведена на рисунке 1. РГПЧ,

используемый для выработки гаммы, является рекуррентной функцией $\Omega_{i+1}= f(\Omega_i)$, где Ω_i – элементы рекуррентной последовательности, f – функция преобразования. Следовательно, неизбежно возникает вопрос о его инициализации, то есть об элементе Ω_0. В действительности, этот элемент данных является параметром алгоритма и называется в криптографии синхропосылкой(S) - начальным заполнением. Для инициализации РГПЧ в ГОСТе используют не непосредственно синхропосылку, а результат ее преобразования по циклу 32-З: $\Omega_0=Ц_{32\text{-}З}(S)$. Последовательность элементов, вырабатываемых РГПЧ, целиком зависит от его начального заполнения, то есть элементы этой последовательности являются функцией своего номера и начального заполнения РГПЧ: $\Omega_i = f_i(\Omega_0)$, где $f_i(X) = f(f_{i-1}(X))$, $f_0(X) = X$. С учетом преобразования по алгоритму простой замены добавляется еще и зависимость от ключа:

$\boldsymbol{Г_i = Ц_{32\text{-}З}(\Omega_i) = Ц_{32\text{-}З}(f_i(\Omega_0)) = Ц_{32\text{-}З}(f_i(Ц_{32\text{-}З}(S))) = \varphi_i(S,K)}$, где $\boldsymbol{Г_i}$ – $\boldsymbol{i}$-тый элемент гаммы, K –ключ.

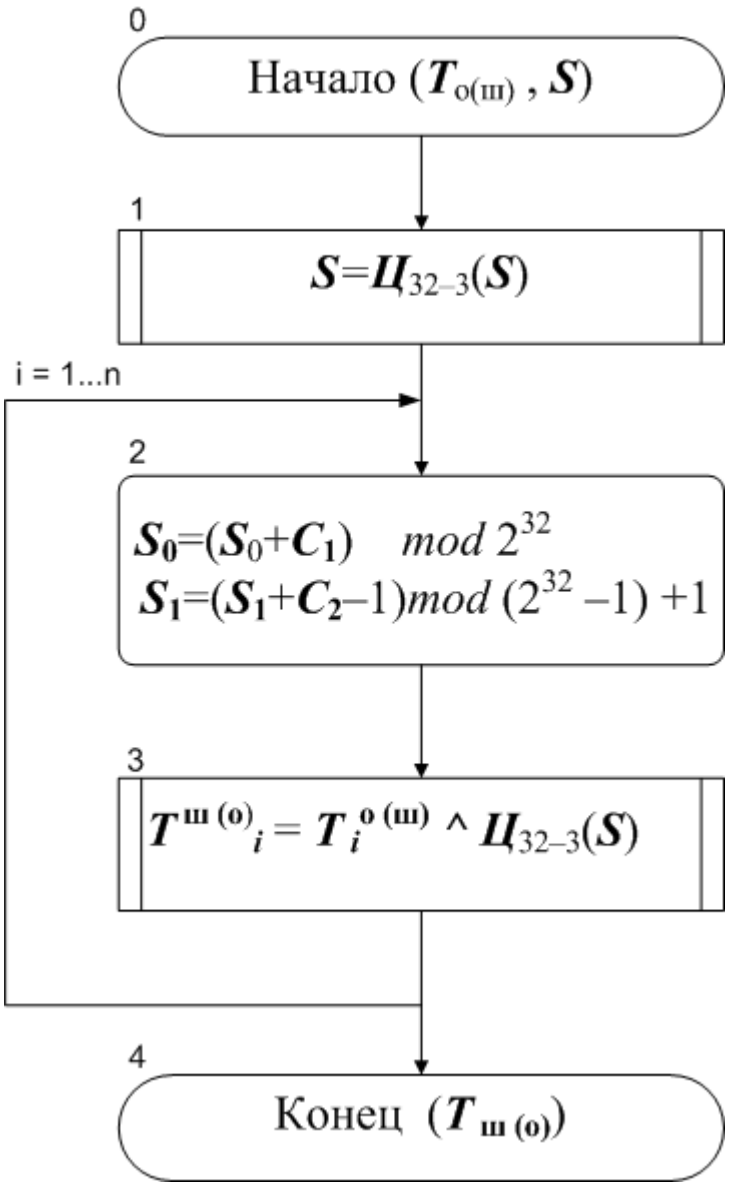

Рис. 1. Алгоритм зашифрования(расшифрования) данных в режиме гаммирования.

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 расшифрования должна использоваться одна и та же синхропосылка. Из требования уникальности гаммы, невыполнение которого приводит к катастрофическому снижению стойкости шифра, следует, что для

шифрования двух различных массивов данных на одном ключе необходимо обеспечить использование различных синхропосылок. Это приводит к необходимости хранить или передавать синхропосылку по каналам связи вместе с зашифрованными данными, хотя в отдельных особых случаях она может быть предопределена или вычисляться особым образом, если исключается шифрование двух массивов на одном ключе.

РГПЧ, спроектированный разработчиками ГОСТа, имеет следующие характеристики:

● в 64-битовом блоке старшая и младшая части обрабатываются независимо друг от друга:

$\boldsymbol{\Omega_i = (\Omega^0_i, \Omega^1_i)}$**;** фактически, существуют два независимых РГПЧ для старшей и младшей частей блока.

● рекуррентные соотношения для старшей и младшей частей следующие:

$\boldsymbol{\Omega^0_i = (\Omega^0_i + C_1) mod\ 2^{32}}$, где $\boldsymbol{C}_1=1010101_{16}$;

$\boldsymbol{\Omega^0_{i+1} = (\Omega^1_i + C_2 - 1)(mod\ 2^{32} - 1) + 1}$, где $\boldsymbol{C}_2=1010104_{16}$;

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

Шаг 0. Определяет исходные данные для основного шага криптопреобразования:

● ***T***о(ш) – массив открытых (зашифрованных) данных произвольного размера, подвергаемый процедуре зашифрования (расшифрования), по ходу процедуры массив подвергается преобразованию порциями по 64 бита;

● ***S*** – ***синхропосылка***, 64-битовый элемент данных, необходимый для инициализации генератора гаммы;

Шаг 1. Начальное преобразование синхропосылки, выполняемое для ее «рандомизации», то есть для устранения статистических закономерностей, присутствующих в ней, результат используется как начальное заполнение РГПЧ;

Шаг 2. Один шаг работы РГПЧ, реализующий его рекуррентный алгоритм. В ходе данного шага старшая (***S***1) и младшая (***S***0) части последовательности данных вырабатываются независимо друг от друга;

Шаг 3. Гаммирование. Очередной 64-битовый элемент, выработанный РГПЧ, подвергается процедуре за- шифрования по циклу 32–З, результат используется как элемент гаммы для зашифрования (расшифрования) очередного блока открытых (зашифрованных) данных того же размера.

Шаг 4. Результат работы алгоритма – зашифрованный (расшифрованный) массив данных. Ниже перечислены особенности гаммирования как режима шифрования.

1. Одинаковые блоки в открытом массиве данных дадут при зашифровании различные блоки шифртекста, что позволит скрыть факт их идентичности.

2. Поскольку наложение гаммы выполняется побитно, шифрование неполного блока данных легко выполнимо как шифрование битов этого неполного блока, для чего используется соответствующие биты блока гаммы. Так, для зашифрования неполного блока в 1 бит можно использовать любой бит из блока гаммы.

3. Синхропосылка, использованная при зашифровании, каким-то образом должна быть передана для использования при расшифровании. Это может быть достигнуто следующими путями:

- хранить или передавать синхропосылку вместе с зашифрованным массивом данных, что приведет к увеличению размера массива данных при зашифровании на размер синхропосылки, то есть на 8 байт;
- использовать предопределенное значение синхропосылки или вырабатывать ее синхронно источником и приемником по определенному закону, в этом случае изменение размера передаваемого или хранимого массива данных отсутствует;

Оба способа дополняют друг друга, и в тех редких случаях, где не работает первый, наиболее употребительный из них, может быть использован второй, более экзотический.

Второй способ имеет гораздо меньшее применение, поскольку сделать синхропосылку предопределенной можно только в том случае, если на данном комплекте ключевой информации шифруется заведомо не более одного массива данных, что бывает в редких случаях. Генерировать синхропосылку синхронно у источника и получателя массива данных также не всегда представляется возможным, поскольку требует жесткой привязки к чему- либо в системе. Так, здравая на первый взгляд идея использовать в качестве синхропосылки в системе передачи зашифрованных сообщений номер передаваемого сообщения не подходит, поскольку сообщение может потеряться и не дойти до адресата, в этом случае произойдет десинхронизация систем шифрования источника и приемника. Поэтому в рассмотренном случае нет альтернативы передаче синхропосылки вместе с зашифрованным сообщением.

С другой стороны, можно привести и обратный пример. Допустим, шифрование данных используется для защиты информации на диске, и реализовано оно на низком уровне, для обеспечения независимого доступа данные шифруются по секторам. В этом случае невозможно хранить синхропосылку вместе с зашифрованными данными, поскольку размер сектора нельзя изменить, однако ее можно вычислять как некоторую функцию от номера считывающей головки диска, номера дорожки (цилиндра) и номера сектора на дорожке. В этом случае синхропосылка привязывается к положению сектора на диске, которое вряд ли может измениться без переформатирования диска, то есть без уничтожения данных на нем.

Режим гаммирования имеет еще одну интересную особенность. В этом режиме биты массива данных шифруются независимо друг от друга. Таким образом, каждый бит шифртекста зависит от соответствующего бита открытого текста и, естественно, порядкового номера бита в массиве. Из этого вытекает,

что изменение бита шифртекста на противоположное значение приведет к аналогичному изменению бита открытого текста на противоположный:

Данное свойство дает злоумышленнику возможность воздействуя на биты шифр-текста вносить предсказуемые и даже целенаправленные изменения в соответствующий открытый текст, получаемый после его расшифрования, не обладая при этом секретным ключом. Это иллюстрирует хорошо известный в криптологии факт, что ***секретность и аутентичность суть различные свойства*** криптографических систем. Иными словами, свойства криптосистемы обеспечивать защиту от несанкционированного ознакомления с содержимым сообщения и от несанкционированного внесения изменений в сообщение являются независимыми и лишь в отдельных случаях могут пересекаться. Сказанное означает, что существуют криптографические алгоритмы, обеспечивающие определенную секретность зашифрованных данных и при этом никак не защищающие от внесения изменений и наоборот, обеспечивающие аутентичность данных и никак не ограничивающие возможность ознакомления с ними. По этой причине рассматриваемое свойство режима гаммирования не должно рассматриваться как его недостаток.

В ГОСТе ключевая информация состоит из двух структур данных. Помимо собственно ключа, необходимого для всех шифров, она содержит еще и таблицу замен.

1. Ключ является массивом из восьми 32-битовых элементов кода, он обозначается символом ***K***. Элементы ключа используются как 32-разрядные целые числа без знака. Таким образом, размер ключа составляет $32 \cdot 8 = 256$ бит или 32 байта.

2. Таблица замен представлена в виде матрицы размера 8*16, содержащей 4-битовые элементы, которые можно представить в виде целых чисел от 0 до 15. Строки таблицы замен называются узлами замен. Каждый узел замен должен содержать 16 различных чисел от 0 до 15 в произвольном порядке. Таблица

замен обозначается символом H. Общий объем таблицы замен равен: 8 узлов * 16 элементов/узел * 4 бита/элемент = 512 бит или 64 байта.

Основной шаг криптопреобразования.

Основной шаг криптопреобразования определяет преобразование 64-битового блока данных. Дополнительным параметром является 32-битовый блок, в качестве которого используется какой-либо элемент ключа. Схема алгоритма основного шага (Шаг($\boldsymbol{N}$,$\boldsymbol{X}$)) приведена на рисунке 2.

Шаг 0. Определяет исходные данные для основного шага криптопреобразования:

● $\boldsymbol{N}$ – преобразуемый 64-битовый блок данных, в ходе выполнения шага его младшая ($\boldsymbol{N}1$) и старшая ($\boldsymbol{N}2$) части обрабатываются как отдельные 32-битовые целые числа без знака. Таким образом, можно записать $\boldsymbol{N}$=($\boldsymbol{N}1$,$\boldsymbol{N}2$).

● $\boldsymbol{X}$ – 32-битовый элемент ключа;

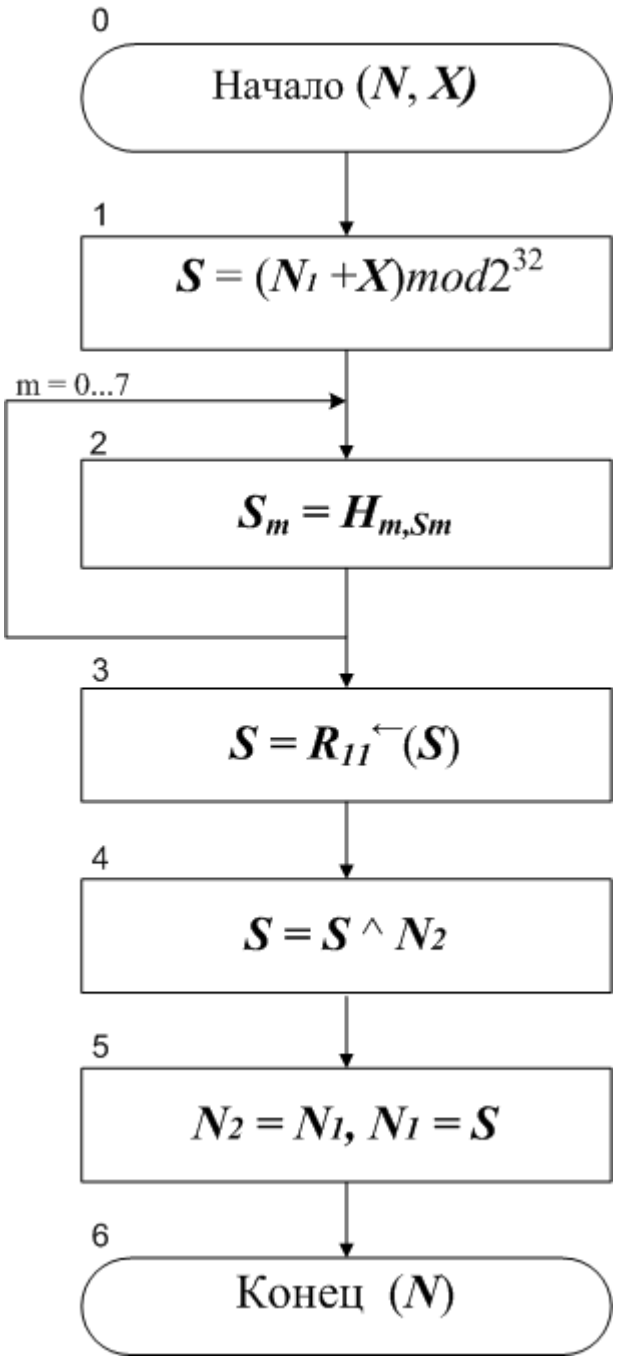

Рис. 2. Схема основного шага криптопреобразования алгоритма ГОСТ 28147-89.

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

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 новым, которое выбирается по таблице замен следующим образом: значение блока S_i меняется 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.

Шаг 3. Циклический сдвиг на 11 бит влево. Результат предыдущего шага сдвигается циклически на 11 бит в сторону старших разрядов и передается на следующий шаг. На схеме алгоритма символом $R_{11}^{\leftarrow}$ обозначена функция циклического сдвига своего аргумента на 11 бит влево, т.е. в сторону старших разрядов.

Шаг 4. Побитовое сложение: значение, полученное на шаге 3, побитно складывается по модулю 2 со старшей половиной преобразуемого блока.

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

Шаг 6. Полученное значение преобразуемого блока возвращается как результат выполнения алгоритма основного шага криптопреобразования.

Базовые циклы криптографических преобразований.

ГОСТ относится к классу блочных шифров, то есть единицей обработки информации в нем является блок данных. Алгоритмы для криптографических преобразований, то есть для зашифрования, расшифрования и «учета» в контрольной комбинации одного блока данных. Именно эти алгоритмы и называются ***базовыми циклами*** ГОСТа.

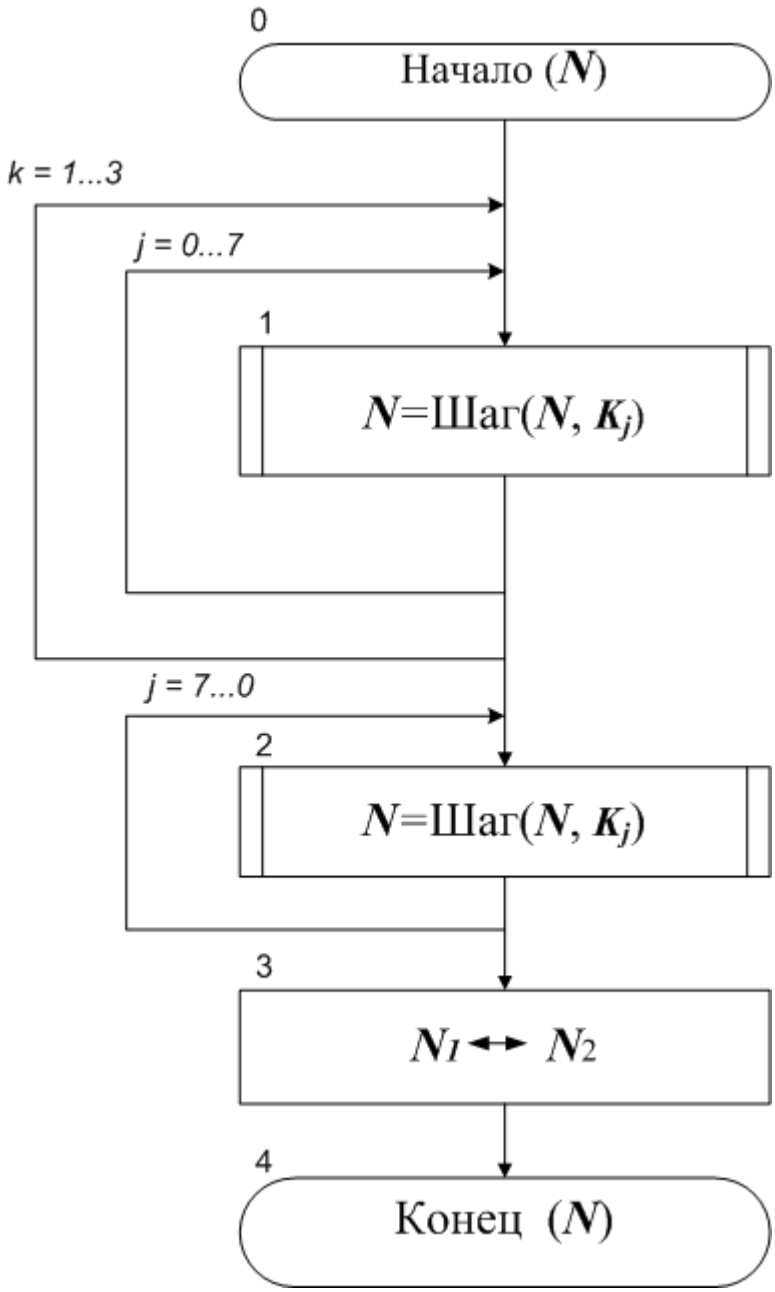

Рис. 3. Схема цикла зашифрования 32-З.

Базовые циклы построены из ***основных шагов*** криптографического преобразования. В процессе выполнения основного шага используется только один элемент ключа, в то время как ключ ГОСТ содержит восемь таких элементов. Базовые циклы заключаются в многократном выполнении ***основного шага*** с использованием разных элементов ключа и отличаются друг от друга только числом повторения шага и порядком использования ключевых элементов. Цикл зашифрования 32-З:

$\boldsymbol{K}0,\boldsymbol{K}1,\boldsymbol{K}2,\boldsymbol{K}3,\boldsymbol{K}4,\boldsymbol{K}5,\boldsymbol{K}6,\boldsymbol{K}7,\boldsymbol{K}0,\boldsymbol{K}1,\boldsymbol{K}2,\boldsymbol{K}3,\boldsymbol{K}4,\boldsymbol{K}5,\boldsymbol{K}6,\boldsymbol{K}7,\boldsymbol{K}0,\boldsymbol{K}1,\boldsymbol{K}2,\boldsymbol{K}3,\boldsymbol{K}4,\boldsymbol{K}5,\boldsymbol{K}6,\boldsymbol{K}7,$
$\boldsymbol{K}7,\boldsymbol{K}6,\boldsymbol{K}5,\boldsymbol{K}4,\boldsymbol{K}3,\boldsymbol{K}2,\boldsymbol{K}1,\boldsymbol{K}0.$

Цикл 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 зашифрования («З») в использовании ключевых элементов.

Схема базового цикла ($\boldsymbol{Ц}_{32\text{-}З}$) приведена на рисунке 3. Базовый цикл принимает в качестве аргумента и возвращает в качестве результата 64- битовый блок данных, обозначенный на схемах $\boldsymbol{N}$. Символ Шаг($\boldsymbol{N},\boldsymbol{K_j}$) обозначает выполнение основного шага криптопреобразования для блока $\boldsymbol{N}$ с использованием ключевого элемента $\boldsymbol{K_j}$.

При расшифровании блока в режиме гаммирования изменения предсказуемы. Непредсказуемые изменения в расшифрованном блоке данных могут быть обнаружены только в случае избыточности этих самых данных, причем, чем больше степень избыточности, тем вероятнее обнаружение искажения. Очень большая избыточность имеет место, например, для текстов на естественных и искусственных языках, в этом случае факт искажения обнаруживается практически неизбежно. Однако в других случаях, например, при искажении сжатых звуковых образов, мы получим просто другой образ, который сможет воспринять наше ухо. Искажение в этом случае останется необнаруженным, если, конечно, нет априорной информации о характере звука. Вывод здесь такой: поскольку способность некоторых режимов шифрования обнаруживать искажения, внесенные в шифрованные данные, существенным образом опирается на наличие и степень избыточности шифруемых данных, эта способность не может рассматриваться как их достоинство.

Для решения задачи обнаружения искажений в зашифрованном массиве данных с заданной вероятностью в ГОСТе предусмотрен дополнительный

режим криптографического преобразования – выработка имитовставки. Имитовставка – это контрольная комбинация, зависящая от открытых данных и секретной ключевой информации. Целью использования имитовставки является обнаружение всех случайных или преднамеренных изменений в массиве информации.