**Алгоритм ГОСТ 28147-89**

Описание алгоритма

Схема алгоритма ГОСТ 28147-89 показана на Рисунке (№ Рисунка). Схема этого алгоритма построена на основе сети Фейстеля, и является достаточно простой, что однозначно упрощает его программную или аппаратную реализацию.



Рисунок 1 Схема работы алгоритма ГОСТ 28147-89

Алгоритм ГОСТ 28147-89 шифрует информацию блоками по 64 бита, которые разбиваются на два субблока по 32 бита (N1 и N2). Субблок N1 определенным образом обрабатывается, после чего его значение складывается со значением субблока N2 (сложение выполняется по модулю 2), затем субблоки меняются местами. Такое преобразование выполняется определенное количество раундов: 16 или 32 в зависимости от режима работы алгоритма (описаны далее). В каждом раунде выполняются следующие операции:

1. Наложение ключа. Содержимое субблока N1 складывается по модулю 232 с частью ключа К*x*.

Ключ шифрования алгоритма ГОСТ 28147-89 имеет размерность 256 битов, а К*x* — это его 32-битная часть, т. е. 256-битный ключ шифрования представляется в виде конкатенации 32-битных подключей **(№ РИСУНКА**):

К0*,* К1, К2, К3,К4, К5, К6*,* К7.



Рисунок 2 Ключ Шифрования ГОСТ-28147-89

В процессе шифрования используется один из этих подключей — в зависимости от номера раунда и режима работы алгоритма.

1. Табличная замена используется таким образом: на вход подается блок данных определенной размерности (в этом случае — 4-битный), числовое представление которого определяет номер выходного значения. Например, имеем S-box следующего вида:

4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1.

Пусть на вход пришел 4-битный блок «0100», т. е. значение 4. Согласно таблице, выходное значение будет равно 15, т. е. «1111» (0 заменяется на 4, 1 — на 11, значение 2 не изменяется и т. д.).

Как видно, схема алгоритма весьма проста, что означает, что наибольшая нагрузка по шифрованию данных ложится на таблицы замен. К сожалению, алгоритм обладает тем свойством, что существуют «слабые» таблицы замен, при использовании которых алгоритм может быть раскрыт криптоаналитическими методами. К числу слабых относится, например, таблица, в которой выход равен входу, то есть:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

1. Побитовый циклический сдвиг влево на 11 битов.

**Режимы работы алгоритма**

Алгоритм ГОСТ 28147-89 имеет 4 режима работы:

* режим простой замены;
* режим гаммирования;
* режим гаммирования с обратной связью;
* режим выработки имитоприставок.

Эти режимы несколько отличаются от общепринятых (описанных в paid. 1.4), поэтому стоит рассмотреть их подробнее.

Данные режимы имеют различное назначение, но используют одно и то же описанное выше шифрующее преобразование.

**Режим простой замены**

В режиме простой замены для зашифровывания каждого 64-битного блока информации просто выполняются 32 описанных выше раунда. 32-битные подключи используются в следующей последовательности:

* К0, К1, К2, К3, К4, К5, К6, К7, K0… в раундах с 1 по 24.
* K7, K6, K5, K4, K3, K2, K1, K0 в раундах с 25 по 32.

Дешифровывание в режиме простой замены производится совершенно так же, но с несколько другой последовательностью применения подключей:

* K7, K6, K5, K4, K3, K2, K1, K0 в раундах с 1 по 8.
* К0, К1, К2, К3, К4, К5, К6, К7, K0… в раундах с 9 по 32.

Аналогично стандартному режиму ЕСВ. по причине раздельного шифрова¬ния блоков режим простой замены категорически не рекомендуется ис¬пользовать для шифрования собственно данных; он должен использоваться только для шифрования других ключей шифрования в многоключевых схемах.

**Режим гаммирования**

В режиме гаммирования (рис. 3.3) каждый блок открытого текста побитно складывается по модулю 2 с блоком гаммы шифра размером 64 бита. Гамма шифра — это специальная последовательность, которая вырабатывается с помощью описанных выше преобразований следующим образом:

1. В регистры N1 и N2 записывается их начальное заполнение— 64-битная величина, называемая «синхропосылкой» (синхропосылка, практически, является аналогом вектора инициализации).



Рисунок 3 Режим Гаммирования

1. Выполняется зашифровывание содержимого регистров N1 и N2 (в данном случае — синхропосылки) в режиме простой замены.
2. Содержимое N1 складывается по модулю (232 - 1 ) с константой Cl = 224 +216 +28 +4, результат сложения записывается в регистр N1.
3. Содержимое N2 складывается по модулю 232 с константой C2 = 224 +216 +28 + 1 , результат сложения записывается в регистр N2.
4. Содержимое регистров N1 и N2 подается на выход в качестве 64-битного блока гаммы шифра (т. е. в данном случае N1 и N2 образуют первый блок гаммы).
5. Если необходим следующий блок гаммы (т. е. необходимо продолжить зашифровывание или дешифровывание), выполняется возврат к шагу 2.

Для расшифровывания аналогичным образом выполняется выработка гаммы, затем снова применяется операция XOR к битам зашифрованного текста и гаммы.

Для выработки той же самой гаммы шифра у пользователя, расшифровывающего криптограмму, должен быть тот же самый ключ и то же значение синхропосылки, которые применялись при зашифровывании информации.

В противном случае получить исходный текст из зашифрованного не удастся.

В большинстве реализаций алгоритма ГОСТ 28147-89 синхропосылка не является секретным элементом, однако синхропосылка может быть так же секретна, как и ключ шифрования. В этом случае можно считать, что эффективная длина ключа алгоритма (256 битов) увеличивается еще на 64 бита синхропосылки, которую можно рассматривать как дополнительный ключевой элемент.

**Режим гаммирования с обратной связью**

В режиме гаммирования с обратной связью в качестве заполнения регистров N1 и N2, начиная со 2-го блока, используется не предыдущий блок гаммы, а результат зашифровывания предыдущего блока открытого текста **(№РИСУНКА)**. Первый же блок в данном режиме генерируется полностью аналогично предыдущему.



Рисунок 4 Выработка гаммы шифра в режиме гаммирования с обратной связью

**Режим выработки имитоприставки**

Имитоприставка — это криптографическая контрольная сумма, вычисляемая с использованием ключа шифрования и предназначенная для проверки целостности сообщений.

Генерация имитоприставки в рамках алгоритма ГОСТ 28147-89 выполняется следующим образом:

1. Первый 64-битный блок информации, для которой вычисляется имитоприставка, записывается в регистры N1 и N2 и зашифровывается в сокращенном режиме простой замены, в котором выполняются первые 16 раундов из 32.
2. Полученный результат суммируется по модулю 2 со следующим блоком информации с сохранением результата в N1 и N2.
3. N1 и N2 снова зашифровываются в сокращенном режиме простой замены и т. д. до последнего блока информации.

Имитоприставкой считается 64-битное результирующее содержимое регистров N1 и N2 или его часть. Чаще всего используется 32-битная имитоприставка. Этого достаточно, поскольку, как и любая контрольная сумма, имитоприставка предназначена, прежде всего, для защиты от случайных искажений информации.

Имитоприставка используется следующим образом:

1. При зашифровывании какой-либо информации вычисляется имитоприставка открытого текста и посылается вместе с шифротекстом.
2. После расшифровывания имитоприставка снова вычисляется и сравнивается с присланной.
3. Если вычисленная и присланная имитоприставки не совпадают— шифр- текст был искажен при передаче или использовались неверные ключи при расшифровывании.

Имитоприставка особенно полезна для проверки правильности расшифровывания ключевой информации при использовании многоключевых схем.

**Криптоанализ ГОСТа 28147-89**

С момента опубликования ГОСТа на нём стоял ограничительный гриф «для служебного пользования», и формально шифр был объявлен «полностью открытым» только в мае 1994 года. История создания шифра и критерии разработчиков по состоянию на 2016 год не обнародованы.

Высокая стойкость алгоритма ГОСТ 28147-89 дости­гается за счет следующих факторов:

* большой длины ключа— 256 битов, и вместе с секретной синхропосылкой эффективная длина ключа увеличивается до 320 битов;
* 32 раундов преобразований; уже после 8 раундов достигается полный эф­фект рассеивания входных данных: изменение одного бита блока откры­того текста повлияет на все биты блока шифртекста, и наоборот, т. е. су­ществует многократный запас стойкости.

Однако в алгоритме ГОСТ 28147-89 существуют и уязвимости. Например, секретные таблицы замен могут быть вычислены с помощью следующей атаки:

1. Устанавливается нулевой ключ и выполняется поиск «нулевого вектора», т. е. значения где — функция раунда алгоритма. Этот этап занимает порядка 232 операций шифрования.
2. С помощью нулевого вектора вычисляются значения таблиц замен, что занимает не более 211 операций.

Указанный выше способ – не единственная возможная атака на ГОСТ 28147-89. В различное время были опубликованы следующие работы, посвящённые криптоатакам на алгоритм ГОСТ 28147-89. Далее в таблице( Таблица 1)приведены краткие сведения по некоторым опубликованным работам, посвящённым криптоатакам на ГОСТ 28147-89.

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| Название Работы | Количество раундов | Количество Текста  (ОТ-открытый Текст, СПТ-специально подобранный текст | Вычислительная Сложность |
| 1R атака на связанных ключах | 32 | 235 СПТ | 236 |
| Метод рационального продолжения | 32 | В среднем 10-100 СПТ | 1010 |
| Атака методом бумеранга | 32 | 210  пар открытого/зашифрованного текста (СПТ) | 226 |
| Атака Исобе | 32 | 232 ОТ | 2224 |
| Атака на шифросистему с 12 связанными ключам | 8 | От 250 до 2106 СПТ | 260 |
| Улучшенная атака скольжения | 24 | 264 ОТ | 263 |

На основе представленных выше данных и в заключении всего сказанного можно сделать вывод о том, что ГОСТ 28147-89 обладает высокой криптостойкостью. В пользу этого утверждения говорит так-же то, что алгоритм шифрования, описанный в ГОСТ 28147-89 ныне вместе с новым алгоритмом «Кузнечик» был опубликован как часть стандарта ГОСТ Р 34.12-2015.

**Алгоритм AES(Rijndael)**

**Структура алгоpитма:**

Алгоритм AES представляет блок данных в виде двумерного байтового массива размером 4x4. Все операции производятся над отдельными байтами массива, а также над независимыми столбцами и строками.

В каждом раунде алгоритма выполняются следующие преобразования :

1. SubBytes.
2. ShiftRows.
3. MixColumns.
4. AddRoundKey.

1. Операция SubBytes, представляющая собой табличную замену каждого байта массива данных согласно Таблице замен алгоритма AES (Рисунок 5 Таблица Замен AES)



Рисунок 5 Таблица Замен AES

Таблица меняет входное значение 0 на 63 (шестнадцатеричное значение), 1 — на 7С и т. д.

Вместо данной табличной замены можно выполнить эквивалентную ей комбинацию двух операций:

* вычисление мультипликативной обратной величины от входного значения в конечном поле GF(28); обратной величиной от 0 является 0;
* выходное значение b вычисляется следующим образом:

Где:

- есть i-ый бит b,

  — i-ый бит константы c=01100011;

2. Операция ShiftRows, которая выполняет циклический сдвиг влево всех строк массива данных, за исключением нулевой (Рисунок 6 ShiftRows). Сдвиг i-й строки массива (для i = 1,2,3) производится на i байтов.

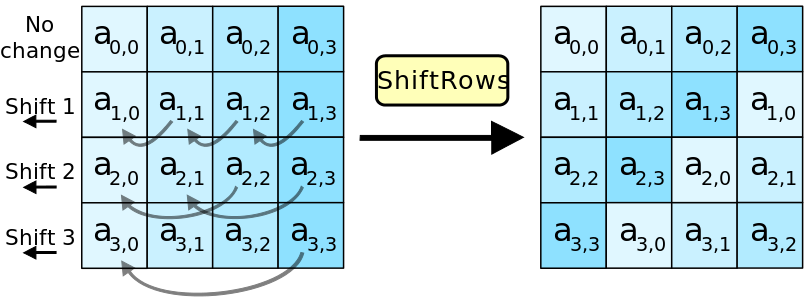


Рисунок 6 ShiftRows

3. Операция MixColumns. Выполняет умножение каждого столбца массива данных, который рассматривается как полином в конечном поле GF( 28), на фиксированный полином а(х):

Умножение выполняется по модулю

4. Операция AddRoundKey (Рисунок 7 Add Round Key). Выполняет наложение на массив данных материала ключа. А именно, на i-й столбец массива данных (i = 0...3) побитовой логической операцией «исключающее или» (XOR) накладывается определенное слово расширенного ключа *WArVt,*, где r— номер текущего раунда алгоритма, начиная с 1.

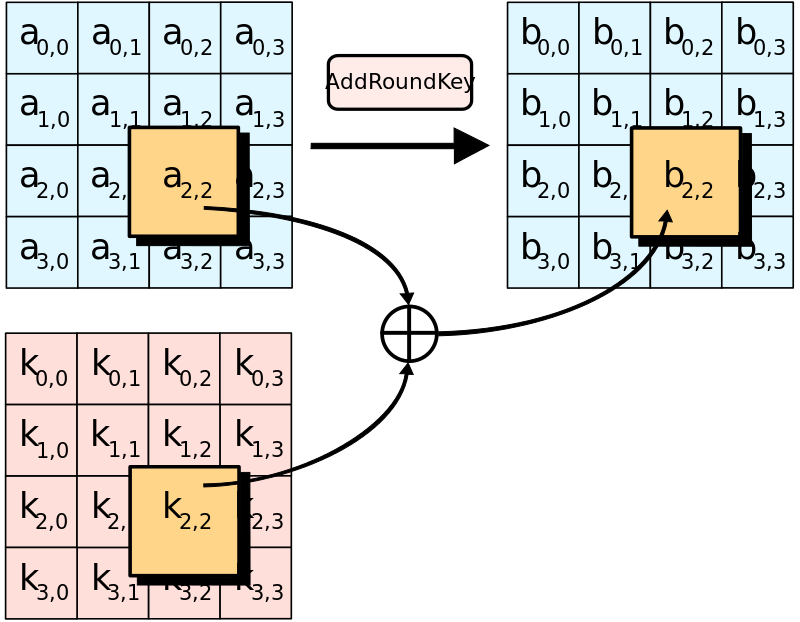


Рисунок 7 Add Round Key

Алгоритм AES имеет несколько стандартных вариаций, отличающихся размером ключа и, соответственно, количеством раундом шифрования. Возможная длина ключей шифрования для алгоритма AES - 128, 192 и 256 битов. Зависимость количества раундов от длины ключа представлена в таблице (Таблица 2).

Таблица 2

|  |  |
| --- | --- |
| Длина ключа, бит | Количество раундов |
| 128 | 10 |
| 192 | 12 |
| 256 | 14 |

Перед первым раундом алгоритма выполняется предварительная модификация ключа с помощью функции AddRoundKey.А в последний раунд Алгоритма над данными не выполняется функция MixColumns.

**Процесс обработки ключа**

Алгоритм обработки ключа состоит из двух процедур:

* Процедура расширения ключа
* Процедура выбора раундового ключа (ключа итерации)

**Процедура расширения ключа**

Алгоритм AES использует ключи шифрования трех фиксированных размеров: 128, 192 и 256 битов. Задача процедуры расширения ключа состоит в формировании нужного количества слов(4-байтный фрагмент) расширенного ключа для их использования в операции AddRoundKey . Таким образом, в процессе расширения ключа формируется 4 \* (R + 1) слов, где R – Количество раундов шифрования.

AES алгоритм, используя процедуру KeyExpansion() и подавая в неё Cipher Key – Исходный криптографический ключ, K, получает ключи для всех раундов. Всего получается Nb\*(Nr + 1) слов: изначально для алгоритма требуется набор из Nb слов, и каждому из Nr раундов требуется Nb ключевых набора данных. Полученный массив ключей для раундов обозначается как .

Первые Nk (Nk — размер исходного ключа шифрования К в словах) слов Wj , т. е. i = 0...(Nk -1), формируются их последовательным заполнением байтами ключа, как можно увидеть исходя из иллюстрации (Рисунок 8 ).



Рисунок 8 Инициализация первых Nk слов

В каждое последующее слово, , кладётся значение полученное при операции XOR между словами и . Для слов, позиция которых кратна Nk, перед операией XOR к w[i-1] применяется трансформация, которая состоит из операций RotWord, и затем операцией SubWord над результатом, полученным в RotWord, после чего производят операцию XOR между результатом операции SubWord и константой раунда Rcon[i].

Rcon[i] представляет собой слова, в которых все байты, кроме первого, являются нулевыми, а первый байт имеет значение mod;

Операция RotWord побайтно сдвигает входное слово на 1 байт влево.

Операция SubWord производит в слове побайтовую нелинейную замену в ключе, аналогичную замене, производимой в операции SubBytes.

**Алгоритм выбора раундового ключа**

На каждой итерации раундовый ключ для операции AddRoundKey выбирается из массива начиная с элемента до , где Nb – число столбцов в шифруемом блоке, и в случае с AES алгоритмом Nb всегда равно 4.

**Расшифровывание**

Расшифровывание выполняется применением обратных операций в обратной последовательности.В ходе расшифровывания данных используются следующие операции

1. AddRoundKey – является обратной по отношению к самой себе, что обеспечивает простоту практической реализации алгоритма AES.
2. InvMixColumns - операция, которая является обратной по отношению к MixColumns.
3. InvShiftRows- операция, которая является обратной по отношению к ShiftRows.
4. InvSubBytes - операция, которая является обратной по отношению к SubBytes.

Перед первым раундом расшифровывания выполняется операция AddRoundKey . Затем выполняется R раундов расшифровывания, в каждый из которых осуществляет указанные преобразования.

1. Операция InvShiftRows производит циклический сдвиг вправо трех последних строк массива данных на то же количество байтов, на которое выполнялся сдвиг операцией ShiftRows при зашифровывании.
2. Операция InvSubBytes производит побайтно замену, обратную замене,произведённой в операции SubBytes.
3. Операция AddRoundKey. Нумерация раундов при расшифровывании производится в обратную сторону — от (R -1) до 0.
4. Операция InvMixColumns выполняет умножение каждого столбца массива данных аналогично прямой операции MixColumns, однако, умножение производится на полином , определенный следующим образом:

При этом, последний раунд расшифровывания не содержит операцию InvMixColumns.

**Исследования криптостойкости Алгоритма AES.**

Согласно оценкам авторов, AES(Rijndael) не подвержен следующим видам криптоаналитических атак:

* У алгоритма отсутствуют слабые ключи, а также возможности его вскрытия с помощью атак на связанных ключах.
* К алгоритму не применим дифференциальный криптоанализ.
* Алгоритм не атакуем с помощью линейного криптоанализа и усеченных дифференциалов.
* Square-атака также не применима к алгоритму AES(Rijndael).
* Алгоритм не вскрывается методом интерполяции.

В ходе оценки криптостойкости алгоритма в рамках конкурса были предложены различные виды криптоаналитических атак на алгоритм AES(Rijndael) с сокращённым количеством раундов. Одной из команд криптологов была предложена атака на связанных ключах на 9- раундовую версию AES(Rijndael) с 256-битным ключом, которой необходимо 277 выбранных открытых текстов, зашифрованных на 256 связанных ключах, и 2224 операций шифрования. И, хотя существует негласное правило, гласящее, что следует модифицировать криптографический алгоритм путём двукратного увеличения количества раундов, по сравнению с максимально атакуемым, не все атаки на AES(Rijndael) являются осуществимыми на практике. Опираясь на эти сведения, а так же учитывая, что в алгоритме AES(Rijndael) выполняется, как минимум, 10 раун­дов, запас криптостойкости алгоритма экспертами был признан адекватным.

В результате, учитывая заключение экпертов о достаточном запасе криптостойкости AES(Rijndael), а так же простоту реализации данного алгоритма, алгоритм, названный создателями Rijndael был выбран в качестве победителя конкурса AES.

После того, как в июне 2003 года Агентство национальной безопасности США постановило, что шифр AES является достаточно надёжным, чтобы использовать его для защиты сведений, составляющих государственную тайну, а алгоритм стал известен широкой общественности в качестве стандарта, количество попыток взлома алгоритма AES увеличилось. Таким образом, в процессе эксплуатации алгоритма были предложены и более сильные атаки, которые, однако, до сих пор не привели к вскрытию алгоритма, что позволяет сомневаться в их практической реализуемости. Например:

* была предложена атака на алгоритм AES-192 с использованием комбинации методов бумеранга и связанных ключей, при котором 9-раундовый AES-192 атакуется при наличии выбранных откры­тых текстов, каждый из которых шифруется на 256 связанных ключах, выполнением операций шифрования. Позднее эта атака была усилена, в частности предлагается атака на 10-раундовый алгоритм AES-192; для новой атаки требуется выбранных открытых текстов (на 256 связанных ключах) и операций.

Кроме атак, нацеленных непосредственно на алгоритмическую структуру AES, алгоритм так-же подвергался атакам по сторонним каналам. Атаки по сторонним каналам используют определённые особенности реализации систем, использующих данный шифр, с целью раскрыть частично или полностью секретные данные, в том числе ключ. Известно несколько подобных атак на системы, использовавшие алгоритм AES:

* В апреле 2005 года Дэниел Бернштейн опубликовал работу с описанием атаки, использующей для взлома информацию о времени выполнения каждой операции шифрования. Данная атака потребовала более 200 миллионов выбранных и заранее подготовленных шифротекстов для нахождения ключа.
* В декабре 2009 года была опубликована работа, в которой использование дифференциального анализа ошибок, искусственно создаваемых в матрице состояния на 8-м раунде шифрования, позволило восстановить ключ за операций

Так же следует привести краткие данные, по некоторым работам, посвящённым криптоанализу AES(Таблица 3).

Таблица 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название Работы | Тип алгоритма | Количество раундов | Количество Текста | Количество Операций |
| Improved cryptoanalysis of Rijndael | AES | 7 | От до СП |  |
| Improved cryptoanalysis of Rijndael | AES-256 | 8 | От до СП | От до |
| Improved cryptoanalysis of Rijndael | AES-256 | 9 | текстов зашифрованных на 256 связанных ключах |  |
| Cache-timing attacks on AES | AES | Не указанно | 200 миллионов выбранных шифротекстов | 2\* |
| A Diagonal Fault Attack on the Advanced Encryption Standar | AES | 8 | Не указанно |  |
| Related-Key Boomerang and Rectangle attack | AES-192 | 9 | СП зашифрованных на 256 связанных ключах |  |
| Combined Differential, Linear and Related-Key Attacks on Block-Ciphers and MAC Algorithms | AES-192 | 10 | СП зашифрованных на 256 связанных ключах |  |

В заключение разбора алгоритма AES хочется сказать, что несмотря на то, что некоторые версии Алгоритма находятся под угрозой взлома, например AES-192, версия с 256-битным ключом до сих пор является в достаточной степени криптостойкой, о чём свидетельствует то, что этот вариант алгоритма AES до сих пор является стандартом в США, а так же тот факт, что большая часть теоретически описанных атак на алгоритм AES является практически нереализуемой. Таким образом, учитывая лёгкость реализации, высокую криптостойкость и повсеместную распространённость, AES является одним из наиболее надёжных и предпочтительных алгоритмов для реализации шифрования в программных и аппаратных системах.

**Twofish**