Blowfish — криптографический алгоритм, реализующий блочное симметричное шифрование с переменной длиной ключа. Разработан Брюсом Шнайером в 1993 году. Представляет собой сеть Фейстеля. Выполнен на простых и быстрых операциях: XOR, подстановка, сложение. Является незапатентованным и свободно распространяемым.

До появления Blowfish существовавшие алгоритмы были либо запатентованными, либо ненадёжными, а некоторые и вовсе держались в секрете (например, Skipjack). Алгоритм был разработан в 1993 году Брюсом Шнайером в качестве быстрой и свободной альтернативы устаревшему DES и запатентованному IDEA. По заявлению автора, критериями проектирования Blowfish были:

* скорость (шифрование на 32-битных процессорах происходит за 26 тактов);
* простота (за счёт использования простых операций, уменьшающих вероятность ошибки реализации алгоритма);
* компактность (возможность работать в менее, чем 5 Кбайт памяти);
* настраиваемая безопасность (изменяемая длина ключа).

Алгоритм состоит из двух частей: расширение ключа и шифрование данных. На этапе расширения ключа исходный ключ (длиной до 448 бит) преобразуется в 18 32-битовых подключей и в 4 32-битных S-блока, содержащих 256 элементов. Общий объём полученных ключей равен (18+256*4)*32  = 33344 бит или 4168байт

**(Более подробное описание – по надобности)**

**Advanced Encryption Standard** (AES), также известный как Rijndael симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарташифрования правительством США по результатам конкурса AES. Этот алгоритм хорошо проанализирован и сейчас широко используется, как это было с его предшественником DES. Национальный институт стандартов и технологий США (англ. National Institute of Standards and Technology, NIST) опубликовал спецификацию AES 26 ноября 2001 года после пятилетнего периода, в ходе которого были созданы и оценены 15 кандидатур. 26 мая 2002 года AES был объявлен стандартом шифрования. По состоянию на 2009 год AES является одним из самых распространённых алгоритмов симметричного шифрования. Поддержка AES (и только его) введена фирмой Intel в семейство процессоров x86начиная с Intel Core i7-980X Extreme Edition, а затем на процессорах Sandy Bridge.

AES является стандартом, основанным на алгоритме Rijndael. Для AES длина input(блока входных данных) и State(состояния) постоянна и равна 128 бит, а длина шифроключа **K** составляет 128, 192, или 256 бит. При этом, исходный алгоритм Rijndael допускает длину ключа и размер блока от 128 до 256 бит с шагом в 32 бита. Для обозначения выбранных длин input, State и Cipher Key в 32-битных словах используется нотация Nb = 4 для input и State, Nk = 4, 6, 8 для Cipher Key соответственно для разных длин ключей.

В начале шифрования input копируется в массив State по правилу  \mathrm{state}[r,c] = \mathrm{input}[r+4c] , для  0 \le r < 4  и  0 \le c < Nb . После этого к State применяется процедура AddRoundKey() и затем State проходит через процедуру трансформации (раунд) 10, 12, или 14 раз (в зависимости от длины ключа), при этом надо учесть, что последний раунд несколько отличается от предыдущих. В итоге, после завершения последнего раунда трансформации, State копируется в output по правилу \mathrm{output}[r+4c] = \mathrm{state}[r,c], для  0 \le r < 4  и  0 \le c < Nb .

В июне 2003 года Агентство национальной безопасности США постановило, что шифр AES является достаточно надёжным, чтобы использовать его для защиты сведений, составляющих государственную тайну (англ. classified information). Вплоть до уровня SECRET было разрешено использовать ключи длиной 128 бит, для уровня TOP SECRET требовались ключи длиной 192 и 256 бит

/\*

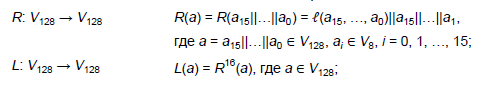
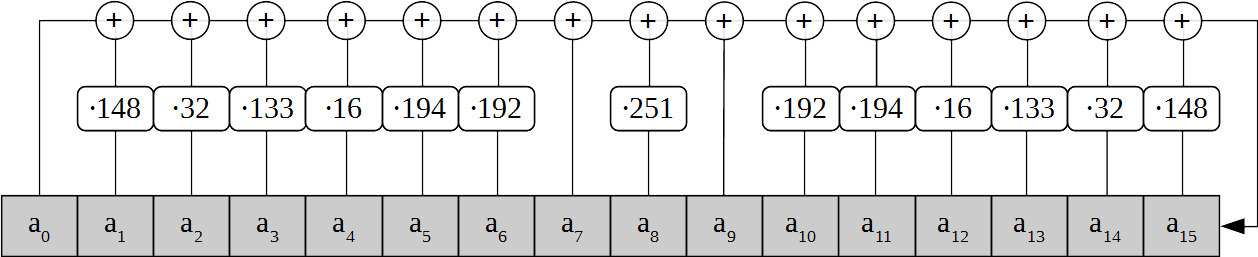
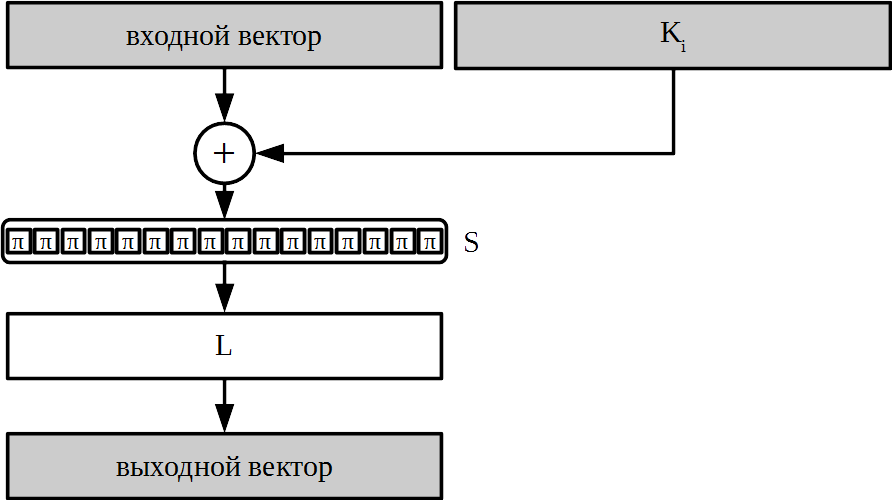
**Блочный шифр «Кузнечик» (Kuznechik, Kuznyechik)** — симметричный алгоритм блочного шифрования с размером блока 128 бит и длиной ключа 256 бит, для генерации которого используется сеть Фейстеля.

В июне 2015 года в России был принят новый стандарт блочного шифрования — ГОСТ Р 34.12-2015. Этот стандарт помимо старого доброго ГОСТ 28147-89, который теперь называется «Магма» и имеет фиксированный набор подстановок, содержит описание блочного шифра «Кузнечик». О нем я и расскажу в этом посте.  
  
В отличие от ГОСТ 28147-89 новый шифр представляет собой не сеть Фейстеля, а т.н. SP-сеть: преобразование, состоящее из нескольких одинаковых раундов, при этом каждый раунд состоит из нелинейного и линейного преобразований, а также операции наложения ключа. В отличие от сети Фейстеля, при использовании SP-сети преобразуется весь входной блок, а не его половина. Такая структура иногда также называется AES-like (похожей на AES), однако, в отличие от последнего у «Кузнечика» есть ряд своих «фишек»:

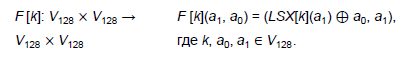
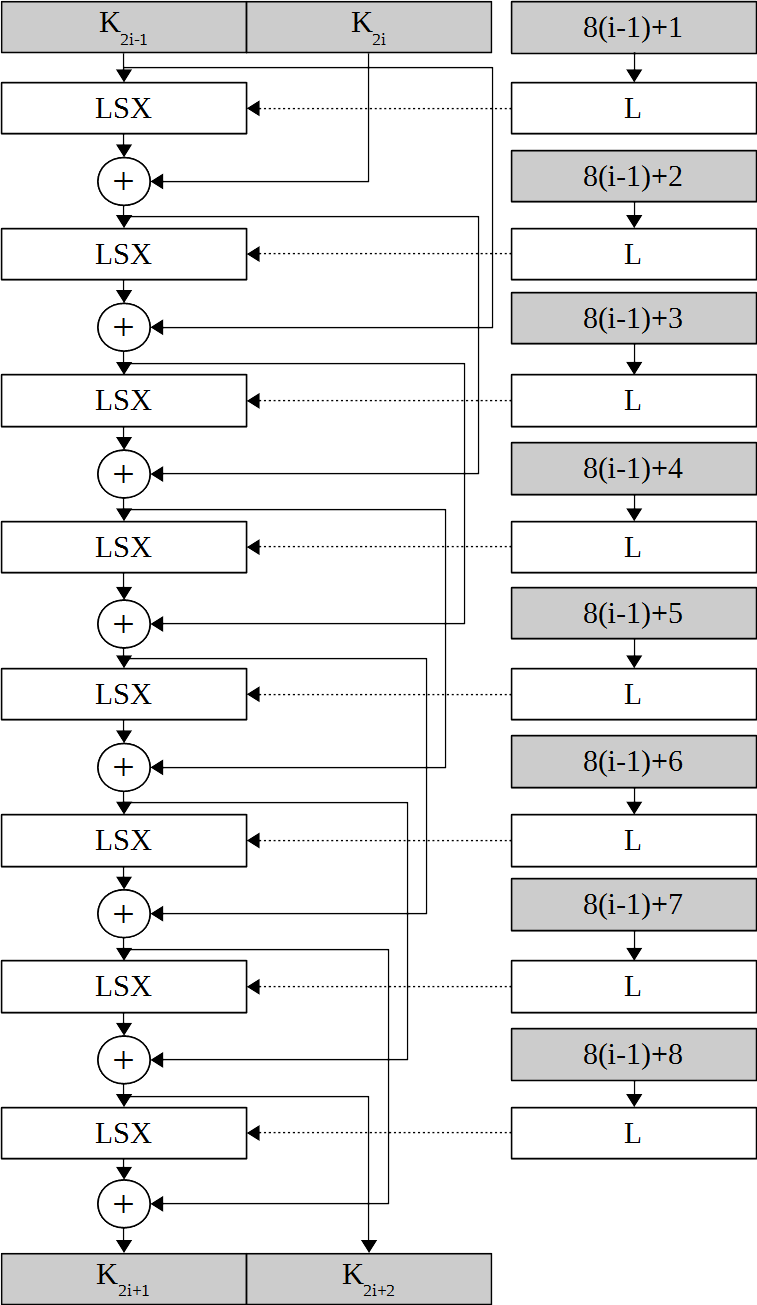
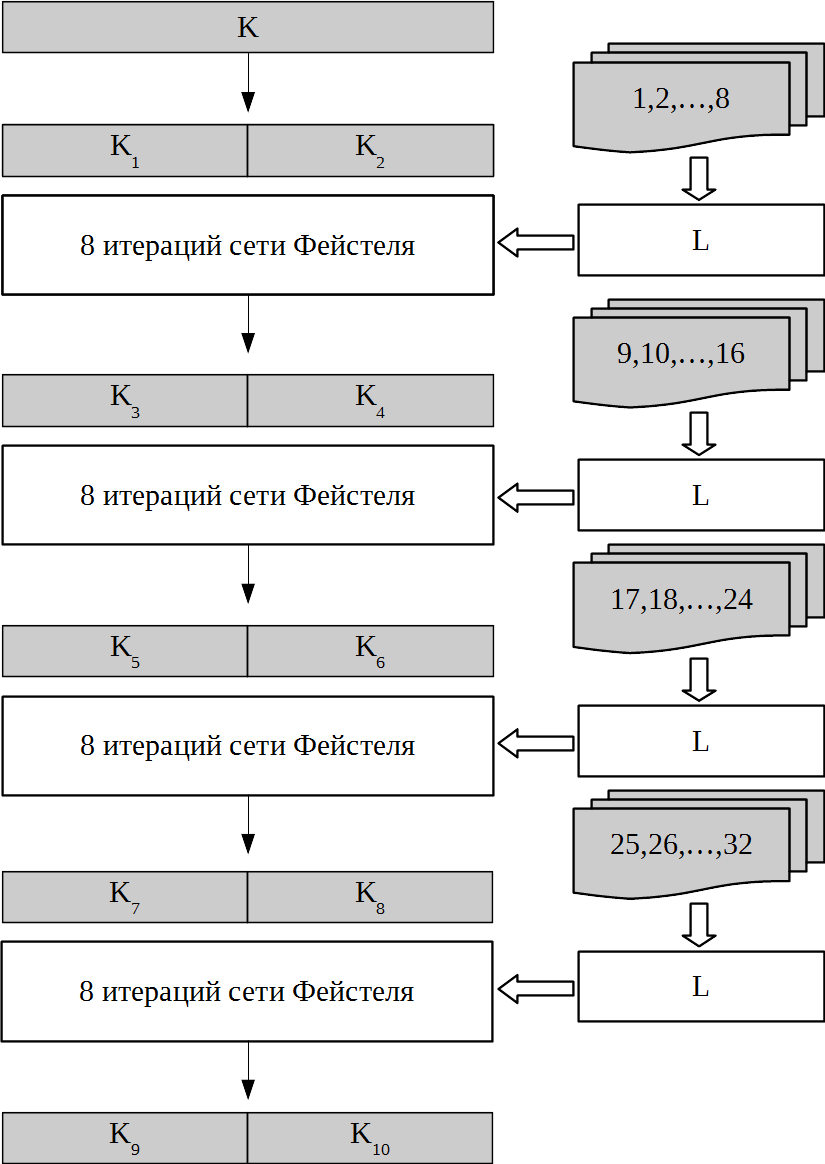
* линейное преобразование может быть реализовано в с помощью регистра сдвига;
* ключевая развертка реализована с помощью сети Фейстеля, в которой в качестве функции используется раундовое преобразование исходного алгоритма.

Длина входного блока «Кузнечика» — 128 бит, ключа — 256 бит.

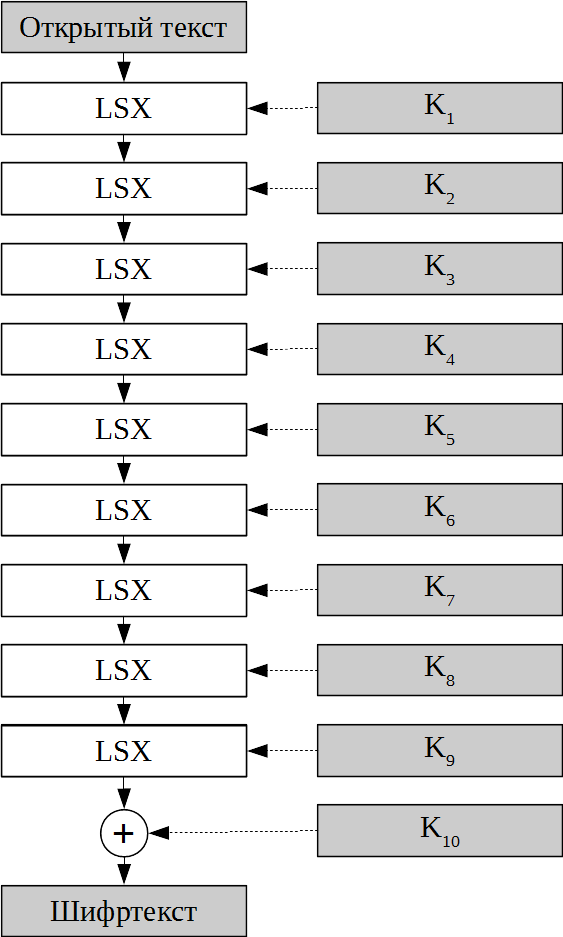
Преобразования

Шифрование основано на последовательном применении нескольких однотипных раундов, каждый из которых содержит три преобразования: сложение с раундовым ключом, преобразование блоком подстановок и линейное преобразование.  
  
128-битный входной вектор очередного раунда складывается побитно с раундовым ключом:  
  
https://habrastorage.org/files/de7/760/18d/de776018de3e4f4b851eaa7c21a63cd9.png  
  
Нелинейное преобразование представляет собой применение к каждому 8-битному подвектору 128-битного входного вектора фиксированной подстановки:  
  
https://habrastorage.org/files/f62/e09/16c/f62e0916c4004ca8bc769475afbf6960.png  
  
В «Кузнечике» используется та же подстановка, что и в хэш-функции «Стрибог».  
  
Линейное преобразование, как я уже сказал, может быть реализовано не только как обычно в блочных шифрах — матрицей, но и с помощью РСЛОС — линейного регистра сдвига с обратной связью, который движется 16 раз.  
  
  
  
Сам регистр реализуется над полем Галуа по модулю неприводимого многочлена степени 8: https://habrastorage.org/files/b0f/a5f/1ae/b0fa5f1aebb14383a09dc7dd493dd6f0.png:  
  
  
  
Раундовое преобразование можно изобразить следующим образом:  
  


Выработка раундовых ключей

Рассмотрим теперь процедуру генерации раундовых ключей из мастер-ключа. Первые два получаются разбиением мастер-ключа пополам. Далее для выработки очередной пары раундовых ключей используется 8 итераций сети Фейстеля, где, в свою очередь, в качестве раундовых ключей используется счетчиковая последовательность, прошедшая через линейное преобразование алгоритма:  
  
  
https://habrastorage.org/files/a28/642/64e/a2864264e3be46f9ab1fc1d676e04fbf.png  
https://habrastorage.org/files/7e0/487/6de/7e04876de820463bbf83926172848589.png  
  
Раунд ключевой развертки можно представить следующим образом:  
  
  
  
А всю процедуру выработки раундовых ключей так:  
  


Шифрование и дешифрование

В результате, шифрование одного 128-битного входного блока описывается следующим уравнением:  
  
https://habrastorage.org/files/c97/5db/8e3/c975db8e31a446fa985a992f101aeb1d.png  
  
А в виде блок-схемы может быть представлено так:  
  
  
  
Расшифрование реализуется обращением базовых преобразований и применением их в обратном порядке:  
  
https://habrastorage.org/files/c30/ac9/a84/c30ac9a84ac1437492cd0d744de7c722.png

ИСТОЧНИК

<https://habrahabr.ru/post/266359/>

<https://habrahabr.ru/post/273895/>

Панасенко С.П. - Алгоритмы шифрования

\*/