Оглавление

[1. Цели и задачи защиты информации 3](#_Toc512007136)

[2. Основные характеристики защищаемой информации 3](#_Toc512007137)

[3. Угрозы безопасности информации – основные типы угроз 3](#_Toc512007138)

[4. Основные группы, составляющие методы и средства защиты информации 3](#_Toc512007139)

[5. Правовое обеспечение информационной безопасности 3](#_Toc512007140)

[6. Инженерно-технические методы и средства защиты информации 3](#_Toc512007141)

[7. Основные принципы построения симметричных шифров 3](#_Toc512007142)

[8. Законодательные и нормативные документы информационной безопасности 4](#_Toc512007143)

[9. Алгоритмы симметричного шифрования: основные понятия и определения. Область применения 4](#_Toc512007144)

[10. Сеть Фейстеля как метод построения алгоритмов симметричного шифрования 4](#_Toc512007145)

[11. Алгоритм шифрования TEA 5](#_Toc512007146)

[12. Алгоритм шифрования DES 6](#_Toc512007147)

[13. Режимы выполнения алгоритмов симметричного шифрования 6](#_Toc512007148)

[14. Алгоритм шифрования ГОСТ 28147-89 8](#_Toc512007149)

[15. Алгоритм шифрования 3DES 9](#_Toc512007150)

[16. Алгоритм шифрования DESX 9](#_Toc512007151)

[17. Алгоритм шифрования AES 9](#_Toc512007152)

[18. Потоковое шифрование 11](#_Toc512007153)

[19. Алгоритм RC4 11](#_Toc512007154)

[20. Криптографические хеш-функции 12](#_Toc512007155)

[21. Хеш-функции на основе блочных шифров 13](#_Toc512007156)

[22. Функция хеширования MD4 13](#_Toc512007157)

[23. Функция хеширования ГОСТ Р 34.11-94 15](#_Toc512007158)

[24. Основные теоремы теории чисел 15](#_Toc512007159)

[25. Наибольший общий делитель. Алгоритмы Евклида 16](#_Toc512007160)

[26. Односторонняя функция 16](#_Toc512007161)

[27. Криптография с открытым ключом 16](#_Toc512007162)

[28. Задача распределения ключей 16](#_Toc512007163)

[29. Метод Диффи – Хеллмана 16](#_Toc512007164)

[30. Алгоритм шифрования Шамира 16](#_Toc512007165)

[31. Алгоритм шифрования Эль-Гамаля 16](#_Toc512007166)

[32. Алгоритм шифрования RSA 16](#_Toc512007167)

[33. Комбинированные криптосистемы 16](#_Toc512007168)

[34. Электронная цифровая подпись 16](#_Toc512007169)

[35. Алгоритм цифровой подписи RSA 16](#_Toc512007170)

[36. Алгоритм цифровой подписи Эль-Гамаля 17](#_Toc512007171)

[37. Алгоритм цифровой подписи DSA 17](#_Toc512007172)

[38. Алгоритм цифровой подписи ГОСТ 3410-94 17](#_Toc512007173)

[39. Инфраструктура открытых ключей 17](#_Toc512007174)

[40. Сертификат открытого ключа 17](#_Toc512007175)

[41. Идентификация, аутентификация, авторизация 17](#_Toc512007176)

[42. Методы аутентификации, использующие одноразовые и многоразовые пароли 17](#_Toc512007177)

[43. Методы аутентификации, использующие симметричные и ассиметричные алгоритмы 17](#_Toc512007178)

[44. Биометрическая аутентификация пользователя 17](#_Toc512007179)

[45. Межсетевые экраны. Функции межсетевых экранов 18](#_Toc512007180)

[46. Основные типы межсетевых экранов 18](#_Toc512007181)

[47. Способы подключения межсетевых экранов 21](#_Toc512007182)

[48. Виртуальные частные сети 22](#_Toc512007183)

[49. Виды виртуальных частных сетей 22](#_Toc512007184)

[50. Протокол SSL/TLS. Применение и основные положения протокола 23](#_Toc512007185)

[51. Протокол SKIP. Применение и основные положения протокола 26](#_Toc512007186)

[52. Протокол IPSec. Применение и основные положения протокола 27](#_Toc512007187)

[53. Протокол SET. Применение и основные положения протокола 29](#_Toc512007188)

### Цели и задачи защиты информации

Информация – сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления. Формы представления информации:

* Речевая
* Телекоммуникационная
* Документированная

### Основные характеристики защищаемой информации

К защищаемой информации относится информация, являющаяся предметов собственности и подлежащая защите.

Информационная безопасность – комплекс мероприятий, которые обеспечивают для охватываемой им информации следующие факторы:

* Конфиденциальность
* Целостность
* Доступность

### Угрозы безопасности информации – основные типы угроз

Угрозы безопасности информации – событие или действие, которое может вызвать нарушение в работе информационной системы. Разделяют два вида угроз:

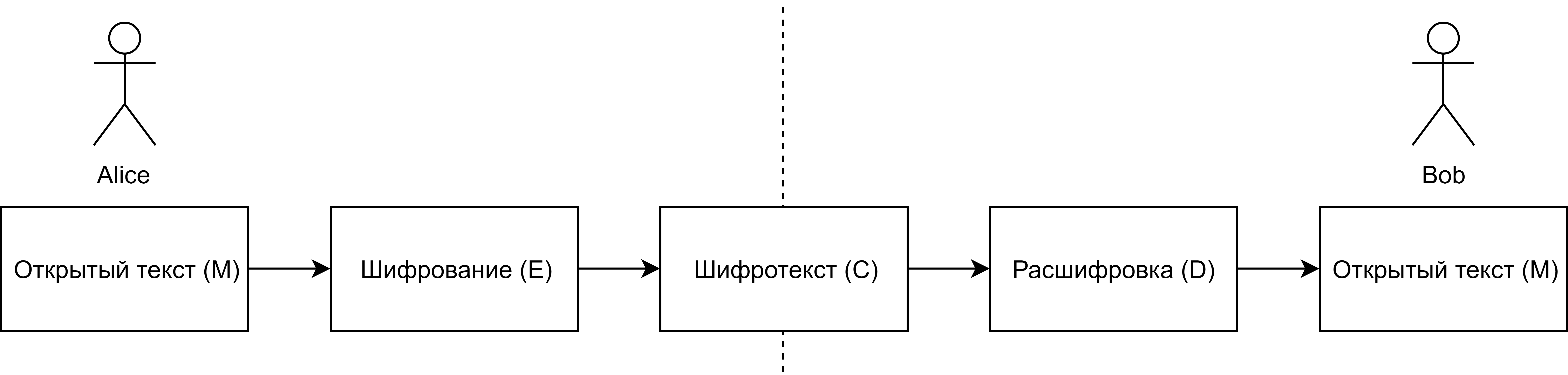
* 1. Естественные угрозы
  2. Искусственные угрозы
* Непреднамеренные
* Преднамеренные

### Основные группы, составляющие методы и средства защиты информации

### Правовое обеспечение информационной безопасности

### Инженерно-технические методы и средства защиты информации

### Основные принципы построения симметричных шифров



Попытка получить открытый текст при неизвестном криптографическом алгоритме называется дешифрацией, а лицо, пытающиеся получить информацию называется атакующим.

Криптографический алгоритм – последовательность математический операций, позволяющих зашифровать открытый тест или расшифровать шифротекст.

Существует два типа криптографических алгоритмов, основанных на использовании ключа: симметричные (один ключ при шифровании и расшифровке) и ассиметричные (разные ключи при шифровании и расшифровке)

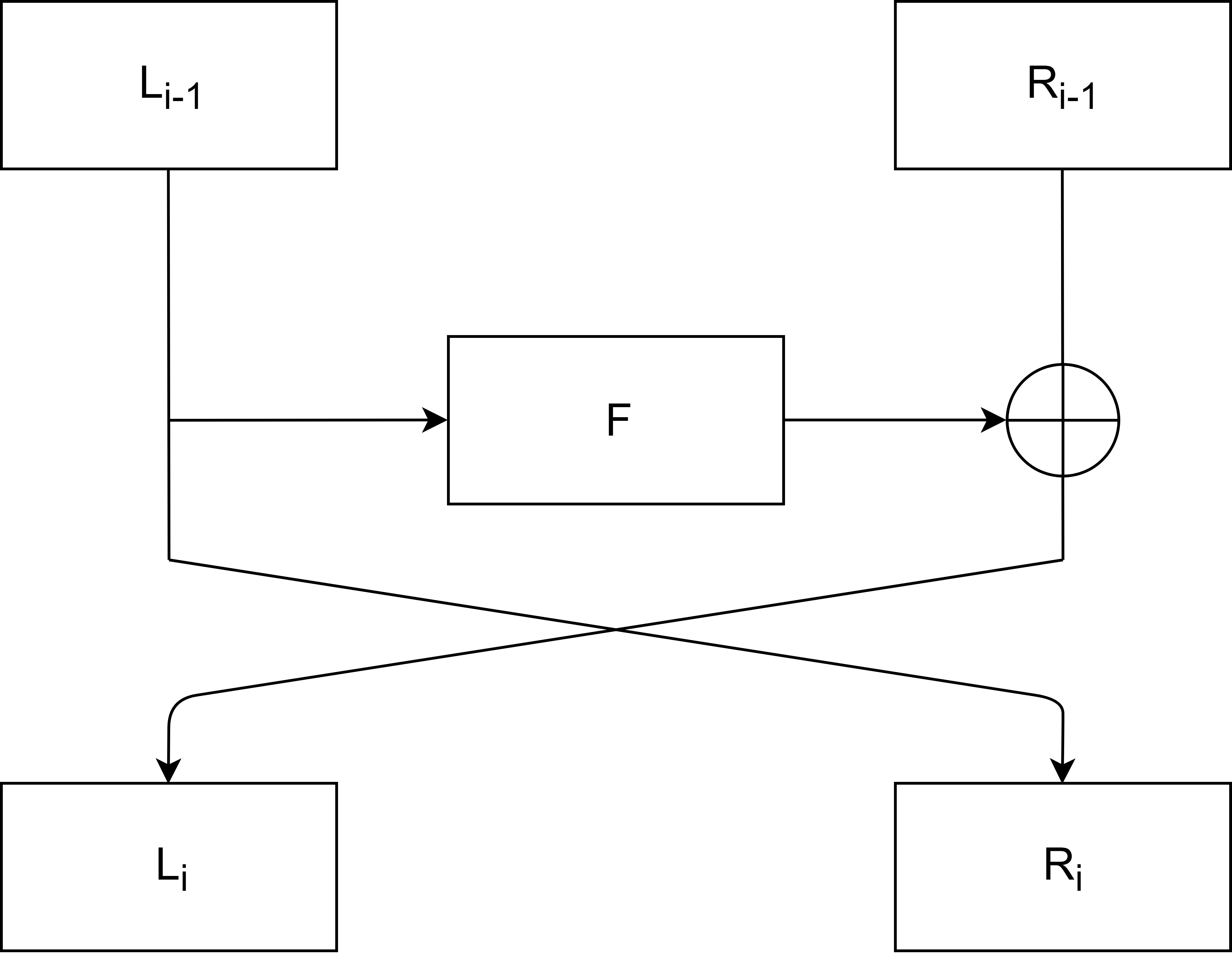
|  |  |  |
| --- | --- | --- |
| Сложение |  |  |
| Сложение по модулю 2 |  |  |
| Умножение по модулю |  |  |
| Циклические сдвиги |  |  |
|  |  |  |
| Табличная подстановка |  |  |

**Надежность схемы шифрования зависит только от секретности ключа и не зависит от алгоритма шифрования (расшифровки)**

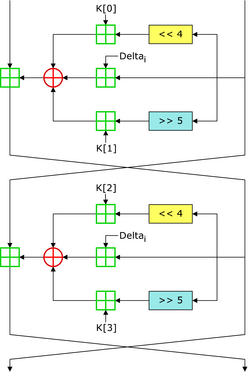
### Законодательные и нормативные документы информационной безопасности

### Алгоритмы симметричного шифрования: основные понятия и определения. Область применения

### Сеть Фейстеля как метод построения алгоритмов симметричного шифрования



### Алгоритм шифрования TEA

Tiny Encryption Algorithm (TEA) — блочный алгоритм шифрования типа «Сеть Фейстеля». Широко используется в ряде криптографических приложений и широком спектре аппаратного обеспечения благодаря крайне низким требованиям к памяти и простоте реализации. Впервые представлен в 1994 году.

Характеристики:

* Основан на сети Фейстеля
* Размер блока 64 бит
* Длина ключа 128 бит
* 32 раунда по умолчанию

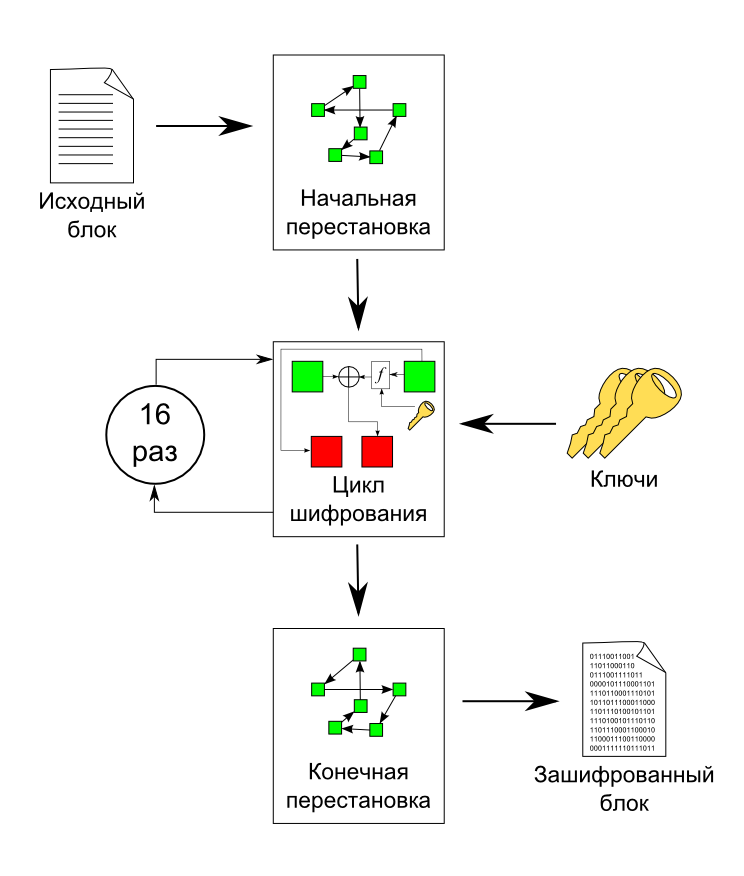
**Шифрование:**

|  |
| --- |
| delta = 2654435769  sum = 0  for i = 1 to 32 do  sum = sum + delta  L = L + ((R << 4) + K[0]) XOR (R + SUM) XOR ((R >> 5) + K[1])  R = R + ((L << 4) + K[2]) XOR (L + SUM) XOR ((L >> 5) + K[3])  return (L, R) |

**Расшифровка:**

|  |
| --- |
| delta = 2654435769  sum = 3337565984  for i = 1 to 32 do  R = R - ((L << 4) + K[2]) XOR (L + SUM) XOR ((L >> 5) + K[3])  L = L - ((R << 4) + K[0]) XOR (R + SUM) XOR ((R >> 5) + K[1])  sum = sum – delta  return (L, R) |

### Алгоритм шифрования DES

Data Encryption Standard (DES) — алгоритм для симметричного шифрования, разработанный фирмой IBM и утверждённый правительством США в 1977 году как официальный стандарт.

Характеристики:

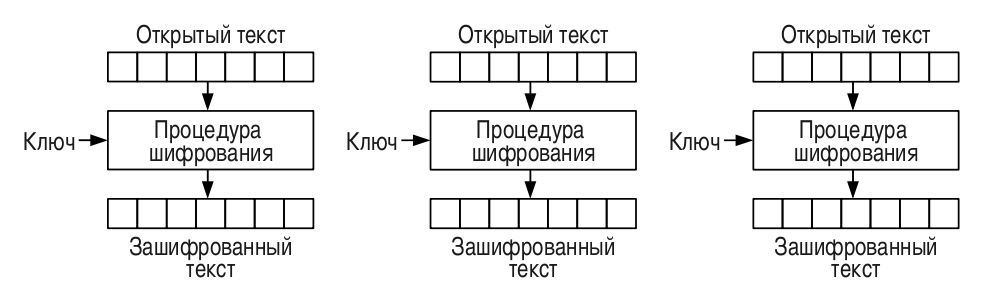
* Основан на сети Фейстеля
* Размер блока 64 бит
* Длина ключа 56 бит
* Количество раундов 16

### Режимы выполнения алгоритмов симметричного шифрования

Режим шифрования — метод применения блочного шифра (алгоритма), позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных. При этом для шифрования одного блока могут использоваться данные другого блока.

Обычно режимы шифрования используются для изменения процесса шифрования так, чтобы результат шифрования каждого блока был уникальным вне зависимости от шифруемых данных и не позволял сделать какие-либо выводы об их структуре. Это обусловлено, прежде всего, тем, что блочные шифры шифруют данные блоками фиксированного размера, и поэтому существует потенциальная возможность утечки информации о повторяющихся частях данных, шифруемых на одном и том же ключе.

**Режим электронной кодовой книги (ECB):**



Сообщение делится на блоки одинакового размера. Размер (длина) блока равен ***n*** и измеряется в битах. В результате получается последовательность. Последний блок при необходимости дополняется до длины ***n***. Каждый блок шифруется алгоритмом шифрования с использованием ключа ***k***. Расшифровка выполняется функцией с использованием того же ключа ***k***.

Особенности:

* каждый блок шифруется/расшифровывается независимо от других блоков.

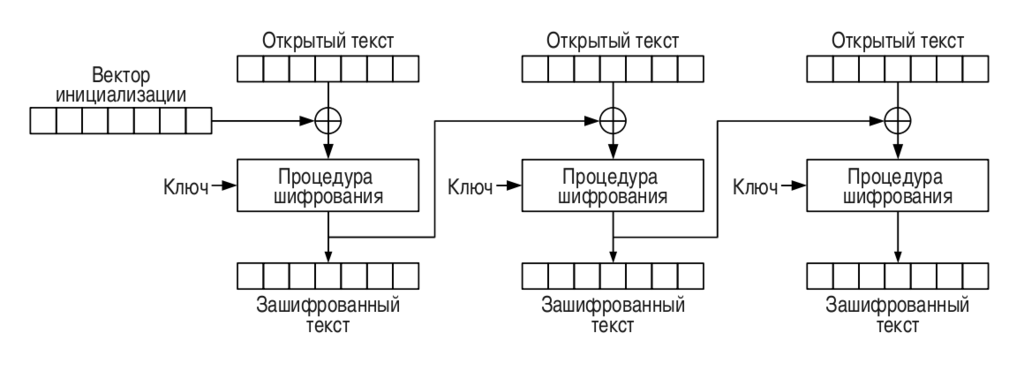
Достоинства:

* постоянная скорость обработки блоков (скорость определяется эффективностью реализации шифра);
* возможно распараллеливание вычислений (так как блоки не связаны между собой).

Недостатки:

* сохранение статистических особенностей открытого текста (поскольку одинаковым блокам шифротекста соответствуют одинаковые блоки открытого текста).

**Режим сцепления блоков (CBC):**



Шифрование очередного (i-го) блока выполняется с использованием предыдущего зашифрованного ((i-1)-го) блока. Для первого блока предыдущего зашифрованного блока не существует, поэтому первый блок шифруют с использованием «вектора инициализации». Размер (длина) вектора равна размеру блока (n). Расшифровка выполняется функцией с использованием тех же ключа k и вектора инициализации.

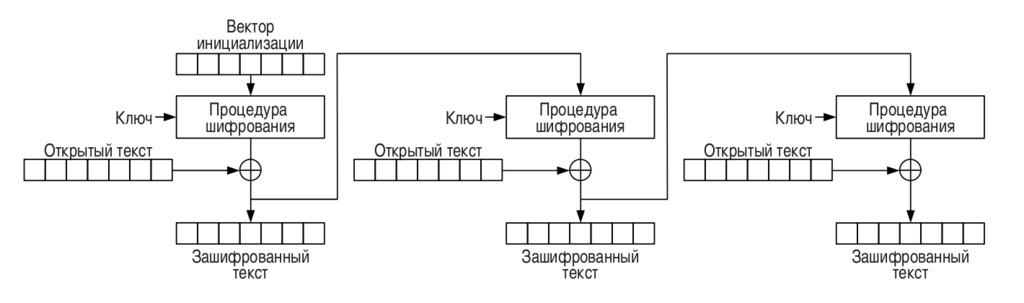
Достоинства:

* постоянная скорость обработки блоков (скорость определяется эффективностью реализации шифра; время выполнения операции «xor» пренебрежимо мало);
* отсутствие статистических особенностей, характерных для режима ECB (поскольку каждый блок открытого текста «смешивается» с блоком шифротекста, полученным на предыдущем шаге шифрования);

Недостатки:

* невозможность распараллеливания шифрования (поскольку для шифрования каждого i-го блока требуется блок, зашифрованный на предыдущем шаге (блоки связаны между собой)).

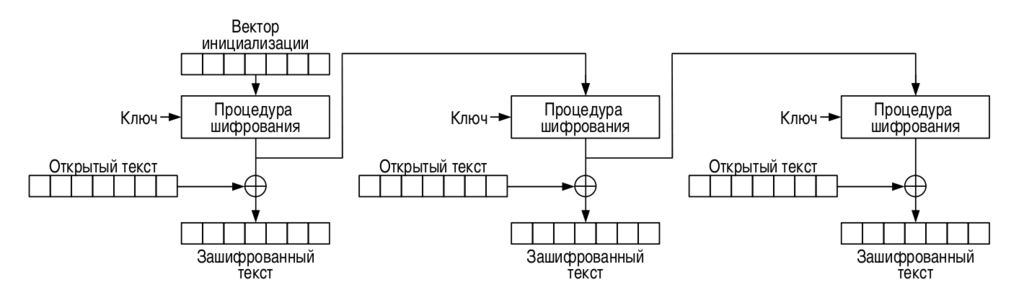
**Режим обратной связи по шифротексту (CFB):**



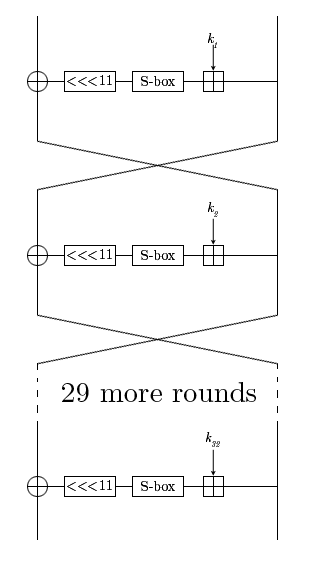
Во время шифрования каждый блок открытого текста складывается по модулю 2 с блоком, зашифрованным на предыдущем шаге. Блоки открытого текста «смешиваются» («маскируются») с блоками шифротекста. Если в режиме СFВ с полноблочной обратной связью имеется два идентичных блока шифротекста, результат, например, шифрования алгоритмом DES на следующем шаге будет тем же. Скорость шифрования режима СFВ с полноблочной обратной связью та же, что и у блочного шифра, причём возможности распараллеливания процедуры шифрования ограничены.

**Режим обратной связи по выходу (OFB):**

Особенностью режима является то, что в качестве входных данных для алгоритма блочного шифрования не используется само сообщение. Вместо этого блочный шифр используется для генерации псевдослучайного потока байтов, который с помощью операции XOR складывается с блоками открытого текста.



### Алгоритм шифрования ГОСТ 28147-89

**ГОСТ 28147-89** — российский стандарт симметричного блочного шифрования, принятый в 1989 году.

Характеристики:

* Основан на сети Фейстеля
* Размер блока 64 бита
* Размер ключа 256 бит (8 раундовых ключей по 32 бита)
* 32 раунда
* 8 таблиц замены

Для генерации подключей исходный 256-битный ключ разбивается на восемь 32-битных блоков: K1…K8.

Ключи K9…K24 являются циклическим повторением ключей K1…K8 (нумеруются от младших битов к старшим). Ключи K25…K32 являются ключами K8…K1.

Ri и Ki складываются по модулю .

Результат разбивается на восемь 4-битовых подпоследовательностей, каждая из которых поступает на вход своего узла таблицы замен (в порядке возрастания старшинства битов), называемого ниже S-блоком. Общее количество S-блоков стандарта — восемь, то есть столько же, сколько и подпоследовательностей. Каждый S-блок представляет собой перестановку чисел от 0 до 15 (конкретный вид S-блоков в стандарте не определен).

Расшифровывание выполняется так же, как и зашифровывание, но инвертируется порядок подключей Ki.

### Алгоритм шифрования 3DES

Triple DES (3DES) — симметричный блочный шифр, созданный в 1978 году на основе алгоритма DES с целью устранения главного недостатка последнего — малой длины ключа (56 бит), который может быть взломан методом полного перебора ключа. Скорость работы 3DES в 3 раза ниже, чем у DES, но криптостойкость намного выше — время, требуемое для криптоанализа 3DES, может быть в миллиард раз больше, чем время, нужное для вскрытия DES.

**Схемы шифрования:**

* **DES – EEE3**. Шифруется три раза с тремя разными ключами (операции шифрование-шифрование-шифрование).
* **DES – EDE3**. Операции шифровка-расшифровка-шифровка с тремя разными ключами.
* **DES – EEE2**. Шифруется три раза с двумя разными ключами (первом и третьем шаге используется одинаковый ключ)
* **DES – EDE2**. Операции шифровка-расшифровка-шифровка с двумя разными ключами (первом и третьем шаге используется одинаковый ключ)

3DES с тремя различными ключами имеет длину ключа, равную 168 бит, но из-за атак «встреча посередине» эффективная криптостойкость составляет только 112 бит. В варианте DES-EDE c двумя ключами номинальный размер ключа также составляет 112 бит. Однако, такой выбор ключей уязвим к определенным атакам на основе открытых текстов, что уменьшает эффективную длину ключа до 80 бит.

### Алгоритм шифрования DESX

DESX (DES extended) - симметричный алгоритм шифрования, разработанный на основе блочного шифра DES (Data Encryption Standard). Данный алгоритм использует метод отбеливания ключа с целью усиления устойчивости к атакам на основе полного перебора.

Суть алгоритма состоит в том, что перед выполнением однократного DES и после него на данные операцией XOR накладываются различные 64-битные фрагменты ключа.

Таким образом, длина ключа увеличивается до 56 + 2 × 64 = 184 бит.

### Алгоритм шифрования AES

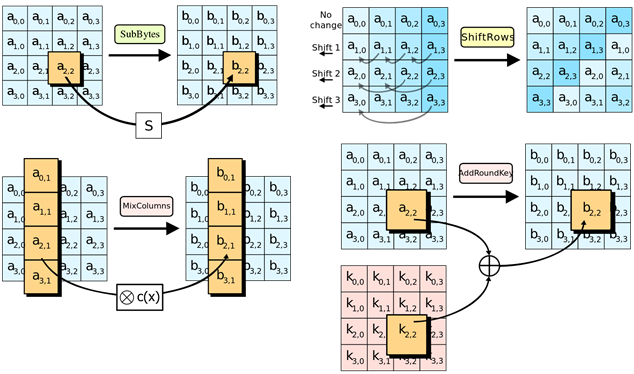
***Advanced Encryption Standard (AES)***, также известный как ***Rijndael*** — симметричный алгоритм блочного шифрования, принятый в качестве стандарта шифрования правительством США по результатам конкурса AES. Этот алгоритм хорошо проанализирован и сейчас широко используется, как это было с его предшественником DES. Национальный институт стандартов и технологий США 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.

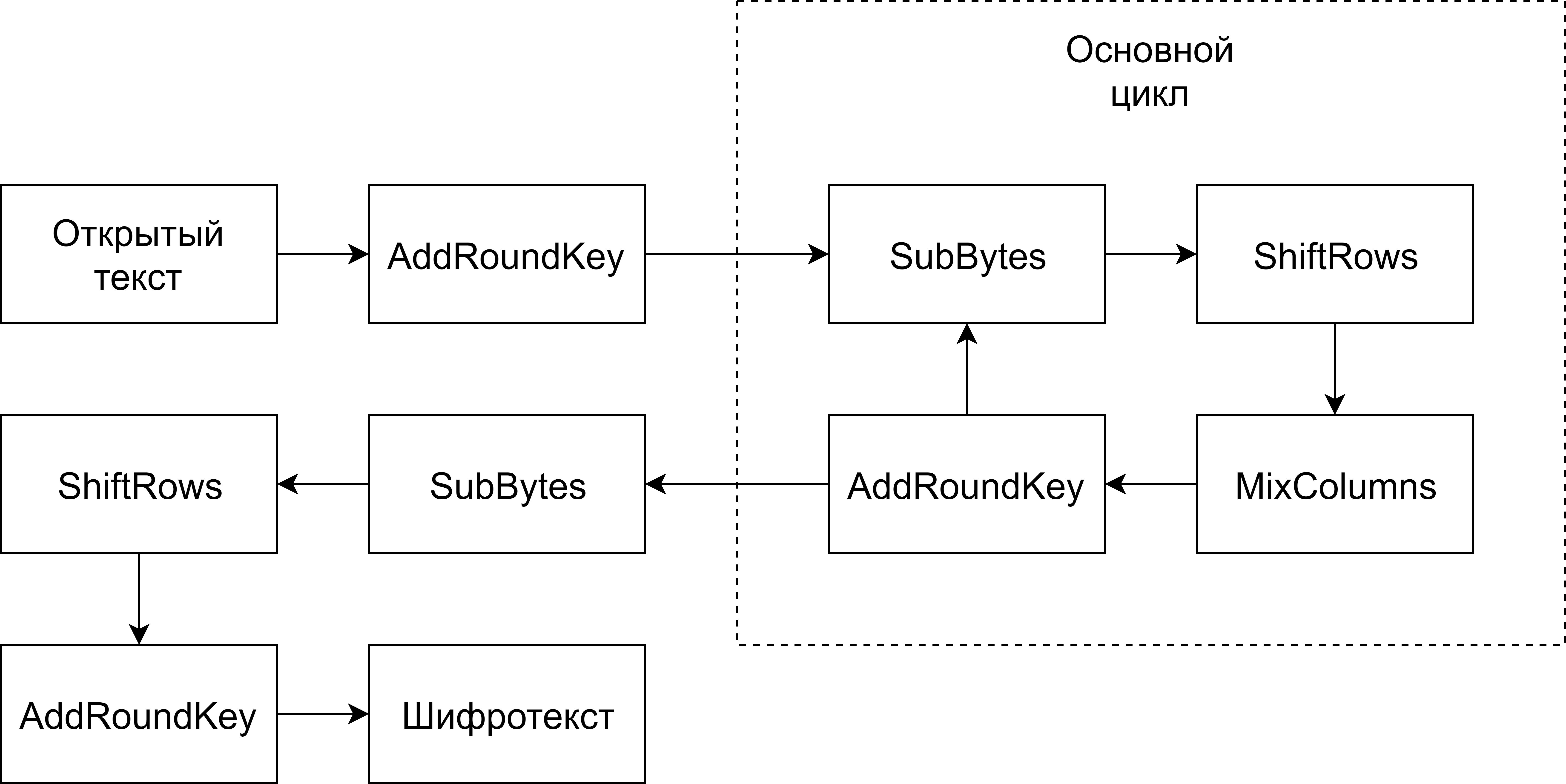
Характеристики:

* Размер блока 128 бит.
* Размер ключа 128/192/256 бит.
* 10/12/14 раундов.

Основные операции:

* SubBytes (SB) – табличная замена каждого байта массива. Обрабатывает каждый байт состояния, независимо производя нелинейную замену байтов используя таблицу замен.
* ShiftRows (SR) – сдвиг строк массива. При этой трансформации строки состояния циклически сдвигаются на ***n*** байт по горизонтали, в зависимости от номера строки. Для нулевой строки ***n*** = 0, для первой строки ***n*** = 1 Б и т. д.
* MixColumns (MC) – каждая колонка перемножается с фиксированным многочленом c(x).
* AddRoundKey (AK) – каждый байт состояния объединяется с ключом, используя операцию XOR.





### Потоковое шифрование

Потоковый шифр — это симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста.

***Шифрование***:

, где

– элемент открытого текста

– элемент ключа

– элемент зашифрованного текста

***Расшифровка:***

*.*

Элементы ключа создаются с помощью генераторов случайных чисел. Основные требования к генераторам:

* Период последовательности должен быть очень большим
* Порожденная последовательность должна быть неотличима от случайной

Виды генераторов:

* Линейный конгруэнтный генератор
* Генератор Парка – Миллера
* Квадратичный генератор
* Кубический генератор

### Алгоритм RC4

RC4 — потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколах SSL и TLS, алгоритмах обеспечения безопасности беспроводных сетей WEP и WPA).

Особенности:

* Работает с n-битными словами (обычно n=8) ()
* L – длина ключа ()
* Состоит из двух этапов: подготовительный (инициализация S-блока) и основной (генерация псевдослучайного ключа )

***Инициализация S-блока:***

|  |
| --- |
| for i = 0 to - 1 S[i] := i  j := 0  for i = 0 to - 1  j := (j + S[i] + Key[i mod L]) mod  поменять местами S[i] и S[j]  endfor |

***Пример: n = 3, K = 25 (L = 2)***

|  |  |  |
| --- | --- | --- |
| ***i*** | ***j*** | ***S*** |
|  | 0 | 0, 1, 2, 3, 4, 5, 6, 7 |
| 0 | 0+0+2=2 | 2, 1, 0, 3, 4, 5, 6, 7 |
| 1 | 2+1+5=0 | 1, 2, 0, 3, 4, 5, 6, 7 |
| 2 | 0+0+2=2 | 1, 2, 0, 3, 4, 5, 6, 7 |
| 3 | 2+3+5=2 | 1, 2, 3, 0, 4, 5, 6, 7 |
| 4 | 2+4+2=0 | 4, 2, 3, 0, 1, 5, 6, 7 |
| 5 | 0+5+5=2 | 4, 2, 5, 0, 1, 3, 6, 7 |
| 6 | 2+6+2=2 | 4, 2, 6, 0, 1, 3, 5, 7 |
| 7 | 2+7+5=6 | 4, 2, 6, 0, 1, 3, 7, 5 |

***Генерация псевдослучайного ключа :***

|  |
| --- |
| i := 0; j := 0  while:  i := (i + 1) mod  j := (j + S[i]) mod  поменять местами S[i] и S[j]  t := ( S[i] + S[j] ) mod  K := S[t]  endwhile |

***Пример (продолжение):***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***i*** | ***j*** | ***t*** | ***S*** | ***K*** |
| 0 | 0 |  | 4, 2, 6, 0, 1, 3, 7, 5 |  |
| 1 | 0+2=2 | 0 | 4, 6, 2, 0, 1, 3, 7, 5 | 4 |
| 2 | 2+2=4 | 3 | 4, 6, 1, 0, 2, 3, 7, 5 | 0 |
| … | | | | |

### Криптографические хеш-функции

Криптографические хеш-функции — это функция, выполняющая необратимое преобразование данных. Обладает следующими свойствами:

* На вход может поступать двоичный блок данных производной длины
* На выходе получается двоичной блок данных фиксированной длины
* Значения на выходе распределяются по равномерному закону по всему диапазону возвращаемых значений
* При изменении хотя-бы 1 бита на входе выход значительно изменяется

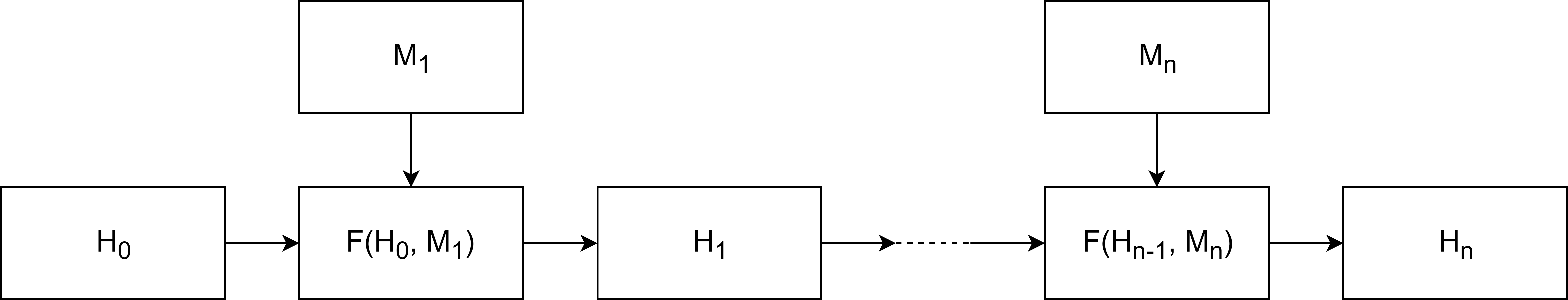
Используются при:

* Формировании контрольных кодов (проверка целостности информации)
* Организация парольных систем
* Генерация ключевой информации
* Реализация протоколов электронной подписи

Требования к хэш-функциям:

* Для любого значения M вычисление ХФ должно выполняться относительно быстро
* При известном значении ХФ должно быть вычислительно трудно найти M для которого
* При известном M трудно найти M’, такое, что
* Вычислительно трудно найти пару M и M’ для которых

***Структура Меркла-Дамгарда*** — метод построения криптографических хеш-функций



### Хеш-функции на основе блочных шифров

Очередной блок текста подается в качестве ключа, а хэш-значение предыдущего шага -- в качестве входного блока. Выход алгоритма блочного шифрования является текущим хэш-значением.

* Схема Майера – Матиаса

|  |
| --- |
| h = 0  for i = 0 to n do    endfor |

* Схема Девиса – Майера

|  |
| --- |
| h = 0  for i = 0 to n do    endfor |

* Схема Миягучи – Пренеля

|  |
| --- |
| h = 0  for i = 0 to n do    endfor |

### Функция хеширования MD4

MD4 (Message Digest 4) — криптографическая хеш-функция, разработанная в 1990 году.

Характеристики:

* Размер блока 512 бит
* Размер хэша 128 бит
* 3 раунда

***Алгоритм:***

***Шаг 1. Добавление недостающих битов***



Сообщение расширяется так, чтобы его длина в битах по модулю 512 равнялась 448. Таким образом, в результате расширения, сообщению недостает 64 бита до длины, кратной 512 битам. Расширение производится всегда, даже если сообщение изначально имеет нужную длину.

Расширение производится следующим образом: один бит, равный 1, добавляется к сообщению, а затем добавляются биты, равные 0, до тех пор, пока длина сообщения не станет равной 448 по модулю 512. В итоге, к сообщению добавляется как минимум 1 бит, и как максимум 512.

***Шаг 2. Инициализация MD буфера***

Для вычисления хеша сообщения используется буфер, состоящий из 4 слов:

|  |
| --- |
| H1 = ‘01 23 45 67’  H2 = ‘89 ab cd ef’  H3 = ‘fe dc ba 98’  H4 = ‘76 54 32 10’ |

***Шаг 3. Обработка блока данных***

Для начала определим три вспомогательные функции, каждая из которых получает на вход три 32-битных слова, и по ним вычисляет одно 32-битное слово.

|  |
| --- |
| *,*  ,  . |

|  |
| --- |
| A, B, C, D = H1, H2, H3, H4  // 1 раунд  for i = 0 to 15 do  T = A + ***F***(B, C, D) + M[Zi] + Ki  A, B, C, D = D, T << Si, B, C  endfor  // 2 раунд  for i = 16 to 32 do  T = A + ***G***(B, C, D) + M[Zi] + Ki  A, B, C, D = D, T << Si, B, C  endfor  // 3 раунд  for i = 0 to 15 do  T = A + ***H***(B, C, D) + M[Zi] + Ki  A, B, C, D = D, T << Si, B, C  endfor  H1 = H1 + A  H2 = H2 + B  H3 = H3 + C  H4 = H4 + D |

***Шаг 4. Формирование хэша***

Результат (хеш-функция) получается как H = (H1, H2, H3, H4).

***Сравнение с MD5:***

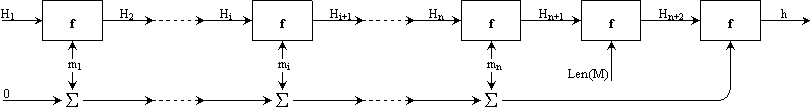
* MD4 использует три цикла из 16 шагов каждый, в то время как MD5 использует четыре цикла из 16 шагов каждый.
* MD5 использует четыре элементарные логические функции, по одной на каждом цикле, по сравнению с тремя в MD4, по одной на каждом цикле.
* В MD5 на каждом шаге текущий результат складывается с результатом предыдущего шага.

### Функция хеширования ГОСТ Р 34.11-94

***ГОСТ Р 34.11-94 –*** российский криптографический стандарт вычисления хэш-функции. Введен в 1994 году.

Характеристики:

* На основе алгоритма Меркла – Дамгарда
* Размер блока 256 бит
* Размер хэша 256 бит
* 1 раунд



### Основные теоремы теории чисел

***Определение 2.*** Два числа называются взаимно простыми, если они не имеют общих делителей кроме 1.

***Теорема 1.*** Любое целое положительное число может быть представлено как произведение простых чисел, причем единственным способом.

***Теорема 2.*** Если ***p*** – простое число, то .

***Теорема 3.*** Пусть ***p*** и ***q*** – простые числа (***p***≠***q***), тогда .

***Теорема 4.*** Для всякого простого числа ***p*** и натурального числа ***a*** (***a***<***p***), взаимно простого с ***p***:

***Теорема 5.*** ***a*** и ***n*** – натуральные взаимно простые числа, тогда

***Теорема 6.*** Пусть ***p*** и ***q*** – простые числа (***p***≠***q***), ***a*** – натуральное число, тогда

***Теорема 7.*** Пусть ***a*** и ***b*** – два целых положительных числа, тогда существуют целые ***x*** и ***y***, такие, что:

### Наибольший общий делитель. Алгоритмы Евклида

### Односторонняя функция

### Криптография с открытым ключом

### Задача распределения ключей

### Метод Диффи – Хеллмана

### Алгоритм шифрования Шамира

### Алгоритм шифрования Эль-Гамаля

### Алгоритм шифрования RSA

### Комбинированные криптосистемы

### Электронная цифровая подпись

### Алгоритм цифровой подписи RSA

### Алгоритм цифровой подписи Эль-Гамаля

### Алгоритм цифровой подписи DSA

### Алгоритм цифровой подписи ГОСТ 3410-94

### Инфраструктура открытых ключей

### Сертификат открытого ключа

### Идентификация, аутентификация, авторизация

### Методы аутентификации, использующие одноразовые и многоразовые пароли

### Методы аутентификации, использующие симметричные и ассиметричные алгоритмы

### Биометрическая аутентификация пользователя

Биометрические системы аутентификации — системы аутентификации, использующие для удостоверения личности людей их биометрические данные.

Биометрическая аутентификация — процесс доказательства и проверки подлинности заявленного пользователем имени, через предъявление пользователем своего биометрического образа и путём преобразования этого образа в соответствии с заранее определённым протоколом аутентификации.

Не следует путать данные системы с системами биометрической идентификации, каковыми являются, к примеру, системы распознавания лиц водителей и биометрические средства учёта рабочего времени. Биометрические системы аутентификации работают в активном, а не пассивном режиме и почти всегда подразумевают авторизацию. Хотя данные системы не идентичны системам авторизации, они часто используются совместно (например, в дверных замках с проверкой отпечатка пальца).

Критерии для биометрических параметров. Они обязаны соответствовать следующим пунктам:

* Всеобщность: Данный признак должен присутствовать у всех людей без исключения.
* Уникальность: Биометрия отрицает существование двух людей с одинаковыми физическими и поведенческими параметрами.
* Постоянство: для корректной аутентификации необходимо постоянство во времени.
* Измеримость: специалисты должны иметь возможность измерить признак каким-либо устройством для дальнейшего занесения в базу данных.
* Приемлемость: общество не должно быть против сбора и измерения биометрического параметра.

### Межсетевые экраны. Функции межсетевых экранов

Межсетевой экран, сетевой экран, брандмауэр (нем. Brandmauer), файрволл (англ. Firewall) — программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами.

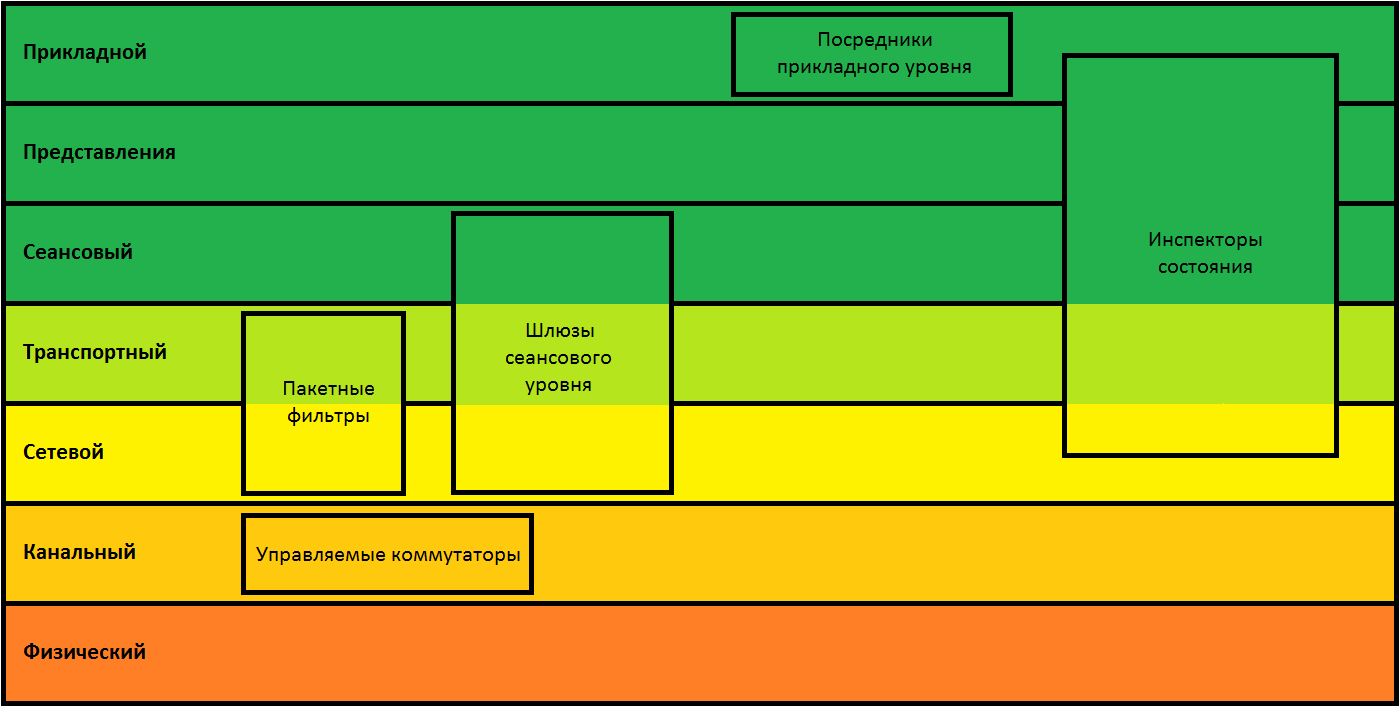
Среди задач, которые решают межсетевые экраны, основной является защита сегментов сети или отдельных хостов от несанкционированного доступа с использованием уязвимых мест в протоколах сетевой модели OSI или в программном обеспечении, установленном на компьютерах сети. Межсетевые экраны пропускают или запрещают трафик, сравнивая его характеристики с заданными шаблонами.

Наиболее распространённое место для установки межсетевых экранов — граница периметра локальной сети для защиты внутренних хостов от атак извне. Однако атаки могут начинаться и с внутренних узлов — в этом случае, если атакуемый хост расположен в той же сети, трафик не пересечёт границу сетевого периметра, и межсетевой экран не будет задействован. Поэтому в настоящее время межсетевые экраны размещают не только на границе, но и между различными сегментами сети, что обеспечивает дополнительный уровень безопасности.

### Основные типы межсетевых экранов

До сих пор не существует единой и общепризнанной классификации межсетевых экранов. Однако в большинстве случаев поддерживаемый уровень сетевой модели OSI является основной характеристикой при их классификации. Учитывая данную модель, различают следующие типы межсетевых экранов:

* Управляемые коммутаторы.
* Пакетные фильтры.
* Шлюзы сеансового уровня.
* Посредники прикладного уровня.
* Инспекторы состояния.



#### Управляемые коммутаторы

Управляемые коммутаторы иногда причисляют к классу межсетевых экранов, так как они осуществляют фильтрацию трафика между сетями или узлами сети. Однако они работают на канальном уровне и разделяют трафик в рамках локальной сети, а значит не могут быть использованы для обработки трафика из внешних сетей (например, из Интернета).

При реализации политики безопасности в рамках корпоративной сети, основу которых составляют управляемые коммутаторы, они могут быть мощным и достаточно дешёвым решением. Взаимодействуя только с протоколами канального уровня, такие межсетевые экраны фильтруют трафик с очень высокой скоростью. Основным недостатком такого решения является невозможность анализа протоколов более высоких уровней.

#### Пакетные фильтры

Пакетные фильтры функционируют на сетевом уровне и контролируют прохождение трафика на основе информации, содержащейся в заголовке пакетов. Многие межсетевые экраны данного типа могут оперировать заголовками протоколов и более высокого, транспортного, уровня (например, TCP или UDP). Пакетные фильтры одними из первых появились на рынке межсетевых экранов и по сей день остаются самым распространённым их типом. Данная технология реализована в подавляющем большинстве маршрутизаторов и даже в некоторых коммутаторах.

При анализе заголовка сетевого пакета могут использоваться следующие параметры:

* IP-адреса источника и получателя;
* тип транспортного протокола;
* поля служебных заголовков протоколов сетевого и транспортного уровней;
* порт источника и получателя.

Пакетные фильтры могут быть реализованы в следующих компонентах сетевой инфраструктуры:

* пограничные маршрутизаторы;
* операционные системы;
* персональные межсетевые экраны.

Так как пакетные фильтры обычно проверяют данные только в заголовках сетевого и транспортного уровней, они могут выполнять это достаточно быстро. Поэтому пакетные фильтры, встроенные в пограничные маршрутизаторы, идеальны для размещения на границе с сетью с низкой степенью доверия. Однако в пакетных фильтрах отсутствует возможность анализа протоколов более высоких уровней сетевой модели OSI. Кроме того, пакетные фильтры обычно уязвимы для атак, которые используют подделку сетевого адреса. Такие атаки обычно выполняются для обхода управления доступом, осуществляемого межсетевым экраном.

#### Шлюзы сеансового уровня

Межсетевой экран сеансового уровня исключает прямое взаимодействие внешних хостов с узлом, расположенным в локальной сети, выступая в качестве посредника (англ. proxy), который реагирует на все входящие пакеты и проверяет их допустимость на основании текущей фазы соединения. Шлюз сеансового уровня гарантирует, что ни один сетевой пакет не будет пропущен, если он не принадлежит ранее установленному соединению. Как только приходит запрос на установление соединения, в специальную таблицу помещается соответствующая информация (адреса отправителя и получателя, используемые протоколы сетевого и транспортного уровня, состояние соединения и т. д.). В случае, если соединение установлено, пакеты, передаваемые в рамках данной сессии, будут просто копироваться в локальную сеть без дополнительной фильтрации. Когда сеанс связи завершается, сведения о нём удаляются из данной таблицы. Поэтому все последующие пакеты, «притворяющиеся» пакетами уже завершённого соединения, отбрасываются.

Так как межсетевой экран данного типа исключает прямое взаимодействие между двумя узлами, шлюз сеансового уровня является единственным связующим элементом между внешней сетью и внутренними ресурсами. Это создаёт видимость того, что на все запросы из внешней сети отвечает шлюз, и делает практически невозможным определение топологии защищаемой сети. Кроме того, так как контакт между узлами устанавливается только при условии его допустимости, шлюз сеансового уровня предотвращает возможность реализации DoS-атаки, присущей пакетным фильтрам.

Несмотря на эффективность этой технологии, она обладает серьёзным недостатком: как и у всех вышеперечисленных классов межсетевых экранов, у шлюзов сеансового уровня отсутствует возможность проверки содержания поля данных, что позволяет злоумышленнику передавать «троянских коней» в защищаемую сеть.

#### Посредники прикладного уровня

Межсетевые экраны прикладного уровня, как и шлюзы сеансового уровня, исключают прямое взаимодействие двух узлов. Однако, функционируя на прикладном уровне, они способны «понимать» контекст передаваемого трафика. Межсетевые экраны, реализующие эту технологию, содержат несколько приложений-посредников (англ. application proxy), каждое из которых обслуживает свой прикладной протокол. Такой межсетевой экран способен выявлять в передаваемых сообщениях и блокировать несуществующие или нежелательные последовательности команд, что зачастую означает DoS-атаку, либо запрещать использование некоторых команд (например, FTP PUT, которая даёт возможность пользователю записывать информацию на FTP сервер).

Посредник прикладного уровня может определять тип передаваемой информации. Например, это позволяет заблокировать почтовое сообщение, содержащее исполняемый файл. Другой возможностью межсетевого экрана данного типа является проверка аргументов входных данных. Например, аргумент имени пользователя длиной в 100 символов либо содержащий бинарные данные является, по крайней мере, подозрительным.

Посредники прикладного уровня способны выполнять аутентификацию пользователя, а также проверять, что SSL-сертификаты подписаны конкретным центром. Межсетевые экраны прикладного уровня доступны для многих протоколов, включая HTTP, FTP, почтовые (SMTP, POP, IMAP), Telnet и другие.

Недостатками данного типа межсетевых экранов являются большие затраты времени и ресурсов на анализ каждого пакета. По этой причине они обычно не подходят для приложений реального времени. Другим недостатком является невозможность автоматического подключения поддержки новых сетевых приложений и протоколов, так как для каждого из них необходим свой агент.

#### Инспекторы состояния

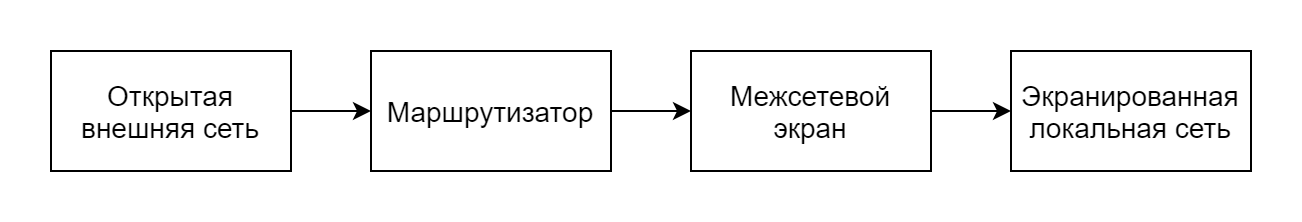
Каждый из вышеперечисленных типов межсетевых экранов используется для защиты корпоративных сетей и обладает рядом преимуществ. Однако, куда эффективней было бы собрать все эти преимущества в одном устройстве и получить межсетевой экран, осуществляющий фильтрацию трафика с сетевого по прикладной уровень. Данная идея была реализована в инспекторах состояний, совмещающих в себе высокую производительность и защищённость. Данный класс межсетевых экранов позволяет контролировать:

* каждый передаваемый пакет — на основе таблицы правил;
* каждую сессию — на основе таблицы состояний;
* каждое приложение — на основе разработанных посредников.

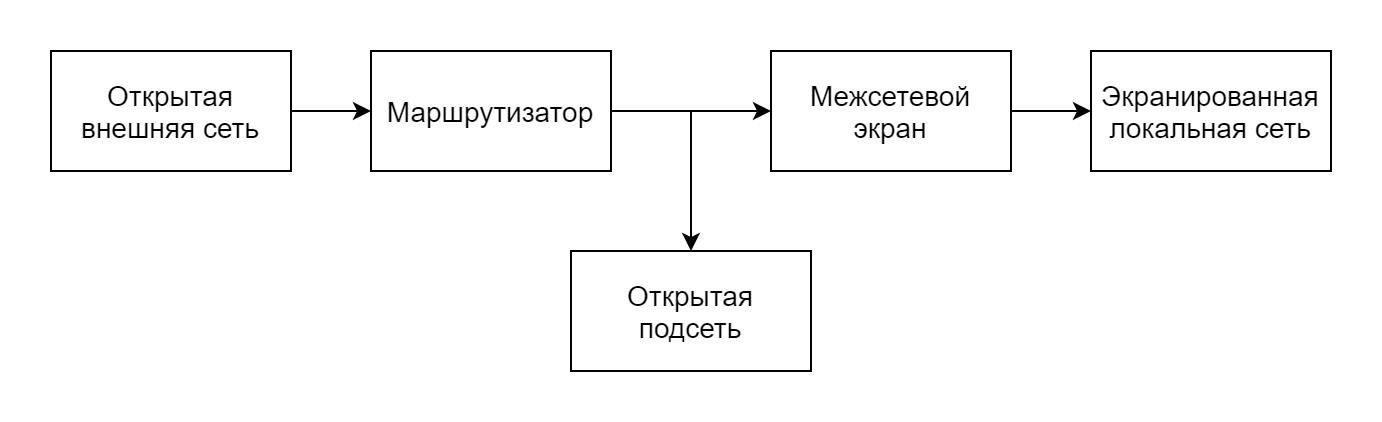
Осуществляя фильтрацию трафика по принципу шлюза сеансового уровня, данный класс межсетевых экранов не вмешивается в процесс установления соединения между узлами. Поэтому производительность инспектора состояний заметно выше, чем у посредника прикладного уровня и шлюза сеансового уровня, и сравнима с производительностью пакетных фильтров. Ещё одно достоинство инспекторов состояния — прозрачность для пользователя: для клиентского программного обеспечения не потребуется дополнительная настройка. Данные межсетевые экраны имеют большие возможности расширения. При появлении новой службы или нового протокола прикладного уровня для его поддержки достаточно добавить несколько шаблонов. Однако инспекторам состояний по сравнению с посредниками прикладного уровня свойственна более низкая защищённость.

### Способы подключения межсетевых экранов

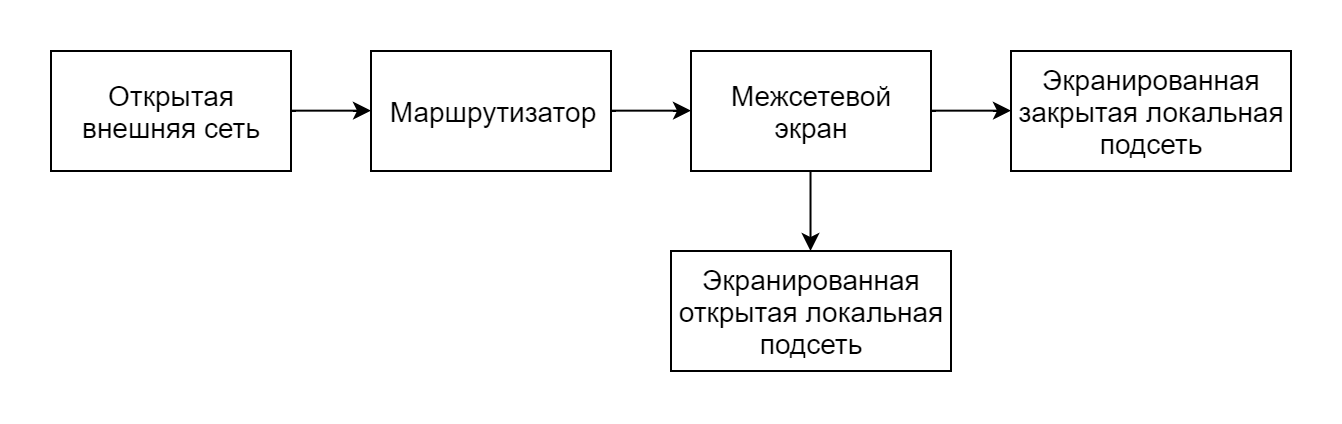
#### Схема единой защищённой локальной сети



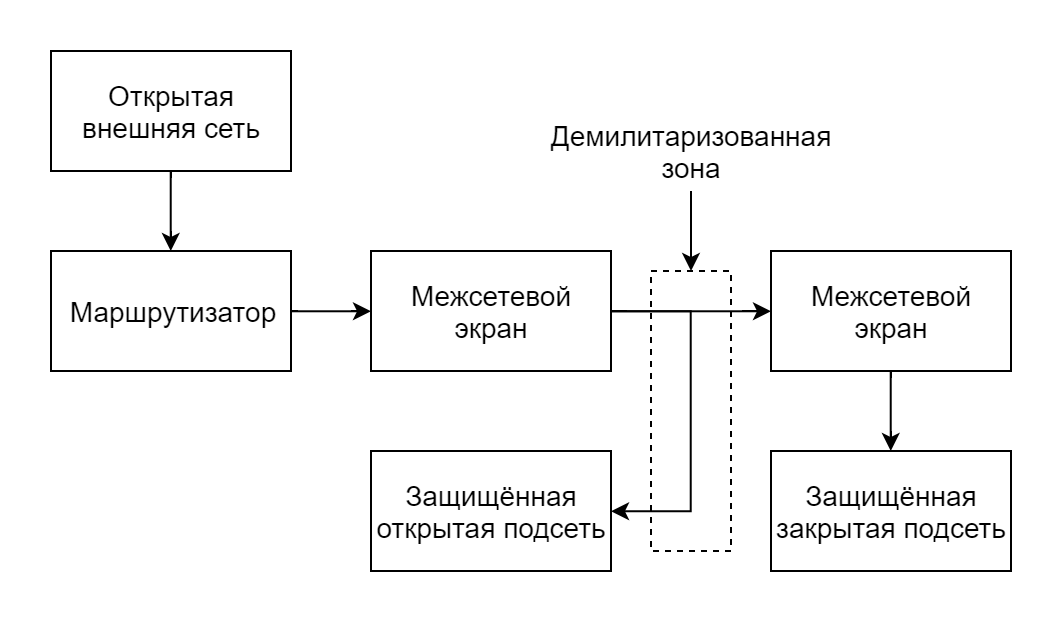
#### Схема с незащищённой открытой и защищённой закрытой подсетями



#### Схема с раздельной защитой закрытой и открытой подсетей на основе межсетевого экрана с тремя сетевыми интерфейсами



#### Схема с раздельной защитой закрытой и открытой подсетей на основе двух межсетевых экранов с двумя сетевыми интерфейсами



### Виртуальные частные сети

VPN (англ. Virtual Private Network — виртуальная частная сеть) — обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например, Интернет). Несмотря на то, что коммуникации осуществляются по сетям с меньшим или неизвестным уровнем доверия (например, по публичным сетям), уровень доверия к построенной логической сети не зависит от уровня доверия к базовым сетям благодаря использованию средств криптографии (шифрования, аутентификации, инфраструктуры открытых ключей, средств для защиты от повторов и изменений передаваемых по логической сети сообщений).

В зависимости от применяемых протоколов и назначения, VPN может обеспечивать соединения трёх видов: узел-узел, узел-сеть и сеть-сеть.

Алгоритм отправки сообщений:

* Извлечь из заголовка IP-адрес, по нему определить алгоритм и ключи;
* Формируется и добавляется цифровая подпись;
* Производится шифрованием всего IP-пакета;
* Зашифрованный IP-пакет вкладывается в новый IP-пакет с адресом получателя следующего пакета.

Алгоритм принятия сообщений:

* Из полученного IP-пакета извлекается адрес отправителя, определяются алгоритм и ключи;
* Зашифрованный пакет извлекается из полученного пакета и расшифровывается;
* Проверяется цифровая подпись;
* Отправка пакета адресату.

### Виды виртуальных частных сетей

Виртуальные частные сети по назначению разделяются на следующие виды:

#### Intranet VPN

Используют для объединения в единую защищённую сеть нескольких распределённых филиалов одной организации, обменивающихся данными по открытым каналам связи.

#### Remote Access VPN

Используют для создания защищённого канала между сегментом корпоративной сети (центральным офисом или филиалом) и одиночным пользователем, который, работая дома, подключается к корпоративным ресурсам с домашнего компьютера, корпоративного ноутбука, смартфона или интернет-киоска.

#### Extranet VPN

Используют для сетей, к которым подключаются «внешние» пользователи (например, заказчики или клиенты). Уровень доверия к ним намного ниже, чем к сотрудникам компании, поэтому требуется обеспечение специальных «рубежей» защиты, предотвращающих или ограничивающих доступ последних к особо ценной, конфиденциальной информации.

#### Internet VPN

Используется для предоставления доступа к интернету провайдерами, обычно если по одному физическому каналу подключаются несколько пользователей. Протокол PPPoE стал стандартом в ADSL-подключениях.

L2TP был широко распространён в середине 2000-х годов в домовых сетях: в те времена внутрисетевой трафик не оплачивался, а внешний стоил дорого. Это давало возможность контролировать расходы: когда VPN-соединение выключено, пользователь ничего не платит. В настоящее время (2012) проводной интернет дешёвый или безлимитный, а на стороне пользователя зачастую есть маршрутизатор, на котором включать-выключать интернет не так удобно, как на компьютере. Поэтому L2TP-доступ отходит в прошлое.

#### Client/Server VPN

Он обеспечивает защиту передаваемых данных между двумя узлами (не сетями) корпоративной сети. Особенность данного варианта в том, что VPN строится между узлами, находящимися, как правило, в одном сегменте сети, например, между рабочей станцией и сервером. Такая необходимость очень часто возникает в тех случаях, когда в одной физической сети необходимо создать несколько логических сетей. Например, когда надо разделить трафик между финансовым департаментом и отделом кадров, обращающихся к серверам, находящимся в одном физическом сегменте. Этот вариант похож на технологию VLAN, но вместо разделения трафика используется его шифрование.

### Протокол SSL/TLS. Применение и основные положения протокола

SSL (Secure Sockets Layer) и его преемник TLS (Transport Layer Security) – криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети интернет. Они используют асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

В 2014 году правительство США сообщило об уязвимости в текущей версии протокола. SSL должен быть исключен из работы в пользу TLS.

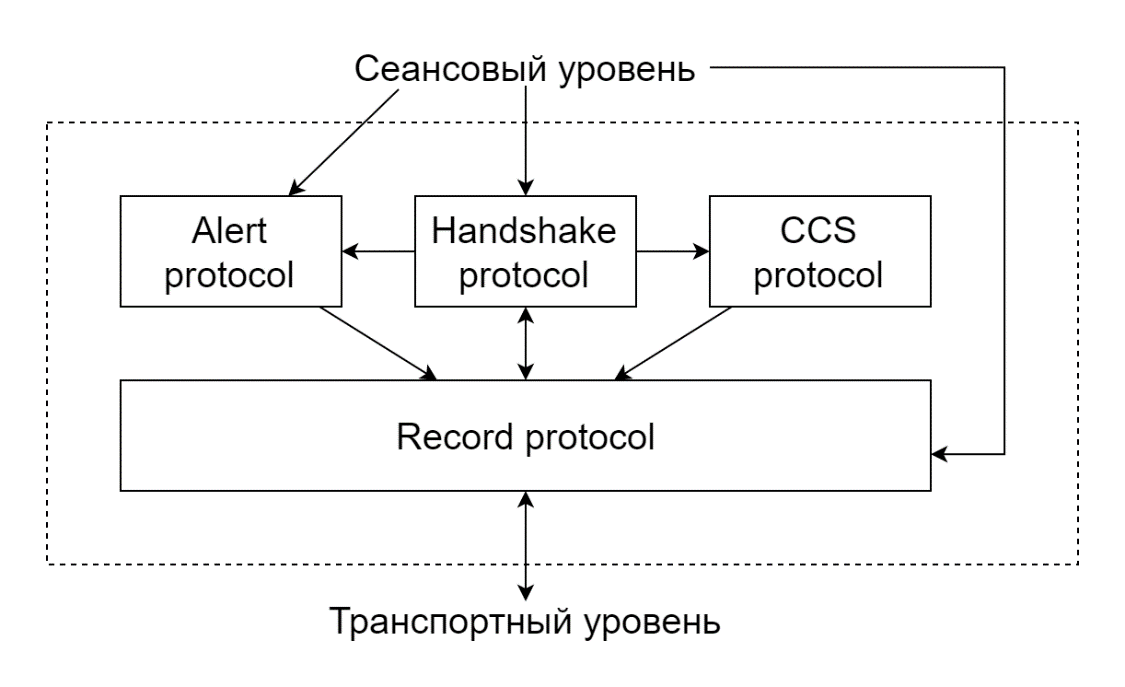
Протокол записи (Record Layer) — это уровневый протокол. На каждом уровне сообщения включают поля для длины, описания и проверки. Протокол записи принимает сообщения, которые нужно передать, фрагментирует данные в управляемые блоки, разумно сжимает данные, применяя MAC (message authentication code), шифрует и передаёт результат. Полученные данные он расшифровывает, проверяет, распаковывает, собирает и доставляет к более верхним уровням клиента.

Существует четыре протокола записи:

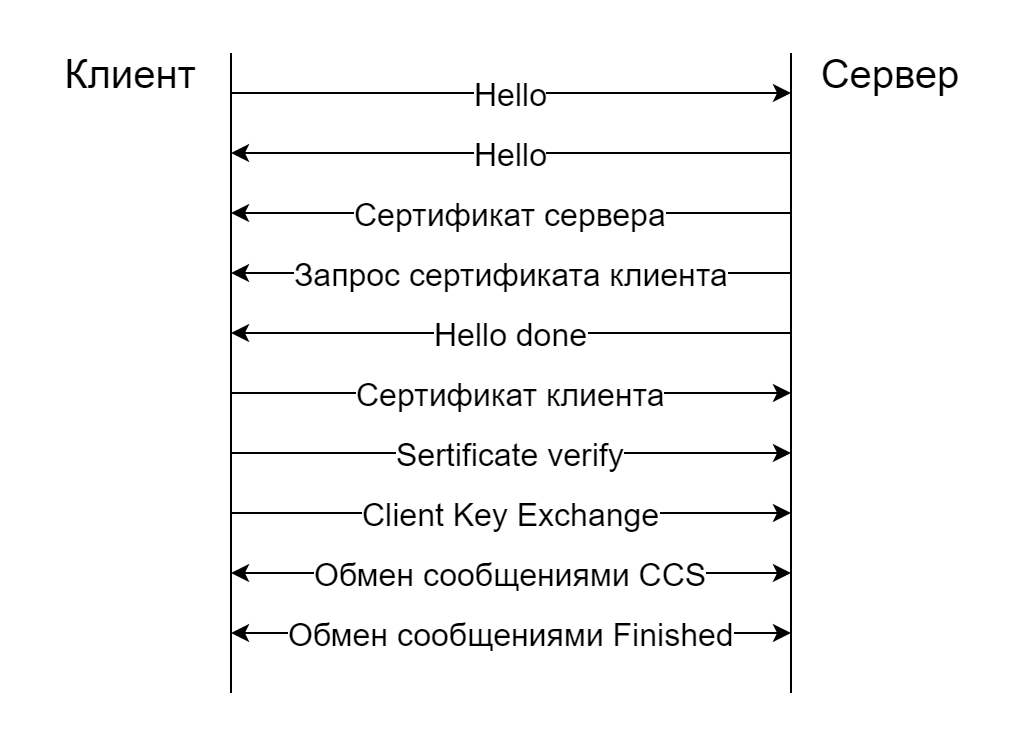
* Протокол рукопожатия (handshake protocol);
* Протокол тревоги (alert protocol);
* Протокол изменения шифра (the change cipher spec protocol);
* Протокол приложения (application data protocol);

Если SSL реализация получает тип записи, который ей неизвестен, то эта запись просто игнорируется. Любой протокол, созданный для использования совместно с SSL, должен быть хорошо продуман, так как будет иметь дело с атаками на него. Заметим, что из-за типа и длины записи, протокол не защищен шифрованием. Внимание следует уделить тому, чтобы минимизировать трафик.

Протокол записи:



Протокол рукопожатия:



Преимущества протоколов TLS/SSL:

* Невидим для протоколов более высокого уровня;
* Популярность использования в Интернет-соединениях и приложениях электронной коммерции;
* Отсутствие постоянного соединения между сервером и клиентом;
* Позволяет создать туннель для приложений, использующих TCP, таких как электронная почта, инструменты программирования и т.д.

Недостатки протоколов TLS/SSL:

* Невозможность использования с протоколами UDP и ICMP;
* Необходимость отслеживания состояния соединения;
* Наличие дополнительных требований к программному обеспечению о поддержке TLS.

При проектировании приложений SSL реализуется «под» любым другим протоколом прикладного уровня, таким как HTTP, FTP, SMTP, NNTP и XMPP, таким образом обеспечивая «прозрачность» его использования. Исторически SSL был использован, в первую очередь, с надёжными транспортными протоколами, такими как Transmission Control Protocol (TCP). Тем не менее, он также был реализован с датаграммными транспортными протоколами, такими как User Datagram Protocol (UDP) и Datagram Control Protocol (DCCP), использование которого было стандартизировано, что привело к появлению термина Datagram Transport Layer Security (DTLS).

Частое использование протокола SSL привело к формированию протокола HTTPS (Hypertext Transfer Protocol Secure), поддерживающего шифрование. Данные, которые передаются по протоколу HTTPS, «упаковываются» в криптографический протокол SSL или TLS, тем самым обеспечивая защиту этих данных. Такой способ защиты широко используется в мире Веб для приложений, в которых важна безопасность соединения, например в платёжных системах. В отличие от HTTP, для HTTPS по умолчанию используется TCP-порт 443.

Поддержка протоколов вебсайтами:

|  |  |  |
| --- | --- | --- |
| **Версия протокола** | **Безопасность** | **Поддержка сайтами** |
| SSL 2.0 | Нет | 4.9 % |
| SSL 3.0 | Нет | 16.6 % |
| TLS 1.0 | Может быть | 94.7 % |
| TLS 1.1 | Да | 82.6 % |
| TLS 1.2 | 85.5 % |

Изначально виртуальные частные сети (VPN) на основе SSL разрабатывались как дополнительная и альтернативная технология удалённого доступа на основе IPsec VPN. Однако, такие факторы, как достаточная надёжность и дешевизна, сделали эту технологию привлекательной для организации VPN. Также SSL получил широкое применение в электронной почте.

Алгоритмы, используемые в SSL:

* Для обмена ключами и проверки их подлинности применяются: RSA, Diffie-Hellman, ECDH, SRP, PSK.
* Для аутентификации: RSA, DSA, ECDSA.
* Для симметричного шифрования: RC2, RC4, IDEA, DES, Triple DES или AES, Camellia.
* Для хеш-функций: SHA, MD5, MD4 и MD2.

Алгоритмы, используемые в TLS:

* Для обмена ключами и проверки их подлинности применяются комбинации алгоритмов: RSA (асимметричный шифр), Diffie-Hellman (безопасный обмен ключами), DSA (алгоритм цифровой подписи), ECDSA;
* Для симметричного шифрования: RC4, IDEA, Triple DES, SEED, Camellia или AES;
* Для хеш-функций: MD5, SHA, SHA-256/384.

Алгоритмы могут дополняться в зависимости от версии протокола. До версии протокола TLS 1.2 были доступны также следующие алгоритмы симметричного шифрования, но они были убраны как небезопасные: RC2, IDEA, DES.

### Протокол SKIP. Применение и основные положения протокола

Simple Key-Management for Internet Protocol (или SKIP) — протокол, разработанный около 1995 года организацией IETF Security Working Group для обмена ключами шифрования.

SKIP является гибридным протоколом распределения ключей, похожим на SSL, кроме того, он один раз устанавливает долгосрочные ключи, а затем не требует предварительной связи в целях установления или обмена ключами. Таким образом, он не требует никаких дополнительных соединений и постоянной генерации новых ключей.

Общий алгоритм:

1. Два пользователя вырабатывают общий секретный ключ методом Диффи-Хеллмана;
2. Для передачи исходного IP-пакета выполняются следующие действия:
   1. Генерируется пакетный ключ ;
   2. Исходный пакет шифруется с помощью пакетного ключа ;
   3. Зашифрованный пакет укладывается в блок данных SKIP-пакета;
   4. Пакетный ключ шифруется с помощью общего секретного ключа и помещается в SKIP-пакет;
   5. Вычисляется контрольное значение исходного пакета.

|  |  |  |  |
| --- | --- | --- | --- |
| Заголовок IP-пакета | – ключ в зашифрованном виде от шифрованного пакета | Контрольное значение | Шифрованный пакет |

SKIP не подвержен атакам человека по середине, поскольку общие параметры DH (алгоритм Диффи-Хеллмана) носят долгосрочный характер и проходят проверку подлинности.

Если пакет ключей аутентификации когда-либо окажется раскрыт, то обновление мастер-ключа алгоритмом, описанным выше, исключает повторное использование раскрытого ключа. Поэтому, даже если конкретный ключ трафика будет под угрозой, текущий неявный ключ будет в безопасности, и поэтому злоумышленник не сможет вычислить алгоритмы шифрования в . «Не зная шифрования с помощью , злоумышленник не может повторно использовать ранее раскрытые ключи для чего бы то ни было».

Кроме того, даже если все ключи , зашифрованные данным , окажутся раскрыты, злоумышленник не сможет ничего узнать о других или узнать . «Даже выборочное использование ключей , с целью нахождения , эквивалентно выборочной атаке по открытому тексту на , и это невозможно даже при очень большом количестве известных ключей до тех пор, пока алгоритм шифрования ключей не окажется незащищенным от данной атаки». А так как алгоритм шифрования ключей является защищенным от общей/выборочной атаки, то и протокол SKIP в целом является защищенным от этого.

По сравнению с существующими системами шифрования трафика, SKIP имеет ряд уникальных особенностей:

* SKIP *универсален*:
  + он шифрует IP-пакеты, не используя информацию о формирующих их пользователях, приложениях или процессах;
  + будучи установленным непосредственно над пакетным драйвером, он обрабатывает весь трафик, не накладывая никаких ограничений ни на программное обеспечение, ни на физические каналы.
* SKIP *сеансонезависим*: не требует дополнительного обмена информацией для организации защищенного взаимодействия (за исключением единовременно запрошенного открытого ключа собеседника).
* SKIP *независим от системы шифрования*:
  + другие системы шифрования могут быть подсоединены к системе в виде внешних библиотечных модулей;
  + пользователь может выбирать алгоритм шифрования информации - как из предлагаемых поставщиком, так и использовать свой;
  + для закрытия пакетного ключа и, собственно, данных могут использоваться различные алгоритмы шифрования разной степени защищенности.

Среди протоколов управления криптографическими ключами наиболее широкое распространение получили протоколы SKIP и ISAKMP. SKIP проще в реализации, но он не поддерживает переговоров по поводу алгоритмов шифрования. Если получатель, использующий SKIP, не может расшифровать пакет, он уже никак не согласует метод шифрования с противоположной стороной. Протокол ISAKMP (вернее, его более поздняя реализация протокол IKE – Internet Key Exchange) поддерживает такие переговоры и выбран в качестве обязательного протокола для управления ключами в IPsec для IPv6. При использовании ISAKMP снижается уязвимость закрытых основных ключей, служащих для распределения временных ключей шифрования. При этом IKE считается более надежным и гибким.

В протоколе IPv4 может применяться как протокол ISAKMP, так и протокол SKIP.

### Протокол IPSec. Применение и основные положения протокола

IPsec (сокращение от IP Security) — набор протоколов для обеспечения защиты данных, передаваемых по межсетевому протоколу IP. Позволяет осуществлять подтверждение подлинности (аутентификацию), проверку целостности и/или шифрование IP-пакетов. IPsec также включает в себя протоколы для защищённого обмена ключами в сети Интернет. В основном, применяется для организации VPN-соединений.

В вопросе выбора уровня реализации защищённого канала несколько противоречивых аргументов: с одной стороны, за выбор верхних уровней говорит их независимость от вида транспортировки (выбора протокола сетевого и канального уровней), с другой стороны, для каждого приложения необходима отдельная настройка и конфигурация. Плюсом в выборе нижних уровней является их универсальность и наглядность для приложений, минусом — зависимость от выбора конкретного протокола (например, PPP или Ethernet). Компромиссом в выборе уровня является IPsec: он располагается на сетевом уровне, используя самый распространённый протокол этого уровня — IP. Это делает IPsec более гибким, так что он может использоваться для защиты любых протоколов, базирующихся на TCP и UDP. В то же время, он прозрачен для большинства приложений.

Преимущества протокола IPsec:

* Безопасность и надёжность защиты данных протокола проверена и доказана, так как протокол был принят как Интернет-стандарт;
* Работа в верхнем слое сетевого протокола и шифрование данных над уровнем сетевого протокола.

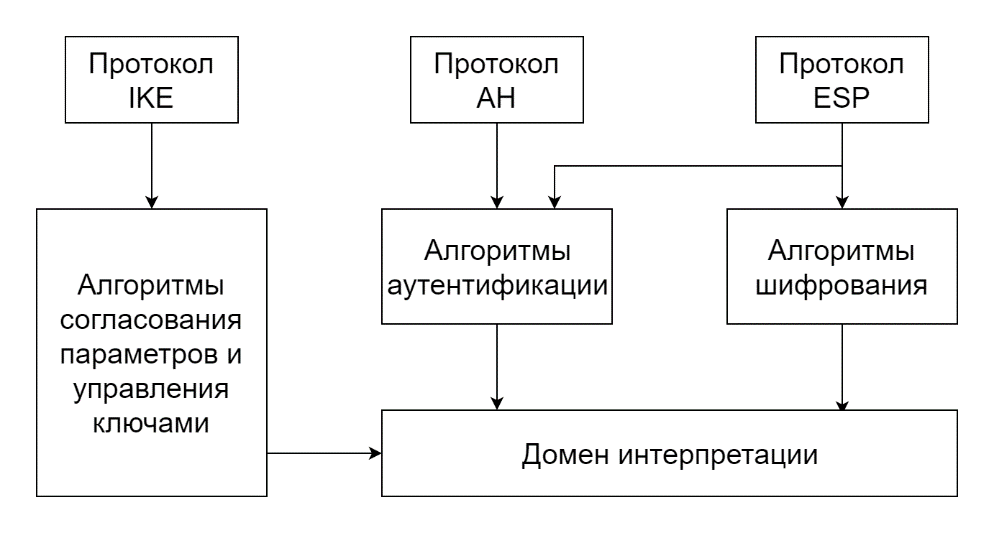
Недостатки протокола IPsec:

* Сложность реализации;
* Дополнительные требования к оборудованию сети (маршрутизаторы и т. п.);
* Существует много различных реализаций, не всегда корректно взаимодействующих друг с другом.

IPsec является набором стандартов Интернета и своего рода «надстройкой на IP-протоколом. Его ядро составляют три протокола:

* Authentication Header (АН) обеспечивает целостность передаваемых данных, аутентификацию источника информации и функцию по предотвращению повторной передачи пакетов
* Encapsulating Security Payload (ESP) обеспечивает конфиденциальность (шифрование) передаваемой информации, ограничение потока конфиденциального трафика. Кроме этого, он может исполнять функции AH: обеспечить целостность передаваемых данных, аутентификацию источника информации и функцию по предотвращению повторной передачи пакетов. При применении ESP в обязательном порядке должен указываться набор услуг по обеспечению безопасности: каждая из его функций может включаться опционально.
* Internet Security Association and Key Management Protocol (ISAKMP) — протокол, используемый для первичной настройки соединения, взаимной аутентификации конечными узлами друг друга и обмена секретными ключами. Протокол предусматривает использование различных механизмов обмена ключами, включая задание фиксированных ключей, использование таких протоколов, как Internet Key Exchange, Kerberized Internet Negotiation of Keys (RFC 4430) или записей DNS типа IPSECKEY (RFC 4025).

Также одним из ключевых понятий является Security Association (SA). По сути, SA является набором параметров, характеризующим соединение. Например, используемые алгоритм шифрования и хэш-функция, секретные ключи, номер пакета и др.

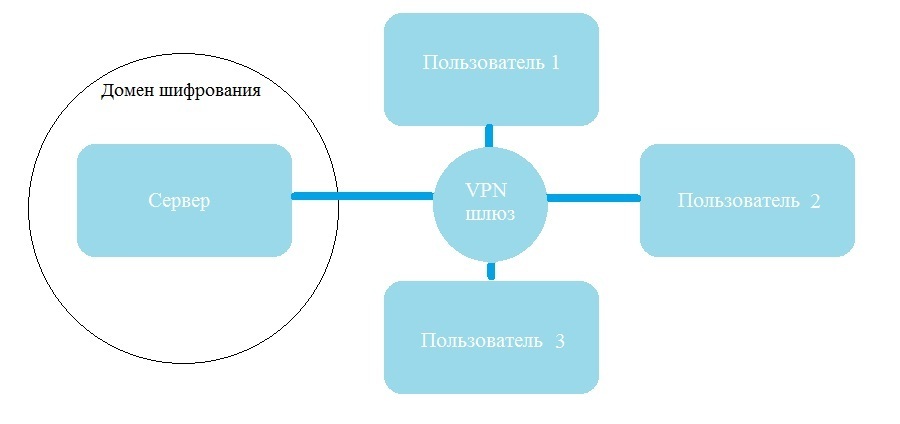


IPsec может функционировать в двух режимах: транспортном и туннельном.

В транспортном режиме шифруются (или подписываются) только данные IP-пакета, исходный заголовок сохраняется. Транспортный режим, как правило, используется для установления соединения между хостами. Он может также использоваться между шлюзами для защиты туннелей, организованных каким-нибудь другим способом (см., например, L2TP).

В туннельном режиме шифруется весь исходный IP-пакет: данные, заголовок, маршрутная информация, а затем он вставляется в поле данных нового пакета, то есть происходит инкапсуляция. Туннельный режим может использоваться для подключения удалённых компьютеров к виртуальной частной сети или для организации безопасной передачи данных через открытые каналы связи (например, Интернет) между шлюзами для объединения разных частей виртуальной частной сети.

Режимы IPsec не являются взаимоисключающими. На одном и том же узле некоторые SA могут использовать транспортный режим, а другие — туннельный.



Протокол IPsec используется, в основном, для организации VPN-туннелей. В этом случае протоколы ESP и AH работают в режиме туннелирования. Кроме того, настраивая политики безопасности определенным образом, протокол можно использовать для создания межсетевого экрана. Смысл межсетевого экрана заключается в том, что он контролирует и фильтрует проходящие через него пакеты в соответствии с заданными правилами. Устанавливается набор правил, и экран просматривает все проходящие через него пакеты. Если передаваемые пакеты попадают под действие этих правил, межсетевой экран обрабатывает их соответствующим образом. Например, он может отклонять определенные пакеты, тем самым прерывая небезопасные соединения. Настроив политику безопасности соответствующим образом, можно, например, запретить веб-трафик. Для этого достаточно запретить отсылку пакетов, в которые вкладываются сообщения протоколов HTTP и HTTPS. IPsec можно применять и для защиты серверов — для этого отбрасываются все пакеты, кроме пакетов, необходимых для корректного выполнения функций сервера. Например, для Web-сервера можно блокировать весь трафик, за исключением соединений через 80-й порт протокола TCP, или через порт TCP 443 в случаях, когда применяется HTTPS.

С помощью IPsec здесь обеспечивается безопасный доступ пользователей к серверу. При использовании протокола ESP все обращения к серверу и его ответы шифруются. Однако за VPN-шлюзом (в домене шифрования) передаются открытые сообщения. Другие примеры использования IPsec:

* шифрование трафика между файловым сервером и компьютерами в локальной сети, используя IPsec в транспортном режиме.
* соединение двух офисов с использованием IPsec в туннельном режиме.

### Протокол SET. Применение и основные положения протокола