1. Симметричные блочные и потоковые шифры.
2. Синхронные и асинхронные потоковые шифры.
3. Сферы применения потоковых шифров.
4. Одноразовый шифр-блокнот
5. Регистры сдвига с линейной обратной связью
6. GSM Протокол A5/2
7. Необходимость вектора инициализации в потоковых шифрах.
8. Шифр Trivium.
9. Шифр Rabbit
10. Свойства конфузии и диффузии в блочных шифрах.
11. Сеть Фейстеля
12. Алгоритм DES. Процедуры шифрования/дешифрования. Расширение ключа.
13. Техника отбеливания ключа
14. 3DES
15. ГОСТ 28147 или блочный шифр с длиной блока 64 бита из ГОСТ 34.12.
16. Режимы шифрования ECB, CBC, CFB, OFB, CTR, GCM. Устойчивость к атакам вставки и удаления блоков. Лавинный эффект. Возможность параллельной обработки блоков.
17. Код аутентификации сообщения
18. Схемы хэширования на основе блочных симметричных криптосистем.

**ВОПРОСЫ К ЗАЩИТЕ**

1. **Симметричные блочные и потоковые шифры.**

Симметричные шифры делятся на блочные и потоковые. **Блочные шифры** обрабатывают данные **блоками фиксированной длины**, а **потоковые** **побитно**.

Блочный шифр использует как путаницу, так и диффузию, в то время как потоковый шифр полагается только на путаницу.

Блочный шифр использует один и тот же ключ для шифрования каждого блока, в то время как потоковый шифр использует разные ключи для каждого байта.

Блочный шифр использует режимы алгоритмов ECB (электронная кодовая книга) и CBC (цепочка блоков шифров) . Напротив, потоковый шифр использует режимы алгоритма CFB (Cipher Feedback) и OFB (Output Feedback).

К достоинствам поточных шифров относятся высокая скорость шифрования, относительная простота реализации и отсутствие размножения ошибок. Недостатком является необходимость использовать для каждого сообщения другой ключ. Это обусловлено тем, что если два различных сообщения шифруются на одном и том же ключе, то эти сообщения легко могут быть расшифрованы.

A diagram of a block diagram

Description automatically generated with medium confidenceОчевидной проблемой, касающейся использования **блочных шифров**, является повторяющийся текст, для которого генерируется тот же шифр. Следовательно, это дало бы подсказку криптоаналитику, что облегчает поиск повторяющихся строк простого текста. В результате он может раскрыть все сообщение.

1. **Синхронные и асинхронные потоковые шифры.**

A diagram of a key stream generator

Description automatically generated

В синхронных потоковых шифрах ключевой поток s\_i зависит только от ключа k (CFB,OFB)

В асинхронных потоковых шифрах ключевой поток s\_i зависит от ключа k и от символов шифротекста y\_i

1. **Сферы применения потоковых шифров.**

На практике, потоковые шифры применяются намного реже, чем блочные симметричные шифры

Благодаря своей скорости и простоте программноаппаратной реализации, потоковые шифры используются в устройствах с малыми вычислительными мощностями (мобильные телефоны, датчики, встроенные устройства и др.)

1. **Одноразовый шифр-блокнот**

В 1917 Гильберт Вернам запатентовал одноразовый шифр-блокнот (шифр Вернама)

Суть шифра – XOR с ключом той же длины, что и сообщение

При этом ключ должен обладать тремя критически важными свойствами:

- иметь случайное равномерное распределение;

- совпадать по размеру с заданным открытым текстом;

- применяться только один раз.

Для данного шифра доказано, что он обладает абсолютной криптостойкостью (т.е. и практической и математической одновременно).

1. **Регистры сдвига с линейной обратной связью**

В регистре сдвига с линейной обратной связью (РСЛОС) выделяют две части (модуля):

* собственно регистр сдвига

Регистр состоит из функциональных ячеек памяти (битов одного или нескольких машинных слов), в каждой из которых хранится текущее состояние (значение) одного бита. Количество ячеек L, называют длиной регистра.

* Функцию обратной связи, вычисляющую значение вдвигаемого бита.

Для линейного регистра – это операция XOR над некоторыми его битами, которые называют отводами.

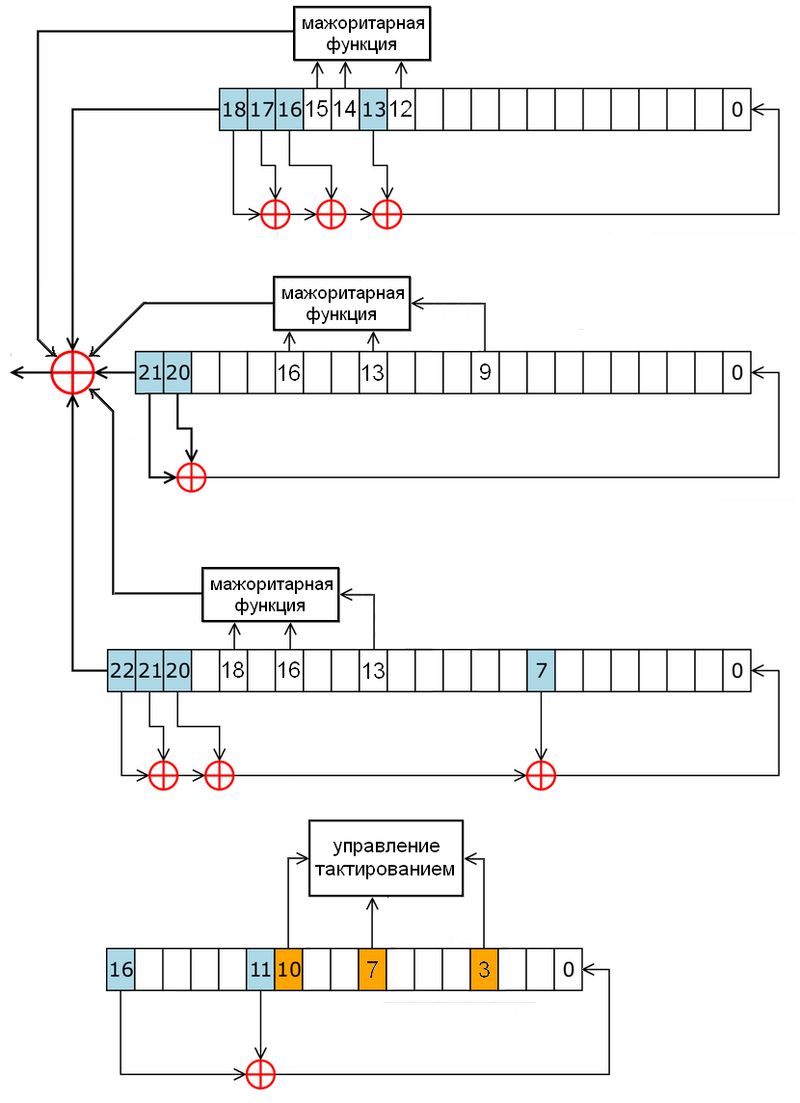
A diagram of a number and numbers

Description automatically generated

A table with numbers and letters

Description automatically generated

1. **GSM Протокол A5/2**



А5 — это поточный алгоритм шифрования

Шифр основан на побитовом сложении по модулю два (булева операция «исключающее или») генерируемой псевдослучайной последовательности и шифруемой информации. В A5 псевдослучайная последовательность реализуется на основе трёх линейных регистров сдвига с обратной связью. Регистры имеют длины 19, 22 и 23 бита соответственно. Сдвигами управляет специальная схема, организующая на каждом шаге смещение как минимум двух регистров, что приводит к их неравномерному движению. Последовательность формируется путём операции «исключающее или» над выходными битами регистров.

Структура алгоритма А5 выглядит следующим образом:

* Три регистра(R1, R2, R3) имеют длины 19, 22 и 23 бита,
* Многочлены обратных связей:

X19 + X18 + X17 + X14 + 1 для R1,

X22 + X21 + 1 для R2,

X23 + X22 + X21 + X8 + 1 для R3,

* Управление тактированием осуществляется специальным механизмом:

в каждом регистре есть биты синхронизации: 8 (R1), 10 (R2), 10 (R3), вычисляется функция

F = x&y|x&z|y&z, где & — булево AND, | - булево OR, а x, y и z — биты синхронизации R1, R2 и R3 соответственно,

сдвигаются только те регистры, у которых бит синхронизации равен F, фактически, сдвигаются регистры, синхробит которых принадлежит большинству,

* Выходной бит системы — результат операции XOR над выходными битами регистров.

В алгоритм А5/2 добавлен ещё один регистр на 17 бит (R4), управляющий движением остальных.

Изменения структуры следующие:

добавлен регистр R4 длиной 17 бит,

многочлен обратной связи для R4: X17 + X12 + 1,

управление тактированием осуществляет R4:

в R4 биты 3, 7, 10 есть биты синхронизации,

вычисляется мажоритарная функция F = x&y|x&z|y&z (равна большинству), где & — булево AND, | - булево OR, а x, y и z — биты синхронизации R4(3), R4(7) и R4(10) соответственно,

R1 сдвигается если R4(10) = F,

R2 сдвигается если R4(3) = F,

R3 сдвигается если R4(7) = F,

выходной бит системы — результат операции XOR над старшими битами регистров и мажоритарных функций от определённых битов регистров:

R1 — 12, 14, 15,

R2 — 9, 13, 16,

R3 — 13, 16, 18.

1. **Необходимость вектора инициализации в потоковых шифрах.**

IV – случайно задаваемая переменная для каждой сессии шифрования (вектор инициализации или случайная метка). IV не обязательно засекречивается и шифруется в процессе работы потокового шифра. IV может быть полностью публичным.

1. **Шифр Trivium.**

A diagram of a machine

Description automatically generated

Trivium — симметричный алгоритм синхронного потокового шифрования, использующий 80-битный ключ. В основе Trivium лежат три сдвиговых регистра: A, B и C. Длины регистров равны 93, 84 и 111 соответственно. Сумма XOR всех трех выходных данных регистра формирует ключевой поток si. Особенностью шифра является то, что выход каждого регистра соединен со входом другого регистра. Таким образом, регистры расположены по кругу. Шифр можно рассматривать как состоящий из одного кольцевого регистра общей длиной 93+84+111 = 288.

Вход каждого регистра вычисляется как сумма XOR двух бит: Выходной бит другого регистра и один бит регистра в определенном месте возвращается на вход.

Позиции приведены в Таблице

A white box with black text

Description automatically generated

Выходные данные каждого регистра вычисляются как сумма XOR трех битов:

* Крайний правый бит регистра.
* Один бит регистра в определенном месте подается на выход. Позиции приведены в Таблице.
* Выходные данные логической функции AND, входные данные которой представляют собой два определенных бита регистра. Опять же, положения входов логического элемента AND приведены в таблице.

1. **Шифр Rabbit**

Rabbit используют 128-битный ключ и 64-битный инициализирующий вектор. Ключ 128 бит делится на 8 подключей длиной 16 бита k\_0, …, k\_7, переменные состояний и счётчики инициализируются при помощи подключей:

**A close-up of a math problem

Description automatically generated**

1. **Свойства конфузии и диффузии в блочных шифрах.**

**Конфузия** (Confusion) – снижение зависимости между ключом и шифротекстом. Обеспечивается подстановками/заменами или S-блоками (substitution box).

**Диффузия** (diffusion) – один символ открытого текста влияет на много символов шифротекста. Цель скрыть статистические свойства открытого текста. Обеспечивается перестановками или P-блоками (permutation box)

1. **Сеть Фейстеля**

Сеть Фе́йстеля — один из методов построения блочных шифров. Сеть состоит из ячеек, называемых ячейками Фейстеля. На вход каждой ячейки поступают данные и ключ. На выходе каждой ячейки получают изменённые данные и изменённый ключ. Все ячейки однотипны, и говорят, что сеть представляет собой определённую многократно повторяющуюся (итерированную) структуру. Ключ выбирается в зависимости от алгоритма шифрования/расшифрования и меняется при переходе от одной ячейки к другой.

* Блок [открытого текста](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82) делится на две равные части: (�0, �0)(L0, R0)
* В каждом [раунде](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%83%D0%BD%D0%B4_(%D0%B2_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%B8)) вычисляются:

A diagram of a algorithm

Description automatically generated

* где ⊕ означает исключающее XOR. Тогда зашифрованный текст будет (Rn, Ln).

A close up of a text

Description automatically generated

Расшифровка зашифрованного текста (Rn, Ln) осуществляется путем вычисления для i = n, n-1,…,1

A diagram of a machine

Description automatically generated

1. **Алгоритм DES. Процедуры шифрования/дешифрования. Расширение ключа.**

DES — алгоритм для симметричного шифрования. Размер блока для DES равен 64 битам. В основе алгоритма лежит сеть Фейстеля с 16 циклами (раундами) и ключом, имеющим длину 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP-1) преобразований.

A diagram of a machine

Description automatically generated

Алгоритм DES состоит из нескольких основных этапов:

* Расширение ключей
* Permutation choice 1

56 бит ключа выбираются из исходных 64 битов с помощью перестановочного выбора 1 (PC-1) — оставшиеся восемь бит отбрасываются или используются в качестве битов проверки четности.

* Key schedule (16 раундов)

Затем 56 бит делятся на две 28-битные половины; каждая половина после этого рассматривается отдельно. В последовательных раундах обе половины поворачиваются влево на один или два бита (указанных для каждого раунда), а затем с помощью Permuted Choice 2 выбираются 48 битов подключа.

* Initial permutation(начальная перестановка)

Перестановка 64 бита с помощью таблица начальная перестановка IP

A table with numbers on it

Description automatically generated

* Function F (16 раундов)

64 бит делятся на две 32-битные половины; каждая половина после этого рассматривается отдельно.

В каждом раунде каждая половина состоит из 32 битов текста, которые будут представлены следующим образом:

A number and a circle

Description automatically generated with medium confidence

* **Final permutation.**

С помощью таблицы финальной перестановки, получаем шифротекст.

**Дешифрования DES**

A diagram of a algorithm

Description automatically generated

1. **Техника отбеливания ключа**

**Отбеливанием (whitening) называется способ, при котором выполняется XOR части ключа с входом блочного алгоритма и XOR другой части ключа с выходом блочного алгоритма.**

[image]Смысл этих действий в том, чтобы помешать криптоаналитику получить пару "открытый текст/шифротекст" для лежащего в основе блочного алгоритма. Метод заставляет криптоаналитика угадывать не только ключ алгоритма, но и одно из значений отбеливания.

Если K1 = K2, то для вскрытия грубой силой потребуется 2^(n+m/p) действий, где n - размер ключа, m - размер блока, и p - количество известных открытых текстов. Если K1 и K2 различны, то для вскрытия грубой силой с тремя известными открытыми текстами потребуется 2^(n+m+1) действий. Против дифференциального и линейного криптоанализа, такие меры обеспечивают защиту только для нескольких битов ключа. Но с вычислительной точки зрения это очень дешевый способ повысить безопасность блочного алгоритма.

1. **3DES**

3DES - симметричный блочный шифр, созданный на основе алгоритма DES с целью устранения главного недостатка последнего — малой длины ключа (56 бит)

3DES использует 3 ключа по 56 бит, поэтому максимальный эффективный размер ключа составляет 168 бит. Данные шифруются первым ключом, расшифровываются вторым ключом и, наконец, снова шифруются третьим ключом. Это делает 3DES медленнее, чем исходный DES, но взамен обеспечивает большую безопасность

**A diagram of a flowchart

Description automatically generated**

1. **ГОСТ 28147 или блочный шифр с длиной блока 64 бита из ГОСТ 34.12.**

**A white paper with black text and numbers

Description automatically generated**

Число раундов S = 32

Длина блока n = 64 бита

Размер ключа k – 256 бит

Правило расширения ключа. Ключ разбивается на 8 частей ki по 32 бита. Подключи k1, k2,…, k8 по 32 бита циклически повторяются 3 раза, с 24 по 31 раунд подключи k1, k2,…, k8 используются в обратном порядке.

1. **Режимы шифрования ECB, CBC, CFB, OFB, CTR, GCM. Устойчивость к атакам вставки и удаления блоков. Лавинный эффект. Возможность параллельной обработки блоков.**

Режимы шифрования

* Electronic Code Book mode (ECB) – режим электронной кодовой книги.

Каждый блок открытого текста заменяется блоком шифротекста.

A diagram of a diagram

Description automatically generated

* прост в обращении, можно работать с блоками независимо и даже распараллелить вычисления.
* не защищен от атак с удалением и вставками
* ошибка в одном бите влияет на целый блок в расшифрованном тексте
* Cipher Block Chaining mode (CBC) – режим сцепления блоков шифротекста

A diagram of a computer system

Description automatically generated

* Защита от атак вставки и удаления блоков.
* ошибки при шифровании и в открытом тексте дают ошибку не только в текущем блоке, но и портит следующие блоки.
* Cipher Feedback mode (CFB) – режим обратной связи по шифротексту

A diagram of a diagram

Description automatically generated

* защита от атак вставки и удаления.
* ошибки в открытом тексте и при шифровании распрастраняются дальше по шифротексту.
* Вероятностное шифрование
* Асинхронный потоковый шифр.
* Output Feedback mode (OFB) – режим обратной связи по выходу

A diagram of a diagram

Description automatically generated

* Ошибка в открытом тексте остается в блоке.
* Ошибка при шифровании распространяется по шифротексту
* Counter mode (CTR) – режим счетчика

A diagram of a blockchain

Description automatically generated

* защита от атак вставки и удаления.
* ошибки в открытом тексте и при шифровании распрастраняются дальше по шифротексту.
* Вероятностное шифрование
* Асинхронный потоковый шифр.
* Galois Counter mode (GCM) – режим счетчика с аутентификацией Галуа

Как и в CTR, блоки нумеруются последовательно, а затем этот номер блока объединяется с IV и шифруется блочным шифром E. Результат этого шифрования затем подвергается операции XOR с открытым текстом для получения зашифрованного текста. Как и все режимы счетчика, это по существу потоковый шифр, поэтому важно, чтобы для каждого зашифрованного потока использовался отдельный IV.

Блоки зашифрованного текста считаются коэффициентами многочлена, который затем оценивается в зависимости от ключа точки H с использованием арифметики конечных полей. Затем результат шифруется, создавая тег аутентификации, который можно использовать для проверки целостности данных. Зашифрованный текст затем содержит IV, зашифрованный текст и тег аутентификации.

A screenshot of a computer

Description automatically generated

**Устойчивость к атакам вставки и удаления блоков**

Атака вставки блоков данных (конфликт выбора) возникает, когда злоумышленник вставляет новый блок данных в шифротекст, а получатель расшифровывает его, как если бы это была часть исходного сообщения. Это может привести к изменению значений и смещению данных и привести к ошибкам при расшифровке остальной части сообщения

Атака удаления блоков данных (атака с использованием блока врага) возникает, когда злоумышленник удаляет один или несколько блоков данных из шифротекста. При расшифровке получатель не сможет обнаружить, что часть данных была удалена, и может получить некорректный результат.

Для обеспечения устойчивости к атакам вставки и удаления блоков данных рекомендуется использовать алгоритмы шифрования, которые обеспечивают целостность данных и обнаружение поддельных/измененных блоков, такие как режимы шифрования аутентифицированных шифров (Authenticated Encryption Modes) или алгоритмы, основанные на научных статьях, такие как Одноразовая подпись блока (Block cipher-based Message Authenticate Code).

Это делает ECB неустойчивым к атакам вставки и удаления блоков, так как атакующий может манипулировать отдельными блоками и изменить или удалить их, не влияя на остальные блоки. (-)

CBC режим шифрования использует предыдущий засшифрованный блок в качестве вектора инициализации для шифрования следующего блока. Это создает зависимость между последовательными блоками данных. Из-за этой зависимости CBC режим устойчив к атакам вставки и удаления блоков (+)

CFB режим также устойчив к атакам вставки и удаления блоков (+)

 OFB также устойчив к атакам вставки и удаления блоков. (+)

CTR устойчив к атакам вставки и удаления блоков.(+)

GCM (Galois/Counter Mode) -  (+)

* Лавинный эффект

 изменение значения малого количества битов во входном тексте или в ключе ведет к «лавинному» изменению значений выходных битов шифротекста. Другими словами, это зависимость всех выходных битов от каждого входного бита.

Криптографический алгоритм удовлетворяет лавинному критерию, если при изменении одного бита входной последовательности изменяется в среднем половина выходных битов.

CBC(+)Лавинный эффект также присутствует, поскольку изменение одного бита входных данных приводит к случайным изменениям в большом количестве выходных данных.

CFB но изменение одного бита входных данных приведет к изменениям только в одном блоке выходных данных.(-)

OFB Лавинный эффект присутствует, поскольку изменение одного бита входных данных приводит к случайным изменениям в выходных данны (+)

CTR(+) Лавинный эффект присутствует(+)

GCM (-)

* Возможность параллельной обработки блоков

ECE

CBC позволяет параллельную обработку блоков данных в режиме шифрования, но требует последовательной обработки в режиме дешифрования.

CFB режим шифрования позволяет параллельную обработку блоков данных как в режиме шифрования, так и в режиме дешифрования

OFB также позволяет параллельную обработку блоков данных.

CTR позволяет параллельную обработку блоков данных.

1. **Код аутентификации сообщения**

MAC (имитовставка, message authentication code — код аутентичности сообщения) — контрольная сумма, которая добавляется к сообщению и предназначена для обеспечения его целостности и аутентификации источника данных.

MAC обычно применяется для обеспечения целостности и защиты от фальсификации передаваемой информации.

1. **Схемы хэширования на основе блочных симметричных криптосистем.**

A table of mathematical equations

Description automatically generated

Схемы хэширования на основе блочных симметричных криптосистем используют блочные шифры для создания хэш-функций.

Схемы хэширования на основе блочных симметричных криптосистем, такие как HMAC и CMAC, предоставляют аутентификацию и целостность сообщений. Они широко используются в протоколах безопасности, таких как IPSec и SSL/TLS, а также в других приложениях, где важна проверка целостности данных и аутентификация отправителя.