
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ

# Криптография. Симметричные алгоритмы шифрования.

# Криптография. Симметричные алгоритмы шифрования.

1. *Введение.*
2. *Терминология.*
3. *Симметричные криптосистемы.*
4. *Алгоритм Цезаря.*
5. *Алгоритм замены полиалфавитный.*
6. *Алгоритм замены с большим ключом.*
7. *Перестановки.*
8. *Гаммирование.*

# Введение

Проблема защиты информации путем ее преобразования, исключающего ее прочтение посторонним лицом, волновала человеческий ум с давних времен. История криптографии - ровесница истории человеческого языка.

# Введение

Первоначально письменность сама по себе была криптографической системой, так как в древних обществах ею владели только избранные. Священные книги Древнего Египта, Древней Индии тому примеры.

# Введение

С широким распространением письменности криптография стала формироваться как самостоятельная наука. Первые криптосистемы встречаются уже в начале нашей эры. Так, Цезарь в своей переписке использовал систематический шифр, получивший его имя.

# Введение

Бурное развитие криптографические системы получили в годы первой и второй мировых войн. Начиная с послевоенного времени, и по нынешний день появление вычислительных средств ускорило разработку и совершенствование криптографических методов.

# Актуальность использования криптографических методов в информационных системах

1. Расширилось использование компьютерных сетей, в частности глобальной сети Интернет, по которым передаются большие объемы информации государственного, военного, коммерческого и частного характера, не допускающего возможность доступа к ней посторонних лиц;

# Актуальность использования криптографических методов в информационных системах

2. появление новых мощных компьютеров, технологий сетевых и нейронных вычислений дискредитировало множество криптографических систем еще недавно считавшихся практически н раскрываемыми.

# КРИПТОЛОГИЯ

- занимается проблемой защиты информации путем ее преобразования (kryptos - тайный, logos - наука).

# Направления криптологии

1. Криптография
2. Криптоанализ.

# ЦЕЛЬ КРИПТОГРАФИИ

Криптография занимается поиском и исследованием математических методов преобразования информации с целью ее защиты от несанкционированного доступа.

# ЦЕЛЬ КРИПТОАНАЛИЗА

- исследование возможности расшифровывания информации без знания ключей.

# РАЗДЕЛЫ КРИПТОГРАФИИ

1. Симметричные криптосистемы.
2. Криптосистемы с открытым ключом.
3. Системы электронной подписи.
4. Управление ключами.

# Основные направления использования криптографических методов

1. Передача конфиденциальной информации по каналам связи (например, электронная почта).
2. Установление подлинности передаваемых сообщений.
3. Хранение информации (документов, баз данных) на носителях в зашифрованном виде.

# Алфавиты, используемые в современных ИС

1. Алфавит Z33 – 33 буквы русского алфавита и пробел.
2. Алфавит Z256 – символы, входящие в стандартные коды ASCII и КОИ-8.
3. Алфавит Z2 – {0,1}.
4. Восьмеричный алфавит.
5. Шестнадцатеричный алфавит.

# ШИФРОВАНИЕ

- преобразовательный процесс: исходный текст, который носит также название открытого текста, заменяется шифрованным текстом.

# ДЕШИФРОВАНИЕ

- обратный шифрованию процесс. На основе ключа шифрованный текст преобразуется в исходный.

# КЛЮЧ

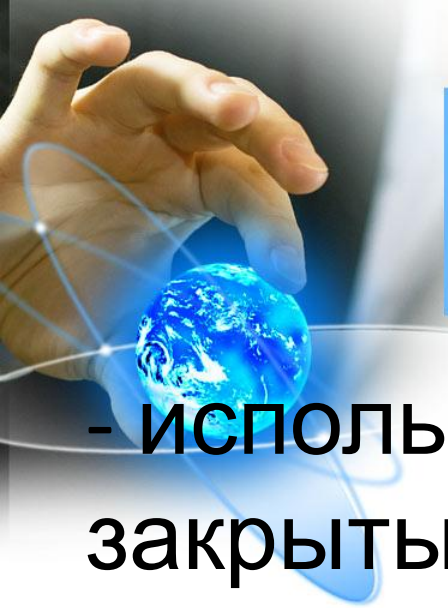
- информация, необходимая для шифрования и дешифрования текстов. Обычно ключ представляет собой последовательн
алфавита.

# Виды криптосистем

1. Симметричные.
2. Асимметричные (другие названия: несимметричные или системы с открытым ключом)

# СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

- для шифрования, и для дешифрования используется один и тот же ключ.



# СИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

- используются два ключа - открытый и закрытый (секретный), которые математически связаны друг с другом. Информация шифруется с помощью открытого ключа и расшифровывается с помощью закрытого. Либо наоборот, шифруется с помощью закрытого ключа и расшифровывается с помощью открытого.

# ПОТОКОВЫЕ АЛГОРИТМЫ

В потоковых алгоритмах способ шифрования отдельного символа не зависит от соседних символов.

# БЛОКОВЫЕ АЛГОРИТМЫ

В блоковых алгоритмах сообщение разбивается на блоки, в которых способ шифрования символов зависит от их положения и окружения.

# ЭЛЕКТРОННАЯ (ЦИФРОВАЯ) ПОДПИСЬ

- присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения.

# КРИПТОСТОЙКОСТЬ

- характеристика шифра, определяющая его стойкость к дешифрованию без знания ключа (т.е. криптоанализу)

# Показатели криптостойкости

1. Количество всех возможных ключей.
2. Среднее время, необходимое для криптоанализа.

# Слабые ключи

- ключи, которые предсказуемо преобразуют некоторый текст (или множество текстов).

**Примеры слабых ключей**: шаг, равный 0 в алгоритме Цезаря; гамма, состоящая из одних нулей в гаммировании.

# Шифр

**Шифром называют пару**: алгоритм и ключ.

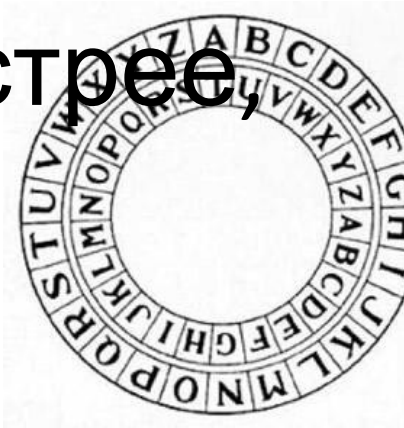
**Эффективность шифрования** зависит от сохранения тайны ключа и криптостойкости шифра.

# Метод грубой силы

Метод «грубой силы» предполагает перебор всех возможных ключей.

# Криптостойкий алгоритм

Алгоритм называют криптостойким, если не существует способов вскрытия зашифрованного текста без знания ключа, дающих результат быстрее, чем метод грубой силы.



# Симметричные криптосистемы

Прежде, чем начать использовать систему, необходимо получить общий секретный ключ так, чтобы исключить к нему доступ потенциального злоумышленника.

# Базовые классы симметричных криптосистем

1. Алгоритмы подстановки или замены.
2. Алгоритмы перестановки.
3. Алгоритмы гаммирования.

# Моноалфавитные подстановки

- это наиболее простой вид преобразований, заключающийся в замене символов исходного текста на другие (того же алфавита) по более или менее сложному правилу. В случае моноалфавитных подстановок каждый символ исходного текста преобразуется в символ шифрованного текста по одному и тому же закону.

# Алгоритм Цезаря

Самый древний алгоритм, предложенный Юлием Цезарем.

В настоящее время не может использоваться, так как его криптостойкость чрезвычайно мала.

# Алгоритм Цезаря

Идея состоит в следующем: задан алфавит и задан шаг (целое число). Шифрование заключается в замене символа исходного текста на символ, отстоящий в алфавите на шаг вправо. Важно, что алфавит рассматривается как «склеенная в кольцо» последовательность символов, то есть вслед за последним символом алфавита идет снова первый.

# Алгоритм замены полиалфавитный

Усложнить шифр можно, используя не один, а несколько алфавитов. Алфавиты могут различаться количеством и порядком символов.

# Алгоритм замены полиалфавитный

Очевидно, что во всех алфавитах должны быть все символы исходного текста и, возможно, еще какие-то. Алфавиты перебираются по некоторому закону, который определен ключом или задан в алгоритме. Одному символу исходного текста соответствует один или несколько символов результирующего.

# Пример алгоритма замены полиалфавитного

Пусть дан первый алфавит «аяздкв бмл»,
второй «яозеадивлкн» и
третий «аколд кмзв».
Пусть алфавиты используются по очереди.
Шаг равен 5.
Исходный текст «задавака».
Тогда результат шифрования:

# Пример алгоритма замены полиалфавитного

Пусть дан первый алфавит «аяздкв бмл»,
второй «яозеадивлкн» и
третий «аколд кмзв».
Пусть алфавиты используются по очереди.
Шаг равен 5.
Исходный текст «задавака».
Тогда результат шифрования: «бквво лк».

# Алгоритм замены с большим ключом

Для обеспечения высокой криптостойкости требуется использование больших ключей. **Большой ключ** приводит к тому, что одна и та же буква исходного текста будет преобразовываться в зашифрованную с разными ключами.

# Алгоритм замены с большим ключом

Можно говорить здесь о криптосистеме с **одноразовым ключом**, то есть ключ настолько большой, что каждый следующий текст шифруется уже с другой его частью. Такой шифр обладает абсолютной теоретической стойкостью, так как взлом одного сообщения не дает никакой информации для взлома другого.

# Проблемы использования алгоритма с большим ключом

Этот способ неудобен для практического применения. Основная его проблема состоит в том, что для расшифрования требуется заранее некоторым секретным способом передать ключ.

# Проблемы использования алгоритма с большим ключом

Чем больше ключ, тем проблематичнее передача. Можно, конечно, передать не сам ключ, а алгоритм его формирования. Но тогда этот алгоритм должен быть секретным.

# Алгоритм замены с большим ключом

Так, например, криптокарта Forteza, используемая агентами национальной безопасности США, базируется на том, что и отправитель и получатель одновременно генерируют одинаковый ключ, пользуясь неким аппаратным средством.

# Схема с одноразовым блокнотом

- применяет алгоритм Диффи-Хэлмана для генерации очередного секретного ключа.

# Пример алгоритма замены

**Пример 1.** Пусть дан алфавит «аяздкв бмл».

Ключ, то есть множество шагов: 1, 5, 3, 7, 2, 6, 4, 1, 7, 3, 3.

Исходный текст «задавака».

Тогда результат шифрования:

# Пример алгоритма замены

**Пример 1.** Пусть дан алфавит «аяздкв бмл».

Ключ, то есть множество шагов: 1, 5, 3, 7, 2, 6, 4, 1, 7, 3, 3.

Исходный текст «задавака».

Тогда результат шифрования: «дв бб мя».

# Пример алгоритма замены

**Пример 2.** Пусть дан алфавит «аяздкв бмл». Задан ключ, то есть алгоритм его вычисления: для шифрования первого символа берется шаг, равный 2, а для каждого последующего – шаг, равный остатку от деления на 10 кода предыдущего исходного символа (номера в таблице ASCII). Исходный текст «задавака».

# Пример алгоритма замены

Коды «з» - 231, «а» - 224, «д» - 228, «в» - 226, «к» - 234.

Тогда результат шифрования:

# Пример алгоритма замены

Коды «з» - 231, «а» - 224, «д» - 228, «в» - 226, «к» - 234.

Тогда результат шифрования: «ккякякмк».

# Пример алгоритма замены

1. После «з» вторая буква «к».
2. Код буквы «а» 224.

   224 mod 10 = 4.

   четвертая буква после «а» - «к».
3. Код буквы «д» 228.

   228 mod 10 = 8.

   восьмая буква после «д» - «я».
4. ……

# Пример алгоритма замены

Т.к. здесь результат шифрования зависим от исходного текста, то такой алгоритм можно отнести к блоковым.

# Перестановки


- несложный метод криптографического преобразования, заключающийся в перестановке местами символов исходного текста по некоторому правилу.

# Перестановки

Блоки информации (байты, биты, более крупные единицы) не изменяются сами по себе, но изменяется их порядок следования, что делает информацию нечитаемой для стороннего наблюдателя. Шифры перестановок в настоящее время не используются в чистом виде, так как их криптостойкость недостаточна.

# Ключ алгоритма перестановки

- множество пар переставляемых символом. Часто его задают в виде таблицы перемешивания.

# Пример алгоритма перестановки

Правило перестановок зададим следующее: первый символ исходного текста меняется местами с символом, номер которого задан в таблице первым. Второй символ уже не совсем исходного текста – с символом, номер которого задан в таблице вторым и т.д. Пусть задана таблица: 4, 3, 2. Если таблица заканчивается, начинается отсчет символов в шифруемом тексте с 1. Исходный текст «задавака».

# Пример алгоритма перестановки

Покажем результат шифрования по шагам: 1 буква меняется местами с 4 «аадзвака», 2 с 3 «адазвака», 3 со 2 «аадзвака», 4 с 4 «аадзвака», 5 с 3 «аавздака», 6 со 2 «аавздака», 7 с 4 «аавкдаза», 8 с 3 «ааакдазв». Расшифрование проводится в обратном порядке.

# Гаммирование

- представляет собой преобразование исходного текста, при котором символы исходного текста складываются (по модулю, равному мощности алфавита) с символами некоторой заданной или генерируемой псевдослучайной последовательности, вырабатываемой по некоторому правилу.

# Гаммирование

В случае если последовательность является истинно случайной (например, снятой с физического датчика) и каждый ее фрагмент используется только один раз, мы получаем криптосистему **с одноразовым ключом. Системы с одноразовым ключом** (как особо надежные) применяются в правительственной связи.

# Гаммирование

Гаммирование является также широко применяемым криптографическим преобразованием. На самом деле граница между гаммированием и использованием бесконечных ключей (и шифров Вижинера) весьма условная.

# Принцип шифрования гаммированием

- заключается в генерации гаммы шифра и наложении полученной гаммы на открытые данные обратимым образом (*например*, используя сложение по модулю 2).

# Принцип дешифрования при гаммировании

- сводится к повторной генерации гаммы шифра при известном ключе (он нужен для того чтобы сгенерировать ту же самую гамму) и наложении такой гаммы на зашифрованные данные.

# Гамма шифра

Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей. По сути дела **гамма шифра должна изменяться случайным образом** для каждого шифруемого слова.

# Гамма шифра

Фактически же, если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (методом «грубой силы»). Криптостойкость в этом случае определяется размером ключа.

# Недостатки метода гаммирования

1. Метод гаммирования становится бессильным, если злоумышленнику становится известен фрагмент исходного текста и соответствующая ему шифрограмма. Простым вычитанием по модулю получается отрезок гаммы.

# Недостатки метода гаммирования

2. Если гамма получена в результате работы генератора псевдослучайных чисел (то есть программно), то знание фрагмента псевдослучайной последовательности может оказаться (почти всегда!) достаточным для восстановления всей последовательности. Фрагмент текста не обязательно должен быть украден.

# Недостатки метода гаммирования

Злоумышленник может сделать предположение о содержании исходного текста. Так, если большинство посылаемых сообщений начинается со слов “СОВ. СЕКРЕТНО”, то криптоанализ всего текста значительно облегчается, т.к. 13 символов гаммы можно определить. Это следует учитывать при создании реальных систем информационной безопасности.

# Виды гаммирования

1. битовое гаммирование;
2. гаммирование в общем виде.

# Битовое гаммирование

- исходный текст складывается с гаммой при помощи операции **XOR** (сложение по модулю)

# Гаммирование в общем виде

- сложение идет по модулю числа, равного длине алфавита.

# Пример модульного сложения букв

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 |

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

*Решение*

TODAY= 19 14 03 00 24

NEVER= 13 04 21 04 17

Сумма = 32 18 24 04 41 06 18 24 04 15 = GSYEP

# Пример модульного вычитания букв

TODAY= 19 14 03 00 24

NEVER= 13 04 21 04 17

Разность = 06 10 -18 -04 07 06 10 08 22 07 = GKIWH

# Пример гаммирования в общем виде

Пусть дан алфавит «абюя 123». В этом алфавите символы пронумерованы от 0 до 7. Длина алфавита равна 8. Пусть у нас есть сообщение «а13» и гамма «ю2». Зашифруем:

а13

ю2ю

# Пример гаммирования в общем виде

Пусть дан алфавит «абюя 123». В этом алфавите символы пронумерованы от 0 до 7. Длина алфавита равна 8. Пусть у нас есть сообщение «а13» и гамма «ю2». Зашифруем:

а13 -> юяб

ю2ю

# Благодарим за внимание!