# <https://ru.bmstu.wiki/DES_(Data_Encryption_Standard)#.D0.9D.D0.B0.D1.87.D0.B0.D0.BB.D1.8C.D0.BD.D0.B0.D1.8F_.D0.BF.D0.B5.D1.80.D0.B5.D1.81.D1.82.D0.B0.D0.BD.D0.BE.D0.B2.D0.BA.D0.B0>

<http://mf.grsu.by/UchProc/livak/b_protect/zok_2.htm>

<https://ozlib.com/1071635/informatika/osnovnye_metody_kriptoanaliza>

# Лабораторная 7

**Блочный шифр** — разновидность симметричного шифра, оперирующего группами бит фиксированной длины — блоками, размер в пределах 64‒256 бит. Если исходный текст меньше размера блока, перед шифрованием его дополняют.

***Различие блочных и поточных***

**Потоковый шифр (иногда говорят «поточный»)** – симметричный шифр, преобразующий каждый символ mi открытого текста в символ шифрованного ci, зависящий от ключа и расположения символа в тексте.

## Сеть Фейстеля

### 4. Охарактеризовать и привести формальное описание сети Фейстеля.

Ячеистая топология

Одна ячейка - соответствует раунду зашифрования(расшифрования)

Сообщение разбивается на блоки одинаковой длины (64 или 128 бит)

Каждый шифруемый блок делится на два подблока одинакового размера(32 или 64 бита) Левый L0 и правый R0

В каждом i-ом раунде выполняется преобразования:

Li=Ri-1

Ri=Li-1+f(Ri-1, Ki)

По какому-либо математическому правилу вычисляется раундовый ключ Ki. В примере + поразрядное суммирование на основе XOR

### 5. Какие стандартные операции используются в блочных алгоритмах шифрования?

***Функция f(Ri-1, Ki):***

блок подстановок (S-блок)

блок перестановок (P-блок)

***Блок подстановок состоит из:***

* дешифратор - преобразует n-разрядное двоичное число в одноразрядное сигнал по основанию 2n;
* внутреннего коммутатора
* шифратора - преобразует сигнал из одноразрядного 2n-ричного в n-разрядный двоичный.

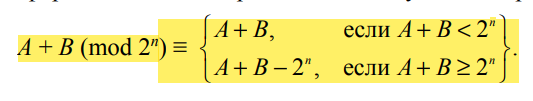
***Блок перестановок*** - изменяет положение цифр, линейное устройство.

### 13. Какие факторы влияют на стойкость блочного алгоритма шифрования?

**Криптостойкость -** шифр должен обладать двумя свойствами **рассеиванием и перемешиванием**

**Рассеивание** - скрывает отношение между зашифрованным и исходным текстами; каждый символ в зашифрованном тексте зависит от одного или всех символов в исходном тексте. Если символ исходного текста изменен - несколько или все символы в зашифрованном тексте будут изменены.

**Перемешивание** - скрывает отношение между зашифрованным текстом и ключем.



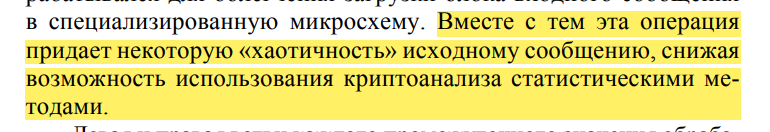
## Алгоритм DES

Строится на основе сети Фейстеля.

Входной блок 64 бита, преобразуется в выходной блок такой же длины. Применяется рассеивание(подстановка) и перестановка

Один блок данных преобразуется в течении 16 раундов

После первоначальной перестановки и разделения 64-битного блока на правую и левую половины длинной по 32 бита выполняются 16 раундов одинаковых действий



* Входные 64-бита
* производится перестановка разрядов
* делится на левую и правую части по 32-бита каждая
* правая часть расширяется до 48 битов для выполнения операции XOR с 48-битным ключом(еще за счет этой операции возрастает зависимость битов результата от битов исходных данных и ключа - лавинный эффект)
* Полученное значение(48-битное) подается на S-блок(блок подстановки)
* При выполнении этой операции - 48 битов делятся на восемь 6-битных подблоков, каждый из которых по таблице замен замещается 4-мя битами.
* Результат 32-битное значение
* Полученное 32-битное значение обрабатывается с помощью перестановки P-блок, не зависит от используемого ключа.
* Результат перестановки объединяется с помощью операции XOR с левой половиной первоначального 64-битного блока данных
* Левая и правая половины меняются местами и начинается следующий раунд.
* После 16-раундов выполняется обратная перестановка.
* Каждый 8-ой бит исходного 64-битного ключа отбрасывается - эти 8 битов - добавлялись чтобы каждый байт имел четное число единиц. Используется для обнаружения ошибок при обмене и хранении ключей.

### 16. Какие ключевые комбинации относятся к слабым (к полуслабым) и почему?

## Слабые и полуслабые ключи

Из-за того что первоначальный ключ изменяется при получении подключа для каждого раунда алгоритма, определенные первоначальные ключи являются ***слабыми***. Первоначальное значение разделяется на две половины которые сдвигаются независимо - если все биты равны 0 или 1, - то используется один и тот же ключ для всех раундово. Происходит это если ключ из 1 или 0, или одна половина из 1 другая из 0.

Некоторые пары ключей при зашифровании переводят открытый текст в идентичный шифртекст. Т.е. один из ключей пары может расшифровать сообщения, зашифрованные другим ключом пары. Это происходит из-за метода, используемого DES для генерации подключей: вместо 16 различных подключей эти ключи генерируют только два различных подключа. В алгоритме каждый из этих подключей используется восемь раз. Эти ключи, называемые ***полуслабыми***.

### 14. В чем состоит сущность дифференциального криптоанализа?

### 15. В чем состоит сущность линейного криптоанализа?

***Криптоанализ: Дифференциальный и Линейный***

***Дифференциальный -***

**Дифференциальный криптоанализ основан на изучении преобразования разностей между шифруемыми значениями на различных раундах шифрования.**

работает с парами шифротекстов, открытый тексты имеют разности, метод анализирует эволюцию этих разностей в процессе прохождения открытых текстов раундов DES при шифровании одним и тем же ключом. Используя разности полученных шифротекстов - присваивают различные вероятности различным ключам. Далее в проц анализа следующих пар шифртекстов один из ключей станет наиболее вероятным - это и есть правильный ключ.

***Линейный***

использующий линейные приближения для описания работы шифра

- находится “хорошие” однораундовые линейные приближения и они объединяются

***Плюсы/Минусы:***

- низкая криптостойкость (полу-слаб. ключи)

+ выс. скорость (малая длина ключа)

+ беспл. распростр. по миру

+ общедоступность

+ нет необх в лицензионных отчислениях

**Модификацией DES является 3DES **

**Несколько алгоритмов DES3**

* DES-EEE3: шифруется 3 раза с 3 разными ключами (операции шифрование-шифрование-шифрование);
* DES-EDE3: 3DES операции шифрование-расшифрование-шифрование с разными ключами;
* DES-EEE2 и DES-EDE2: как и предыдущие, однако на первом и третьем шаге используется одинаковый ключ.

## Стандарт AES

**AES (Advanced Encryption Standard)** – алгоритм шифрования, действующий в качестве государственного стандарта в США с 2001 г. В основу стандарта положен шифр Rijndael. Характеризуется размером блока 128 битов, длиной ключа 128, 192 или 256 битов и количеством раундов 10, 12 или 14 в зависимости от длины ключа.

Основу Rijndael составляют так называемые линейно-подстановочные преобразования. В алгоритме широко используются табличные вычисления, причем все необходимые таблицы задаются константно, т. е. не зависят ни от ключа, ни от данных.

## Стандарт ГОСТ 28147-89

Три режима шифрования:

* простая замена
* гаммирование
* гаммирование с обратной связью

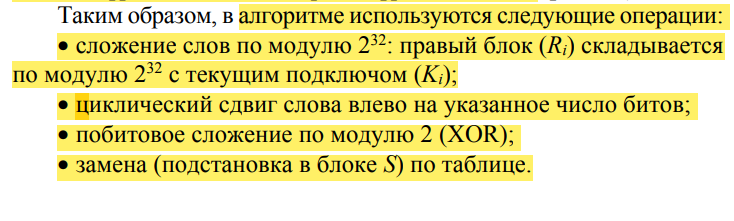
Построен по тем же принципам что и DES - но применяется более длинный ключ(256 бит), 32 раунда шифрования. Размер блока составляет 64 бита.

Блок разбивается на две одинаковые части правая и левая

Правая часть складывается по модулю 2^32 с подключом раунда

Посредством принятого алгоритма шифрует левую часть

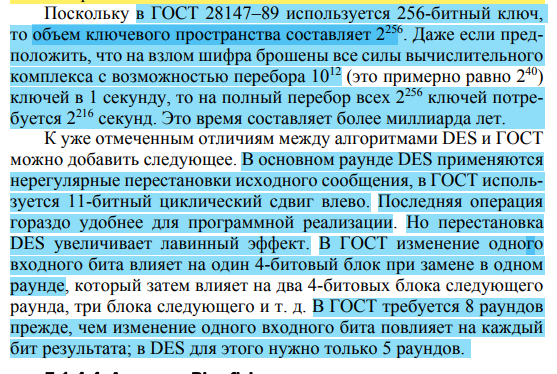
Перед следующим раундом левая и правая части меняются местами.



### 

### 10. Выбрать два произвольных блочных алгоритма. В чем состоят отличия между ними?

Сравнение ГОСТ и DES - на странице 83 выделено голубым цветом.



## Алгоритм Blowfish

Длина блока 64-бита, ключ до 448-бит. Перед зашифрование выполняется сложная фаза инициализации, зашифрование происходит быстро.

Каждый раунд состоит из перестановки зависящей от ключа, и подстановки, зависящей от ключа и данных. Операции XOR и 2^32

Blowfish - использует много подключей, они должны быть вычислены заранее, до начала зашифрования

Элементы алгоритма:

* P-массив - восемнадцать 32-битных подключей.
* S-блоки - каждый из четырех 32-битных блоков содержит 256 элементов
* функция F

### 1. Какие простейшие операции применяются в блочных алгоритмах шифрования?

сложение по модулю 2^n, XOR

### 2. В чем отличие блочных алгоритмов шифрования от потоковых?

сообщение разбивается на равные блоки, к которым применяте однотипная процедура шифрования(раунды). Потоковые - шифруемые символы - одна буква, бит, байт

### 3. Что понимается под «раундом» алгоритма шифрования?

повторяющийся набор преобразований при зашифровании и расшифровании

### 

### 6. В чем состоит особенность сложения чисел по модулю 2n ?

* самое большое слагаемое <2^n
* результат сложения n-разрядное число
* побитовое сложение предусматривает известную взаимосвязь между соседними символами
* A+B(mod2^n) = остаток от (A+B)/2^n

### 7. Сложить по модулю 102 пары чисел: 55 и 14; 76 и 24; 99 и 99.

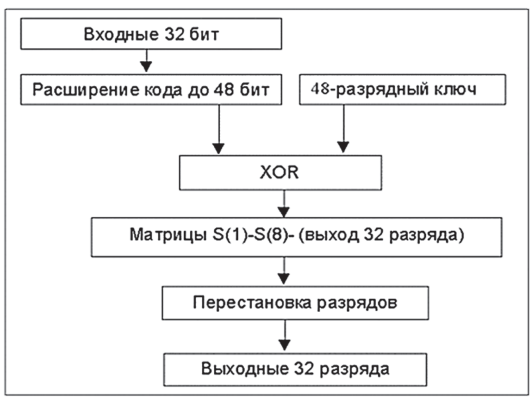
### 8. Сложить по модулю 28 : двоичные числа 10101100 и 11001010; 01111111 и 01101101; шестнадцатеричные числа 0В5 и 37.

### 9. Дать пояснение принципам реализации «лавинного» эффекта.

Возрастание зависимости всех битов результата от исходных битов и ключа.

### 11. Представить графически и пояснить функционал одного раунда блочного алгоритма DES (АES, ГОСТ 28147–89, Blowfish).





### 12. Сколько можно реализовать (теоретически) разновидностей алгоритма 3DES?

### 17. Где применяются блочные криптоалгоритмы?

**3DES** с тремя ключами реализован во многих Интернет-приложениях

**ГОСТ** - удобен для программной реализации

**Blowfish** - предназначен в основном для приложений, в которых ключ меняется нечасто

# Лабораторная 8

### 1. В чем состоит особенность потоковых шифров?

***Потоковый шифр*** (иногда говорят «поточный») – симметричный шифр, преобразующий каждый символ mi открытого текста в символ шифрованного ci, зависящий от ключа и расположения символа в тексте.

Все потоковые шифры делятся на 2 класса: **синхронные** и **асинхронные** (или самосинхронизирующиеся)

Основная задача потоковых шифров - является выработка некоторой последовательности(гамма) для зашифрования, т.е. выходная гамма является ключевым потоком для сообщения.

### 2. В чем состоят преимущества и недостатки синхронных и асинхронных потоковых шифров?

**Синхронные -**  поток ключей генерируется независимо от открытого текста и шифртекста

Свойства:

* Искажение одного символа в шифртексте искажает только один символ в расшифрованном текста
* Защита от любых вставок и удалений шифртекста, т.к. они приведут к потере синхронизации и будут обнаружены
* Нарушение синхронизации приводит к искажению всех символов после синхронизации

**Самосинхронизирующиеся** - ключ зависит от исходного сообщение или от шифротекста. Поток ключей создается функцией ключа и фиксированного числа знаков шифротекста(N): внутреннее состояние генератора является функцией предыдущих N битов шифротекста - генератор потока ключей, приняв N битов, автоматически синхронизируется с шифрующим генератором

Свойства:

* каждый знак открытого текста влияет на следующий шифртекста - статистические свойства открытого текста распространяются на весь шифртекст
* удаленный или добавленный символ вызывает ограниченное кол-вл ошибочных символов в дешифрованном тексте, после чего правильный текст восстанавливается
* каждому неправильному биту шифртекста соответствуют N ошибок в открытом тексте

## Генераторы ключевой информации

Потоковый шифр должен имитировать одноразовый блокнот. Ключ должен по своим свойствам максимально походить на случайную числовую последовательность.

Ключевые последовательности(или псевдослучайные последовательности) - вырабатываются специальными блоками систем потокового шифрования - генераторами

***Случайные числа(последовательности)*** - последовательность элементов, каждый из которых не может быть предсказан только на основе знания предшествующих ему элементов данной последовательности

***Псевдослучайные*** - последовательность элементов, полученная в результате выполнения некоторого алгоритма и используемая в конкретном случае вместо последовательности случайных чисел.

### 4. Дать характеристику линейным конгруэнтным генераторам. Области их применения.

## Линейный конгруэнтный генератор

Соотношение xt+1 = (axt + c ) mod n

xt и xt+1 - члены числовой последовательности

a, c, n - константы

Период не превышает n

Если параметры a, b, c выбраны правильно то генератор будет продолжать случайные числа с max периодом, равным c. При программной реализации значение с устанавливается равным 2^(b-1) или 2^b, b - длина слова в битах

Генератор не используется в криптографии - из-за низкой криптостойкости.

### 6. Представить общую структурную схему генератора ПСП на основе регистров сдвига с линейной обратной связью. Пояснить особенности его функционирования.

## Генератор ПСП на основе регистров сдвига

Состоит из двух частей - РС(регистры сдвигов), функции обратной связи(реализуются с помощью сумматоров сложения по модулю две операция xor). Генератор строится на основе - примитивных порождающих полиномов(многочленов)

Многочлены могут быть приводимыми и неприводимыми

например x^3+x+1 - неприводимый

Многочлен приводимый если он может быть представлен как произведение двух многочленов меньших степеней с коэффициентами из данного поля

Если многочлен приводимой то период ПСП будет максимально возможным 2^L-1(L-наибольшая степень полинома)

### 9. Как устроен генератор ПСП на основе RSA? На чем основана криптостойкость реализуемого алгоритма?

## Генератор псевдослучайных чисел на основе алгоритма RSA

Разработан для систем асимметричного зашифрования/расшифрования

Последовательность генерируется с использованием соотношения

xt = (xt-1)^e mod n

Начальный параметр - n, большие простые числа p и q (n=pq), целое число e, взаимно простое с произведением (p-1)(q-1), некоторое случайное начальное значение x0

Выход генератора - на t-м шаге - младший бит числа xt.

Безопасность генератора опирается на сложность взлома алгоритма RSA, т.е. на разложение числа n на простые сомножители.

## Генератор псевдослучайных чисел на основе алгоритма BBS

Генератор на основе квадратичных вычетов

Начальное значение генератора вычисляется на основе соотношения

x0 = x^2 mod n

n - произведение чисел p и q(p/4 и q/4 - остаток 3

x - взаимно простое с n

Выходом генератора на t-м шаге является младший бит числа xt

xt = (xt-1)^2 mod n

Безопасность - при известных p и q xt-й бит легко вычислить без учета предыдущего (xt-1) бита

xt = (x0)^a mod n

a = 2^t mod ((p-1)(q-1))

Алгоритм медленный, для ускорения использовать не последний а несколько последних битов, при этом алгоритм менее криптостойкий.

### 11. Пояснить базовый алгоритм, реализованный в шифре RC4.

## Потоковый шифр RC4

Потоковый шифр с переменным размером ключа. Гамма не зависит от открытого текста.

Строится на основе генератора псевдослучайных битов(генератор ПСП). На вход генератора записывается ключ, а на выходе читаются псевдослучайные биты. Длина ключа от 40 до 2048 битов.

Ядро - состоит из функции генератора ключевого потока.

Вторая часть - функция инициализации, использует ключ переменной длины, Ki для создания начального состояния генератора ключевого потока.

Размер блока или слова - параметр n. Обычно n=8

Внутреннее состояние шифра определяется массивом слов размером 2^n. При n=8 элементы блока представляют собой перестановку чисел от 0 до 255, перестановка зависит от ключа переменной длины.

Два счетчика с нулевыми начальными значениями.

В основе вычислений операция по mod 2^n

Генератор переставляет значения, хранящиеся в S и каждый раз выбирает различное значение из S в качестве результата.

В одном цикле определяется одно n-битное слово K из ключевого потока, которое в последующем суммируется с исходным текстом для получения зашифрованного текста.

При n=8

i = (i+1) mod 256

j = (j+1) mod 256

меняем местами Si и Sj

a = (Si +Sj) mod 256

K = Sa

Байт K исп в операции XOR с открытым текстом для получения 8-битного шифртекста или для его расшифрования.

Инициализация S-блока

Использует ключ, который подается на вход пользователем.

Заполняется линейно s0=0, s1=1, … s255=255

Другой массив заполняется ключем( размер массива 256-бит ключ повторятся столько раз сколько нужно)

Массив S перемешивается путем перестановок определяемых ключем.

1. j=0, i=0
2. j = (j + Si +Ki) mod 256
3. поменять местами Si и Sj
4. i = i+1
5. если i<256, то перейти на п2

Применяется в некоторых широко распространенных стандартах и протоколах шифрования, таких как WEP, WPA, TLS, Kerberos

# Лабораторная 9

В основе асимметричной криптографии положена идея использования ключей парами - один для зашифрования(публичный), второй для расшифрования(тайный)

Все алгоритмы шифрования с открытым ключом основаны на использовании односторонних функций(например вычисление дискретного логарифма)

***Односторонняя функция -*** математическая функция, которую относительно легко вычислить, но трудно найти значение функции, т.е. зная x легко вычислить f(x), но зная f(x) - трудно найти подходящее значение x

Алгоритмы с открытым ключом можно использовать для решения:

* зашифрования/расшифрования передаваемых и хранимых данных в целях их защиты от несанкционированного доступа
* формирования цифровой подписи
* распределения секретных ключей, исп при шифровании документов симметричными методами

По Диффи-Хелману алгоритм шифрования должен обладать:

* вычислительно легко создавать пару (открытый и закрытый ключи)
* вычислительно легко зашифровать сообщение Mi открытым ключом
* вычислительно легко расшифровать сообщение Ci тайным ключом
* непреодолимой сложностью вычислить тайный ключ зная открытый
* непреодолимой сложностью расшифровать исходное сообщение зная открытый ключ и зашифрованное сообщение

## Криптоалгоритм на основе задачи об укладке ранца

### 1. Что такое «ранцевый (рюкзачный) вектор»? Дать определение.

Рюкзачный вектор S` = (s1, s2, … sz) - это упорядоченный набор из z, z>=3, различных натуральных чисел si. Входом задачи о ранце называется пара (S`, S) - где S` - рюкзачный вектор, S - натуральное число.

Решением для входа (S`, S) - будет такое подмножество из S`, сумма элементов которого равняется S.

### 2. Сформулировать задачу укладки ранца.

Дано множество предметов общим числом z различного веса.

Спрашивается, можно ли положить некоторые из этих z в ранец так, чтобы его вес стал равен определенному значению S

Более формально:

Дан набор значений k1, k2, … kz и суммарное значение S. Требуется вычислить значения bz такие, что

S = b1k1 +b2k2 + … +bzkz

bi - 0/1, если b=0 значит предмет не кладут в рюкзак, b=1 кладут

Суть метода для шифрования состоит в том, что существует две различные задачи укладки ранца, одна из них решается легко, другая решается трудно.

Трудный для укладки ранец применяется в качестве открытого ключа, который легко используется для зашифрования, и невозможен для расшифрования.

Закрытый ключ - легкий для укладки ранец, который предоставляет простой способ для расшифрования сообщения.

**Закрытый ключ d** - легкий для укладки ранец - сверх возрастающая последовательность, состоящая из z элементов: d1, d2, … dz: d = {di}, i=1, … z

### 4. Что такое сверхвозрастающая последовательность? Привести примеры.

***Сверхозростающая последовательность*** каждый последующий член больше суммы всех предыдущих

## Алгоритм укладки ранца на основе сверхвозрастающей последовательности

Необходимо по очереди анализировать некоторый “текущий вес” S предметов, составляющих сверхвозрастающую последовательность; в результате анализа нужно упаковать ранец.

* Текущим числом выбирается S сравнивается с весом самого тяжелого предмета, если вес меньше веса предмета то его кладут в ранец
* Если предыдущий предмет попал в рюкзак от текущего веса отнимаем вес того что положили в рюкзак, и сравнивается уже с новым предметом
* И так до того момента пока S=0, решение найдено - в противном случае нет

*Пример*

Последовательность {2, 3, 6, 13, 27, 52, 105, 210}

S = 270

270 = 210 + 52 + 6 + 2

тайный ключ 10100101

Открытый ключ формируется на основе тайного:

ei = di\*a mod n

a - некоторое число

n - больше суммы всех чисел последовательности НОД(a, n) = 1

*Продолжение примера*

Сумма всех чисел равна 418, пусть n = 420, a =31 НОД(420, 31) = 1

e1 = 2\*31 mod 420 = 62 и т.д.

## Зашифрование сообщения

* Сообщение разбивается на блоки(по размерам равные числу z элементов посл в ранце)
* 1 - указывает на присутствие элемента, 0 - отсутствие - вычисляются полные веса рюкзаков по одному ранцу для каждого блока сообщения с использованием открытого ключа e получателя

*Продолжение пример*

Сообщение из 7 букв представляем в бинарном виде

e: {62, 93, 186, 403, 417, 352, 315, 210}

Первый символ в бинарном виде 11010000

Укладка ранца 62+93+403

Вес ранца 558

и т.д. Зашифрованное сообщение С = 558 470 155 365 1239 155 924

## Расшифрование сообщения

Вычисляем a^-1

a\*a^-1 mod n = 1 - применяем алгоритм Евклида

После определения обратного числа, каждое значение шифрограммы преобразуется в соответствии с соотношением:

Si = cia^-1 mod n

По итогу этих вычислений полученные значения будут считаться весом ранца

*Продолжение примера*

a^-1 = 217/31 \* 271 mod 420 = 1

d = {2, 3, 6, 13, 27, 52, 105, 210}

n =420, a =31

С = 155 365 558 155 924 1239 470

S1 = 155 \* 271 mod 420 = 5

используя d получаем 11000000

и этой последовательности соответствует символ алфавита в используемой таблице кодировки.

### 9. Что такое «секретная лазейка»?

### 10. Охарактеризовать криптостойкость алгоритма на основе задачи об укладке ранца.

## Безопасность криптоалгоритма на основе задачи об укладке ранца

Криптостойкость во многом определяется скоростью поиска нужного варианта укладки ранца.

Ранцевые криптосистемы не являются криптостойкими.

Обнаружили что зная числа a и a^-1 и n, можно восстановить свехвозростающую последовательность по нормальной последовательности. Числа a и n не обязательно должны быть теми же, что использовались при создании системы легальным пользователем.

# Лабораторная 10

Асимметричная криптография основана на сложности решения некоторых математических задач:

* разложение больших чисел на простые сомножители(задача факторизации)
* вычисление дискретного логарифма в конечном поле, а также вычислительные операции над точками эллиптической кривой.

Эти задачи используют операцию получения остатка от целочисленного деления

**Основная теорема арифметики**. Всякое натуральное число N, кроме 1, может быть представлено как произведение простых множителей

**Задача дискретного логарифма** для данных целых чисел a и b, a > 1, b < n, найти логарифм - такое целое число x, что a^x = b(mod n) - x logab

Китайская теорема об остатках. Если разложение N на простые множители представляет собой p1p2…pt(некоторые простые числа могут встречаться несколько раз) то система уравнений (x mod pi) = ai

i = 1, 2, … t - имеет единственное решение: x, меньшее N

Число однозначно определяется своими вычетами по модулю от этих простых чисел.

### 1. Охарактеризовать алгоритмы RSA и Эль-Гамаля. Для каких целей они могут применяться? 2. На чем основана криптостойкость алгоритмов RSA и Эль-Гамаля?

## Алгоритм RSA

Безопасность основана на сложности - разложения на множители больших чисел.

Открытый и тайны ключи - функции двух больших простых чисел.

* Генерация ключей используется два больших случайных простых числа p и q. Для max криптостойкости p и q - равной длины
* n=pq - один из трех компонентов n, e, d
* Случайном образом выбирается e, такой что e и (p-1)(q-1) - взаимно простые
* Используем расширенный алгоритм Евклида ed = 1 (mod фи(n))
* Ключ состоит из трех чисел, которые образуют (e, n) - открытых ключ, (d, n) - тайный ключ

Для з/р используется ключ получателя: отправитель шифрует сообщение открытым ключом получатель расшифровывает своим тайным)

Зашифрование - ci = (mi)^e mod n

Расшифрование - mi = (ci)^d mod n

### 1. Охарактеризовать алгоритмы RSA и Эль-Гамаля. Для каких целей они могут применяться? 2. На чем основана криптостойкость алгоритмов RSA и Эль-Гамаля?

## Алгоритм Эль-Гамаля

Можно использовать для:

* з/р данных,
* формирование цифровой подписи
* согласования общего ключа

Безопасность основана на трудности вычисления дискретных логарифмов

Формируем общий секретный ключ для абонентов и сообщение шифруется путем умножения его на этот ключ

Отличие от RSA:

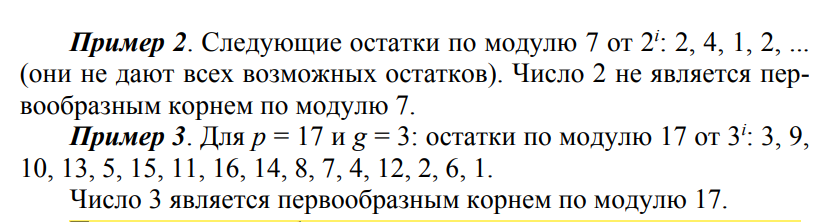
* генерацией ключевой инф и числом компонент составляющих ключ
* каждому блоку открытого текста соответствует два блока зашифрованной
* Используется число k которое практически никак не связано с ключевой информацией получателя и которое принимет различные значения при шифровании различных блоков

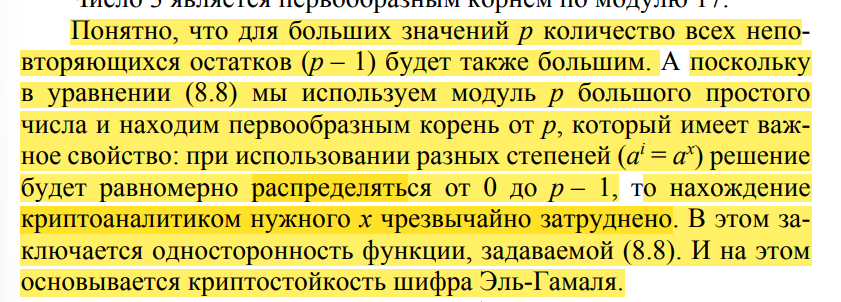
*Генерация ключевой информации*

* Выбираем простое число p
* Выбираем число g (g<p) - первообразный корень p
* Выбираем число x (x<p)
* Вычисляется компонент кл инф y = g^x mod p
* Владельцу кл инф(состоит из 4 чисел) может посылаться некоторый шифртекст созданный с исп открытого ключа получателя: p, g, y. Расшифрование - тайным ключом p, g, x

### 3. Что такое первообразный корень?

Первообразный корень по модулю p является таким числом, что его степени (g^i, 1<=i<=p-1) дают все возможные по модулю p вычеты(остатки), которые взаимно простые с p





*Зашифрование*

Блок шифртекста состоит из двух чисел ai и bi

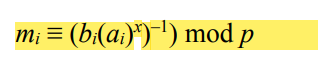
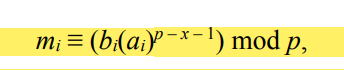
ai = g^k mod p

bi = (y^kmi) mod p

Длина зашифрованного текста увеличится примерно в 1,5

k - после вычислений уничтожается

*Расшифрование*

 или 

(a^x)^-1 - обратное значение числа a^x по модулю p