### Сущность проблемы информационной безопасности систем. Характеристики и параметры ИС и ИВС.

Информационная безопасность систем – свойство информационной системы или реализуемого в ней процесса, характеризующее способность обеспечить необходимый уровень своей защиты.

**Информация** – сведения (данные) о внутреннем и окружающем нас мире, событиях, процессах, явлениях и т. д., воспринимаемые и передаваемые людьми или техническими устройствами.

**Информационная система** – организационно упорядоченная совокупность документов и информационных технологий, реализующая информационные процессы.

**Информационные процессы** – процессы сбора, накопления, хранения, обработки, передачи и использования информации.

**Безопасность ИС** – свойство ИС, выражающееся в способности ис противодействовать попыткам несанкционированного доступа.

**Защита информации** – организационные, правовые, программно-технические и иные меры по предотвращению угроз безопасности.

**Надежность ИС** – характеристика способности программного, аппаратного средства выполнить при определенных условиях требуемые функции в течение определенного периода времени.

Надежность является свойством, включающим в себя: безотказность, ремонтопригодность, сохраняемость, долговечность.

**Безотказность** – это свойство технического объекта непрерывно сохранять работоспособное состояние в течение некоторого времени.

**Ремонтопригодность** – это свойство технического объекта, заключающееся в приспособленности к восстановлению работоспособного состояния путем технического обслуживания.

**Сохраняемость** – характеризует способность технического объекта сохранять в пределах значения после его хранения или транспортировки.

**Долговечность** – это свойство технического объекта сохранять в заданных пределах работоспособное состояние до наступления предельного состояния при установленной системе ремонта.

**Методы повышения надежности**

1) уменьшение времени работы;

2) снижение интенсивности отказов;

3) улучшение восстанавливаемости;

4) резервирование.

безопасность, криптография, методы защиты информации, классы криптографических методов, первую лабораторную (правовые, технические, аппаратно-программные), политика безопасности и ее разработка

# Характеристика угроз безопасности современным ИС и ИВС.

1. Анализ сетевого трафика.

2. Сканирование сети.

3. Угроза выявления пароля.

4. Подмена доверенного объекта сети и передача по каналам связи сообщений от его имени с присвоением его прав доступа.

5. Внедрение ложного объекта сети.

6. Отказ в обслуживании.

7. Удаленный запуск приложений.

Фактор, воздействующий на ИВС – это явление, действие или процесс, результатом которых может быть утечка, искажение, уничтожение данных, блокировка доступа к ним, повреждение или уничтожение системы защиты. Все многообразие дестабилизирующих факторов можно разделить на два класса: внутренние и внешние.

Внутренние дестабилизирующие факторы, влияющие:

1) на программные средства (ПС):

а) некорректный исходный алгоритм;

б) неправильно запрограммированный исходный алгоритм (первичные ошибки);

2) на аппаратные средства (АС):

а) системные ошибки при постановке задачи проектирования;

б) отклонения от технологии изготовления комплектующих изделий и АС в целом;

в) нарушение режима эксплуатации, вызванное внутренним состоянием АС.

Внешние дестабилизирующие факторы, влияющие:

1) на программные средства:

а) неквалифицированные пользователи;

б) несанкционированный доступ к ПС с целью модификации кода;

2) аппаратные средства:

а) внешние климатические условия;

б) электромагнитные и ионизирующие помехи;

в) перебои в электроснабжении;

г) недостаточная квалификация обслуживающего персонала;

д) несанкционированный доступ с целью нарушения работоспособности АС.

# Политика безопасности современных ИС и ИВС.

**Основными этапами программы обеспечения безопасности являются следу­ющие:**

* определение ценности технологических и информационных активов орга­низации;
* оценка рисков этих активов (сначала путем идентификации тех угроз, для которых каждый актив является целевым объектом, а затем оценкой веро­ятности того, что эти угрозы будут реализованы на практике);
* установление уровня безопасности, определяющего защиту каждого акти­ва, то есть мер безопасности, которые можно считать рентабельными для применения;
* формирование политики безопасности организации на базе предыдущих этапов;
* привлечение необходимых финансовых ресурсов для реализации политики безопасности, приобретение и установка требуемых средств безопасности;
* проведение разъяснительных мероприятий и обучения персонала для под­держки сотрудниками и руководством требуемых мер безопасности;
* регулярный контроль пошаговой реализации плана безопасности с целью выявления текущих проблем, учета изменения внешнего окружения и вне­сение необходимых изменений в состав персонала.

**К политикам безопасности предъявляются следующие основные требования:**

* политики безопасности должны:

-     указывать цели и причины, по которым нужна политика;

-     описывать, что именно охватывается этими политиками;

-     определить роли, обязанности и контакты;

-     определить, как будут обрабатываться нарушения                 безопасности;

политики безопасности должны быть:

-     реальными и осуществимыми;

- лаконичными и доступными для понимания;

-  сбалансированными по защите и производительности.

**Первыми шагами по разработке политики безопасности являются следующие:**

-     создание команды или выбор подрядной организации по разработке политики;

-     принятие решения об области действия и целях политики. В области дей­ствия должно быть указано, кто охватывается этой политикой;

-     принятие решения об особенностях разрабатываемой политики;

-  определение лица или органа для работы в качестве официального интер­претатора политики.

# Энтропия источника сообщения. Энтропия Шеннона.

**Алфавит** – это конечная совокупность символов (знаков) с помощью которых можно представить любое сообщение в ИС: -i-ый символ алфавита

ИС технического плана использует бинарный алфавит. Число символов в любом алфавите называется **мощностью алфавита N(A)**.

Символ *с* будет символом равным конкретному символу алфавита есть **вероятностная характеристика данного символа**.

Вероятность того, что произвольный символ ξ произвольного документа (текст, база данных, текст программы) будет буквой «»:

Пример 1. Как часто в английском алфавите повторяются буквы:

– в среднем 13 случаях из 100 будет буква е.

- самая маленькая вероятность у буквы z.

Чем больше выборка (генеральная выборка), тем точнее от частоты события (частности) мы можем прийти к характеристике вероятности.

**Энтропия алфавита** (источника сообщения) – информационная характеристика алфавита (источника сообщения на основе этого алфавита).

Понятие энтропии введено для характеристики термодинамических систем. Оно по большому счету характеризует всякие неравновесности, мы смотрим на это как меру неопределённости или хаоса, а в информационной системе – меру информативности каждого символа.

Понятие энтропии применительно к ИС ввел К. Шеннон (весь мир считает его за отца теории информатики). Шеннон определил энтропию алфавита в виде следующего соотношения:

Энтропия выражается в битах. **Bit** (binary digit – двоичная цифра) – наименьшая и основная единица измерения количества информации.

С физической точки зрения энтропия показывает, какое количество информации (бит) приходится в среднем на один символ алфавита. И с её помощью можно измерять количество информации.

# Энтропия источника сообщения. Энтропия Хартли.

*H(A) = - (\*)*

Энтропия Хартли есть частный случай энтропии Шеннона. Она рассматривается в предположении неизменности появления символа. Все они одинаковы. При одинаковой вероятности появления символов алфавита имеем такую вероятность при N символов:

*P(ai)=const=1/N*

Если такую вероятность подставить в формулу Шеннона, то получим:

***HH(A)*** *=* ***-****-log2*

Энтропия Хартли находится в битах. **Бит** — наименьшая и основная единица измерения количества информации. **Бит** — двоичная цифра.

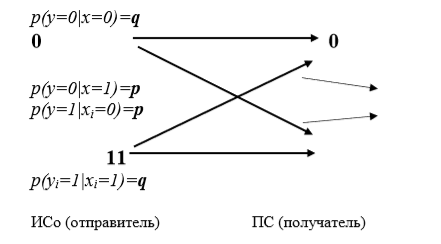
Чем больше мощность алфавита, тем информативнее каждый символ этого алфавита, поскольку **энтропия** — это количество информации, соответствующей одному символу алфавита. Если алфавит двоичный, то мы математически получим подтверждение, что количество информации, приходящейся на 1 символ будет 1 бит, если 4 символа — 2 бита, если 8 — 3 бита.

В основании логарифма стоит цифра 2, если вместо нее поставить число b, то мы будем говорить, что энтропия измеряется не в битах, а в других единицах. Если b=3, то энтропия измеряется не в битах, в тритах, если b=e, то в натах. Мы в основном будем использовать основание два, но мы должны знать и другие основания. Понятно, что существуют однозначные соответствия, если брать алгоритм по разным основаниям.

# Двоичный канал передачи информации.

Процесс возникновения ошибок описывает модель под названием двоичный симметричный канал (ДСК). Двоичный канал передачи информации строится на основе двоичного алфавита: А={0,1}.

В общем случае, если передается сообщение Хk=х1, х2, …, хk, а принимается сообщение Yk= y1, y2, …., yk, то рассмотренные условные вероятности можно рассматривать с двух точек зрения: p(хi|yj) и p(yj|хi).



На рис. обозначены: q– вероятность безошибочной передачи бита сообщения, р– вероятность передачи бита с ошибкой. Понятно, что p + q = 1.

ДСК состоит в том, чтобы построить лучший канал. Если на каком-либо этапе дальнейшее улучшение канала оказывается неэкономичным или технически невозможным, то передача повторяется столько раз, сколько окажется нужным для того, чтобы результирующая [вероятность](http://edu.sernam.ru/book_kiber1.php?id=227) ошибки стала ниже некоторой удовлетворяющей проектировщика границы. Трудность, связанная с классическим подходом, состоит в том, что когда эта граница вероятности ошибки стремится к нулю, то или канал становится несоразмерно дорогим, или доход от его использования оказывается несоразмерно низким.   
Теоремы Шенноны

1. Для заданного канала возможно при помощи кодирования вести передачу с вероятностью ошибки, меньшей любого наперед заданного значения, если скорость передачи информации не превышает некоторого предела, известного под названием [пропускной способности канала](http://sernam.ru/book_p_net.php?id=115) С.

2. Обратно, для скоростей передачи информации, больших С, невозможно вести передачу со сколь угодно малой вероятностью ошибки.

# Энтропия двоичного алфавита.

Бинарный алфавит состоит из двух символов: 0 и 1 (А{0,1}). Соответственно, есть две вероятности p(ai=0)=р(0) и р(ai=1)=р(1).

*р(0)+р(1)=1 (\*\*)*

***H(A) = -***

**Вопрос:** каким условиям соответствует максимальное значение энтропии двоичного алфавита?

Возьмём производную:

Вычислив производную, получим: p(1)=1/2=0,5, а потом посчитаем p(0)=1/2=0,5. Подставляем это в (\*):

Мы получили математическое подтверждение, что информативность одного символа двоичного алфавита равна 1 биту.

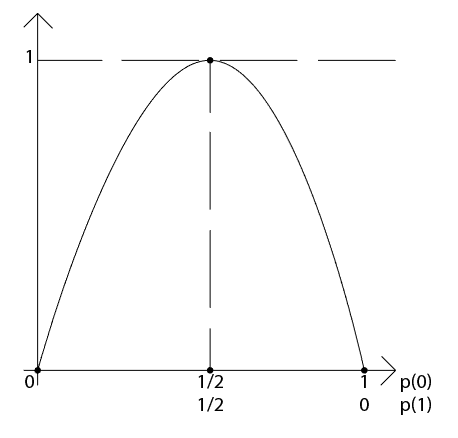
Каждый символ будет весить 1 бит только в том случае, если в сообщении содержится одинаковое количество 0 и 1.

Когда нажимаем на клавишу клавиатуру генерируется последовательность длиной 1 байт.

Как меняется энтропия двоичного алфавита?

Если в сообщении есть только 0, то вероятность появления нуля – 1, а энтропия равна 0, значит, ни в одном символе алфавита нет никакой информации. **Энтропия** — это мера неопределённости, мера хаоса. Клод Шеннон писал, что если сообщение не снимает неопределённости (не даёт новой информации, новых сведений), то тогда оно не информативно.

**Зависимость энтропии двоичного алфавита от вероятности появления 0 и 1:**



Мы работаем не в самой точке, а в окрестности этой точки, энтропия равна 1 бит (мы так считаем).

**Количество информации в сообщении**

Количество информации I в произвольном сообщении Хk, где k — число символов в сообщении, определяется соотношением:

*I(Xk) = H(A)\*k*

**Пример.** Подсчитать I(Xk=’I love you’) при H(A)=4,7 бит.

# Условная энтропия. Энтропийная оценка потерь при передаче информации.

Раз информация теряется, то должна быть причина. Основная причина потерь информации помимо того, что мы пересылаем тайное сообщение по открытому каналу оно может быть прочитано кем бы то ни было (никакого шифрования нет), но есть ещё и другой вид потерь: передача информации по несовершенному (ненадёжному) каналу. Есть провода медные, алиминиевые, можно поставить золотую и серебряную жилу, и качество канала будет разным, так как разное сопротивление.

Пусть в ИС сообщение Xk=x1, x2, …, xi, …, xk на входе канала формируется на основе А={ai}, i=1 … N. Сообщение на выходе канала (Yk=y1, y2, …, yi, …, yk) формируется на основе того же алфавита А.

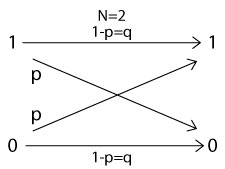
При передаче сообщения по каналу могут появляться ошибки (изначально передавали один символ, а при получении символ заменился на другой).

Отправляем сообщение: Xk=1010

Получаем сообщение: Yk=1110

Если вероятность превращения (трансформирования) 0 в 1 и 1 в 0 равна, условно говоря, p, то тогда мы говорим о симметричных каналах.

**Пример**. Двоичный симметричный канал (ДСК), так как мощность алфавита = 2:



Анализ основывается на использовании теоремы или **формулы Байеса**, которая в своей основе позволяет рассчитать вероятность совместного события, используя *условные вероятности* и *вероятности отдельных событий.*

Обозначим *условную вероятность* события:

*P(xi|yj) : P(0|0) = P(1|1) = q; P(1|0) = P(0|1)=p*

P (0|1) — это условная вероятность того, что получатель имеет 1 при условии, что источник сообщения передавал 0 или вероятность того, что получатель получает 0, а передавалась 1. Не вдаваясь в математику, вспомним формулу Байеса. Рассматривается совместная вероятность событий (подробнее в малиновой книге). Анализ основывается на использовании теоремы, или формулы Байеса, используя условные вероятности или вероятности отдельных событий.

**Стоит задача:** определить количественно потери информации, вызванные несовершенством ИС (канала), т.е. при р>0.

Задача относится к области *проверки гипотез и принятия статистических решений*. **Математическая основа** — теорема Байеса. Совместная вероятность случайных событий А и В:

P(A, B) = P(A|B) P(B) = P(B|A) P(A)

P(A|B)=

Для ДСК:

P(X=0|Y=0)=[P(Y=0|X=0)\*P(X=0)]/[P(Y=0|X=0)\*P(X=0)+P(Y=0|X=1)\*P(X=1)]

P(X=1|Y=0)=[P(Y=0|X=1)\*P(X=1)]/[P(Y=0|X=0)\*P(X=0)+P(Y=1|X=1)\*P(X=1)]

P(X=0|Y=1)=[P(Y=1|X=0)\*P(X=0)]/[P(Y=1|X=0)\*P(X=0)+P(Y=1|X=1)\*P(X=1)]

P(X=1|Y=1)=[P(Y=1|X=1)\*P(X=1)]/[P(Y=1|X=0)\*P(X=0)+P(Y=1|X=1)\*P(X=1)]

Если р>0, то это можно трактовать как неоднозначность (по Шеннону) между переданным и принятым сообщениями. Это Неоднозначность определяется как условная энтропия сообщения Xk, обусловленная полученным сообщением Yk:

H(xi|yj)=-

Или в конечном виде:

H(X|yj)=-

В соответствии с этим можно определить, какому количеству информации соответствует один символ сообщения Xk, если на выходе получен 0. Другими словами, соотношение позволяет определить условную энтропию или количество информации на один символ сообщения, если на выходе или на стороне получателя получен 0.

H(X|Y=0)=–P(X=0|Y=0)\*logP(X=0|Y=0)–P(X=1|Y=0)\*logP(X=1|Y=0)=

=–q log q-p log p

То же, если на выходе получена 1:

H(X|Y=1)=–P(X=0|Y=1)\*logP(X=0|Y=1)–P(X=1|Y=1)\*logP(X=1|Y=1)=

=–q log q-p log p

**Условной энтропией источника** дискретного сообщения Х в ДСК называем величину:

H(X|Y)=P(Y=0)\*P(X|Y=0)+p(Y=1)\*H(X|Y=1)=

=–p log p–q log q

H(X|Y) означает среднее количество информации для входного символа относительного полученного сообщения Y или потерю информации на каждый символ переданного сообщения.

*q=1-p, так как p+q=1*

p — вероятность того, что символ будет передан с ошибкой

q — вероятность того, что произвольный символ будет передан правильно

q и p составляют полную группу событий

Еще есть такой тип ошибок как **стирание** (не будем вдаваться в подробности).

Мы получили общее соотношение для условной энтропии источника дискретного сообщения. Дискретной означает принимающей только определённые значения (если циферблат можно плавно крутить, то это не дискретные величины). В нашем случае: 0 и 1.

!!!! Эта **условная энтропия** означает количество информации, которая может быть потеряна если р>0 (вероятность ошибки больше 0), а нет ни одного канала, где это бы не выполнялось (нет ничего абсолютно надёжного).

Хорошим считается канал, где вероятность ошибки не превышает p<=10-5.

Если скорость передачи 100Мбит/с и мы считаем, что это хороший канал, то при вероятности ошибки 10-5, сколько ошибочных бит в секунду будет передано? 100Мбит — это 106, за одну секунду будет передано 107 бит. Чтобы получилась вероятность ошибки 10-5, на нужно число ошибочных бит поделить и мы должны получить величину (дома досчитать).

**Пример.** Пусть известно, что P(X=0)=P(X=1)=0,5 и p=0,01.

H(X|Y)=–p log p–q log q=–0,01\*log0,01–0,99\*log0,99=0,081 бит

Эффективная энтропия: *He=0,919 бит.*

## Эффективная энтропия

*He=H(X)-H(X|Y)*

Это понятие введено Шенноном. Это разница между энтропией алфавита источника и условной энтропией. Условная энтропия показывает потери информации, приходящуюся на 1 бит, а эффективная энтропия показывает, сколько реально информации передает каждый бит с учётом потерь.

**Пример.** Какое количество информации будет передано по каналу связи за 1 час при скорости передачи 1 Мбит/с, если вероятность ошибки равна 0,5?

1. Нам нужно посчитать, какое число бит (символов двоичных) будет передано за 1 час. Как мы оценивали количество информации в сообщении? Количество информации в сообщении есть энтропия умножить на k, если нет потерь. А если есть потери, то мы ставит эффективную энтропию, которая показывает, сколько на самом деле весит каждый двоичный символ. А Эффективная энтропия равняется энтропия обычная минус энтропия условная. Энтропия двоичного алфавита равна 1. H(X|Y)=–p log p–q log q (q и p равны ½). Мы получим 1.
2. Если вероятность ошибки равна 0,5, то условная энтропия равно одному биту, то эффективная энтропия равна 0. Если подставляем в формулу, то выходит 0\*1Мбит\*3600секунд=0 (количество информации, которое передано).

**Пример.** Какое количество информации будет передано по каналу связи за 1 час при скорости передачи 1 Мбит/с, если вероятность ошибки равна 1?

Если нам известно, что каждый символ будет передан с ошибкой, то тогда полученное сообщение нужно инвертировать.

Условная энтропия равна 0 и эффективная равна 1. И тогда вероятность будет равна 1 (получим столько же, сколько и передавали).

# Базовые понятия криптографии. Основы теории больших чисел. Проблема дискретного логарифма.

**Криптография** (с греч criptos «тайный» и grafo «писать») — наука (раздел математики) о методах обеспечения конфиденциальности или безопасности информации, связанной с тайной, интегральностью данных и идентификацией.

**Криптоанализ** — наука о методах получения исходного значения зашифрованной информации без наличия секретного и неизвестного аналитику ключа.

**Криптология** — наука, объединяющая криптографию и криптоанализ.

**Криптосистема** — понятие, относящееся к совокупности программно-технических средств, функционирующих на основе установленных криптографических алгоритмов и осуществляющих *зашифрование* и *расшифрование* данных.

Огюст Керкгоффс (1883) – «Военная криптография».

Описал 6 требований, которым должна удовлетворять защищенная система:

1. Шифр должен быть физически или не математически невскрываемым;
2. Система не должна требовать секретности, на случай, если она попадет в руки врага;
3. Ключ должен быть простым, храниться в памяти без записи на бумаге, а также легко изменяемым по желанию корресподентов;
4. Зашифрованный текст должен передаваться по телеграфу;
5. Аппарат для шифрования должен быть легко переносимым, работа с ним не должна требовать помощи нескольких лиц;
6. Аппарат для шифрования должен быть относительно прост в использовании, не требовать значительных умственных усилий или соблюдения большого количества правил.

**Зашифрование и расшифрование информации:**

С

М

М’

Расшифрование

Зашифрование

К1

К2

Канал (интруз)

Прямое преобразование называют шифрованием или зашифрованием (**encrypt**), обратное – расшифрование или дешифрование (**decrypt**).

**Исходное сообщение** называется открытым текстом (**M -** **message**).

**Зашифрованное сообщение** – шифртекст или шифрограмма (**С -** cipher).

Ключ — секретный параметр, управляющий ходом преобразования. Ключ определяет конкретный вариант преобразования.

**М** обозначает множество сообщений, состоящих из символов алфавита.

Элемент из М открытый текст (явный):

**М = m1, m2,……mn**

**С** означает множество сообщений, состоящих из символов того же или иного алфавита. Элемент из С (шифрограмма):

**С = с1, с2……сn.**

**К** означает множество ключей (элемент этого множества - ключ).

Каждый элемент e∈К определяет взаимно однозначное отображение (биекцию) М на С обозначается Ее (Ее: М→С)

**Ее**– функция зашифрования.

Каждый элемент d∈K, Dd определяет взаимно-однозначное отображение (биекцию) C на М и обозначается (Dd : С→М).

**Dd**— функция расшифрования.

Процесс вычисления функции Eе  для аргумента m∈М называется зашифрованием явного текста m.

**E** – сам процесс зашифрования, e – ключ для этого процесса (пример: процесс входа в дверь).

**D** – процесс расшифрования, d – ключ для расшифрования (пример: процесс выход на из помещения наружу).

**Классификация систем шифрования (шифров)**

1. На основе процедуры шифрования:

* Ш. подстановочные;
* Ш. перестановочные.

1. На основе генерирования и использования ключа:

* Ш. блочные;
* Ш. поточные.

1. На основе типа ключа:

* Ш. симметричные (с тайным ключом): e=d;
* Ш. ассиметричные (с открытым или публичным ключом): e≠d.

Подстановочные и перестановочные шифры:

Все современные шифры как правило являются подстановочно-перестановочными. Сущность **подстановочного шифрования** состоит в том, что, как правило, исходный (М) или зашифрованный текст (С) используют один и тот же алфавит, а ключом является алгоритм подстановки. Такой шифр называется простым, или моноалфавитным. Классический пример – шифр Цезаря.

Биалфавитный шифр — каждую букву заменят всегда определенное число.

**Перестановочные шифры** используют перестановку символов исходного сообщения в соответствии с установленным правилом, открытый текст остается неизменным, но символы в нем «перетасовываются» (подвергаются пермутации). Так, в простом вертикальном перестановочном шифре 70 открытый текст пишется по горизонтали на разграфленном листе бумаги фиксированной длины, а шифртекст считывается по вертикали.

Рассмотрим это на примере:

М = «ВАСЯ ЛЮБИТ МАШУ».

Считывание по столбцам снизу вверх приводит к такому шифр-

тексту: С = «\_ЛВМЮААБСШИЯУТ\_».

Наиболее уязвимое место подстановочно-перестановчных шифров заключается в статистических характеристиках символов алфавита, поскольку это сообщение тайно и для шифрованных и зашифрованных алгоритмов.

Блочные и поточные шифры:

В блочных шифрах шифрованию и расшифрованию одновременно подвергаются блоки данных, в потоковых шифрах как правило преобразуется одиночный или несколько таких одиночных символов.

Симметричные и ассиметричные шифры:

Симметричные шифры с одинаковым ключом K1 = K2, ассиметричные с публичным ключом, в случае K1≠K2

В симметричных шифрах каждый пользователь должен иметь ключ для переписки с одним пользователем или группой. Вы с кем-то договариваетесь о ключе и передаете его друг другу.

Недостаток симметричных систем заключается в хранении и обмене ключевой информации.

## Проблема дискретного логарифма

При известных **a, x** и **n** и легко вычисляется .

Обратная задача: при известных **y, a,** и **n** найти **х**.

Это задача вычисления дискретного логарифма:

(см. соотношение (2))

*Пример 5.* Если 3х=15 mod 17, то x=6.

Решение существует не для всех дискретных значений.

*Пример 6.* Уравнение 3х=7 mod 13 решения не имеет.

При 1024-битовых и более значениях решение задачи может занять десятки и сотни лет

## Большие числа

Вероятность погибнуть в автокатастрофе – 1 из 27

Время до превращения Солнца в сверхновую звезду – 230 лет

Возраст Вселенной – 234 лет

Число атомов нашей планеты – 2170 атомов

# Основная теорема арифметики. Алгоритм Евклида нахождения НОД.

**Основная теорема арифметики**

Любое составное число можно составить из некоторого количества простых с помощью умножения. Например, составное число 2009 можно получить так:

2009 = 7 \* 7 \* 41

В математике рассматривается так называемая основная теорема арифметики, которая утверждает, что любое натуральное число ( n>1 ) либо само является простым, либо может быть разложено на произведение простых делителей, причем единственным способом (если не обращать внимания на порядок следования сомножителей).

**Элементы Теории чисел**

Теория чисел – изучение свойств целых чисел

Множество целых чисел: Z = {…, -2, -1, 0, 1, 2, …}

Множество натуральных чисел N = {1, 2, 3, …}

Целое число a есть кратное числа b, если a = b\*m для некоторого числа m (числа не равны 0).

Обозначается b|a, например 9|27, т.к. 27 = 9 \* 3. Число b – делитель числа а.

Теорема 1 (алгоритм деления). Для положительного целого числа а и целого числа б существуют единственные целые положительные числа q и p, где 0 <= p < b такие что a=b\*q + р, a < b, p – остаток, q – частное. Если a < b, то q = 0.

Например: а = 4, b = 7, тогда q = 0, p = 4, т.к. 4 = 7 \* 0 + 4

а = 15, b = 7, тогда q = 2, p = 1, т.к. 15 = 7 \* 2 + 1

а = 23, b = 8, тогда q = 2, p = 7, т.к. 23 = 8 \* 2 + 7

Положительное целое число d называется общим делителем чисел а и b, если d|a и d|b.

Положительное целое число d называется наибольшим общим делителем чисел a и b, НОД(a, b), если d|a и d|b, и если из c|a и c|b следует c|d.

Теорема 2. Если а = b \* q + c, то НОД(a, b) = НОД(b, c).

Например: 16 = 6 \* 2 + 4, а = 16, b = 6, q = 2, c = 4.

НОД(a, b) = НОД (16, 6) = 2, НОД(b, c) = НОД(6, 4) = 2

Если НОД (a, b) = 1, то числа a, b называются взаимно простыми.

**Алгоритм Евклида нахождения НОД**

Даны два числа a и b, a > b, a > 0, b > 0. Находим ряд равенств:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Нас интересует последний остаток, который положительный и не равен нулю. b записывается как число, r1 записывается как число.

Ряд неравенств заканчивается, когда получаем некоторое rn+1 = 0. Тогда rn – наибольший общий делитель чисел a и b.

**Пример**. есть числа 525 и 231. Надо найти НОД этих чисел.

525 = 231 \* 2 + 63

231 = 63 \* 3 + 42

63 = 42 \* 1 + 21

42 = 21 \* 2 + 0

Получаем последний положительный остаток r3 = 21. Таким образом, *НОД(525,231) = 21.*

эквивалентно

**Пример**. Есть числа 1234 и 54. Найдем НОД.

1234 = 54 \* 22 + 46

54 = 46 \* 1 + 8

46 = 8 \* 5 + 6

8 = 6 \* 1 + 2

6 = 2 \* 3 + 0

Последний ненулевой остаток равен 2, *НОД(1234, 54) = 2.*

**Пример***.* Найдем НОД (109, 63).

*109 = 63 \* 1 + 46*

*63 = 46 \* 1 + 17*

*46 = 17 \* 2 + 12*

*17 = 12 \* 1 + 5*

*12 = 5 \* 2 + 2*

*5 = 2 \* 2 + 1*

*2 = 1\*2 + 0*

Последний ненулевой остаток = 1, значит числа взаимно простые.

# Основы модулярной арифметики. Вычеты. Китайская теорема об остатках.

## Модулярная арифметика (теория вычетов)

Понятие «модулярная арифметика» ввел немецкий ученый Гаусс. В этой арифметике мы интересуемся остатков от деления числа a на число n. Если таким остатком является число b, то можно записать:

Числа 23 и 11 равны по модулю 12: 23 = 11 mod 12

**Определение 1**. В операции b называют вычетом по модулю n.

**Определение 2**. Множество целых чисел от 0 до 1 образует *полную систему вычетов* по модулю n.

## Правила модулярной арифметики

(a + b) mod n = ((a mod n) + (b mod n)) mod n

(a – b) mod n = ((a mod n) – (b mod n)) mod n

(a \* b) mod n = ((a mod n) \* (b mod n)) mod n

(a \* (b + c)) mod n = (((a \* b) mod n) + ((a \* c) mod n)) mod n

a + 0 = a, a + = 1, a + a = 0

Проиллюстрируем то, что выше, на примерах:

(27 + 15) mod 12 = (27 mod 12 + 15 mod 12) mod 12 = 6.

(27 – 15) mod 12 = (27 mod 12 – 15 mod 12) mod 12 = 0.

(27 \* 15) mod 12 = ((27 mod 12) \* (15 mod 12)) mod 12 = 9.

(27 \* (15 + 3)) mod 12 = (((27 \* 15) mod 12) + ((27 \* 3) mod 12)) mod 12 = 6

**Пример**. Вычислить

Простое решение: .

**Пример**. . X не является степенью 2.

X =25.

В модулярной арифметике существует простое правило, упрощающее операции над большими числами, а именно аддитивная цепочка.

**Пример** (**МОЖЕТ И ПРАВИЛЬНО**). Представим двоичное число иксами. 1011 = 1⋅23 + 0⋅22 + 1⋅21 + 1⋅20 = x3 + x + 1

**Пример** (**МОЖЕТ И ПРАВИЛЬНО**):

x5 + x + 1 = 1⋅25 + 0⋅24 + 0⋅23 + 0⋅22 + 1⋅21 + 1⋅20 = 100011.

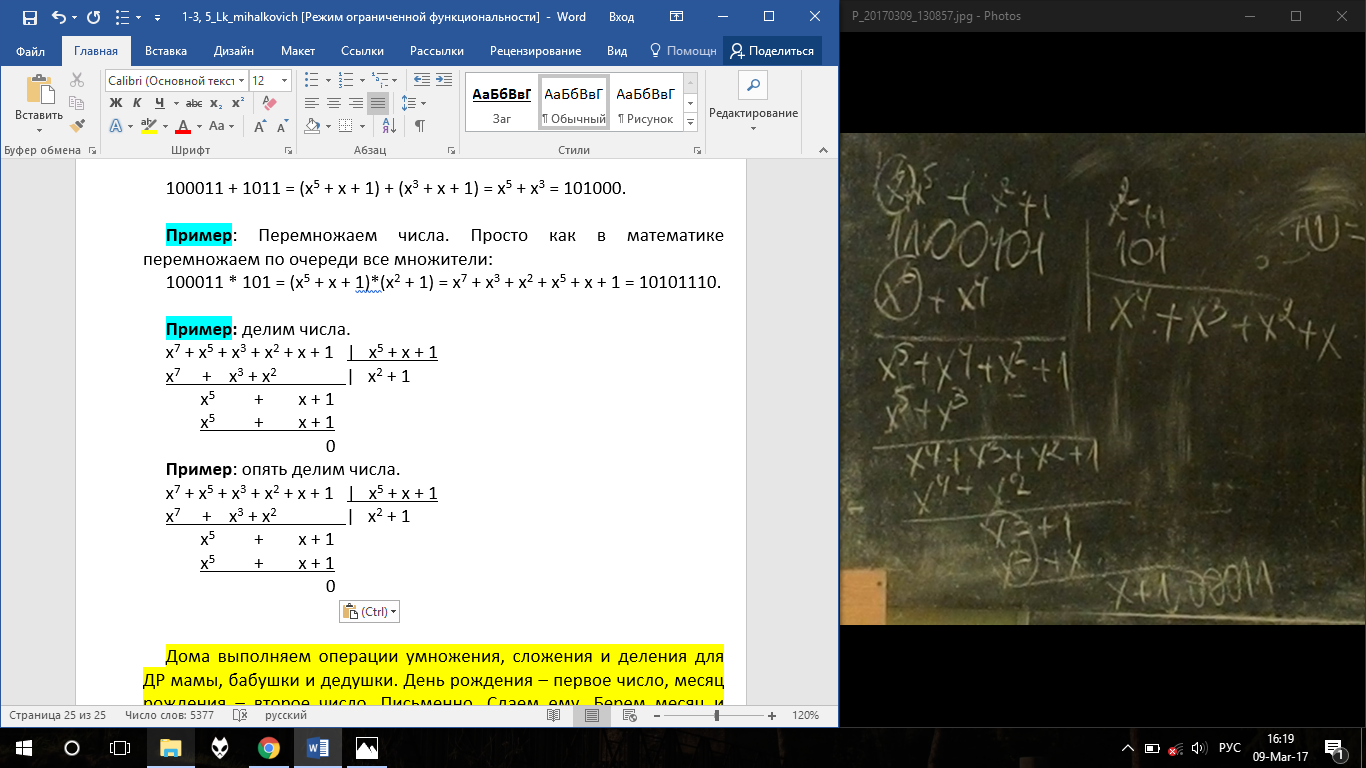
**Пример (НЕПРАВИЛЬНЫЙ)**: Складываем числа. Если икс какой-то степени только один раз повторяется, то записываем его (0 + 1 = 1). Если икса какой-то степени нет вообще, то не записываем его (0 + 0 = 0). Если есть два икса какой-то степени, то тоже не записываем такой икс (1 + 1 = 0).

100011 + 1011 = (x5 + x + 1) + (x3 + x + 1) = x5 + x3 = 101000.

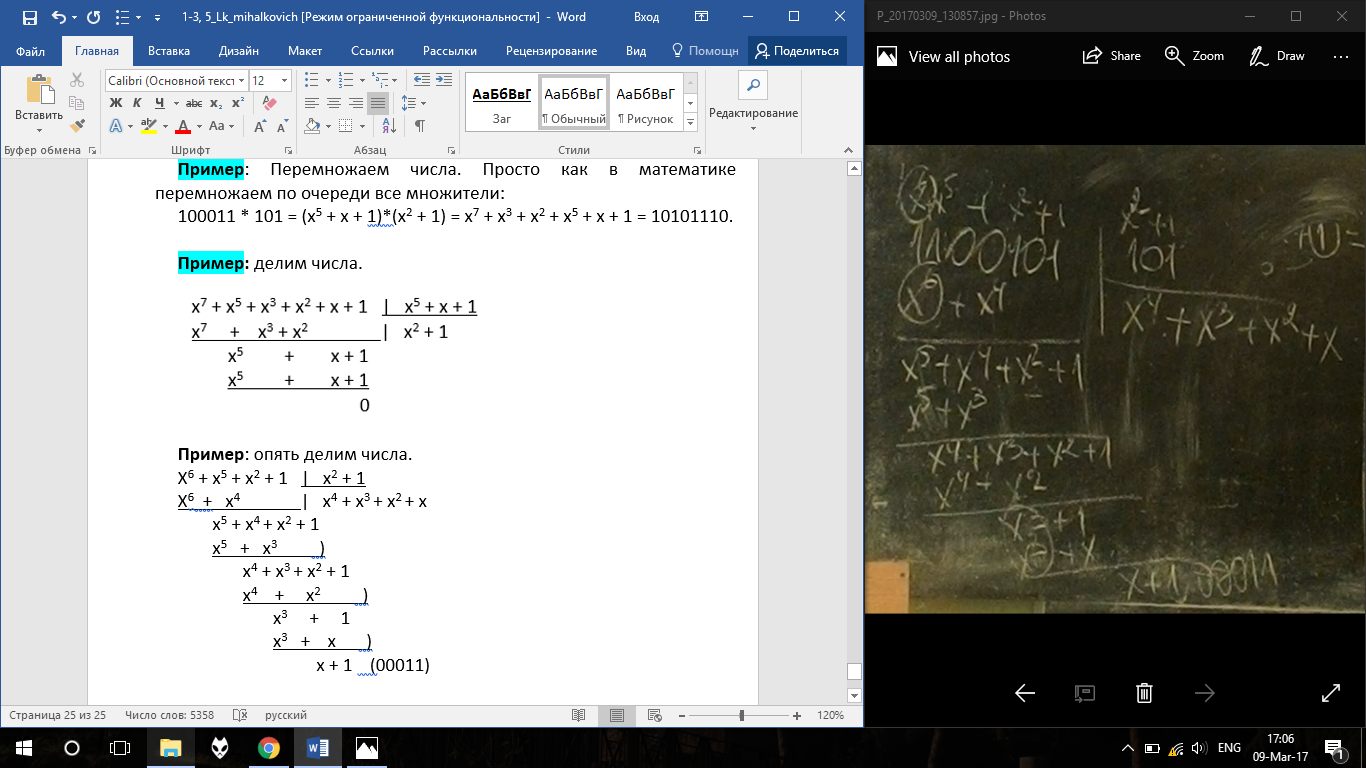
**Пример (НЕПРАВИЛЬНЫЙ)**: Перемножаем числа. Просто как в математике перемножаем по очереди все множители:

100011 \* 101 = (x5 + x + 1)\*(x2 + 1) = x7 + x3 + x2 + x5 + x + 1 = 10101111.

**Пример (ПРАВИЛЬНЫЙ):** делим числа.



**Пример (ПРАВИЛЬНЫЙ)**: опять делим числа.



## Функция Эйлера

Леонард Эйлер (1707-1783)-швейцарский математик.

*Определение 4.* **Приведенной системой вычетов** по модулю **n** называют подмножество полной системы вычетов, члены которого взаимно просто с n.

*Пример 3.* Приведённой системой вычетов по модулю **12** будет подмножество (1, 5,7,11).

Если n – простое число, то ПСВ по модулю **n** входят числа от 1 до n-1.

Функция Эйлера (обозначается ) определяет число элементов в ПСВ по модулю n, то есть количество целых положительных чисел, меньших **n** и взаимно простых с n; n>1.

Если n – простое число, то

Если n=p\*q, где числа p и q – простые числа, то

(3)

Малая терема Ферма. Если n – просто число и не кратно n, то справедливо:

или (4)

Обобщение Эйлера МТФ: Если НОД (а,n)=1, то

(5)

Нетрудно вычислить а-1 (разделив обе части (5) на а)

*Пример 4.* Найти число, обратное 5 по модулю 7(n).

Ответ:

# Обратные вычисления по модулю в криптографии. Расширенный алгоритм Евклида.

## Обратное значение по модулю

*Традиционно: обратное от 4 равно ¼, так как 4\*1/4=1.*

В модулярной арифметике

эквивалентно поиску таких значений x и k, что

Общая задача: найти такое х, что

(1)

(2)

Уравнение (2) имеет **единственное** решение если а и n – взаимно простые числа, в противном случае (2) решений не имеет.

*Пример 2:* a=5 и n=14, x=3: , так как .

**Алгоритм Евклида нахождения НОД**

Даны два числа a и b, a > b, a > 0, b > 0. Находим ряд равенств:

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Нас интересует последний остаток, который положительный и не равен нулю. b записывается как число, r1 записывается как число.

Ряд неравенств заканчивается, когда получаем некоторое rn+1 = 0. Тогда rn – наибольший общий делитель чисел a и b.

**Пример**. есть числа 525 и 231. Надо найти НОД этих чисел.

525 = 231 \* 2 + 63

231 = 63 \* 3 + 42

63 = 42 \* 1 + 21

42 = 21 \* 2 + 0

Получаем последний положительный остаток r3 = 21. Таким образом, *НОД(525,231) = 21.*

эквивалентно

**Пример**. Есть числа 1234 и 54. Найдем НОД.

1234 = 54 \* 22 + 46

54 = 46 \* 1 + 8

46 = 8 \* 5 + 6

8 = 6 \* 1 + 2

6 = 2 \* 3 + 0

Последний ненулевой остаток равен 2, *НОД(1234, 54) = 2.*

**Пример***.* Найдем НОД (109, 63).

*109 = 63 \* 1 + 46*

*63 = 46 \* 1 + 17*

*46 = 17 \* 2 + 12*

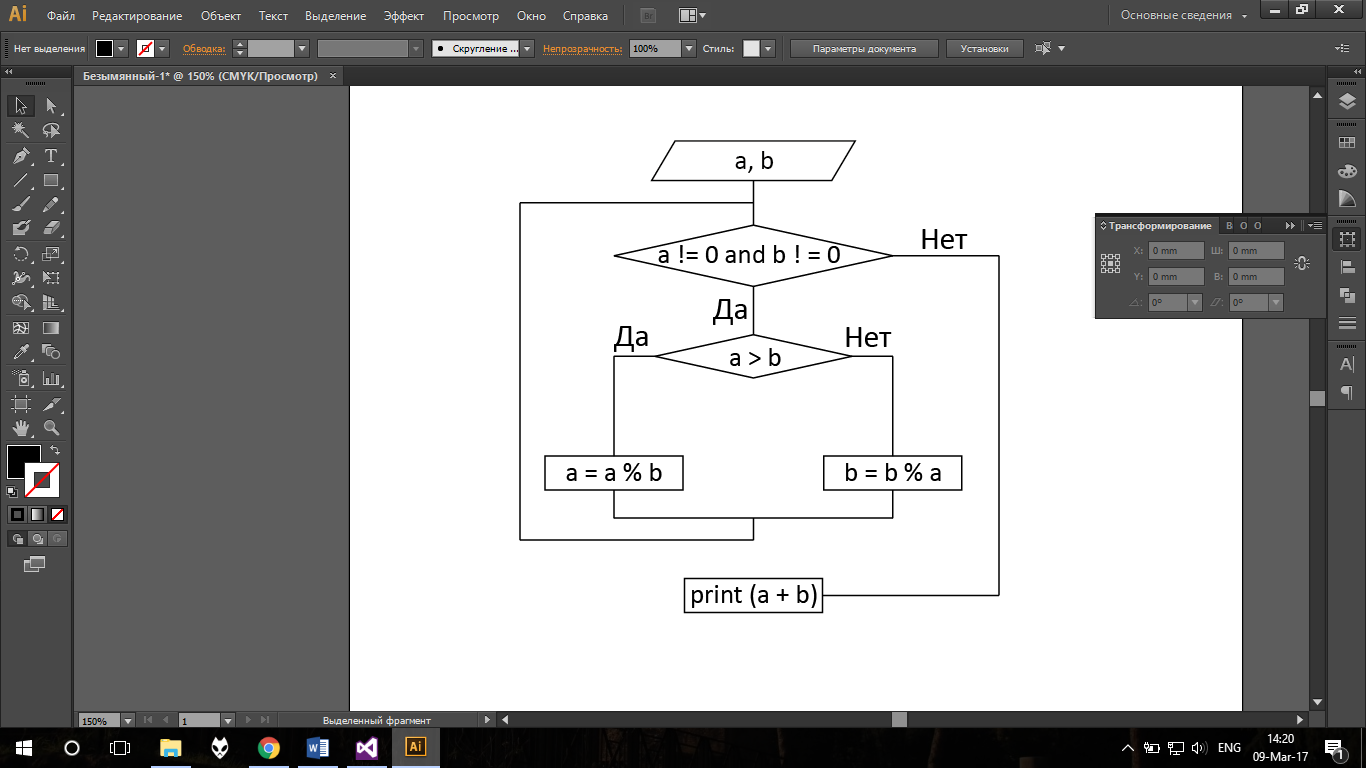
*17 = 12 \* 1 + 5*

*12 = 5 \* 2 + 2*

*5 = 2 \* 2 + 1*

*2 = 1\*2 + 0*

Последний ненулевой остаток = 1, значит числа взаимно простые.



**Пример**. а = 27, b = 15

а = a mod b = 27 mod 15 = 12

a = 12, b = 15

b = b mod a = 15 mod 12 = 3

a = 12, b = 3

a = a mod b = 12 mod 3 = 0

a = 0, b = 3

print NOD = a + b = 0 + 3 = 0

# Функция Эйлера в криптографии.

## Функция Эйлера

Леонард Эйлер (1707-1783)-швейцарский математик.

*Определение 4.* **Приведенной системой вычетов** по модулю **n** называют подмножество полной системы вычетов, члены которого взаимно просто с n.

*Пример 3.* Приведённой системой вычетов по модулю **12** будет подмножество (1, 5,7,11).

Если n – простое число, то ПСВ по модулю **n** входят числа от 1 до n-1.

Функция Эйлера (обозначается ) определяет число элементов в ПСВ по модулю n, то есть количество целых положительных чисел, меньших **n** и взаимно простых с n; n>1.

Если n – простое число, то

Если n=p\*q, где числа p и q – простые числа, то

(3)

Малая терема Ферма. Если n – просто число и не кратно n, то справедливо:

или (4)

Обобщение Эйлера МТФ: Если НОД (а,n)=1, то

(5)

Нетрудно вычислить а-1 (разделив обе части (5) на а)

*Пример 4.* Найти число, обратное 5 по модулю 7(n).

Ответ:

## Проблема дискретного логарифма

При известных **a, x** и **n** и легко вычисляется .

Обратная задача: при известных **y, a,** и **n** найти **х**.

Это задача вычисления дискретного логарифма:

(см. соотношение (2))

*Пример 5.* Если 3х=15 mod 17, то x=6.

Решение существует не для всех дискретных значений.

*Пример 6.* Уравнение 3х=7 mod 13 решения не имеет.

При 1024-битовых и более значениях решение задачи может занять десятки и сотни лет

# Хеш-функция и ее свойства. Области использования хеш-функций.

## Понятие хэш-функции

**Однонаправленная функция** предполагает простоту её вычисления (вычисления f(x) по известному аргументу x) и сложность обратного вычисления (вычисления х по известному f(x)).

**Хэш-функция** – математическая или иная функция, которая принимает на входе строку символов переменной (произвольной) длины и преобразует её в выходную строку фиксированной (обычно-меньшей) длины, называемой *значением* ***хэш-функции*** или её ***свертком***.

**Однонаправленная хэш-функция** – основа многих протоколов.

## Свойства хэш-функции

* Формальная запись**: h=H(M)**
* Зная **М**, легко вычислить **h**
* Зная **h**, трудно определить **М**, для которого **Н(М)=h**
* Зная **М**, трудно определить **М’ (М≠М’)**, для которого **Н(М)=Н(М’)** – коллизия 1-го рода
* Трудно найти два случайных сообщения (**М** и **М’**), для которых **Н(М)=Н(М’)** – коллизия 2-го рода.

# Общая характеристика алгоритмов хеширования классов MD и SHA.

Цифровая подпись предназначена для того же, что и обычная.

Цифровая подпись является элементом криптографического преобразования информации.

Понятие «ЭЦП» была введена в 1976 году У. Диффи и М. Хеллманом.

**Функции ЭЦП:**

* Подпись позволяет аутентифицировать отправителя. ЭЦП получается в результате криптографического преобразования электронных данных документа с использованием личного ключа ЭЦП;
* Позволяет контролировать целостность сообщения: ЭЦП вычислена на основе исходного состояния документа и соответствует лишь ему, поэтому при любом случайном или преднамеренном изменении документа подпись станет недействительной. Если вы поставили подпись цифровую, то уже ничего изменить нельзя, гарантирует целостность;
* Защищает сообщение от подделок: любая подделка должна быть выявлена путем операции сравнения советующих атрибутов подписанного и полученного адресатом сообщений;
* Доказывает авторство лица, подписавшего сообщение: создать корректную ЭЦП можно лишь зная закрытый ключ, известный только его владельцу (лицу, подписавшему документ). Может доказываться в пользу того, кто поставил подпись.

**Важнейшими отличительными особенностями ЭЦП являются:**

* ЭЦП представляет собой бинарную последовательность (в отличие от графического образа, каковым является подпись от руки);
* Указанная бинарная последовательность зависит от содержания подписываемого сообщения.

**Электронная цифровая подпись** — последовательность символов, являющаяся реквизитом электронного документа, зависящая от содержания этого документа и предназначенная для подтверждения целостности и подлинности электронного документа.

## Основные типы ЭЦП

1. На основе симметричных криптосистем; авторизацией документа является сам факт зашифрования его секретным ключом
2. На основе симметричных криптосистем и посредника
3. На основе ассиметричных криптосистем; ЭЦП на основе RSA
4. На основе ассиметричных криптосистем и однонаправленных хэш-функций.

В первых 3 случаях ЭЦП как самостоятельный атрибут электронного документа отсутствует. Все функции, при отсутствии ЭЦП реализуются через ключевую информацию (принадлежность ключевой информации).

## ЭЦП на основе симметричной криптографии

Генерация ЭЦП: **E=F(M,K) → C=F(M,K)** (осуществляется простое шифрование), т.е. **E ≡C;** **физически ЭЦП нет**.

Проверка подлинности сообщения (верификация подписи):

**M=F (C,K)** **→** **ЭЦП = F (C,K)** (осущ. обычное расшифрование).

Если сообщение расшифровано тайным ключом **(К)**, известным только **А** и **В**, то это подтверждает аутентичность документа.

## ЭЦП на основе симметричных криптосистем и посредника

1. **П** (Посредник, СЦ – сертификационный центр) вырабатывает разные ключи для двух сторон **А** и **В**: **Ка**и **Кв**.



1. Сторона **А** шифрует **М** ключом **Ка**и отсылает его **П**.



1. **П** расшифровывает **С** ключом **Ка**, включает в него информацию (свое сообщение **М** посредника), что оно получено от **А (М’)**.



1. **П** шифрует **М’**ключом **Кв**и отправляет его **В**.



1. **В** расшифровывает ключом **Кв**и читает **М’**с гарантией **П**:



**Таким образом:**

* Подпись достоверна (П-гарант);
* Подпись неподдельна (только А и В знают ключ: П— абсолютное доверие);
* Подпись невозможно использовать повторно;
* Подписанный документ нельзя изменить;
* Подпись нельзя отрицать.

## ЭЦП с открытым ключом

В классическом алгоритме RSA используются ключи стороны B или получателя Публичный абонент А использует для зашифрования тайный ключ использует сторона В для расшифровывания.

1. На основе алгоритма **RSA:**

* **А** шифрует документ **своим закрытым ключом:**

**С=F(M; da, na)**

* **B** расшифровывает **С открытым ключом А:**

**М = F(C; ea, na)**

1. На основе алгоритма **Эль-Гамаля:**

* Генерация ключа **p** – простое число; выбираем два случайных числа: **g** и **x (g, x<p)**; вычисляем: **y = gx mod p.**

Открытый ключ — **p, y, g**

Закрытый ключ — **x**.

В данном случае, ключевую информацию генерирует **отправитель (сторона А)**.

* **A** шифрует документ **М своим закрытым ключом**:

Выбирается случайное число **k** — взаимно простое с **(p-1)** и генерируется подпись, состоящая из двух чисел: **a** и **b:**

**a = gk mod p такое, что M = (x\*a+k\*b) mod (p-1)**

* **A** отправляет **В** подписанное сообщение**: M, a, b.**
* **B** получает **M, a, b** и сверяет подпись: подпись принадлежит **А,** если **(ba\*ab) mod p=gM mod p.**

**Пример: p =11, g = 2, x = 8,** вычисляем **y=gx mod p = 28 mod 11 = 3;**

Открытый ключ **p=11, g = 2, y =3.** Закрытый ключ **x=8.**

**Подпись сообщения: M=5:** выбираем **k=9** взаимнопростое с **p-1=10;** вычисляем **a=gk mod p = 29 mod 11 = 6;**

**На основе M = (x\*a+k\*b) mod (p-1)** вычисляем **b: 5=(8\*6+9\*b) mod 10,** решение: **b=3.**

**Подписанное сообщение: M=5, a=6, b=3**

**Проверка подписи: (36\*63) mod 11 = 25 mod 11 = 10.**

Пример:

*Таня взяла и зашифровала ключом сообщение и подписалась как «Урбанович», она может это сделать. Любой может прочесть сообщение, если не скрывать его, т.к. оно зашифровано открытым ключом. Но Иван уверен, что прислал Урбанович. Почему? Потому что Урбанович зашифровал своим тайным ключом, а Иван расшифровал публичным ключом Урбановича сообщение и получил, семантический логический правильный текст. Это означает, что этот документ мог быть зашифрован только тайным ключом Урбановича.*

В случае шифрования сообщения на основе ассиметричной системы или генерации ЭЦП используется исключительно информация **отправителя**: **тайный ключ при шифровании (сторона А) и публичный ключ при расшифровании (сторона В).**

Алгоритм RSA и алгоритм Эль-Гамаля обычно используются для зашифрования не целого сообщения М, а хэш-сообщения h(M).

## ЭЦП на основе ассиметричных криптосистем и однонаправленных хэш-функций

Типы хэш-функций – **h(M).**

MD4 (message digest 4) разработан Роном Ривестом.

Длина h(M) -128 бит, длина М –k (произвольная).

Начальное преобразование М: дополняется символами так, чтобы (k + r+ 64) mod 512 = 0.

64 бита — двоичное представление числа k; 100 … 00 получаем m блоков преобразованного M длины 512 бит. Каждый из этих блоков разбивается на 16 подблоков длины 32 бита (16\*32 = 512).

Весь алгоритм состоит из 3-х раундов.

По свертке можно восстановить входное сообщение, односторонности нет. В системах класса MD длина хэша составляет 128 бит.

Secure Hash Algorithm — алгоритм безопасного хэширования. Модификаций его очень много, здесь базовый – SHA1. Длина хэша – 160 бит. Есть SHA256, SHA512.

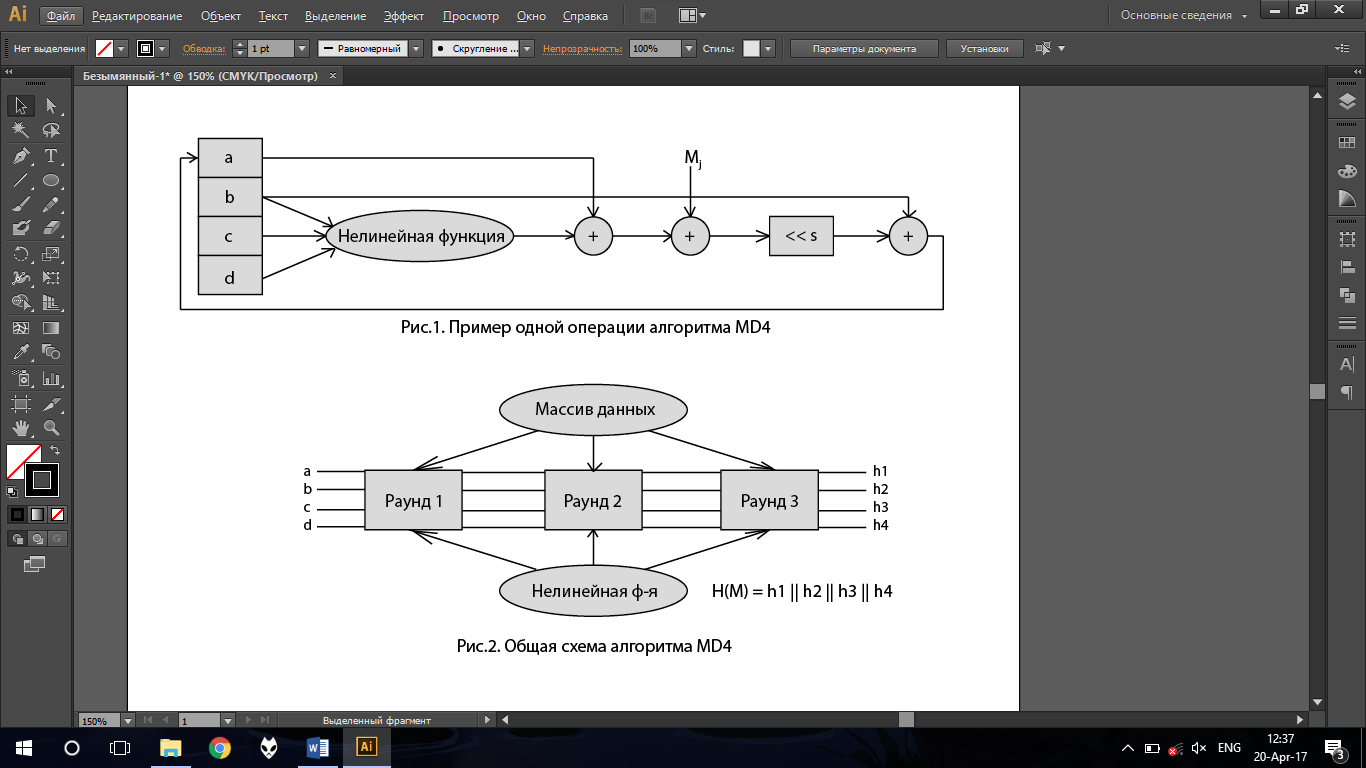
На входе произвольной длины строка, а на выходе всегда 128 бит или 160 бит, в зависимости от алгоритма.

# Алгоритмы хеширования класса MD. Области использования.

## Алгоритм хэширования MD4

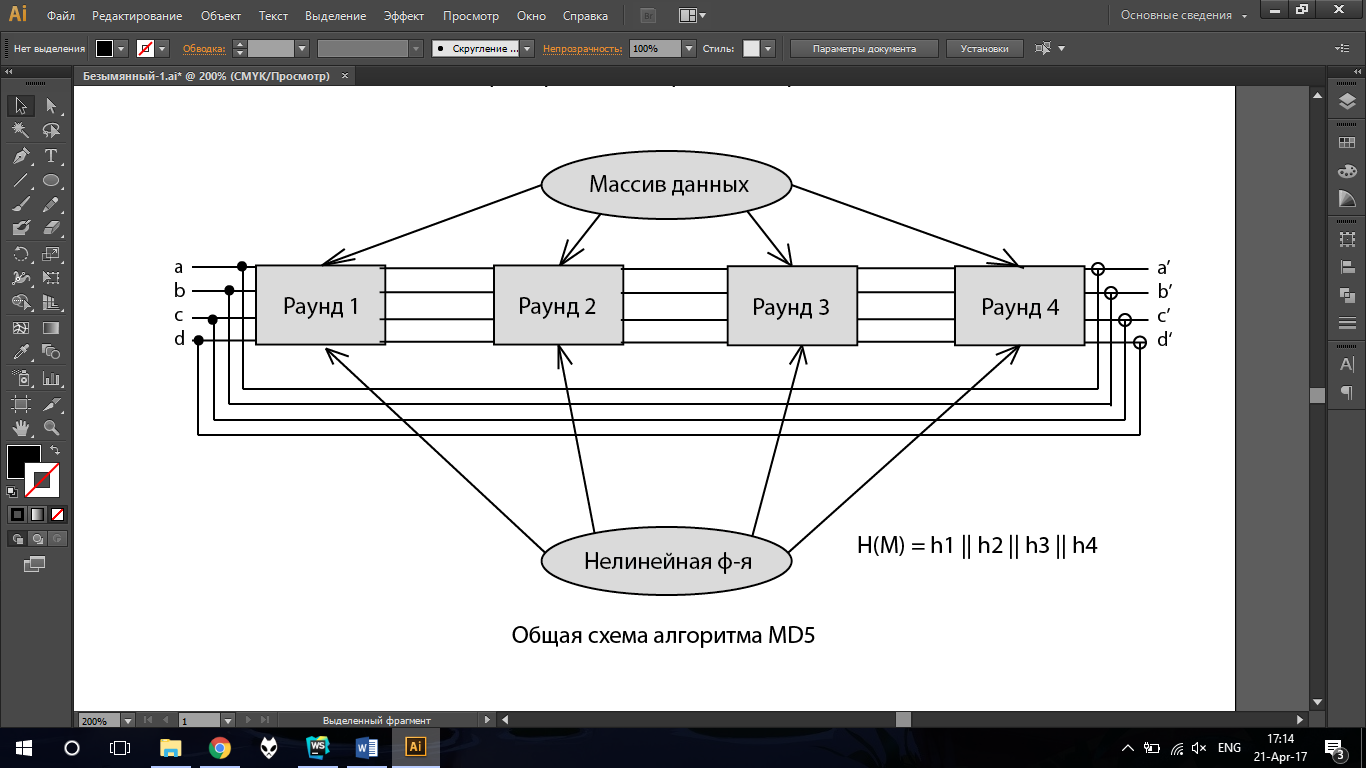
1. Предварительное преобразование сообщения М предусматривает его увеличение таким образом (только увеличение), чтобы длина преобразованного сообщения была кратна определенному числу. Длина кратна числу 512. Сообщение представлено в бинарном виде и после преобразования сообщения его длина должна быть кратна определенному числу, как правило, длина кратна числу 512. Сообщение представляется в бинарном виде, а после преобразование длина должна быть кратна 512.
2. Разделение сообщения на блоки. Сообщение хэшируется поблочно. Сообщение делится на большие блоки по 512 бит, в некоторых алгоритмах, например, по 32 бита.
3. Инициализация переменных. Начальные их значения (это 32-разрядные числа) всегда известны.
4. Процедура хэширования, которая заканчивается выводом хэша (шестнадцатеричными цифрами).

Каждый 32-битный массив данных обрабатывается в течение трех раундов. Эту схему надо уметь рисовать на экзамене.



## Алгоритм хэширования MD5

Хэш-функция MD5 отличается от MD4 тем, что в ней четыре раунда.



Применение MD5

* *Поиск дублирующихся файлов* на компьютере или в интернете (сравнивая MD5 файлы)

Пример: графическая программа dupliFinder под Windows и Linux; это утилита для поиска дубликатов фотографий в папках на диске компьютера путем сравнения их контрольных сумм MD5.

* *Проверка целостности скачанных файлов* – некоторые прораамы идут вместе со значением хеша.

Пример: диски для инсталляции

* *Хеширование паролей*

Пример: (“md5”) = 1bc29b36f623ba82aaf6724fd3b16718

(“”) = d41d8cd98f00b204e9800998ecf8427e (ненулевой хеш получился для нулевой строки)

# Алгоритмы хеширования класса SHA. Области использования.

## Алгоритм хеширования SHA1

Secure Hash Algorithm 1

Длина входного сообщения – максимум 264-1 бит (2 эксабайта – 2\*1018 бит). Алгоритм генерирует 160-битное хеш-значение.

Сходства SHA1 и MD5

* Четыре этапа
* Каждое действие прибавляется к рнее поулченному результату
* Размер блока обработки равный 512 бит
* Оба алгоритма выполняют сложение по модулю 232. Рассчитаны на 32-битную архитектуру

Основное отличие:

Увеличение длины хеша со 128 бит до 160 бит в SHA. Соответственно, число констант увеличивается на одну: a b c d f.

Другие отличия:

* В MD5 длина дайджеста составляет 128 б, в SHA1 – 160 б.
* В MD5 четыре различных элементарных логических функции, в SHA1 – Три.
* SHA1 содержит больше раундов (80 вместо 64) и выполняется на 160-битном буфере по сравнению со 128-битным буфером MD5. SHA1 приблизительно на 25% медленнее, чем MD5.
* SHA1 использует циклический под исправления ошибок

Применение SHA1

* ЭЦП
* Системы управления версиями – ПО для облегчения работы с изменяющейся информацией. СУВ позволяет хранить несколько версий одного и того же документа (коды программ), возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение. Пример – в Git для идентификации каждого объекта по SHA1-хешу от хранимой в объекте информации.
* Для построения кодов аутентификации (процедура проверки подлинности: путем сравнения введенного пароля с паролем в БД пользователей).

Для обнаружения коллизий нужно выполнить 252 операций.

Криптоанализ хеш-функций SHA1

Направлен на исследование уязвимости к различного вида атакам.

Основные атаки:

* *Нахождение коллизий* – двум различным исходным сообщениям соответствует одно и то же хеш-значение. При решении методом «грубой силы» требует в среднем 2160/2 = 280 операций.
* *Нахождение прообраза* – исходного сообщения – по его хешу. При решении методов «грубой силы» требует 2160 операций.

# Общая классификация криптографических методов защиты информации.

**Криптография** (с греч criptos «тайный» и grafo «писать») — наука (раздел математики) о методах обеспечения конфиденциальности или безопасности информации, связанной с тайной, интегральностью данных и идентификацией.

**Криптоанализ** — наука о методах получения исходного значения зашифрованной информации без наличия секретного и неизвестного аналитику ключа.

**Криптология** — наука, объединяющая криптографию и криптоанализ.

**Криптосистема** — понятие, относящееся к совокупности программно-технических средств, функционирующих на основе установленных криптографических алгоритмов и осуществляющих *зашифрование* и *расшифрование* данных.

Огюст Керкгоффс (1883) – «Военная криптография».

Описал 6 требований, которым должна удовлетворять защищенная система:

1. Шифр должен быть физически или не математически невскрываемым;
2. Система не должна требовать секретности, на случай, если она попадет в руки врага;
3. Ключ должен быть простым, храниться в памяти без записи на бумаге, а также легко изменяемым по желанию корресподентов;
4. Зашифрованный текст должен передаваться по телеграфу;
5. Аппарат для шифрования должен быть легко переносимым, работа с ним не должна требовать помощи нескольких лиц;
6. Аппарат для шифрования должен быть относительно прост в использовании, не требовать значительных умственных усилий или соблюдения большого количества правил.

**Зашифрование и расшифрование информации:**

С

М

М’

Расшифрование

Зашифрование

К1

К2

Канал (интруз)

Прямое преобразование называют шифрованием или зашифрованием (**encrypt**), обратное – расшифрование или дешифрование (**decrypt**).

**Исходное сообщение** называется открытым текстом (**M -** **message**).

**Зашифрованное сообщение** – шифртекст или шифрограмма (**С -** cipher).

Ключ — секретный параметр, управляющий ходом преобразования. Ключ определяет конкретный вариант преобразования.

**М** обозначает множество сообщений, состоящих из символов алфавита.

Элемент из М открытый текст (явный):

**М = m1, m2,……mn**

**С** означает множество сообщений, состоящих из символов того же или иного алфавита. Элемент из С (шифрограмма):

**С = с1, с2……сn.**

**К** означает множество ключей (элемент этого множества - ключ).

Каждый элемент e∈К определяет взаимно однозначное отображение (биекцию) М на С обозначается Ее (Ее: М→С)

**Ее**– функция зашифрования.

Каждый элемент d∈K, Dd определяет взаимно-однозначное отображение (биекцию) C на М и обозначается (Dd : С→М).

**Dd**— функция расшифрования.

Процесс вычисления функции Eе  для аргумента m∈М называется зашифрованием явного текста m.

**E** – сам процесс зашифрования, e – ключ для этого процесса (пример: процесс входа в дверь).

**D** – процесс расшифрования, d – ключ для расшифрования (пример: процесс выход на из помещения наружу).

**Классификация систем шифрования (шифров)**

1. На основе процедуры шифрования:

* Ш. подстановочные;
* Ш. перестановочные.

1. На основе генерирования и использования ключа:

* Ш. блочные;
* Ш. поточные.

1. На основе типа ключа:

* Ш. симметричные (с тайным ключом): e=d;
* Ш. ассиметричные (с открытым или публичным ключом): e≠d.

Подстановочные и перестановочные шифры:

Все современные шифры как правило являются подстановочно-перестановочными. Сущность **подстановочного шифрования** состоит в том, что, как правило, исходный (М) или зашифрованный текст (С) используют один и тот же алфавит, а ключом является алгоритм подстановки. Такой шифр называется простым, или моноалфавитным. Классический пример – шифр Цезаря.

Биалфавитный шифр — каждую букву заменят всегда определенное число.

**Перестановочные шифры** используют перестановку символов исходного сообщения в соответствии с установленным правилом, открытый текст остается неизменным, но символы в нем «перетасовываются» (подвергаются пермутации). Так, в простом вертикальном перестановочном шифре 70 открытый текст пишется по горизонтали на разграфленном листе бумаги фиксированной длины, а шифртекст считывается по вертикали.

Рассмотрим это на примере:

М = «ВАСЯ ЛЮБИТ МАШУ».

Считывание по столбцам снизу вверх приводит к такому шифр-

тексту: С = «\_ЛВМЮААБСШИЯУТ\_».

Наиболее уязвимое место подстановочно-перестановчных шифров заключается в статистических характеристиках символов алфавита, поскольку это сообщение тайно и для шифрованных и зашифрованных алгоритмов.

Блочные и поточные шифры:

В блочных шифрах шифрованию и расшифрованию одновременно подвергаются блоки данных, в потоковых шифрах как правило преобразуется одиночный или несколько таких одиночных символов.

Симметричные и ассиметричные шифры:

Симметричные шифры с одинаковым ключом K1 = K2, ассиметричные с публичным ключом, в случае K1≠K2

В симметричных шифрах каждый пользователь должен иметь ключ для переписки с одним пользователем или группой. Вы с кем-то договариваетесь о ключе и передаете его друг другу.

Недостаток симметричных систем заключается в хранении и обмене ключевой информации.

# Подстановочные шифры. Шифр Цезаря. Его криптостойкость.

Подстановочные и перестановочные шифры:

Все современные шифры как правило являются подстановочно-перестановочными. Сущность **подстановочного шифрования** состоит в том, что, как правило, исходный (М) или зашифрованный текст (С) используют один и тот же алфавит, а ключом является алгоритм подстановки. Такой шифр называется простым, или моноалфавитным. Классический пример – шифр Цезаря.

Биалфавитный шифр — каждую букву заменят всегда определенное число.

## Шифр Цезаря

Данный шифр является подстановочным. Шифр Цезаря основан на подстановке одних символов на другие.

Этот алгоритм циклически сдвигается на 3 символа влево.

Если есть сообщение М = СBA, то зашифрованное будет P = FED. Ключ в данном алгоритме будет K=3.

При мощности алфавита в n-символов можно создать n-1 шифров, подобных на шифр Цезаря. (Всего в английском алфавите 26 символов, можно создать 25 шифров).

Шифр на основе аффинной системы подстановки Цезаря отличается от простого шифра Цезаря тем, что используется дополнительная цифра множителя.

В аффинной системе a, b должны быть связаны ограничением НОД(а,b) = 1. Эти 2 числа делятся вместе на 1. НОД (10,8) = 2. Но в задании это не страшно, можно использовать.

## Шифр Цезаря с ключевым словом.

Ключ состоит из 2 элементов:

* Само слово
* Индекс алфавита или номер символа исходного алфавита начиная от которого вписывается это ключевое слово.

Особенностью этой системы является использование ключевого слова для смещения и изменения порядка символов в алфавите подстановки (желательно, чтобы все буквы ключевого слова были различными) и некоторого числа а: 0<= а < N. Рассмотрим систему на примере.

Выберем некоторое число a и слово или короткую фразу в качестве *ключевого слова.* Пусть выбраны слово DIPLOMAT в качестве ключевого и число а=5.

Ключевое слово записывается под буквами алфавита, начиная с буквы, индекс которой совпадает с выбранным числом а, как это показано ниже:

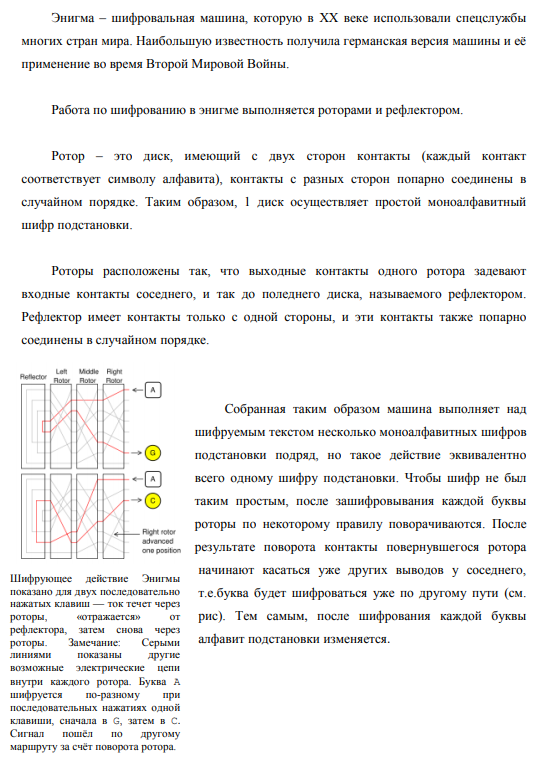
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |  |  |  |  | 10 |  |  | 15 |  |  |  |  | 20 |  |  |  |  |  |  |  |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|  |  |  |  |  | D | I | P | L | O | M | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |

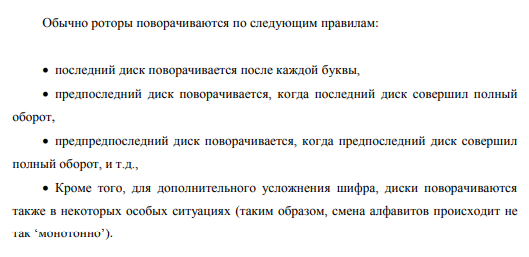
Оставшиеся буквы алфавита подстановки записываются после ключевого слова в алфавитном порядке

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| V | W | X | Y | Z | D | I | P | L | O | M | A | T | B | C | E | F | G | H | J | K | N | Q | R | S | U |

Если открытым сообщением будет BELSTU, то зашифрованным — WZAHJK.

# Особенности реализации шифровальной машины Энигма.





# Шифр на основе аффинной системе подстановок Цезаря. Его криптостойкость.

**Аффинная система подстановок Цезаря.** В данном методе используется ключ шифрования в виде пары целых чисел (*A, K*). Число *A* задает переход при шифровании вперед на *A?J* букв, а число *K* – дополнительное смещение по алфавиту на *K* букв. Следовательно, аффинную систему подстановок Цезаря можно описать следующей формулой:

*I = (А?J+K) mod M. (3)*

Формула (3) может быть использована только при выполнении следующих условий: *0 ? (A, J)? (M-1),  0 ? K ? (M-1),*НОД*(A, M)=1.*

Наибольший общий делитель чисел *A*и*M* должен быть равен единице, чтобы избежать ситуации повтора, когда разным символам открытого текста соответствует один и тот же символ шифртекста.

*Пример 8.*Создадим таблицу замен для аффинной системы подстановок Цезаря с ключом (5, 4) на примере русского алфавита. Возьмем алфавит из 32 букв (все кроме буквы «Ё»). Таким образом,  *А = 5, К = 3, M = 32*и все условия (в том числе и НОД(5, 32) = 1) необходимые для использования (3) выполняются. Код буквы шифртекста находим из соотношения  *I = (5?J+4) mod 32.*

Сведем числовые коды букв открытого и зашифрованного текстов в таблицу (табл. 2).

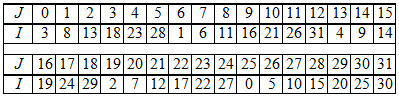


Таблица 2. Таблица кодов для аффинных подстановок при A=5, K=3, M=32

Преобразуем числовые коды в соответствующие буквы русского алфавита и получим соответствие для символов открытого текста и шифртекста (табл. 3).

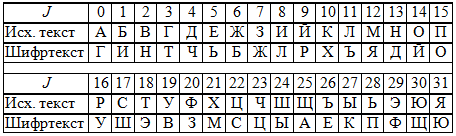


Таблица 3. Таблица символов для аффинных подстановок при A=5, K=3, M=32

С помощью табл. 3 или формулы (3) слово «МИР» преобразуется в шифртекст «ЯЛУ».

# Система шифрования Цезаря с ключевым словом. Его криптостойкость.

## Шифр Цезаря с ключевым словом.

Ключ состоит из 2 элементов:

* Само слово
* Индекс алфавита или номер символа исходного алфавита начиная от которого вписывается это ключевое слово.

Особенностью этой системы является использование ключевого слова для смещения и изменения порядка символов в алфавите подстановки (желательно, чтобы все буквы ключевого слова были различными) и некоторого числа а: 0<= а < N. Рассмотрим систему на примере.

Выберем некоторое число a и слово или короткую фразу в качестве *ключевого слова.* Пусть выбраны слово DIPLOMAT в качестве ключевого и число а=5.

Ключевое слово записывается под буквами алфавита, начиная с буквы, индекс которой совпадает с выбранным числом а, как это показано ниже:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |  |  |  |  | 10 |  |  | 15 |  |  |  |  | 20 |  |  |  |  |  |  |  |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|  |  |  |  |  | D | I | P | L | O | M | A | T |  |  |  |  |  |  |  |  |  |  |  |  |  |

Оставшиеся буквы алфавита подстановки записываются после ключевого слова в алфавитном порядке

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| V | W | X | Y | Z | D | I | P | L | O | M | A | T | B | C | E | F | G | H | J | K | N | Q | R | S | U |

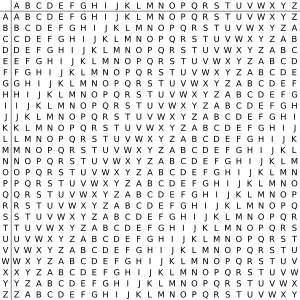
Если открытым сообщением будет BELSTU, то зашифрованным — WZAHJK.

# Шифр Виженера. Его криптостойкость.

## Шифр Виженера

Когда мы говорили про шифр Цезаря, мы упоминали, что для любого алфавита мощность N можно создать n-1 шифров. Это и использовал Виженер.

Вот таблица Виженера (квадрат Виженера), вверху записан алфавит над линией, в первой строке этот алфавит повторяется ещё раз, слева записывается тоже этот алфавит, а в каждой последующей строке есть циклический сдвиг предыдущей строки на одну позицию влево. Четвёртая строка является случаем классического шифра Цезаря. У по итогу 26 строк и 26 столбцов.



Процедура зашифрования:

Есть сообщение, которое нужно зашифровать. m=BELSTU. Выбираем ключевое слово. Например TIR. Используя метод шифрования на основе матрицы Виженера, следует стремиться к тому, чтобы длина ключевого слова примерно соответствовала длине сообщения и чтобы количество повторений одинаковых символов ключевой комбинации (соответствует длине шифруемого сообщения) было минимальным.

Запишем сообщение, ключ и шифротекст в виде таблицы:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сообщение | B | E | L | S | T | U |
| Ключ | T | I | R | T | I | R |
| Шифртекст | U | M | C | L | B | L |

Открываем таблицу Виженера, и далее выбирается символ подстановки по 2 параметрам: берется пара: буква B берётся из строки (д.б. началом строки), а T – из столбца (д.б. началом столбца). На пересечении столбца и строки находим букву в нашем случае это U. И так далее. В таблицу символов можно добавлять другие символы, например, точку, тире, запятую. Надо добавлять в строку и в столбец. Тогда увеличиться мощность алфавита.

Даже из этого простого примера видно, что разным буквам сообщения (S,U) соответствует одна и та же буква в шифротексте. Следует отметить, что для выполнения эффективного частотного криптоанализа аналитику важно знать длину ключа, если его начальная длина меньше длины шифруемого сообщения.

# Перестановочные шифры.

**Перестановочные шифры** используют перестановку символов исходного сообщения в соответствии с установленным правилом, открытый текст остается неизменным, но символы в нем «перетасовываются» (подвергаются пермутации). Так, в простом вертикальном перестановочном шифре 70 открытый текст пишется по горизонтали на разграфленном листе бумаги фиксированной длины, а шифртекст считывается по вертикали.

Рассмотрим это на примере:

М = «ВАСЯ ЛЮБИТ МАШУ».

Считывание по столбцам снизу вверх приводит к такому шифр-

тексту: С = «\_ЛВМЮААБСШИЯУТ\_».

Наиболее уязвимое место подстановочно-перестановчных шифров заключается в статистических характеристиках символов алфавита, поскольку это сообщение тайно и для шифрованных и зашифрованных алгоритмов.

Блочные и поточные шифры:

В блочных шифрах шифрованию и расшифрованию одновременно подвергаются блоки данных, в потоковых шифрах как правило преобразуется одиночный или несколько таких одиночных символов.

Симметричные и ассиметричные шифры:

Симметричные шифры с одинаковым ключом K1 = K2, ассиметричные с публичным ключом, в случае K1≠K2

В симметричных шифрах каждый пользователь должен иметь ключ для переписки с одним пользователем или группой. Вы с кем-то договариваетесь о ключе и передаете его друг другу.

Недостаток симметричных систем заключается в хранении и обмене ключевой информации.

# Методы симметричного криптопреобразования. Стандарт DES. Общая характеристика.

## Алгоритм DES

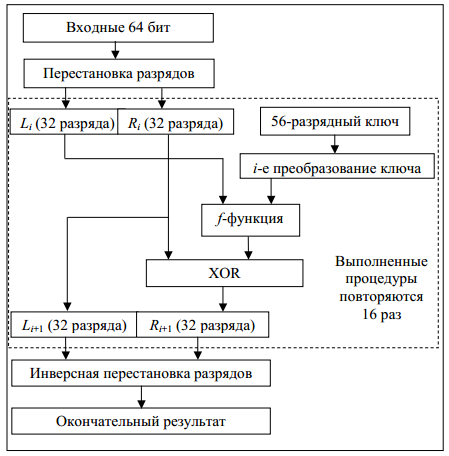
CRC – Check Redundancy Code — код контроля чётности.

Допустим, есть 1 байт, 1000 1101, для каждого полубайта создаю бит чётности. Для расчёта бита чётности равный сумме четырех элементов по модулю 2, считаем: 1+0+0+0=1 mod 2=1, вписываем после первого блока 1. Теперь код равен: 1000 1 1101. Далее считаем: 1+1+0+1=1 mod 2=1. Вписываем в код: 1000 1 1101 1. Вместо 8 исходных мы получаем 10. Мы вставили 2 бита чётности. При расшифровке не получается исходное сообщение.

**Как работает TCP протокол**. IP – протокол отвечает за доставку сообщения по адресу. Как письмо, взяли конверт написали письмо и оно поехало. А другой написал письмо, которое не вмещается в один конверт, он разложил письмо на 3 конверта, оказывается, что на почтовой станции одно письмо пошло по одному направлению (через один маршрутизатор), другое по другому направлению, третье – по ещё одному, они всё доходят дедушке на деревню, причём не обязательно в том порядке, в котором дедушка отправлял, так вот основная задача TCP – эти письма разложить в нужной последовательности. И вторая важнейшая особенность протокола, обнаруживать ошибки, которые появляются при передаче (вдруг конверт промочило по дороге).

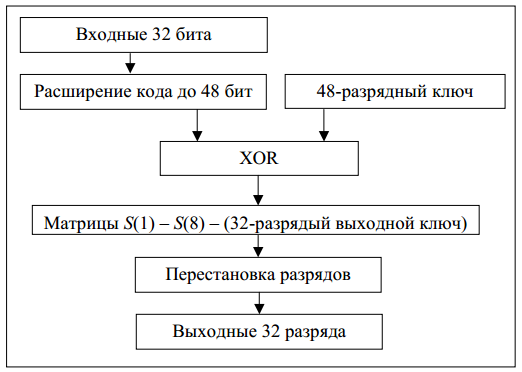
Здесь 56 бит и на чётных позициях, кратных степени 2 (находятся биты чётности, которые являются свёрткой (суммой) не по порядку, а определённых символов.

**DES** — блочный алгоритм (64-битовый блок) является подстановочно-перестановочным. Классический алгоритм. Блок данных шифруется и расшифровывается в течение **16 раундов**. Но эффективная длина ключа – 48 бит.



Имеется ключ 64 бита, 8 бит чётности, которые предназначены для контроля ошибок, отбрасывается, остается 56 бит. Много блоков и много раундов. Далее 56 бит делятся на 2 части, в каждой части используется сжимающая перестановка. Тут 28 28 из 56, потом их перемешивают и из них остаётся 48 рабочих, а после сдвигов это является входом до следующего раунда. !Знать блок-схему преобразования ключевой информации.

Есть блок данных из 64 бита, он делится на 2 части, левую и правую (каждый имеет длину 32 бита), в текущем раунде выполняется операция только над текущей частью, допустим у нас только над правой.



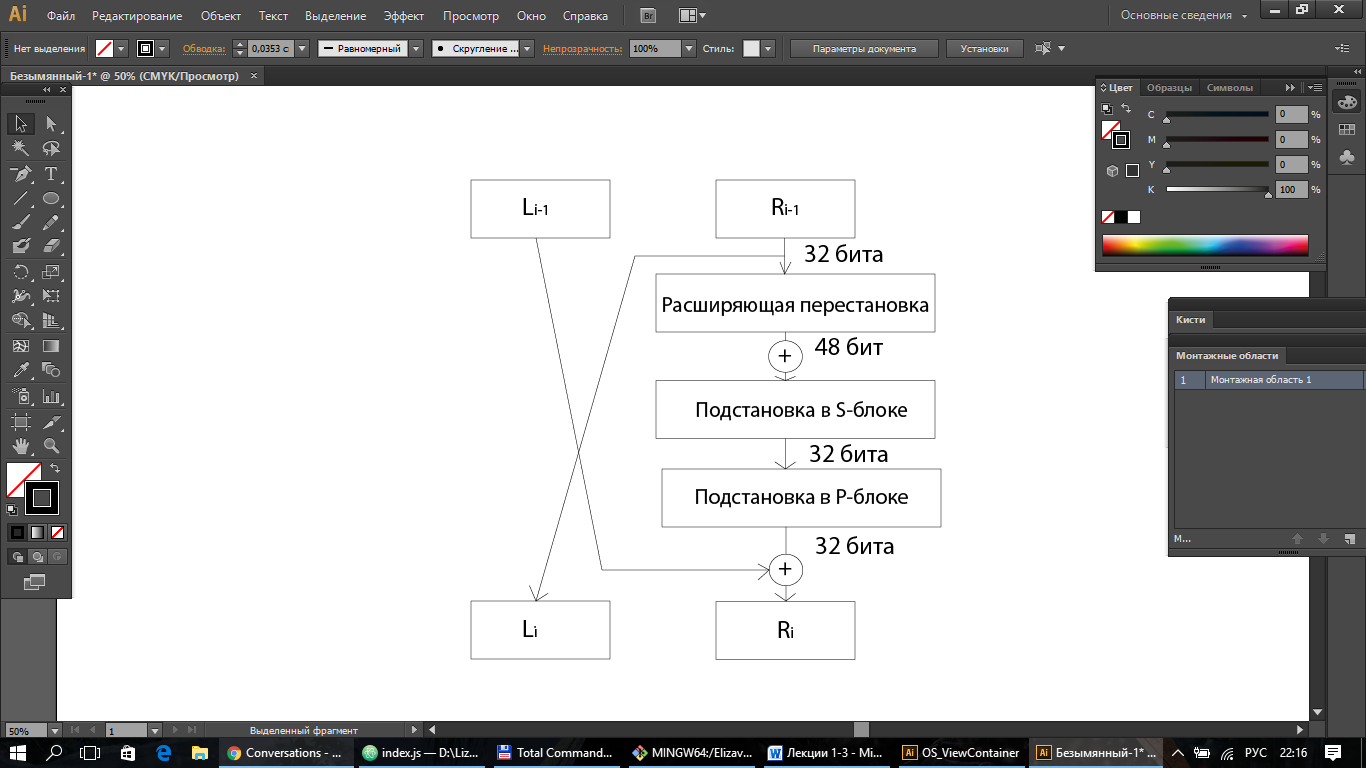
Имеется 32 бита, выполняется расширяющая перестановка, и из 32 бит за счет повторений сообщение увеличивается сообщение в полтора раза (48 бит), и тут используется операция сложения по модулю 2, далее выполняется операция подстановки и из 48 бит имеем 32 бита, потом ещё одна перестановка, имеем те же 32 бита, которые складываются по модулю 2 с битами левой части блока и это является входом для следующего раунда. !Знать и уметь пояснить блок-схему раунда шифрования/расшифрования в DES.

Было сообщение 32 бита, я условно их обозначил индексами от 1 до 32, самый первый символ повторяется 4 раза, он будет на первой позиции, на второй позиции, на 19 позиции и на 48 позиции. Второй символ будет на 3 позиции, 25 позиции. За счет повторений мы получили комбинацию из 48 бит, всё это есть в госте (в каком раунде, сколько бит мы повторяем), не известна только ключевая информация.

Длина ключа составляет 64 бита, из них 8 – биты четности: расположены в позициях 8, 16, 24, 32, 40, 48, 56, 64.

# Методы симметричного криптопреобразования. Стандарт DES. Структура одного цикла. Криптостойкость алгоритма.

## Один раунд DES



## Расширяющая перестановка

Основная цель расширяющих перестановок – создание «лавинного эффекта», который означает увеличение влияния отдельных бит шифруемого текста на конечный результат. Это затрудняет выполнение операций криптоанализа (взлома шифра).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 1 | 2 | 3 | 4 | 5 | 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 | 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 | 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 | 28 | 29 | 3 | 31 | 32 | 1 |

Результирующий 48-разрядный код преобразуется далее в 32-разрядный с помощью S-матриц.

## Подстановка в S-блоке

Производится в 8 блоках (каждый имеет 6-битовый вход и 4-битовый выход). Каждый блок – матрица из 4 строк и 16 столбцов: по 6-ти входным битам определяются номера столбцов и строк, под которыми ищем выходные значения: биты b1 и b6 объединяются (b1b6) и соответствуют № строки таблицы, остальные биты - № столбца.

Каждая таблица из 8 – известна.

Пример: на входе 6-го S-блока: 110011. № строки – 11 – третья строка, № столбца – 1001 – 9-й столбец. На их пересечении в таблице – число 14 – 1110 - эти данные на выходе блока

На вход каждого из 8-ми S-блоков поступает 6-битовое число (6\*8 = 48 бит), которое определяет адрес ячейки в этом блоке: 1-й и 6-й бит определяют адрес строки, четыре остальных - адрес столбца ячейки данного блока, в которых записано 4-х разрядное число, которое и является выходом данного блока.

S-Блок 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

S-Блок 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |

S-Блок 3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |

S-Блок 4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |

S-Блок 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 15 | 13 | 0 | 14 | 9 |
| 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |

S-Блок 6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |

S-Блок 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| 13 | 0 | 11 | 7 | 4 | 9 | 1 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |

S-Блок 8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |
| 2 | 1 | 14 | 7 | 4 | 10 | 8 | 13 | 15 | 12 | 9 | 0 | 3 | 5 | 6 | 11 |

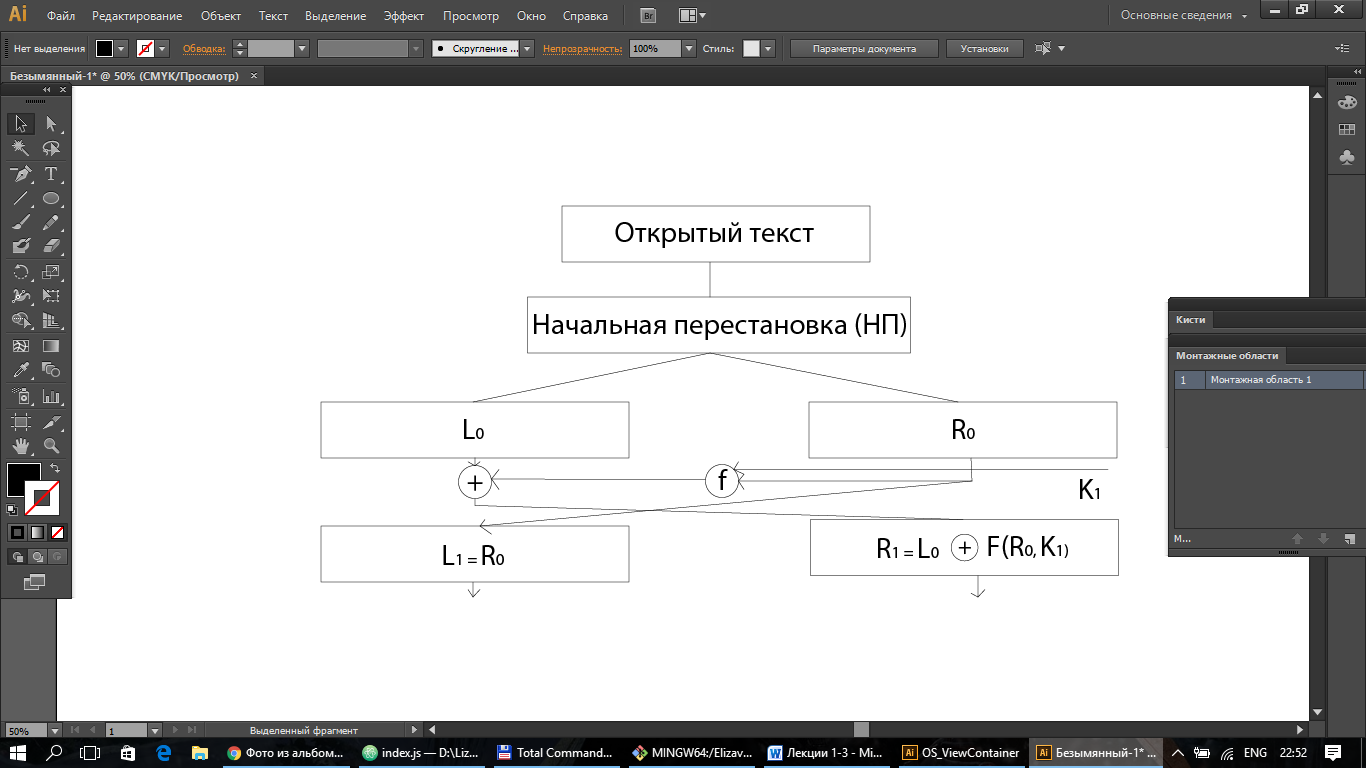
## Перестановка в P – блоке

Просто переставляются биты.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 | 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 | 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

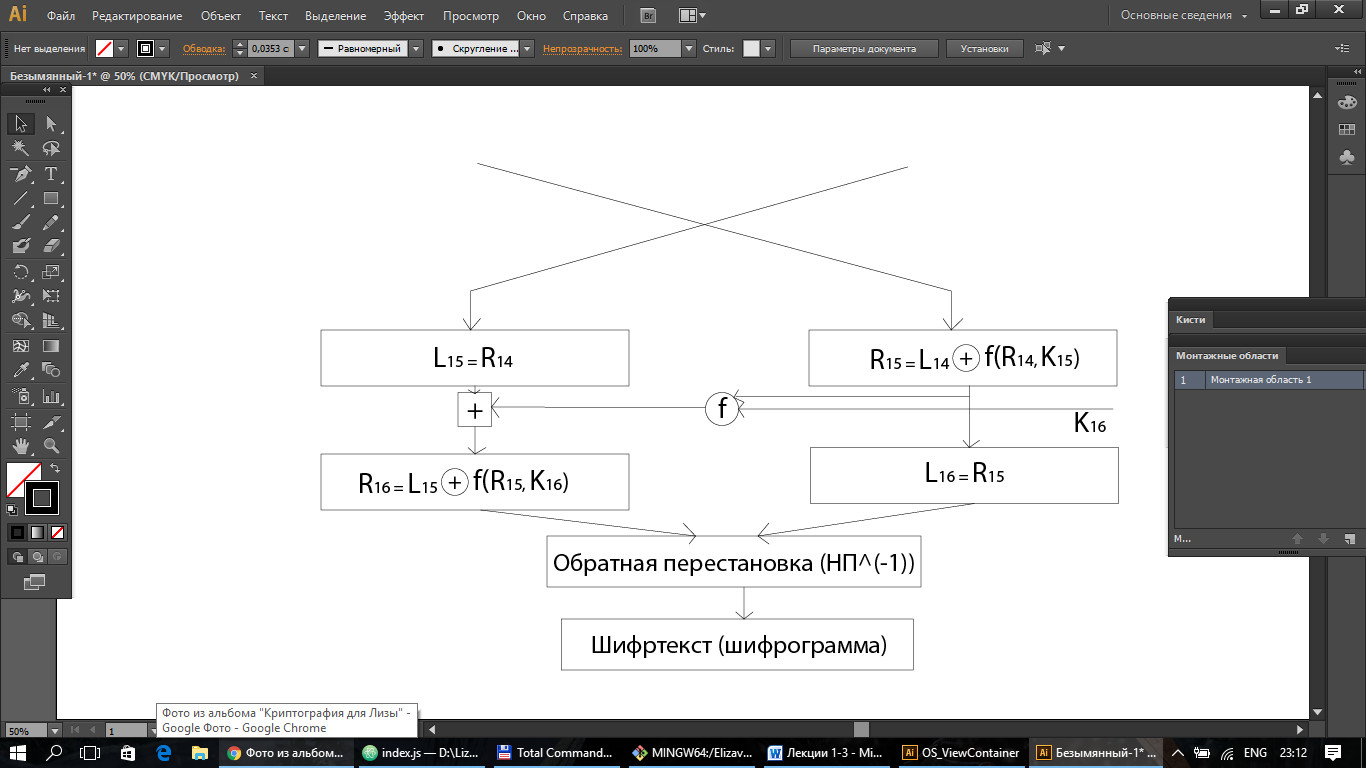
Последняя операция XOR – результаты перестановки в P-блоке суммируются с левой половиной исходного текста и образуют правую часть входной информации для следующего раунда.

Алгоритм DES (начало)



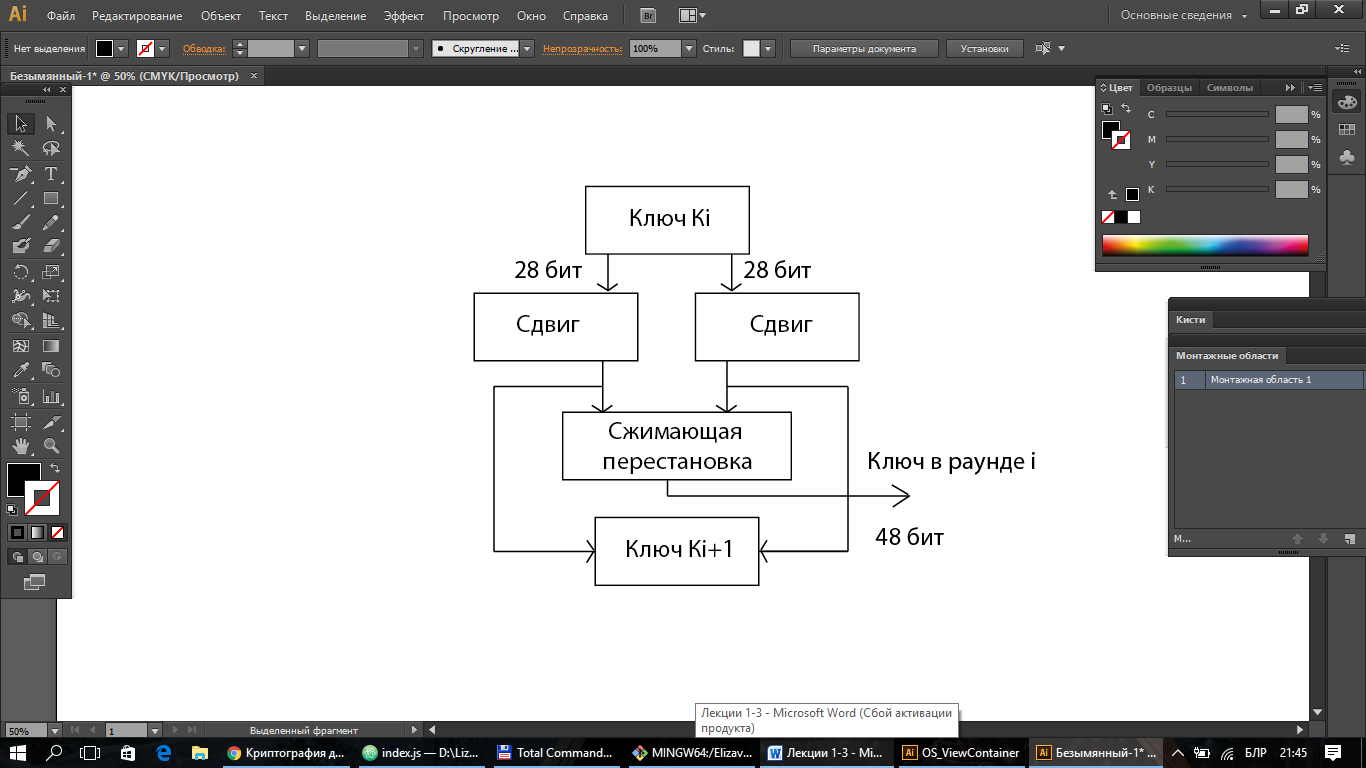
Один из недостатков алгоритма DES – в каждом раунде шифрованию подвергается только половина 8-байтового блока.

Алгоритм DES (окончание)



## Преобразования ключей

Преобразование ключа в первом раунде



Первоначальный 64-битный ключ сокращается до 56-битного путем отбрасывания битов четности. Остаются 56 битов, последовательность которых показана в таблице:

Таблица начальной перестановки битов 56-разрядного ключа

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

После этого для каждого из 16 раундов генерируется свой 48-битный подключ. Для этого 56 битов сначала делятся пополам. Затем над каждой половиной выполняется операция сдвига на 1 либо на 2 бита, в зависимости от раунда.

Циклический сдвиг

Влево на одну позицию:

111010 -> 110101

Влево на две позиции:

111010 -> 101011

Число битов сдвига ключа в каждом раунде

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Раунд | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Число | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Полученная комбинация ключа будет исходной для следующего раунда. А в текущем раунде выполняется основная операция – сжимающая перестановка: из 56 битов будут выбраны 48 и их последовательность будет изменена. Принцип такого преобразования в таблице:

Сжимающая перестановка ключа

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 14 | 17 | 11 | 24 | 1 | 5 | 3 | 28 | 15 | 6 | 21 | 10 |
| 23 | 19 | 12 | 4 | 26 | 8 | 16 | 7 | 27 | 20 | 13 | 2 |
| 41 | 52 | 31 | 37 | 47 | 55 | 30 | 40 | 51 | 45 | 33 | 48 |
| 44 | 49 | 39 | 56 | 34 | 53 | 46 | 42 | 50 | 36 | 29 | 32 |

Расшифрование осуществляется в обратной последовательности с ключами обратной последовательности.

## Реализация DES

* Аппаратная. Самая скоростная реализация – на основе числа DEC: скорость зашифрования и расшифрования – более 1 Гб/с (16.8 млн блоков за 1 секунду).
* Программная. На мэйнфрейме IBM 3090 – более 32 тыс блоков за 1 сек.

Оценка времени лобового взлома шифра DES 1995 г.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Стоимость в долларах | Длина ключа, бит | | | | |
| 40 | 58 | 64 | 80 | 128 |
| 100 тыс | 2 с | 35 ч | 1 г | 70000 л | 1019 л |
| 1 млн | 0.2 с | 3.5 ч | 37 д | 7000 л | 1018 л |
| 100 млн | 2 мс | 2 мин | 9 ч | 70 л | 1016 л |
| 10 млрд | 0.02 мс | 1 с | 5.4 мин | 245 д | 1014 л |

Основной подход при защите: Стоимость информации >= Стоимость взлома.

Основное достоинство алгоритма – небольшая длина ключа. Это позволяет производить шифрование и расшифрование с высокой скоростью.

## Особенности DES

DES работает с бинарными числами с 0 и 1. Каждая группа из 4-х бит преобразовывается в 16-ричное число: бинарное 0001 равно 16-ричному 1, 1000 – 8, 1111 – F.

DES шифрует группы 64-битных сообщений (16-ричных чисел)

DES использует ключи (16 hex или 64 бита; каждый 8-й ключевой бит игнорируется в DES алгоритмом, так что эффективный размер ключей 56 бит).

Пример:

M = 0123456789ABCDEF (M – в 16-ричном формате). М в бинарном формате (64 бит блок текста):

Достоинства DES – относительно высокая скорость (из-за малой длины ключа), бесплатное

М = 000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

L = 0000 0001 0010 0011 0100 0101 0110 0111

R = 1000 1001 1010 1011 1100 1101 1110 1111

K = 122457799BBCDFF1

Или R = 0001 0011 00110100 01010111 011111001 10011011 10111100 11011111 **1**1110001

Каждый 8-й ключевой бит игнорируется DES в соответствии с таблицей

57 49 41 33 25 17 9

1 58 50 42 34 26 18

10 2 59 51 43 35 27

19 11 3 60 52 44 36

63 55 47 39 31 23 15

7 62 54 46 38 30 22

14 6 61 53 45 37 29

21 13 5 28 20 12 4

Достоинства DES – распространение по всему миру, общедоступность и отсутствие лицензионных отчислений.

Недостатки DES – низкая криптостойкость (из-за малой длины ключа), проблема хранения и распределения ключевой информации.

На основе DES строится протокол Kerberos.

# Методы симметричного криптопреобразования. Стандарты 3DES. Реализация и криптостойкость. Характеристика других известных симметричных алгоритмов.

3-DES

Формально зашифрованный блок данных получается на основе трех процедур преобразования. Существует три типа алгоритма:

DES-EEE3 - Шифруется 3 раза с 3 разными ключами (шифрование-шифрование-шифрование)

DES-EDE3 - Операции шифрование-расшифрование-шифрование с разными ключами

DES-EEE2 и DES-EDE2 - Как и предыдущие, однако, на первом и третьем шаге используется одинаковый ключ.

Криптостойкость 3-DES

3-DES с различными ключами имеет длину ключа равную 168 бит, но из-за атак «встреча посередине» (известны М и С, найти К) эффективная криптостойкость составляет только 112 бит. Криптоаналитик имеет расшифрованные и соответствующие им зашифрованные сообщения. Чем больше будет таких пар, тем выше вероятность взлома шифра, т.е. вычисления ключа.

В варианте DES-EDE, в котором К1=К2, эффективный ключ имеет длину 89 бит.

Применение 3-DES

3-DES с тремя ключами реализован во многих интернет приложениях.

3-DES используется при управлении ключами в стандартах ANSI и ISO 8732

Не известны успешные атаки на 3-DES.

Современная альтернатива 3-DES – AES Rijndael.

# Сравнительная характеристика алгоритмов Lucifer, IDEA, ГОСТ 28147-89, Blowfish.

Для сравнения с IDEA выбраны [DES](https://ru.wikipedia.org/wiki/DES), [Blowfish](https://ru.wikipedia.org/wiki/Blowfish) и [ГОСТ 28147-89](https://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2_28147-89). Выбор [DES](https://ru.wikipedia.org/wiki/DES) обусловлен тем, что IDEA проектировался как его замена. [Blowfish](https://ru.wikipedia.org/wiki/Blowfish) выбран потому, что он быстр, и был придуман известным криптологом Брюсом Шнайером. Для сравнения также выбран [ГОСТ 28147-89](https://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2_28147-89), блочный шифр, разработанный в [СССР](https://ru.wikipedia.org/wiki/%D0%A1%D0%A1%D0%A1%D0%A0). Как видно из таблицы, размер ключа у IDEA больше, чем у DES, но меньше, чем у ГОСТ 28147-89 и Blowfish. Скорость шифрования IDEA на [Intel486SX](https://ru.wikipedia.org/wiki/Intel486SX)/33МГц больше в 2 раза, чем у DES, выше чем у ГОСТ 28147-89, но почти в 2 раза меньше, чем у Blowfish.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Таблица параметров** | | | | | |
| **Алгоритм** | **Размер ключа, бит** | **Длина блока, бит** | **Число раундов** | **Скорость шифрования на**[**Intel486SX**](https://ru.wikipedia.org/wiki/Intel486SX)**/33МГц (Кбайт/с)** | **Основные операции** |
| [**DES**](https://ru.wikipedia.org/wiki/DES) | 56 | 64 | 16 | 35 | Подстановка, перестановка, побитовое исключающее ИЛИ |
| **IDEA** | 128 | 64 | 8 | 70 | Умножение по модулю {\displaystyle 2^{16}+1}, сложение по модулю {\displaystyle 2^{16}}, побитовое исключающее ИЛИ |
| [**Blowfish**](https://ru.wikipedia.org/wiki/Blowfish) | 32-448 | 64 | 16 | 135 | Сложение по модулю {\displaystyle 2^{32}}, подстановка, побитовое исключающее ИЛИ |
| [**ГОСТ 28147-89**](https://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2_28147-89) | 256 | 64 | 32 | 53 | Сложение по модулю {\displaystyle 2^{32}}, подстановка, побитовое исключающее ИЛИ, циклический сдвиг |

# Криптографические системы с открытым (публичным) ключом. Задача об укладке ранца.

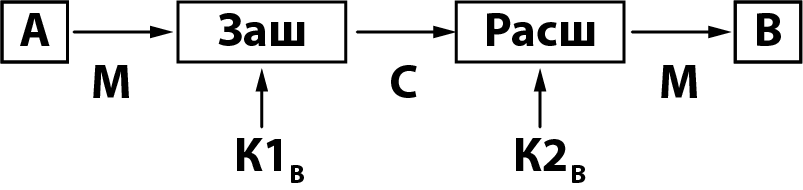
## Ассиметричная криптография

Авторы – У. Диффи, М. Хеллман -1976 г.

Идея – использовать парами (К1: для зашифрования и К2 для расшифрования), которые очень трудно вычислить один из другого; ключ К1 известен и доступен для всех, ключ К2 – тайный.

Особенность ассиметричной криптографии в том, что ключи зашифрования и расшифрования разные. Идея систем состоит в том, что не имея тайного ключа можно так же узнавать информацию.

Упрощённая схема преобразования (Аня – **А** передаёт зашифрованное сообщение Васе – **В**)



Криптография придумана для того, что передавать тайную информацию по публичному каналу.

## Ранцевый алгоритм

Особенностью асиметричных алгоритмов является то, что для зашифрования и расшифрования используется два разных ключа. Одним из простых примеров реализации асимметричной системы является **ранцевый алгоритм**. Алгоритм называется ранцевый и означает, что в основе его решения лежит укладывание ранца. Например, у нас есть много вещей, которые нужно сложить в чемодан, но есть оговорка, что вес чемодана не должен превышать какое-то число килограммов, которые, допустим, разрешает перевозить бесплатно данный авиаперевозчик. Вы хотите взять то ли какие-то конкретные вещи, то ли как можно больше вещей увезти с  собой. Вот и есть объяснение алгоритма на пальцах.

Алгоритм разработан Мартином Хеллманом и Робертом Мерклём (иногда говорят Меркелем). Существует множество предметов различного веса. Вопрос: можно ли положить какие-то конкретно предметы в чемодан, чтобы его вес стал равным определённому значению?

*Как формулируется задача?* В основе алгоритма лежит число некоторых предметов (чисел) n и есть суммарное число предметов из этого множества, равное S.

*Задача*: требуется вычислить значения коэффициентов bn (bi=0 или bi=1, bi принадлежит {0, 1}). Если bi=0, то предмет в чемодан не кладем, если bi=1, значит, кладём.

*Пример*. Есть веса предметов 1, 5, 6, 11, 14 и 20, при этом нужно упаковать рюкзак так, чтобы его вес стал равен 22, использовав предметы весом 5, 6 и 11. Как это сделать? Невозможно упаковать рюкзак так, чтобы вес его стал 24. Время, необходимое для решения этой проблемы, в зависимости от объёма в общем случае возрастает как экспоненциальная функция от количества предметов.

Меркль рекомендовал использовать n порядка 100, где n — число элементов супервозрастающей последовательности («размер»-рюкзака). Таким образом, он использовал не произвольную последовательность Mi, а супервозрастающую, то есть такую, что каждый последующий член Mi больше суммы всех предыдущих:

Каждый элемент должен быть больше, чем сумма всех предыдущих элементов. Пример такой последовательности можем записать (2, 3, 6, 12, …) — 2+3=5<6, 2+3+6=11<12.

Итак, в основе ассиметричного шифра лежит идея об укладке ранца.

Пример текста, зашифрованного с помощью задачи укладки ранца, показан в таблице (сумма последующих здесь не равна, но главное идея):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Открытый текст | 111001 | 010110 | 000000 | 011000 |
| Ранец | 1 5 6 11 14 20 | | | |
| Вычисление шифра | 1+5+6+20= | 5+11+14= | 0 | 5+6= |
| Шифртекст | 32 | 30 | 0 | 11 |

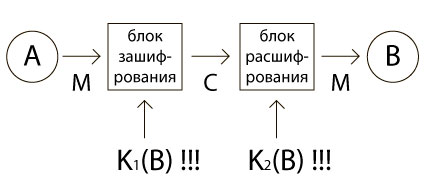
Если мы первый, второй. третий и последний элементы, то в открытом тексте мы берем бинарную последовательность: b1=1, прибавляем второй элемент (b2=1), третий элемент прибавляем (b3=1), потом два элемента мы не используем (11, 14), соответственно, стоят нули (b4=0 и b5=0), и наконец используем последний (20), стоит единица (b6=1). Значит, открытым текстом будет вот эта бинарная комбинация — 111001. А если бинарную последовательность отождествлять с кодом бинарного определенного символа последовательности, то все становится понятно. Как мы говорили окодах ASCII, каждый символ любой раскладки имеет свой бинарный код. Если известна только комбинация, а неизвестен сам символ, то в этом и заключается тайна. Если складываем только 5 и 6, то будет такая комбинация — 011000.

*Что же тут является открытым ключом. а что является закрытым?* Предположим, как реализуется идея шифрования асимметричной системы? В симметричной понятно: я отправляю кому-то сообщение, используя один и тот же ключ, который никто не знает. А в асимметричной системе есть 2 ключа:

*K1 -> e (encrypt) -> открытый, публичный (для зашифрования)*

*K2 -> d (decrypt) -> закрытый, тайный (для расшифрования)*

При расшифровании и при зашифровании используются ключи одной стороны В, стороны получателя. Отправитель никаких своих ключей не использует. Если я хочу отправить какое-то сообщение, то я должен узнать открытый ключ стороны B (стороны получателя) и затем расшифровать это сообщение сможет только сам получатель с помощью своего тайного ключа.



*А — отправляет (отправитель)*

*B — получает (получатель)*

Для такого набора чисел решение задачи является тривиальным. Чтобы избавиться от этой тривиальности понадобилось ввести «секретный ключ», который создают два числа:

* q должно быть больше суммы всех Mi
* r такое, что НОД(q, r)=1, то есть эти числа взаимно простые.

Вместо первоначального набора чисел Mi будем использовать числа (дополнительные коэффициенты):

*ci=r\*Mi mod q*

Понятно, что M никто знать не должен. Это текст, который мы шифруем. Открытый текст — это те коэффициенты, которые мы вычислили из ci. А закрытый, это те вещи с конкретными весами, которые мы выбираем, а также числа q и r. Сообщение — это двоичные числа b, то есть бинарные числа, которые соответствуют символам открытого текста, как в предыдущей таблице. Коэффициенты b (числа b) соответствуют значениям из таблицы на странице 72 учебника. Сам шифртекст есть сумма произведений:

*ci\*bi, где ci — открытый ключ получателя*

Я взял q и r, вычислил ci и выложил их в открытую базу данных (разрешил брать всем, кому захочется). Дальше получатель расшифровывает сообщение, находит С, используя следующее тождество (формулу):

С= *c1, c2, …, cn*

Серьезного практического применения ранцевый алгоритм не нашел, он просто является примером реализации идеи асимметричного шифрования. Но серьезное практическое применение нашел другой алгоритм — алгоритм RSA.

# Управление криптографическими ключами. Алгоритм рукопожатия.

## Управление криптографическими ключами

**Управление ключами** – информационный процесс, включающий в себя три элемента:

* генерацию ключей;
* накопление ключей;
* распределение ключей.

Алгоритм Диффи-Хелмана позволяет значительно упростить процедуру обмена ключевой информацией через публичные каналы.

В серьезных ИС используются специальные аппаратные и программные **методы генерации случайных ключей** (датчики ПСЧ).

Под **накоплением ключей** понимается организация их хранения, учета и удаления.

Вся информация об используемых ключах должна храниться в зашифрованном виде. Ключи, зашифровывающие ключевую информацию, называются **мастер-ключам** (мастер-ключи каждый пользователь должен знать наизусть).

## Распределение ключей

**Распределение ключей (РК)** – самый ответственный процесс в управлении ключами.

К нему предъявляются два требования:

* оперативность и точность распределения;
* скрытность распределяемых ключей.

Введем обозначения:

* IA — идентификатор стороны А;
* DA — секретное криптопреобразование стороны А (с использованием секретного ключа асимметричной криптосистемы);
* ЕА — открытое криптопреобразование стороны А (с использованием открытого ключа асимметричной криптосистемы);
* TА — временной штамп (метка) стороны А;
* RА — случайное число, выбранное стороной А.

## Протокол рукопожатия (основа протокола SSL)

А и В желают определить общий секретный ключ (**K**). Они знают открытые ключи друг друга.

1. **А** посылает **В** сообщение **С= ЕB (IА , RА )**; **ЕB** – процедура зашифрования с открытым ключом **В**, **IА** – идентификатор **А** и **RА** – случайное число.
2. **В** расшифровывает **С** и получает **IА**и **RА.**

**В** посылает **С’= ЕA (IB , RA )** в адрес **А**;

После расшифрования **С‘** **А** может проверить, что **В** получил **RА**, поскольку только **В** может расшифровать **С**.

1. **А** посылает **В** сообщение **C″ = EB(KB),**

**В** расшифрует **С"** и сможет проверить, что **А** получил

**IB**, поскольку только **А** может расшифровать **С'**.

Тем самым А и В **аутентифицировали друг друга**.

Теперь **А** посылает **В**: **С’’’=EB(K), В** расшифровывает сообщение и получает **К**.

**Алгоритм обеспечивает как секретность, так и аутентичность при обмене ключом K.** Может использоваться **TА** – временной штамп (метка) стороны **А.** Стороны А и В завершили процедуру взаимной аутентификации (пожали друг другу руки).

# Распределение ключей на основе симметричных систем.

## Распределение ключей

**Распределение ключей (РК)** – самый ответственный процесс в управлении ключами.

К нему предъявляются два требования:

* оперативность и точность распределения;
* скрытность распределяемых ключей.

Введем обозначения:

* IA — идентификатор стороны А;
* DA — секретное криптопреобразование стороны А (с использованием секретного ключа асимметричной криптосистемы);
* ЕА — открытое криптопреобразование стороны А (с использованием открытого ключа асимметричной криптосистемы);
* TА — временной штамп (метка) стороны А;
* RА — случайное число, выбранное стороной А.

## РК на основе симметричных систем.

1. Получение двумя пользователями общего ключа от центрального органа — центра распределения ключей (ЦРК) или центра сертификации (ЦС). ЦС играют важную роль.

Проблемы:

* Одно проникновение в систему злоумышленника компрометирует ЦРК;
* ЦРК может долгое время участвовать в пассивном подслушивании, прежде чем это будет обнаружено.

*Возможное решение:* иерархическая (древовидная) система с пользователями, находящимися на листьях, и ЦРК в промежуточных узлах.

1. Используя открытые сети.

Проблема: **атака человек (встреча) посередине.** Пример: Павел Павлович попросил у Вероники в долг денег, Вероника сказала, что не доверяет ему. Тогда он просит у Наташи, она говорит, что у неё нет денег, но она ему доверяет, тогда он спросил доверяет ли Вероника Наташе, она сказала, что да. Тогда Павел Павлович попросил Наташу, чтобы она взяла деньги у Вероники и дала ему деньги. Тогда между Павлом Павловичем и Вероникой возникают транзитивные доверительные отношения через третью сторону (Наташу). Потом он отдаёт Наташе деньги, и между ними абсолютные доверительные отношения. Но может быть так, что Наташа использует ситуацию по-другому.

Точно так же, например, мы передаём по сети сообщение зашифрованное и человек по середине знает ключ. Тогда допустим Павел Павлович выслал сообщение: «Вероника я тебе вышлю деньги послезавтра». А Наташа изменила его на «Отдал Наташе». Это называется атака по середине или подмена.

# Алгоритм передачи ключа по Диффи-Хеллману.

## Управление криптографическими ключами

**Управление ключами** – информационный процесс, включающий в себя три элемента:

* генерацию ключей;
* накопление ключей;
* распределение ключей.

Алгоритм Диффи-Хелмана позволяет значительно упростить процедуру обмена ключевой информацией через публичные каналы.

В серьезных ИС используются специальные аппаратные и программные **методы генерации случайных ключей** (датчики ПСЧ).

Под **накоплением ключей** понимается организация их хранения, учета и удаления.

Вся информация об используемых ключах должна храниться в зашифрованном виде. Ключи, зашифровывающие ключевую информацию, называются **мастер-ключам** (мастер-ключи каждый пользователь должен знать наизусть).

## Распределение ключей

**Распределение ключей (РК)** – самый ответственный процесс в управлении ключами.

К нему предъявляются два требования:

* оперативность и точность распределения;
* скрытность распределяемых ключей.

Введем обозначения:

* IA — идентификатор стороны А;
* DA — секретное криптопреобразование стороны А (с использованием секретного ключа асимметричной криптосистемы);
* ЕА — открытое криптопреобразование стороны А (с использованием открытого ключа асимметричной криптосистемы);
* TА — временной штамп (метка) стороны А;
* RА — случайное число, выбранное стороной А.

# Алгоритм шифрования RSA. Реализация и криптостойкость.

## Алгоритм RSA

Алгоритм, созданный **R**ivest-ом, **S**hamir-ом и **A**delman-ом.

*Идея*: есть открытый текст M, который состоит из элементов m1, m2, …, mn. Есть ключ k1=e и k2=d. Как создаётся ключ? Каждая группа пользователей должна **сгенерировать** для себя *ключевую информацию*. Вот эти числа e и d. Первоначально для генерации ключа выбираются два *целых* простых числа p и q, значения которых являются тайными для всех, кроме их владельца. Дальше вычисляется число n:

*n=p\*q*

**Элементы ключа** — большие простые числа e, d, n.

Находим функцию Эйлера: φ(n)=(p-1)\*(q-1).

Далее случайно выбирается любое число e *(e и φ(n)=(p-1)\*(q-1) — взаимно простые числа)*. С помощью функции Эйлера находим d:

e\*d=1 mod φ(n) или d=e-1 mod φ(n)

Число е является вторым элементом ключа. Далее считаем d; e и d — также взаимно простые числа:

*d\*e=1 mod φ(n)*

В результате, пользователь сгенерировал три числа e, d, n. Это ключ, который состоит из двух частей: e и n — открытый (публичный) ключ (K1), d (d и n) —закрытый (тайный) ключ К2. Числа e и d можно поменять местами. Если мы выбрали так, то е – делаем общедоступным, а d— тайным, известным только пользователю.

**Зашифрование:** ci=(mi)e mod n

**Расшифрование:** mi=(ci)d mod n

*Пример:* p=11 q=3 два простых числа

*n=p\*q=11\*3=33*

*φ(n)=(p-1)\*(q-1)=10\*2=20*

n - первый элемент ключевой информации. Какая трудность вычисления n? Заключается в разложении числа на простые сомножители.

Выбираем число e, которое должно быть взаимно простым с φ(n). Пусть е=7 (можно быдло взять d=7), это второе значение ключа. И находим d:

*d\*e=1 mod φ(n)*

*d\*7=1 mod 20*

Получили ключ K=(e=7, d=3, n=33). Два ключа К1: e, n и К2: d, n.

Процедура шифрования предусматривает использование информации той стороны, для которой создаётся зашифрованное сообщение, то есть используется ключевая информация принимающей стороны, то есть стороны B.

*Пример*: M=’ВЕГА’. Мы оперируем не значками, а числами, которые являются, например индексами этих чисел в алфавите.

*m1=В (03), m2=Е (06), m3=Г (04), m4=А (01),* еслипредставлять это сообщение в цифровой форме (в десятичной системе счисления) или *m1=В=000011* если перевести в бинарное представление (в алфавите русском 33 буквы, то есть нам нужно 6 разрядов для того, чтобы закодировать все символы разными значениями).

Теперь будем кодировать эти числа с использованием ключевых значений.

**Зашифрование**

c=c1c2c3c4

c1=(m1)e mod n=(3)7 mod 33 = 2187 mod 33 = 9 -> “З” буква з

c2=(m2)e mod n=(6)7 mod 33 = 279936 mod 33 = 30 -> “Ь” буква ь

c3=(m3)e mod n=(4)7 mod 33 = 16 -> “О” буква о

c4=(m4)e mod n=(1)7 mod 33 = 1 -> “А” буква а

*Зашифрованное сообщение:* С = «3ЬОА»

**Расшифрование**

Подставляем d:

mi=(ci)d mod n

m1=(c1)3 mod 33 = (9)3 mod 33 = 729 mod 33 = 3 -> (“В”)

Подставляем e:

mi=(ci)e mod n

m1=(9)7 mod 33 = 15 (вышли разные ответы)

Если криптоаналитик знает эти числа, знает ключ, то есть знает, как проводилось вычисление. Приходим к выводу, что в основе алгоритма RSA лежит трудно разрешаемая *проблема вычисления дискретного логарифма в конечном поле*.

Сейчас в современных системах длина ключа составляет L(K)=1024 bit, а в некоторых системах 2048 bit. Решена проблема факторизации чисел длиной до 512 bit.

## Стойкость RSA

Криптостойкость машины «Энигма» оценивается в 159\*1018 вариантов комбинаций шифра.

**Стойкость RSA основывается на большой вычислительной сложности известных алгоритмов разложения произведения простых чисел на сомножители.**

**Пример:** легко найти произведение двух простых чисел 7 и 13 – 91. Попробуйте два простых числа, произведение которых равно 91. (7 и 13) или 323 (17\*19).

Для надежного шифрования алгоритмом RSA, как правило, выбираются простые числа, количество двоичных разрядов которых составляет тысячи.

**Атаки на RSA**

Алгоритм машины Энигма — сложный подстановочный алгоритм, никаких вычислительный операций в нем нет. Сложность взлома заключается в подстановках, каждый символ — новая подстановка.

Алгоритм RSA — классический вычислительный алгоритм, никаких подстановок и перестановок нет.

В худшем случае имеется шифртекст, в лучшем — имеются открытые символы, соответствующие закрытым.

Криптоаналитик имеет шифртекст **C = E(M)**.

Цель — восстановить **М** (расшифровать **С**).

Метод: Шифрует известным открытым ключом произвольное сообщение **М’**. Получает **С’**. Если **С=С’**, то получено **М (М= М’)**, в ином случае шифруется другое сообщение **М”** и т.д.

Основные методы взлома шифра:

1. Наличие только зашифрованных сообщений

2. Наличие зашифрованных и нескольких фрагментов

***ХВ (06.04)***

Общая схема системы анализа с перехватом

Такую возможность исключает **вероятностное шифрование** (алгоритмы Эль-Гамаля, алгоритмы на основе эллиптических кривых) — зашифрование одного и того же сообщения одним и тем же открытым ключом даёт различные шифртексты. Но при расшифровании — получаются одинаковые сообщения.

Зная вероятностные свойства символов, мы имеем доп аргументы, которые упрощают процедуру взлома шифра или расшифрования смс. Это относится к проблемам вероятностного шифрования.

Пример: криптоаналитик перехватил какой-то текст: **мтутпти** и знает (или предполагает), что он был зашифрован при помощи шифра простой замены.

Этот шифртекст свидетелствует о том, что открытый текст состоит из 7 букв, причем вторая, четвертая и шествая буквы совпадают. Можно предположить, что открытым текстом (словом) является **«дорогой»** — это наиболее вероятное значение начального слова сообщения (или **молоток**, или **потолок**, или..).

**Апостериорные вероятности** таких открытых текстов возрастают относительно их априорных вероятностей.

Криптоаналитик уверен, что таким словом не может быть **«призрак»** или **«сатирик»**, т. е. апостериорная вероятность обоих этих открытых текстов сокращается до нуля вне зависимости от их априорных вероятностей.

**Апостериорная вероятность** — условная вероятность случайного события при условии того, что известны апостериорные данные, т. е. полученные после опыта (при условии, что зашифрованное слово стоит вначале, значит это скорее всего дорогой).

**Априорная вероятность** (безусловная вероятность) связана с некоторым событием А, представляет собой степень уверенности в том, что данное событие произошло, в отсутствии любой другой информации.

В машине энигма сообщение начинается «Hi Gitler!» вероятность этого является достаточно высокой.

# Алгоритм шифрования Эль-Гамаля. Реализация и криптостойкость.

## Ассиметричный алгоритм шифрования Эль-Гамаля

* Стойкость алгоритма базируется на сложности решения задачи дискретного логарифмирования
* Предложен в 1985

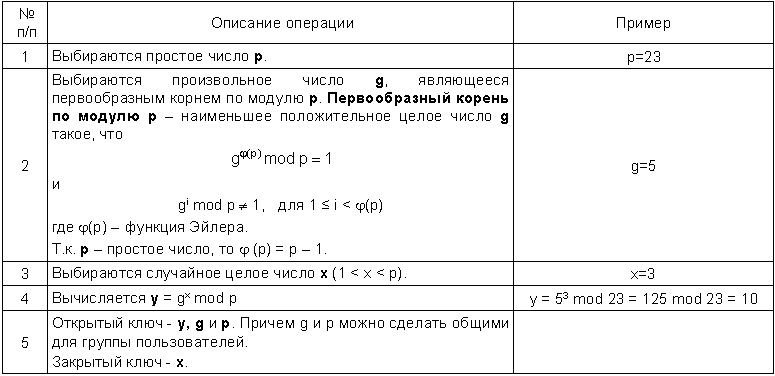
Основные отличия от RSA:

1. Открытый ключ в алгоритме Эль-Гамаля состоит из 3 чисел, закрытый из 1.
2. Каждый зашифрованный блок данных состоит из 2 чисел (в RSA из одного)
3. В алгоритме Эль-Гамаля для повышения криптостойкости шифра (вероятностное шифрование) используется дополнительный параметр — **некоторое случайное число k.**

!все шифры строятся на основе ключевой информации получателя!

Этапы алгоритма:

1. Генерация ключа



Тот, кто шифрует смс:

* Выбирает случайное число k (1<k<p-1). Дополнительный параметр для усложнения алгоритма.
* Шифрограмма (С) генерируется по следующим формулам:

**a = gk mod p,           (5)**

**b = (yk \*М) mod p,       (6)**

где M — исходное сообщение;

(a, b) – зашифрованное сообщение (С).

* Расшифрование С выполняется по следующей формуле

**М = (b \*(ax)-1) mod p          (7)**

**или**

**М = (b \*ap-1-x) mod p,          (8)**

где (ax)-1 – обратное значение числа ax по модулю p.

Пример.

М= «Абрамов» (1 2 18 1 14 16 3)

Ключ:

* открытый: y=10, g=5, p=23
* тайный: х=3

Зашифрование.

Выбираем k=7 (для шифрования каждого символа следует выбирать разные k – принцип вероятностного шифра).

Для упрощения примем k неизменным. Далее каждый блок шифртекста шифруется 2 числами.

Используем (5): a = gk mod p = 57 mod 23 = 17 (для каждого символа д.б. подсчитано отдельное знач. а

Используем (6): **b = (yk \*М) mod p** для вычисления второй части шифртекста:

* Для первой буквы (А): **b1 = (yk1 \*М1) mod p = (107\*1) mod 23 = 14,**
* Для второй буквы (б): **b2 = (yk2 \*М2) mod p = (107\*2) mod 23 = 5,**
* Для третьей буквы (р): **b3 = (yk3 \*М3) mod p = (107\*18) mod 23 = 22**
* И т.д.

Шифртекст **С**=171417051722…..

Длина сообщения **удваивается (основной недостаток алгоритма).**

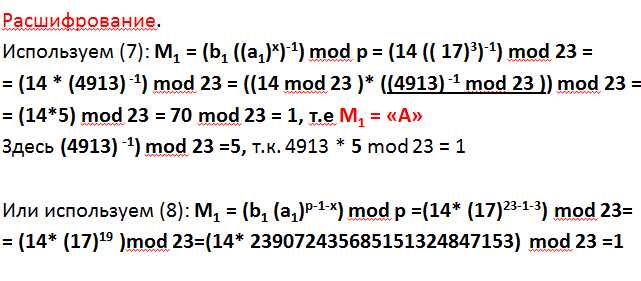
Расшифрование.

**Используем (7):** М1 = (b1 ((a1)x)-1) mod p = (14 (( 17)3)-1) mod 23 =

= (14 \* (4913) -1) mod 23 = ((14 mod 23 )\* ((4913) -1 mod 23 )) mod 23 = (14\*5) mod 23 = 70 mod 23 = 1, т.е **М1 = «А»**

Здесь **(4913) -1) mod 23 =5**, т.к. 4913 \* 5 mod 23 = 1

Или используем (8): М1 = (b1 (a1)p-1-x) mod p =(14\* (17)23-1-3) mod 23= (14\* (17)19 )mod 23=(14\* 239072435685151324847153) mod 23 =1



и.т.д.

!!! В схеме Эль-Гамаля необходимо использовать различные значения случайной величины k для зашифрования различных сообщений М и М’. Если использовать одинаковые k, то для соответствующих шифртекстов (a, b) и (a’, b’) выполняется соотношение b (b’)-1 = М (М’)-1 (mod p). Из этого выражения можно легко вычислить М, если известно М’.

# Потоковое шифрование. Типы. Гаммирование в потоковом шифровании.

**Особенности:**

1.Операции зашифрования и расшифрования выполняются поразрядно.

2.Каждый символ шифртекста получается в рез-те поразрядной операции сложения по модулю два символа открытого текста и символа ключа

3.Поточный шифратор и дешифратор требует задания *начального значения ключа*

4.Потоковые шифры используются в специальных приложениях и редко обсуждаются

Важнейшее достоинство ПШ перед блочными — высокая скорость шифрования — обеспечивается шифрование практически в реальном масштабе времени.

Классический потоковый шифр – **Шифр Вернама** (*One-time pad* — **схема одноразовых блокнотов,** 1917 г):

Если взять сообщение и ключ (должен быть случайной последовательностью)

Зашифрование - открытый текст объединяется операцией «XOR» с ключом (одноразовым блокнотом или шифроблокнотом).

**Ключ** (**гамма**) должен обладать тремя критически важными свойствами:

* быть истинно случайным (последовательность, полученная с использованием любого алгоритма, является не истинно случайной, а псевдослучайной. Это значит, что они не являются чисто случайными, потому что здесь предсказать можно, т.к. последовательность эта через какой-то период выполняется);
* совпадать по размеру с заданным открытым текстом;
* применяться только один раз.

При соблюдении данных свойств расшифровать сообщение нельзя.

В 1949 году К. Шеннон доказал абсолютную стойкость шифра Вернама - шифр Вернама является самой безопасной криптосистемой из всех возможных.

## Идея гаммирования для ПШ

Классическим примером шифров такого типа – шифр Вернама (или шифр одноразовых блокнотов).

Рандомный генератор — это псевдослучайные числа и предсказать их можно, т.к. через определенный цикл последовательность будет повторяться.

Генератор гаммы выдаёт ключевой поток (гамму): K= k_1,k_2,k_3,\dots ,k_L

Поток битов открытого текста: c_i=m_i\oplus k_i

M = m_i=c_i\oplus k_i

Поток битов шифртекста:

Расшифрование производится операцией XOR между той же самой гаммой и зашифрованным текстом:

Если последовательность битов гаммы не имеет периода и выбирается случайно, то **взломать шифр невозможно.**

Шифр Вернама характеризуется абсолютной криптографической стойкостью.

## Типы потоковых шифров

**1. Синхронные шифры**

* поток гаммы (ключ) генерируется независимо от открытого текста и шифртекста;
* для успешного расшифрования необходимо синхронизировать ключ с шифртекстом;

Когда сообщение передается, оно может быть перехвачено и в него могут вставляться биты и извлекаться. В зависимости от того, какая работает система — синхронная или асинхронная получаются различные результаты. В синхронных системах любого шифртекста искажается только 1 символ в расшифрованном сообщении или искажается несколько символов. Любые вставки или удаления нарушают синхронизацию, это обеспечивает обнаружение факта таких операций. Вместе с тем, такое преимущество одновременно является и недостатком, поскольку потери выпадения символов сразу приводят к возможности расшифрования всего документа.

*Свойства*:

1. Искажение одного символа в шифротексте искажает только один символ в расшифрованном тексте (**+**),
2. Защита от любых вставок и удалений шифртекста, так как они приведут к потере синхронизации и будут обнаружены (**+**)

3. Нарушение синхронизации (добавление или удаление символа) приводит к искажению всех сим-в после потери синхронизации (**-**)

**2.** **Самосинхронизирующиеся**  (***асинхронные****)* (1946 г) –

предусматривают генерацию ключевой информации на основе какого-то числа символов зашифрованного сообщения.

* значение ключа зависит либо от исходного текста, либо 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 (при расшифровании) , приняв **N** битов, автоматически синхронизируется с шифрующим генератором

*Преимущества и недостатки*:

* Так как каждый знак открытого текста влияет на следующий шифротекст, статистические свойства открытого текста распространяются на весь шифротекст (**+**),
* ошибочно удаленный или добавленный символ (бит) вызывает только ограниченное кол-во ошибочных символов в дешифрованном тексте, после чего правильный текст восстанавливается (**+**)
* каждому неправильному биту шифротекста соответствуют **N** ошибок в открытом тексте) (**-**). Ошибки фактически размножаются.

Наибольшей хаотичностью обладают генераторы, которые формируются на основе внутренних флуктуаций полупроводниковых приборов. Полупроводниковый прибор — транзистор или диод. Полупроводниковыми приборами называются приборы, действие которых основано на использовании свойств полупроводниковых материалов.

# Генерация ключевой информации для потокового шифрования. Генераторы ПСП на основе регистров сдвига.

## Генератор ключа (ГК)

**Эффективный ГК** – главная проблема ПШ: генерирование длинных ПСП.

Наиболее частый алгоритм – на основе линейного конгруэнтного генератора; описыв-ся рекуррентным соотн-м:

**xt+1 = (a\*xt + c) mod N,**

**х0**– начальное значение ПСП,

**a** – множитель,

**с** – приращение,

**N** - мощность алфавита

При **с=0** – мультипликативный конгруэнтный ген-р ПСП

*Примеры параметров* для РС с 32-разрядной архитектурой:

**N= 231 -1 = 2 147 483 647, а=16807; 630360016; 10783183814**

**1203248318; 397204094**

*Часто исп-е ГПСП:*

**xt+1 = (1176\*xt + 1476\*xt-1 + 1776\*xt-2) mod 232 - 5,**

**xt+1 = (213 (xt + xt-1 + xt-2) mod 232 - 5,**

**xt+1 = (1995\*xt + 1998\*xt-1 + 2001\*xt-2) mod 232 -849,**

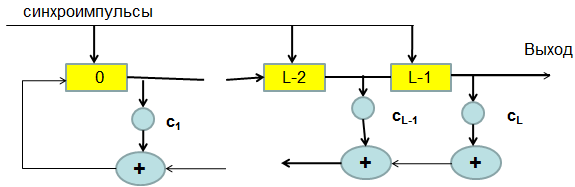
**xt+1 = (219 (xt + xt-1 + xt-2) mod 232 - 1629**

## Генераторы ПСП на основе регистров сдвига

Регистр – связанные цепочкой ячейки памяти.

РС – важнейший структурный компонент ЭЦВМ

РС состоит из **триггеров** из **функции обратных связей (ФОС)**



Выходная послед-ть определяется начальным состоянием

каждого Тг (общее число – **L**: от **0** до **L-1**) и видом ФОС,

**Чаще всего ФОС** – XOR – РСЛОС (регистр сдвига с линейной обратной связью),

**Период регистра сдвига**— длина получаемой последовательности до начала её повторения.

**Важный параметр** – количество триггеров в цепочке, который мы обозначаем символом L. Важно начальное состояние каждого элемента. Если внутреннее состояние ячеек = 0, то мы получим генератор константы 0.

**Свойства:**

1. В течение каждой единицы времени (за такт) выполняются следующие операции:

содержимое ячейки L-1 формирует часть выходной последовательности;

содержимое i-й ячейки перемещается в ячейку i+1   
новое содержимое ячейки 0 определяется битом обратной связи, который вычисляется сложением по модулю с определёнными коэффициентами ci битов ячеек.

1. Так как существует 2L-1 разных ненулевых состояний регистра, то период последовательности, генерируемой РСЛОС при любом ненулевом начальном состоянии, не превышает 2L-1.   
   3. Свойства ПСП зависят от ассоциированного многочлена:

C(x) = 1+c1x+c2x2+ … + cLxL

Его ненулевые коэффициенты называются отводами, (как и соответствующие ячейки регистра, поставляющие значения аргументов функции обратной связи).

* Важное свойство многочлена **C(x)**  - приводимость.
* Многочлен называется *приводимым*, если он может быть представлен как произведение двух многочленов меньших степеней с коэффициентами из данного поля (в нашем случае с двоичными коэффициентами).

Если нет, то многочлен называется *неприводимым*.

* Если многочлен является неприводимым, то период ПСП будет максимально возможным : **2L - 1**

# Особенность шифра Вернама.

**Особенности:**

1.Операции зашифрования и расшифрования выполняются поразрядно.

2.Каждый символ шифртекста получается в рез-те поразрядной операции сложения по модулю два символа открытого текста и символа ключа

3.Поточный шифратор и дешифратор требует задания *начального значения ключа*

4.Потоковые шифры используются в специальных приложениях и редко обсуждаются

Важнейшее достоинство ПШ перед блочными — высокая скорость шифрования — обеспечивается шифрование практически в реальном масштабе времени.

Классический потоковый шифр – **Шифр Вернама** (*One-time pad* — **схема одноразовых блокнотов,** 1917 г):

Если взять сообщение и ключ (должен быть случайной последовательностью)

Зашифрование - открытый текст объединяется операцией «XOR» с ключом (одноразовым блокнотом или шифроблокнотом).

**Ключ** (**гамма**) должен обладать тремя критически важными свойствами:

* быть истинно случайным (последовательность, полученная с использованием любого алгоритма, является не истинно случайной, а псевдослучайной. Это значит, что они не являются чисто случайными, потому что здесь предсказать можно, т.к. последовательность эта через какой-то период выполняется);
* совпадать по размеру с заданным открытым текстом;
* применяться только один раз.

При соблюдении данных свойств расшифровать сообщение нельзя.

В 1949 году К. Шеннон доказал абсолютную стойкость шифра Вернама - шифр Вернама является самой безопасной криптосистемой из всех возможных.

## Идея гаммирования для ПШ

Классическим примером шифров такого типа – шифр Вернама (или шифр одноразовых блокнотов).

Рандомный генератор — это псевдослучайные числа и предсказать их можно, т.к. через определенный цикл последовательность будет повторяться.

Генератор гаммы выдаёт ключевой поток (гамму): K= k_1,k_2,k_3,\dots ,k_L

Поток битов открытого текста: c_i=m_i\oplus k_i

M = m_i=c_i\oplus k_i

Поток битов шифртекста:

Расшифрование производится операцией XOR между той же самой гаммой и зашифрованным текстом:

Если последовательность битов гаммы не имеет периода и выбирается случайно, то **взломать шифр невозможно.**

Шифр Вернама характеризуется абсолютной криптографической стойкостью.

# Особенности использования нейронных сетей в криптографии.

**Нейрокриптография** предполагает криптографические расшифрование/зашифрование информации на основе искусственных нейронных сетей.

**Искусственная нейронная сеть** моделирует работу головногомозга человека. Головной мозг человека состоит из связанных нейронов. Нейроны имеют входы, нейрон, который преобразует информацию, и есть выход, который связан с другими нейронами, управляет мыслительной деятельностью, отвечает за принятые решение и т. д.

**Искусственная нейронная сеть (ИНС)** — математическая модель, а также её программное или аппаратное воплощение, построенное по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

**Применение НС:**

1. **Распознавание образов** (символы текста, изображения, образцы звуков). **Задача классификации**. Можно сфотографировать текст, затем преобразовать его в файл Word.
2. **Принятие решений и управление** (управление роботами).
3. **Кластерный анализ** (разбиение множества входных сигналов на классы, при том, что ни количество, ни признаки классов заранее неизвестны).
4. **Прогнозирование** (предыдущие изменения действительно в какой-то степени предопределяют будущее — котировка акций, погода …).
5. **Оптимизация**.

## Нейрокриптография

Кантер и Кинцель предложили использовать нейронные сети для приложений. Для этого нужно обучить нейронную сеть.

**Нейрокриптография** — раздел криптографии, изучающий применение нейронных сетей, для шифрования и криптоанализа (в общем случае речь идет о применении стохастических алгоритмов)

Основная идея НС (50-60-х годов 20 в): создание искусственного мозга, который позволил бы симулировать некоторые функции человеческого мозга:

* распознавание образцов (букв, форм, сигналов),
* классификация объектов,
* ассоциация.

Мозг человека состоит, главным образом, из большого числа, соединенных друг с другом элементарных нервных клеток, называемых **нейронами.**

Чтобы научить робота что-то делать, можно заложить в него какие-то алгоритмы.

Мы говорили, что в криптографии одной из важных проблем является проблема передачи ключевой информации. Решение проблемы предложили Диффи и Хелман. Они сказали, что эту задачу можно решать с помощью нейронных сетей, которые имеют много входов. Каждый вход имеет разные входные весовые коэффициенты.

Системы обмениваются информацией, чтобы достичь состояние синхронизации (то есть одинаковых значений весовых коэффициентов). Когда такое состояние достигнуто, то передача информации прекращается и искомая последовательность является ключевой информацией.

## Искусственный нейрон

* Впервые модель искусственного нейрона была представлена Маккалоком – Питтсом в 1943 г.:

1 – входы,

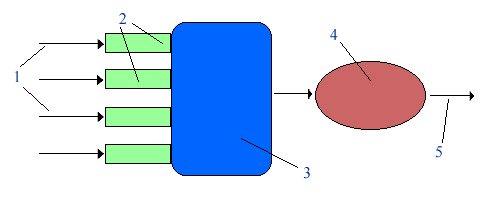
2 – веса,

3 – суммирующий блок,

4 –блок активации,

5 – выход (см рис.)

* эквиваленты: входные значения – дендриты, весовые коэффициенты – синапсы, суммирующий блок – ядро, функция активации – основание аксона, выходное значение – аксон



Математически, работу нейрона можно описать так:

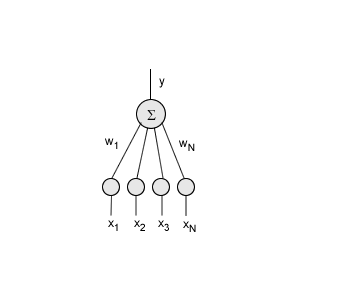
**S = ∑ *xi wi,***

*где* ***xi*** и ***wi*** – соответственно *i*-е входной сигнал и весовой коэффициент нейрона.

В сети (НС) каждый из **m** нейронов получает одинаковый набор входных сигналов (***x1* , *x2 ,* … , *xN*** ) , и каждый из них имеет свой собственный вектор весов (***w1 ,w2 ,…,wN***) (*m = 1, 2, ...,k*).

Система взаимодействующих друг с другом нейронных сетей – **это множество односторонних нейронных сетей, в основе строения которых лежит персептрон (кибернетическая модель элемента мозга)**.

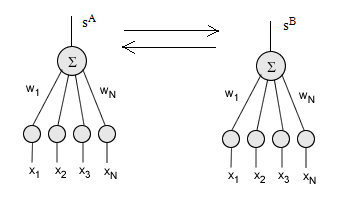
S обозначена y-ком. Внизу входы (x), w – весовые коэффициенты. Выходные и входные значения — это либо 0, либо 1.



## Метод Кантора-Кинцеля

**Идея:** Соединить две НС, которые обмениваются выходной информацией (**У**), «подстраивая» значения своих весовых коэффициентов по определенным правилам (Взаимное обучение сетей).

В режиме синхронизации весовые коэффициенты обеих НС полностью совпадают и могут использоваться сторонами как **ключевая информация**.



Есть две сети, которые имеют одинаковое число входов. Информация может подаваться по открытому каналу, но вот значения весовых коэффициентов изначально задаются случайно. Дальше системы принимают решение и обмениваются информацией. Происходит процесс взаимного обучения. Состояние синхронизации начинается, когда обе сети установили одинаковые значения весовых коэффициентов wi (i изменяется от 1 до n).

Правила взаимного обучения сетей (2 основных):

* правило Хебба
* правило Ойя

## Процесс синхронизации ТРМ

Две TPM (А и В).

1. Задаются случайные (тайные) значения весовых коэффициентов обеих НС

2. Выполняются шаги, пока не наступит синхронизация :

2.1. Генерируют случайный входной вектор X

2.2 Вычисляют значения скрытых нейронов

2.2.Вычисляют значение выходного нейрона

2.3.Сравнивают выходы TPM: А и В

* Выходы разные: переход к п.2.1
* Выходы одинаковые: применяют выбранное правило изменения весовых коэффициентов (Хебба, Ойя и др)

**Это альтернатива методу Диффи-Хелмана**

В нейрокриптографии нейронная сеть может функционировать на основе различных алгебр:

* действительных чисел,
* комплексных чисел,
* октонионов,
* кватеринионов.

В зависимости от алгебры, получается различная криптостойкость.

Чем выше количество используемых чисел, или состояний (от целых чисел до октонионов), тем более криптостойкая система получается.

# Стеганографические методы защиты информации. Классификация и области использования.

Стеганография – это способ передачи или хранения информации с учетом сохранения в тайне самого факта передачи/хранения

Стеганосистема состоит из 4 компонентов: 1. Сообщение, 2. Контейнер (это любая информация, данные, оболочка, помогающая скрыть сообщение), 3. Стеганографический ключ (это секретная информация, известная только законному пользователю, которая определяет конкретный вид алгоритма сокрытия), Ключи в стегосистемах бывают двух типов: закрытые (секретные) и открытые. 4. Канал передачи

**Методы текстовой стреганографии:** 1) Синтаксические, 2) Лингвистические, 3) На основе избыточности

Синтаксические – те, которые не затрагивают синтаксис. Примеры: изменение интервалов между строк, различная длина пробелов между словами, внесение специфических изменений в шрифты, Метод увеличения длины строки (с помощью пробелов), Использование регистра букв, Метод невидимых символов

Лингвистические - те, которые затрагивают синтаксис. Примеры: метод переменной длины слова, метод первой буквы, метод синонимов, мимикрия, спам

На основе избыточности - Метод LSB (Least Significant Bit – наименее значащий бит) основывается на ограниченных способностях зрения или слуха человека, вследствие чего людям тяжело различать незначительные вариации цвета или звука.

Суть метода: заменяем младшие биты в байтах, отвечающих за кодирование цвета. Сначала разбиваем байт секретного сообщения на 4 2-х битовые части **11|00|10|11**. Исходные байты в изображении: 11101100, 01001110, 01111100, 0101100111. Вставляем секретное сообщение: 111011**11**, 010011**00**, 011111**10**, 01011001**11**.

**Методы компьютерной стеганографии:**



**Использование** стеганографических систем является наиболее эффективным при решении проблемы защиты информации с ограниченным доступом. Это означает возможность скрытой передачи информации. Кроме указанного направления, стеганография является одним из перспективных средств для аутентификации и маркировки авторской продукции с целью защиты авторских прав на цифровые объекты от пиратского копирования.

# Текстовая стеганография. Многоключевая модель стеганографической системы.

Стеганографическая система – совокупность средств и методов, которые используются для формирования скрытого канала передачи информации.

Стегосистема образует стегоканал, по которому передается (или в котором хранится) заполненный контейнер. Этот канал считается подверженным воздействиям со стороны нарушителей.

При построении стегосистемы должны учитываться следующие положения:

– противник имеет полное представление о стеганографической системе и деталях ее реализации; единственной информацией, которая остается неизвестной потенциальному противнику, является ключ, с помощью которого только его держатель может установить факт присутствия и содержание скрытого сообщения;

Любая стегосистема должна отвечать следующим требованиям:

– стегосообщение должно быть устойчиво к искажениям

– для сохранения целостности встраиваемого сообщения необходимо использование кода с исправлением ошибки;

– для повышения надежности встраиваемое сообщение должно быть продублировано.

Многоключевая модель информационной системы предполгает интегрированное использование различных методов стеганографии, криптографии для повышения криптостойкости системы.

**Текстовая стеганография**

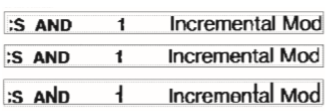
**Виды:**

* Синтаксические — не затрагивают семантику текстового сообщения.
* Лингвистические — трансформируют текстовый файл, сохраняя смысловое содержание текста.

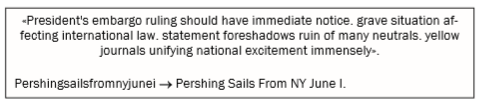
**Синтаксические методы**

**Line-shift coding** (изменение расстояния между строками электронного текста). Его сущность заключается в том, что используется текст с различными межстрочными расстояниями. Очевидным недостатком метода является его низкая эффективность: размер в битах осаждаемой информации не может превысить количество строк в контейнере.

**Feature coding** (внесение специфических изменений в шрифты (начертания отдельных букв)). Этот метод заключается в изменении написания отдельных букв используемого стандартного шрифта (рис. 7.5).



Так, букву «А» можно модифицировать, незначительно укорачивая длинную нижнюю часть буквы. При этом можно закодировать стегосообщение так, что модифицированная буква будет означать «1», а немодифицированная – «0». Модифицировать можно несколько букв.

**Null chipper** (дословно – несуществующий, нулевой лепет). Предполагает размещение тайной информации на установленных позициях слов или в определенных словах текста-контейнера, который, как правило, лишен логического смысла (как видно, действительно лепет). На рис. 7.7 показан пример реализации метода – скрытой информацией являются первые символы слов. 

**Лингвистические методы текстовой стеганографии**

Лингвистическая стеганография занимается скрытым кодированием произвольной информации, представленной в двоичном виде, в текстах.

Необходимо отметить, что осмысленность и внешняя «безобидность» текста должна сохраниться. Одним из наиболее обсуждаемых методов является метод, основанный на системе синонимов языка, используемого для написания электронного текста. Проведенные исследования для случая английского языка показали, что среднее количество синонимов в одном подмножестве синонимов равняется 2,56. Минимальное количество синонимов в одном множестве синонимов равняется 2, а максимальное 13

**метода переменной длины** пользователю, который хочет послать секретное сообщение, необходимо сгенерировать (набрать) текст, в котором слова должны иметь соответствующую длину. Длина этих слов зависит от секретного сообщения и способа кодирования. Обычно одно слово текстаконтейнера определенной длины кодирует два бита информации из стегосообщения. Например, слова текста длиной в 4 и 8 символов могут означать комбинацию бит «00», длиной в 5 и 9 – «01», 6 и 10 – «10», 7 и 11 букв – «11». Слова короче 4 и длиннее 11 букв можно вставлять где угодно для лексической и грамматической связки слов в предложении.

**метода первой буквы** можно передавать еще больше скрытой информации в одном слове: обычно это три или четыре бита. Программа-помощник в этом методе накладывает ограничение уже не на длину слова, а на первую (можно на вторую) букву. Обычно одну и ту же комбинацию могут кодировать несколько букв, например, комбинацию «101» означают слова, начинающиеся с «А», «Г» или «Т».

**Мимикрия** генерирует осмысленный текст, используя синтаксис, описанный в Context Free Grammar (CFG), и встраивает информацию, выбирая из CFG определенные фразы и слова. Грамматика CFG – это один из способов описания языка, который состоит из статических слов и фраз языка, а также узлов. Узлы в простейшем случае представляют собой места в генерируемом тексте, где может быть принято решение, какое слово или фразу дальше необходимо вставить в текст. Мимикрия создает бинарное дерево, которое основано на возможностях CFG, и составляет текст, выбирая те листья дерева, которые кодируют нужный бит.



Под «плотностью заполнения» (третий столбец таблицы) будем понимать отношение стегознаков в пустом контейнере к общему числу символов в пустом контейнере. Например, для метода изменения регистра буквы и метода feature coding стегознаками являются буквы любого алфавита, т. е. пробелы, цифры, специальные знаки и символы не учитываются. А в методах изменения цвета символов, изменения масштаба символов стегознаками являются все символы документа, в том числе специальные знаки и символы.

# Технология блокчейн. Криптовалюта.

# Понятие эллиптической кривой. Принципы построения криптосистемы на эллиптических кривых.

# Представление и описание эллиптической кривой на основе алгебраической геометрии.

# Арифметические операции в эллиптической криптографии.

# Система распределения криптографических ключей на основе эллиптической кривой.

# ЭЦП. Назначение и свойства.

# ЭЦП. Основные методы генерации. Атаки на ЭЦП.

# ЭЦП на основе симметричной криптографии.

# ЭЦП на основе алгоритма RSA.

# ЭЦП на основе симметричной криптосистемы и посредника.

# ЭЦП DSS.

# ЭЦП на основе алгоритма Эль-Гамаля.

# ЭЦП на основе алгоритма RSA и хеш-функции.

# ЭЦП на основе эллиптической кривой.

# Алгоритм К. Шнорра. Стандарт ЭЦП в РБ.

# Защита веб-приложений.