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

**Информационная** (информационно-вычислительная) **система** – организационно упорядоченная совокупность документов, технических средств и информационных технологий, реализующая информационные (информационно-вычислительные) процессы.

**Информационная** (информационно-вычислительная) **система** – организационно упорядоченная совокупность документов, технических средств и информационных технологий, реализующая информационные (информационно-вычислительные) процессы.

**Информационные ресурсы** – отдельные документы или массивы документов в информационных системах.

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

**Безопасность информации** – защищенность информации от нежелательного (для соответствующих субъектов информационных отношений) ее разглашения (нарушения конфиденциальности), искажения (нарушения целостности), утраты или снижения степени доступности информации, а также незаконного ее тиражирования

***Надежность*** - [свойство](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE) [объекта](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82) сохранять во времени в установленных пределах значения всех [параметров](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80), характеризующих способность выполнять требуемые функции в заданных условиях применения, [технического обслуживания](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BE%D0%B1%D1%81%D0%BB%D1%83%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), хранения и [транспортирования](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)

*Надежность* — комплексное свойство, которое в зависимости от назначения объекта и условий его эксплуатации может включать в себя свойства [**безотказности**](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%BE%D1%82%D0%BA%D0%B0%D0%B7%D0%BD%D0%BE%D1%81%D1%82%D1%8C),[**долговечности**](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BB%D0%B3%D0%BE%D0%B2%D0%B5%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C), [**ремонтопригодности**](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BC%D0%BE%D0%BD%D1%82%D0%BE%D0%BF%D1%80%D0%B8%D0%B3%D0%BE%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D1%8C) и [**сохраняемости**](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%85%D1%80%D0%B0%D0%BD%D1%8F%D0%B5%D0%BC%D0%BE%D1%81%D1%82%D1%8C).

**Безотказность** — свойство объекта непрерывно сохранять работоспособное состояние в течение некоторого времени.

**Долговечность** — свойство элемента или системы длительно сохранять [работоспособность](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%BE%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%BD%D0%BE%D1%81%D1%82%D1%8C) до наступления [предельного состояния](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5) при определенных условиях [эксплуатации](https://ru.wikipedia.org/w/index.php?title=%D0%AD%D0%BA%D1%81%D0%BF%D0%BB%D1%83%D0%B0%D1%82%D0%B0%D1%86%D0%B8%D1%8F_(%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)&action=edit&redlink=1).

**Ремонтопригодность** — свойство объекта техники, характеризующее его приспособленность к восстановлению [работоспособного состояния](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%82%D0%BE%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%BD%D0%BE%D1%81%D1%82%D1%8C) после [отказа](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D0%B0%D0%B7) или повреждения.

**Сохраняемость** — свойство объекта непрерывно сохранять требуемые эксплуатационные показатели в течение (и после) срока хранения и транспортировки.

Безопасность любого ресурса информационной системы складывается из обеспечения трех его характеристик: конфиденциальности, целостности и доступности, также могут быть включены другие, такие как аутентичность, подотчетность, надежность; или иначе: информационная безопасность – все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности информации или средств ее обработки:

**конфиденциальность** (англ. confidentiality) компонента системы заключается в том, что он доступен только тем субъектам доступа (пользователям, программам, процессам), которым предоставлены на то соответствующие полномочия;

**целостность** (англ. integrity) компонента предполагает, что он может быть модифицирован только субъектом, имеющим для этого соответствующие права; целостность является гарантией корректности (неизменности, работоспособности) компонента в любой момент времени;

**доступность** (англ. availability) компонента означает, что имеющий соответствующие полномочия субъект может в любое время без особых проблем получить доступ к необходимому компоненту системы (ресурсу).

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 влияют:

1) на программные средства (ПС):

• некорректный исходный алгоритм;

• неправильно запрограммированный исходный алгоритм (первичные ошибки);

2) на аппаратные средства (АС):

• системные ошибки при постановке задачи проектирования;

• отклонения от технологии изготовления комплектующих изделий и АС в целом;

• нарушение режима эксплуатации, вызванное внутренним состоянием АС.

Внешние дестабилизирующие факторы влияют:

1) на программные средства:

• неквалифицированные пользователи;

• несанкционированный доступ к ПС с целью модификации кода;

2) на аппаратные средства:

• внешние климатические условия;

• электромагнитные и ионизирующие помехи;

• перебои в электроснабжении;

• недостаточная квалификация обслуживающего персонала.

# 2. Характеристика угроз безопасности современным ИС и ИВС.

**Информационная** (информационно-вычислительная) **система** – организационно упорядоченная совокупность документов, технических средств и информационных технологий, реализующая информационные (информационно-вычислительные) процессы.

Все многообразие потенциальных угроз безопасности информации по природе их возникновения разделяются на два класса: естественные (объективные) и искусственные (субъективные).

* **Естественные угрозы** – это угрозы, вызванные воздействиями на информационную систему и ее компоненты объективных физических процессов техногенного характера или стихийных природных явлений, независящих от человека.
* **Искусственные угрозы** – это угрозы, вызванные деятельностью человека.

Источники угроз по отношению к самой информационной системе могут быть как внешними, так и внутренними (о чем мы вспоминали выше).

Основные источники угроз безопасности информации можно классифицировать следующим образом:

* **непреднамеренные** (ошибочные, случайные, без злого умысла и корыстных целей) нарушения установленных регламентов сбора, обработки и передачи информации, а также требований безопасности информации и другие действия пользователей ИВС (в том числе сотрудников, отвечающих за обслуживание и администрирование компонентов корпоративной информационной системы), приводящие к непроизводительным затратам времени и ресурсов, разглашению сведений ограниченного распространения, потере ценной информации или нарушению работоспособности компонентов ИВС;
* **преднамеренные** (в корыстных целях, по принуждению третьими лицами, со злым умыслом и т. п.) действия легально допущенных к информационным ресурсам пользователей (в том числе сотрудников, отвечающих за обслуживание и администрирование компонентов корпоративной информационной системы), которые приводят к непроизводительным затратам времени и ресурсов, разглашению сведений ограниченного распространения, потере ценной информации или нарушению работоспособности компонентов информационной системы
  + деятельность преступных групп и формирований, политических и экономических структур, разведок иностранных государств, а также отдельных лиц по добыванию информации, навязыванию ложной информации, нарушению работоспособности ИВС в целом и ее отдельных компонентов;
  + удаленное несанкционированное вмешательство посторонних лиц из территориально удаленных сегментов корпоративной информационной системы и внешних информационно-телекоммуникационных сетей общего пользования (прежде всего сеть Интернет) через легальные и несанкционированные каналы подключения к таким сетям, используя недостатки протоколов обмена, средств защиты и разграничения удаленного доступа к ресурсам;
* ошибки, допущенные при разработке компонентов информационной системы и системы ее защиты, ошибки в программном обеспечении, отказы и сбои технических средств (в том числе средств защиты информации и контроля эффективности защиты).

Отдельно можно отметить «человеческий фактор», классифицировав физических лиц, которые могут получить (а часто и реализуют) несанкционированный доступ к информации. К ним следует отнести:

1) сотрудников организации (учреждения):

• программисты, системные администраторы и даже администраторы информационной безопасности;

• технический персонал;

2) лиц, не являющихся сотрудниками:

• посетители офиса;

• ранее уволенные сотрудники (особенно «обиженные» увольнением);

• хакеры.

Основные факторы (угрозы) ресурсам можно идентифицировать следующим образом:

1) действия внутреннего или внешнего злоумышленника (несанкционированный, в том числе удаленный доступ с целью нарушения работоспособности ИВС, кражи, удаления или модификации информации, несанкционированного распространения материальных носителей за пределами организации);

2) наблюдение за источниками информации;

3) подслушивание конфиденциальных разговоров и акустических сигналов работающих механизмов;

4) перехват электрических, магнитных и электромагнитных полей, электрических сигналов и радиоактивных излучений;

5) разглашение информации компетентными людьми;

6) утеря носителей информации;

7) несанкционированное распространение информации через поля и электрические сигналы, случайно возникшие в аппаратуре;

8) воздействие стихийных сил (наводнения, пожары и т. п.);

9) сбои и отказы в аппаратуре сбора, обработки и передачи информации;

10) отказы системы электроснабжения;

11) воздействие мощных электромагнитных и электрических помех (промышленных и природных).

# 3. Политика безопасности современных ИС и ИВС

**Информационная** (информационно-вычислительная) **система** – организационно упорядоченная совокупность документов, технических средств и информационных технологий, реализующая информационные (информационно-вычислительные) процессы.

Концепция политики информационной безопасности (ИБ) разрабатывается в соответствии с законодательством по информационной безопасности РБ, соответствующими нормативными документами министерства или ведомства, к которому относится организация или учреждение, а также решениями Оперативно-аналитического центра при Президенте Республики Беларусь

Обеспечение ИБ на предприятиях и в учреждениях, как правило, является неотъемлемой частью общей системы управления, необходимой для достижения уставных целей и задач.

Основными разделами концепции информационной безопасности могут быть следующие:

* определение ИБ (или СУИБ);
* структура информационной системы организации (учреждения) и вытекающая из этого структура системы обеспечения информационной безопасности;
* безопасность информации: принципы и стандарты;
* оценка рисков информационным ресурсам в организации (учреждении);
* описание основных механизмов контроля безопасности;
* обязанности и ответственность каждого отдела, управления или департамента, каждого сотрудника в реализации разработанной и утвержденной политики безопасности;
* обязанности лица (администратора безопасности), ответственного за организацию оперативного контроля и управления политикой безопасности;
* ссылки на документы об информационной безопасности, действующие на территории РБ.

Для построения политики информационной безопасности рассматривают следующие направления защиты ИС:

* защита объектов ИС;
* защита процессов, процедур и программ обработки информации;
* защита каналов связи;
* подавление побочных электромагнитных излучений;
* управление системой защиты.

Политика информационной безопасности систем, как и во всех подобных случаях, должна строиться на основе системного подхода, предусматривающего всесторонний анализ причин и угроз безопасности, оценки их последствий, необходимости, экономической или иной целесообразности и адекватности принимаемых противодействий.

Методы и средства защиты можно разделить на три класса:

1. **законодательная и нормативно-правовая база;**
   1. Акты национального законодательства:
      1. международные договоры РБ
      2. Конституция РБ
      3. законы РБ, например Закон Республики Беларусь от 10 ноября 2008 г. № 455-З «Об информации, информатизации и защите информации»;
      4. указы Президента Республики Беларусь;
      5. постановления Правительства Республики Беларусь;
      6. нормативные правовые акты министерств и ведомств;
      7. нормативные правовые акты субъектов, органов местного самоуправления и т. д.
   2. Международные стандарты, например:
      1. BS 7799-1:2005 – Британский стандарт BS 7799 Part 1 – Code of Practice for Information Security Management;
      2. BS 7799-2:2005 – Британский стандарт BS 7799 Part 2 – Information Security Management – Specification for Information Security Management Systems;
      3. ISO/IEC 17799:2005;
      4. ISO/IEC 27001:2005;
      5. ISO/IEC 27002;
      6. ISO/IEC 27005;
      7. SO/IEC 27040:2015.
2. **организационно-технические и режимные меры и методы (политика информационной безопасности);**
   1. организацию охраны, режима, работу с кадрами, с документами;
   2. использование технических средств безопасности (например, простейших дверных замков, магнитных или иных карт и др.), информационно-аналитическую деятельность по выявлению внутренних и внешних угроз.
3. **аппаратные, программно-аппаратные и программные способы и средства обеспечения ИБ.**
   1. средства защиты от несанкционированного доступа:
      1. средства авторизации;
      2. аудит;
   2. системы мониторинга:
      1. системы мониторинга сетей;
      2. анализаторы протоколов;
   3. антивирусные средства:
      1. антивирусные программы;
      2. программные и иные антиспамовые средства;
      3. межсетевые экраны;
   4. криптографические средства:
      1. шифрование данных;
      2. электронная цифровая подпись;
   5. системы бесперебойного питания;
   6. системы аутентификации:
      1. пароль;
      2. ключ доступа (физический или электронный);
      3. биометрия (анализаторы отпечатков пальцев, анализаторы сетчатки глаза, анализаторы голоса, анализаторы геометрии ладони и др.).

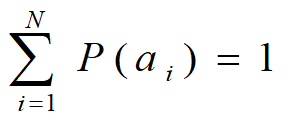
# 4. Энтропия источника сообщения. Энтропия Шеннона.

**Алфавит** - конечная совокупность символов(знаков), с помощью которых можно представить любое сообщение в ИС: А{ai} ai - i-ы символ алфавита

**Мощность алфавита** - количество символов, составляющих алфавит: N(A)

Минимальное число элементов алфавита Nmin = 2, А = {0, 1} – двоичный код. Один дискретный знак представляет собой элементарное сообщение, последовательность знаков – сообщение.

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

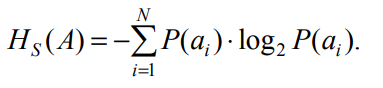


Информационной характеристикой алфавита (источника сообщений на основе этого алфавита) является **энтропия**.

С физической точки зрения энтропия показывает, какое количество информации (бит) приходится в среднем на один символ алфавита.

Этот термин применительно к техническим системам был введен К. Шенноном и Р. Хартли

Энтропию алфавита А = {ai} по К. Шеннону рассчитывают по следующей формуле:



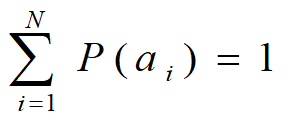
# 5. Энтропия источника сообщения. Энтропия Хартли.

**Алфавит** - конечная совокупность символов(знаков), с помощью которых можно представить любое сообщение в ИС: А{ai} ai - i-ы символ алфавита

**Мощность алфавита** - количество символов, составляющих алфавит: N(A)

Минимальное число элементов алфавита Nmin = 2, А = {0, 1} – двоичный код. Один дискретный знак представляет собой элементарное сообщение, последовательность знаков – сообщение.

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



Информационной характеристикой алфавита (источника сообщений на основе этого алфавита) является **энтропия**.

С физической точки зрения энтропия показывает, какое количество информации (бит) приходится в среднем на один символ алфавита.

Этот термин применительно к техническим системам был введен К. Шенноном и Р. Хартли

Частным случаем энтропии Шеннона считается энтропия Хартли. Дополнительным условием при этом является то, что все вероятности одинаковы и постоянны для всех символов алфавита. С учетом этого формулу (2.1) можно преобразовать к виду:



Сообщение Хk, которое состоит из k символов, должно характеризоваться определенным количеством информации I(Хk):



Здесь Н(А) – энтропия алфавита с соответствующим распределением вероятностей р(аi).

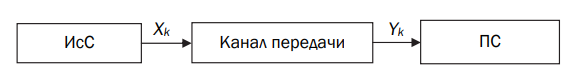
Если принять, что р(аi = 1) = р(1) и р(аi = 0) = р(0), используя выражение (2.1), вычислим энтропию бинарного алфавита:



К примеру, если сообщение Хk состоит только из единиц (Хk = 11…1) и имеет длину k, то вероятность того, что произвольный символ равен единице, составляет единицу (р(аi = 1) = 1), и другая вероятность р(аi = 0) = 0 для i =1,N . Фактически здесь имеет место использование моноалфавита: алфавита, состоящего из одного символа.

# 6. Двоичный канал передачи информации.

Передача информации (данных) осуществляется между двумя абонентами, называемыми источником сообщения (ИcС) и получателем сообщения (ПС). Третьим элементом информационной системы является канал (среда) передачи, связывающий ИсС и ПС.

Отображение сообщения обеспечивается изменением какойлибо физической величины, характеризующей процесс (например, амплитуда, частота, фаза). Эта величина является информационным параметром сигнала (в общем случае – информационной системы). 

Сигналы, как и сообщения, могут быть непрерывными и дискретными. Информационный параметр непрерывного сигнала с течением времени может принимать любые мгновенные значения в определенных пределах. Непрерывный сигнал часто называют аналоговым, а каналы и устройства, функционирующие на основе такого типа сигналов, – аналоговыми.

**Дискретный сигнал** (устройство или канал передачи) характеризуется конечным числом значений информационного параметра.

**Дискретные сообщения** состоят из последовательности дискретных знаков. Часто этот параметр принимает всего два значения (0 или 1). Сообщение или канал его передачи на основе этих двух значений сигнала называют **двоичным** или **бинарным**.

Построение сигнала по определенным правилам, обеспечивающим соответствие между сообщением и сигналом, называют **кодированием**.

**Кодирование в широком смысле** – преобразование сообщения в сигнал.

**Кодирование в узком смысле** – представление исходных знаков, называемых символами, в другом алфавите с меньшим числом знаков. Оно осуществляется с целью повышения надежности и преобразования сигналов к виду, удобному для передачи по каналам связи. Последний тип кодирования относится к так называемой прикладной теории кодирования информации, занимающейся поиском и реализацией методов и средств обнаружения несоответствий (ошибок) между переданным Xk и принятым Yk сообщениям

Оно осуществляется с различными целями: повышение надежности передачи, снижение физического объема сообщения, повышение уровня конфиденциальности или безопасности. Указанные цели соответствуют трем базовым способам преобразования сообщения (Xk) до его передачи по каналу связи:

* кодирование или помехоустойчивое кодирование;
* сжатие или архивирование сообщений;
* криптографическое преобразование10.

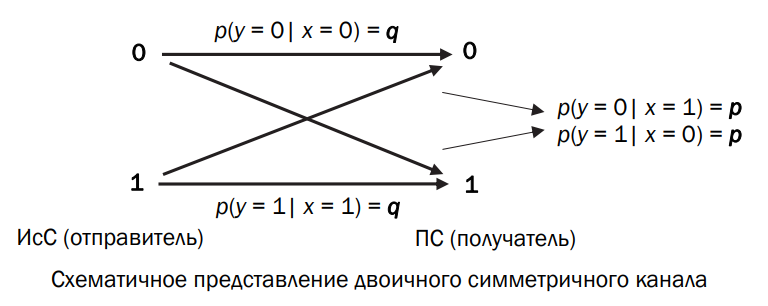
Основные характеристики и параметры двоичных систем.

Важнейшая характеристика источника, получателя или канала – алфавит.

**Алфавит, А** – это общее число знаков или символов (N), используемых для генерации или передачи сообщений. Символы алфавита будем обозначать через {аi}, где 1 ≤ i ≤ N; N – мощность алфавита.

**Двоичный канал** передачи информации строится на основе двоичного алфавита: А = {0, 1}. При этом канал, в котором вероятности искажения переданного 0 (принята соответственно 1; этому событию соответствует условная вероятность р(1|0)) и переданной 1 (принят соответственно 0; этому событию соответствует условная вероятность р(0|1)) равны, как и равны вероятности передачи 0 (р(0)) и 1 (р(1)), называют двоичным симметричным каналом (ДСК).

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



q – вероятность правильной (безошибочной) передачи бита сообщения, р – вероятность передачи бита с ошибкой. Понятно, что p + q = 1.

# 7. Энтропия двоичного алфавита.

Информационной характеристикой алфавита (источника сообщений на основе этого алфавита) является **энтропия**.

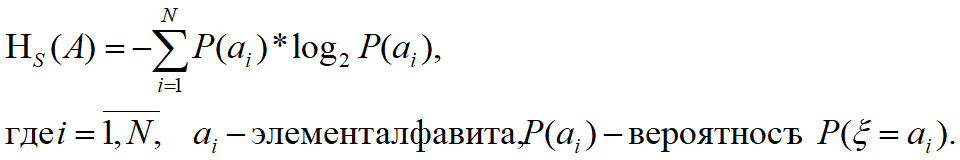
С ***физической точки зрения энтропия*** показывает, какое количество информации (бит) приходится в среднем на один символ алфавита.

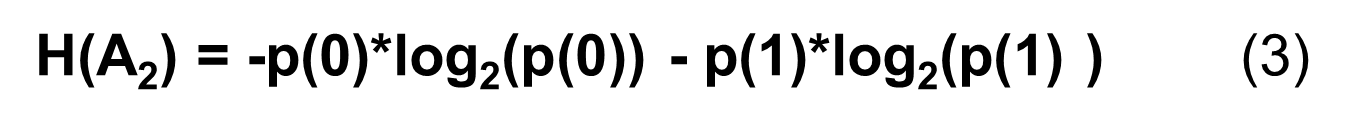
A{0,1} - алфавит

N=2 - мощность алфавита

Вероятность встречи 0 - P(ξ=0)=p(0), 1 - P(ξ=1)=p(1)

Вероятность двоичного алфавита на основе формулы энтропии Шеннона

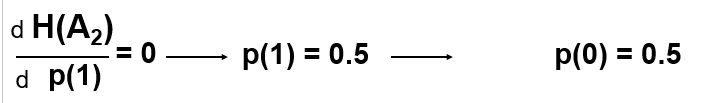




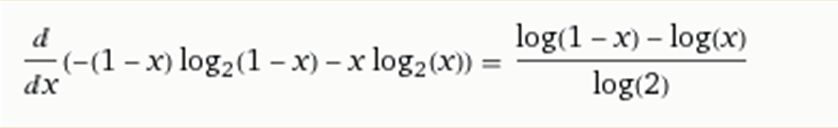
Обозначим p(0) = 1- p(1)

С учетом этого получаем:

**H(A2) = -(1-р(1))\*log2(1-р(1)) - р(1)\*log2(р(1))**



вместо p(1) - в формуле х -



Все это уравнение равно 0 тогда когда log2(1-x) = log2(x)

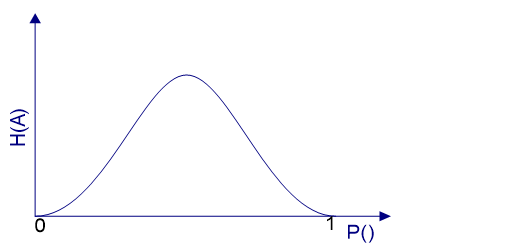
Они равны когда 1-x=x; 1=2\*x; x=0.5 Значит вероятность **р(1) = 0.5, а р(0) = 1 - 0.5 =0.5**

Подставляем значения в формулу (3)

H(A2) = - 0.5 log2 0.5 - 0.5 log2 0.5 = 1 (бит )

По формуле энтропии Хартли: Hc(A) = log2N H(A2) = log22 = 1

График H(A2) = f (p(1), p(0))



# 8. Условная энтропия. Энтропийная оценка потерь при передаче информации.

**Количество информации I** в произвольном сообщении Xk, где k – число символов в сообщении определяется соотношением

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

## ***Энтропийная оценка информации при ее передаче***

Пусть в ИС сообщение Xk = x1, x2, … xi, …. xk на входе канала формируется на основе A = {ai }, i = 1… N

Сообщение на выходе канала ( Yk=y1, y2, … yj, …. yk) формируется на основе того же алфавита : A

При передаче сообщения по каналу могут появляться ошибки.

Обозначим условную вероятность события:

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

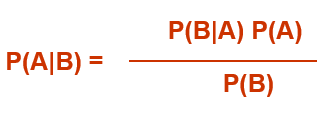
Определим количественно потери информации, вызванные несовершенством ИС, т.е. при р>0

Совместная вероятность случайных событий А и В:

Формула (1)

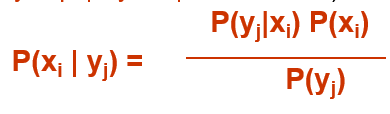
**Р(А,B) = P(A|B) P(B) = P(B|A) P(A) или**

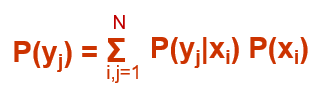
Формула (2)



В соответствии с формулой (2) для ДСК можно записать (используя дискретную форму теоремы Байеса):

формула(3)



где  формула(4)

В общем случае i и j могут принимать различные

значения.

В соответствии с (3) и (4) 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 между переданным и принятым сообщениями.

Эта неоднозначность определяется как условная энтропия (частная) сообщения xi, обусловленная полученным сообщением Y:

формула(5)



В соответствии с (5):

H(Y|x=0)= - P(y=0|x=0) \* log P(y=0|x=0) – P(y=1|y=1) \* log P(y=1|y=1) = **- q log q - p log p**

аналогично:

(Y|x=1)= - P(y=0|x=1) \* log P(y=0|x=1) – P(y=1|x=1) \* log P(y=1|x=1) =  **- p log p - q log q**

**Условной энтропией** Источника дискретного сообщения X в ДСК называем величину

формула(6)

H(Y|X) = - Σ P(xi) H(Y|xi)= Σi P(xi) Σi, j P(yi|xj) \* log P(yi|xj) = P(x=0)\*(- р log2 р – q log2 q) + +P(x=1)\* (- р log2 р – q log2 q) = - р log2 р – q log2 q ‏

так как P(x=0) + P(x=1) = 1

H(Y|X) – энтропия со стороны источника сообщения

H(Y|Х) означает потерю информации на каждый символ переданного сообщения

**Условная энтропия H(X|Y)** источника дискретного сообщения в ДСК рассчитывается на основе формулы(7)

H(X|Y) = P(y=0)H(X|y=0) + P(y=1)H(X|y=1)

H(X|Y) ‏ соответствует энтропии со стороны получателя сообщения

H(X|Y) означает средний объем информации, соответствующей одному из символов сообщения X, относительно принятого сообщения Y

или потерю информации на каждом символе отправленного сообщения

Пример . Пусть известно, что Р(Х=0) = Р(Х=1) =0.5 и р=0.01.

Из (6) определим

H(X|Y) = - р log р - q log q = -0.01 \* log 0.01 – 0.99 \* log 0.99 = = 0.081 бит

Шеннон показал, что эффективная информация на выходе канала относительно входной в расчете на 1 символ (Эфф энтропия алфавита) составляет:

Не = H(X) – H(X|Y)

Для примера He = 0.919 бит

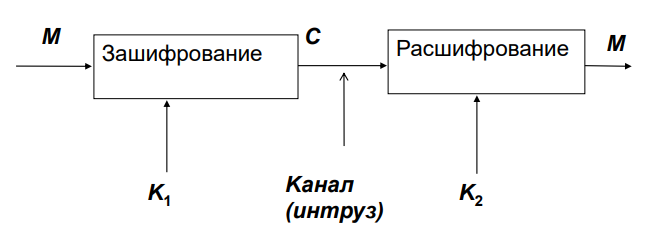
# 9. Базовые понятия криптографии. Основы теории больших чисел. Проблема дискретного логарифма.

**Криптография** (с греч. κρσπτός криптос «тайный"и γράφω графо «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 не известного аналитику ключа

**Криптология** (с греч. κρσπτός и λόγος – логос – «слово")– наука, объединяющая криптографию и криптоанализ

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



**Прямое преобразование** называют шифрованием или зашифрованием, **обратное** - расшифрованием или дешифрованием.

**Исходное сообщение** называется открытым текстом (М, от английского message). **Зашифрованное сообщение** – шифртекстом или шифрограммой

Огюст Керкгоффс - «Военная криптография»

Описал шесть требований, которым должна удовлетворять защищённая система:

1. шифр должен быть физически, если не математически, невскрываемым;
2. система не должна требовать секретности, на случай, если она попадёт в руки врага;
3. ключ должен быть простым, храниться в памяти без записи на бумаге, а также легко изменяемым по желанию корреспондентов;
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.

**M** означает множество сообщений, состоящих из символов определенного алфавита. Элемент из M открытый текст (явный): M=m1 , m2 , ..., m

**C** означает множество сообщений, состоящих из символов того же или иного алфавита. Элемент из С (шифрограмма): C=c1 , c2 , ..., сn

**K** означает множество ключей (элемент этого множества - ключ)

Каждый элемент e K определяет взаимно однозначное отображение (биекцию) M нa C и обозначается Ee (Ee : M → C)

Ee - функция зашифрованиия,

Каждый элемент d K, Dd определяет взаимно однозначное отображение (биекцию) С нa М и обозначается Dd Dd : C → M)

Dd - функция расшифрованиия

Процесс вычисления значения функции Ee для аргумента m M называется зашифрованием явного текста m

Процесс вычисления значения функции Dd для аргумента c C называется расшифрованием криптограммы C

Чтобы сконструировать схему шифрования, нужно выбрать множества M, C, K и определить множества {Ee : e K} i {Dd : d K}

В проблематике современной криптографии можно выделить следующие три типа основных задач:

1. обеспечение конфиденциальности (секретности),
2. обеспечение анонимности (неотслеживаемости),
3. обеспечение аутентификации информации и источника сообщения.

Если известны три некоторых числа (а, х, n), то достаточно легко можно вычислить число y:

Обратная задача: найти х, если известны а, у, n. Эта задача решается гораздо труднее. Ее называют задачей (проблемой) дискретного логарифмирования, по аналогии с вещественными числами, для которых х = loga y

Решения существуют не для всех дискретных логарифмов

Рассматриваемые вычисления относятся к числу так называемых однонаправленных функций. Однонаправленная функция – одно из центральных понятий в асимметричной криптографии. Наглядным примером однонаправленной функции может служить разбиение чашки: разбить чашку на мелкие кусочки достаточно просто, однако очень не просто собрать чашку из кусочков.

# 10. Основная теорема арифметики. Алгоритм Евклида нахождения НОД

Если число не имеет делителей, кроме самого себя и единицы, то оно называется простым, а если у числа есть еще делители, то составным.

Натуральное число n называется простым, если n > 1 и не имеет положительных делителей, отличных от 1 и n.

**Свойство 1.** Любое составное число представляется уникальным образом в виде произведения простых чисел; иначе еще говорят, что разложение числа на простые множители однозначно.

Это свойство вытекает из основной теоремы арифметики.

**Основная теорема арифметики**. Всякое натуральное число n, кроме 1, можно представить как произведение простых множителей:

****

Целое число 39616304 = 2·13·7·2·23·13·2·13·2·7 = = 2·2·2·2·7·7·13·13·13·23.

Для того, чтобы представить относительно небольшое число в виде простых сомножителей, достаточно уметь делить числа столбиком. Однако при этом следует придерживаться некоторых простых правил. Для первого деления нужно выбрать наименьшее простое число большее 1, которое делит исходное число без остатка. Частное от первого деления также нужно разделить с учетом указанных ограничений. Процесс деления продолжаем до тех пор, пока частным не будет 1.

**Свойство 2.** Простых чисел бесконечно много, причем существует примерно n/ln(n) простых чисел, меньших числа n

**Свойство 3.** Наименьший простой делитель составного числа n не превышает √n, поэтому для проверки простоты числа достаточно проверить его делимость на 2 и на все нечетные (а еще лучше простые) числа, не превосходящие √n; как видим, данное свойство коррелирует со свойством 1 собственного делителя.

Сложность решения задачи разложения больших чисел на простые сомножители, известной как «проблема факторизации», определяет криптостойкость некоторых алгоритмов асимметричной криптографии, в частности алгоритма RSA.

**Свойство 4.** Любое четное число, большее 2, представимо в виде суммы двух простых чисел, а любое нечетное, большее 5, представимо в виде суммы трех простых чисел.

**Свойство 5.** Для любого натурального n, большего 1, существует хотя бы одно простое число на интервале от n до 2n.

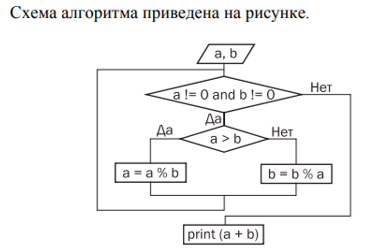
Наибольшее целое число, которое делит без остатка числа a и b, называется наибольшим общим делителем этих чисел – НОД (a, b).

Простым и эффективным средством вычисления НОД (a, b) является алгоритм Евклида. В основе алгоритма лежит определение - Всякое целое число а можно представить с помощью положительного целого числа b равенством вида а = bq + r, 0 ≤ r ≤ b. Число q называется неполным частным, а число r – остатком от деления а на b. В соответствии с этим определением используется цепочка вычислений двумя исходными (начальными) числами а и b: 

При i = 0 в выражении (1.2) аi и bi соответствуют как раз числам а и b. Последний ненулевой остаток (ri, i ≥ 0) соответствует НОД (a, b).

Чтобы найти НОД нескольких чисел (например, a, b, c), достаточно найти НОД двух чисел (например, НОД(a, b) = d) потом НОД полученного (НОД(a, b)) и следующего числа (НОД(c, d) и т. д.

Таким образом, чтобы вычислить НОД k чисел, нужно последовательно вычислить (k–1) НОД. Последнее вычисление дает искомый результат.



# 11. Основы модулярной арифметики. Вычеты.

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

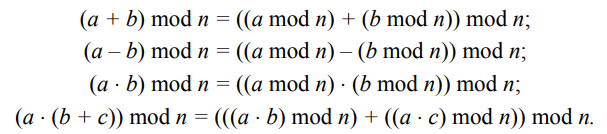
Такая формальная запись читается как «a сравнимо с b по модулю n».

При целочисленном (в том числе и нулевом) результате k деления числа а на число n справедливо a = b + k ⋅ n.

Иногда b называют вычетом по модулю n.

Модулярная арифметика так же коммутативна, ассоциативна и дистрибутивна, как и обычная арифметика

Приведение каждого промежуточного результата по модулю n дает такой же результат, как приведение всего результата вычисления по модулю n:



Модулярная арифметика, как видим, ограничивает диапазон промежуточных и конечного результатов вычислений, т. е. эти вычисления проще организовать и выполнить на компьютере.

Вычисление степени некоторого числа по модулю другого числа представляет собой последовательность операций умножения и деления. Однако существуют методы ускорения таких вычислений.

**Китайская теорема об остатках.** В общем случае, если разложение числа n на простые множители представляет собой p1 ⋅ p2 ⋅ … ⋅ pt (некоторые простые числа могут встречаться несколько раз), то система уравнений



имеет единственное решение: x, меньшее n.

Иными словами, число (меньшее, чем произведение нескольких простых чисел) однозначно определяется своими вычетами по модулю от этих простых чисел.

Китайской теоремой об остатках можно воспользоваться для решения полной системы уравнений в том случае, если известно разложение числа n на простые множители.

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

Взаимно простые числа – это два или несколько целых чисел, наибольший общий делитель которых равен единице. Таким образом, если НОД чисел p и q равен 1, то эти числа называются взаимно простыми.

**Теорема 1.** Целые числа a и b взаимно просты тогда и только тогда, когда существуют такие целые u и v, что выполняется равенство 

**Теорема 2.** Если НОД (a, b) = d , то справедливо следующее соотношение (соотношение Безу): 

Обратное значение числа 4 есть ¼. Это означает, что их произведение должно равняться 1.

В модулярной арифметике обратное значение является понятием более сложным.

Запись a ⋅ х ≡ 1 mod n эквивалентна поиску таких значений х и k, которые удовлетворяли бы тождеству: a ⋅ х = n ⋅ k + 1.

Общая задача вычисления обратного значения по модулю формируется следующим образом: нужно найти такое значение х, которое бы удовлетворяло уравнению: (a ⋅ х) mod n = 1.

Последнюю формулу можно представить в таком виде: 

Число х –1 является обратным значением по модулю n числа а.

Уравнения имеют единственное решение, если числа х и n являются взаимно простыми;

если n – простое число, то любое число от 1 до n – 1 является взаимно простым с n и имеет только одно обратное значение по модулю n.

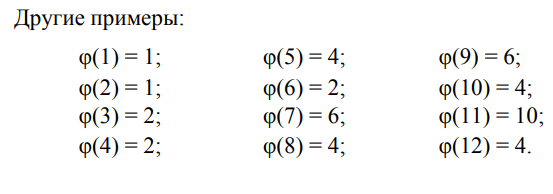
Обратное значение по модулю можно вычислить, воспользовавшись расширенным алгоритмом Евклида.

# 13. Функция Эйлера в криптографии.

Взаимно простые числа – это два или несколько целых чисел, наибольший общий делитель которых равен единице. Таким образом, если НОД чисел p и q равен 1, то эти числа называются взаимно простыми.

Количество натуральных чисел, меньших некоторого числа n и взаимно простых с ним, можно подсчитать на основе известной функции Эйлера, иногда называемой «фи-функцией», φ(n). Например, для числа 24 (n = 24) существует 8 взаимно простых с ним чисел (1, 5, 7, 11, 13, 17, 19, 23), поэтому φ (24) = 8.

Если n – простое число, то 



Любое положительное целое число р может быть выражено с помощью положительных целых чисел, не превосходящих и взаимно простых с каждым делителем числа р.

Например, 6 = 2 ⋅ 3 имеет четыре делителя: 1, 2, 3 и 6. φ(1) + φ(2) + φ(3) + φ(6) = 1 + 1 + 2 + 2 = 6.

Если n = p ⋅ q, то φ(n) = (p – 1) ⋅ (q – 1).

Если числа р и q – взаимно простые, то φ(p ⋅ q) = φ(p) ⋅ φ(q).

С другой стороны, если p и q – очень большие простые числа и известен результат их перемножения (число n), то обратная задача – найти p и q по известному n (задача факторизации) – даже для современных вычислительных средств представляется практически неразрешимой. Эта особенность используется, в частности, в некоторых алгоритмах асимметричной криптогра.

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

**Хеш-функция** – математическая или иная ф., h=H(М) которая принимает на входе строку символов переменной (произвольной) длины и преобразует ее в выходную строку фиксированной (обычно – меньшей) длины

**Хеширование** (или хэширование, англ. hashing ) – это преобразование входного массива данных определенного типа и произвольной длины в выходную битовую строку фиксированной длины.

Преобразования называются хеш-функциями или функциями свертки, а их результаты называют хешем, хеш-кодом, хештаблицей или дайджестом сообщения (анг. message digest)

**Криптографическая хеш-функция** — это специальный класс хеш-функций, который имеет различные свойства, необходимые для криптографии:

* аутентификация (хранение паролей),
* проверка целостности данных,
* защита файлов,
* обнаружение зловредного ПО,
* криптовалютные технологии

**Свойства хеш-функций:**

**Свойство 1:** Детерминированность: независимо от того, сколько раз вычисляется H(M), M - const , при использовании одинакового h всегда получается тот же результат.

**Свойство 2:** Быстрое вычисление H(M): если процесс вычисления не достаточно быстрый, система просто не будет эффективна.

**Свойство 3:** Сложность обратного вычисления: для известного H (М) невозможно (практически) определить М, это свойство односторонности преобразования.

**Свойство 4:** Небольшие изменения в вводимых данных (М) изменяют хеш h(M)

**Свойство 5:** Коллизионная устойчивость (стойкость)

Зная М, трудно определить М’ (М≠ М’ ), для которого H(М)= H(М’) – **коллизия 1-го рода**

Трудно найти два случайных сообщения (М и М’), для которых H(М)= H(М’) - **коллизия 2-го рода**

**Для хеш-функций универсальным методом поиска коллизий является метод, основанный на известной статистической задаче – «парадоксе дня рождения».**

**Парадокс дней рождения** - это кажущееся парадоксальным утверждение, что вероятность совпадения дней рождения (даты) хотя бы у двух членов группы из 23 и более человек, превышает 0,5. Для 60 и более человек вероятность такого совпадения превышает 0,99 (1,0 она достигает, только когда в группе не менее 367 чел.)

**Свойство 6:** Головоломка (для майнеров): для каждого известного h, если k выбран из распределения с высокой мин-энтропией, невозможно найти вводные данные x такие, что H (k|x) = h. (| - конкатенация)

***Опр***. Если некоторое число выбирается из диапазона от 1 до бесконечности, это - высокое распределение мин-энтропии.

При создании хэш-функции используют следующие операции:

* Операция логического суммирования(дизъюнкция)
* Операция логического умножения(конъюнкция)
* Операция отрицания
* Операция логического сдвига на s разрядов
* Операция суммирования по модулю 2

# 15. Общая характеристика алгоритмов хеширования классов MD и SHA.

## MD

Входное сообщение максимальной длины 2^64-1 битов преобразуют в хеш длиной l=128 битов. Исключение 6-я версия длины хеша от 1 до 512 бит

**MD4**

Весь алгоритм состоит из 3-х раундов.

В каждом из раундов выполняется 16 шагов(по числу подблоков).

Каждый шаг вычисляется нелинейную функцию над 3-мя переменными из {a, b, c, d}.

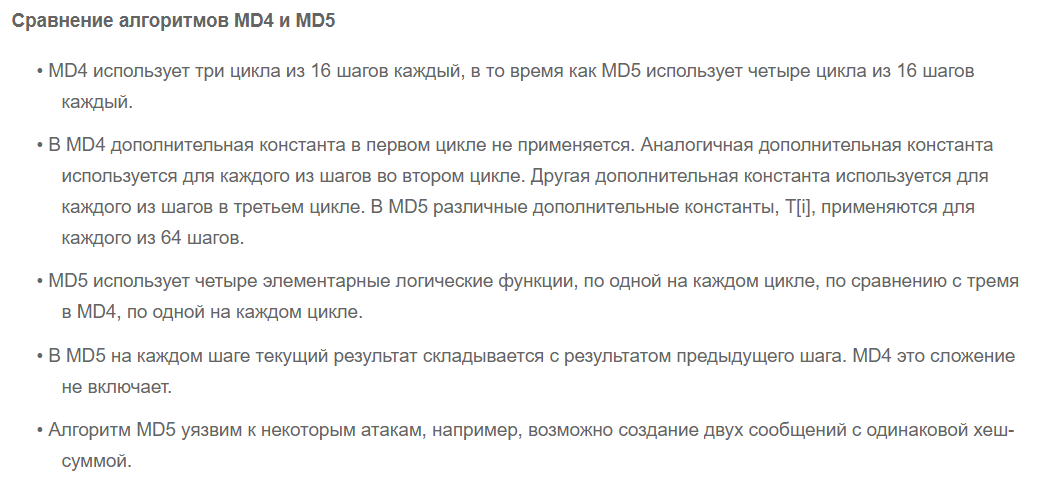
В каждом раунде своя нелинейная функция

**MD5**

добавлен 4-ый раунд, на каждом раунде и шаге используется уникальная константа t

результат каждого раунда добавляется к предыдущему

порядок следования подблоков изменяется в 2 и 3 раундах



## SHA

Максимальный объем хешируемых сообщений такой же что и для MD.

Длина хешей: SHA-1 - 160 битов, SHA-2 - соответствует числу дополняющему через дефис название алгоритма. Максимальная длина SHA-512, 384, 512/256, 512/224 - соответствует 2^128-1 битов

**SHA1**

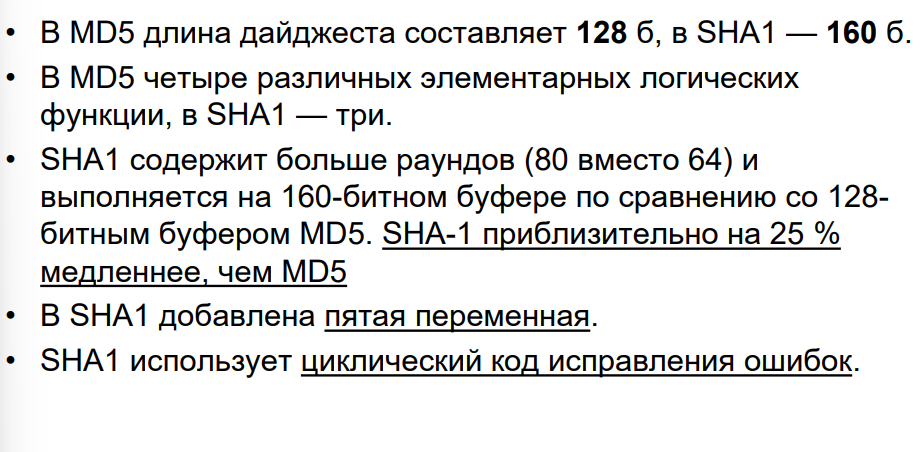
Длина сообщения максимум 2^64-1

Генерирует 160-битное хеш-значение

Четыре этапа

Каждое действие прибавляется к ранее полученному результату

Размер блока обработки равен 512 бит



Оба алгоритма разделяются на 5 стадий:

* расширение входного сообщения
* разбивка расширенного сообщения на блоки
* инициализация начальных констант
* обработка сообщения поблочно
* вывод результата

Входное сообщение «дополняется» (расширяется) так, чтобы его длина (в битах) была конгруэнтной к 448 по модулю 512. Это значит, что сообщение начальной длиной L битов расширяется так, что остаются незаполненными всего лишь 64 бита, чтобы итоговая длина L' была кратной 512. В указанные 64 бита записывается двоичная длина.

Расширение происходит всегда, даже если длина сообщения уже соответствует 448, по модулю 512. Эта операция выполняется следующим образом: один бит «1» добавляется к сообщению, а затем добавляются биты «0», так что длина в битах дополненного сообщения стала конгруэнтной 448 по модулю 512. Добавляется не менее одного бита, но не более 448 битов.

Основой рассматриваемых базовых алгоритмов является модуль, состоящий из циклических преобразований каждого 512-битного блока, который делится на подблоки длиной 32 либо 64 бита (в алгоритмах SHA-512, SHA-384, SHA-512/256, SHA-512/224).

При длине подблока в 16 битов каждый 512-битный блок должен состоять из 32 подблоков.

# 16. Алгоритмы хеширования класса MD. Области использования.

Алгоритм разделяется на 5 стадий:

* расширение входного сообщения
* разбивка расширенного сообщения на блоки(каждый 512-битный блок разделяется на 16 32-разрядных слов)
* инициализация начальных констант (компоненты A, B, C, D или MD-бефер)
* обработка сообщения поблочно(основная процедура алгоритма хеширования)
* вывод результата(конкатенация четырех 32-разрядных слов, начиная с младшего байта регистра А и заканчивая старшим байтом регистра D)

Входное сообщение «дополняется» (расширяется) так, чтобы его длина (в битах) была конгруэнтной к 448 по модулю 512. Это значит, что сообщение начальной длиной L битов расширяется так, что остаются незаполненными всего лишь 64 бита, чтобы итоговая длина L' была кратной 512. В указанные 64 бита записывается двоичная длина.

Расширение происходит всегда, даже если длина сообщения уже соответствует 448, по модулю 512. Эта операция выполняется следующим образом: один бит «1» добавляется к сообщению, а затем добавляются биты «0», так что длина в битах дополненного сообщения стала конгруэнтной 448 по модулю 512. Добавляется не менее одного бита, но не более 448 битов.

MD5

В сравнении с MD4:

добавлен 4-ый р-д с нелинейной функцией

на каждом раунде и шаге используется уникальная константа t

результат каждого раунда добавляется к предыдущему

порядок следования подблоков изменяется во 2-м и в 3-м раундах

Применение:

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

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 программы идут вместе со значением хеша

Пример. Диски для инсталляции.

* Хеширование паролей

# 17. Алгоритмы хеширования класса SHA. Области использования.

Алгоритм разделяется на 5 стадий:

* расширение входного сообщения
* разбивка расширенного сообщения на блоки
* инициализация начальных констант
* обработка сообщения поблочно
* вывод результата

SHA1:

Состоит из четырех этапов

Каждое действие прибавляется к ранее полученному результату

Размер блока обработки равный 512 бит

80 раундову и выполняется на 160-битном буфере по сравнению со 128-битным буфером.

Пять переменных

Циклический код исправления ошибок

Применение:

* ЭЦП
* Системы управления версиями (Version Control System, VCS — 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.
* Для построения кодов аутентификации (процедура проверки подлинности: путем сравнения введенного пароля с паролем в БД пользователей;

# 18. Общая классификация криптографических методов защиты информации.

1. На основе процедуры шифрования
   * **подстановочные** - замена символов открытого текста соответственно символами того же или иного алфавита - подстановка (простой пример шифр Цезаря, Шифр Виженера)
     1. моноалфавитные - шифр однозначной замены или простые подстановочные: шифр Цезаря,
     2. полиграммные шифры - одна подстановка соответствует сразу нескольким подстановкам: шифр Порты, Плейфера и Хилла
     3. Омофонические ширфы - создавались с целью увеличить сложность частотного анализа шифртекстов путем маскировки реальных частот появления символов текста с помощью омофонии
     4. полиалфавитные шифры - состоит из нескольких шифров однозначной замены. Выбор варианта алфавита для зашифрования одного символа зависит от особенностей метода шифрования: Диск Альберти, таблица Трисемуса, шифр Виженера, Роторные машины
   * **перестановочные** - основан на перестановке символов открытого текста
2. На основе генерирования и использования ключа
   * **блочные** - шифрование производится над блоками шифруемого текста - Lucifer, Blowfish, IDEA, ГОСТ 28147-89
   * **поточные** - обрабатывается поток символов по одному - обрабатывают открытый текст побитово - шифр Вернама
3. На основе типа ключа
   * **симметричные** - алгоритмы используют одинаковые значения ключа для зашифрования и расшифрования(K1=K2=K) - алгоритмы DES, 3DES, Lucifer, IDEA, Blowfish, ГОСТ 28147-89

Симметричные криптосистемы называются также криптосистемами с тайным ключом, поскольку значение ключа должно быть известно только отправителю и получателю сообщений.

**Ключ** – секретный параметр, управляющий ходом преобразования. Ключ определяет конкретный вариант преобразования. Ключ используется в обеих операциях: как зашифрования, так и расшифрования. Таким образом, теперь функции зашифрования и расшифрования принимают следующий вид:

функция зашифрования Е: ЕK(М) = С,

функция расшифрования D: DK(C) = M

* + **ассиметричные** - для зашифрования и расшифрования используются открытый и закрытый ключи - RSA, Эль-Гамаля, Диффи-Хелмана, ранцевый алгоритм

Ключи К1 != К2

функция зашифрования Е: ЕK1 (М) = С,

функция расшифрования D: DK2 (C) = M

*Ключи K1 и K2* являются разными, но взаимозависимыми (один из них тайной не является). Поэтому асимметричные криптосистемы называют также криптосистемами с открытым или публичным ключом.

# 19. Подстановочные шифры. Шифр Цезаря.

Сущность **подстановочного шифрования** состоит в том, что исходный текст (из множества М) и зашифрованный текст (из множества С) основаны на использовании одного и того же или разных алфавитов, а тайной или ключевой информацией является алгоритм подстановки.

Подстановочные шифры:

1. ***моноалфавитные*** - замена производится раздельно над каждым одиночным символом сообщения(***шифр Цезаря***(с ключевым словом), ***аффинный шифр цезаря*** на основе уравнения y = ax+b mod N) - низкая криптостойкость
2. ***полиграммные шифры*** - одна подстановка соответствует нескольким символам исходного текста(шифра Порты(АВВА - 003 067))
3. ***омофонические***
4. ***полиалфавитные*** - состоит из нескольких шифров однозначной замены(***Таблица Трисемуса*** - таблица NxN первая стока это алфавит, последующие сдвигаются на один символ, первая буква в первой строке, вторая во второй и т.д; ***Шифр Виженера*** - используется таблица Трисемуса, сообщение шифруется с помощью ключевого слова. Ключ смотрится в верхней строке которая соответствует исходному алфавиту, шифруемое слово в левом столбце соответствующем исходному алфавиту, Если ключ действительно случайный, его длина равна длине сообщения и он использовался единожды, то шифр Виженера теоретически будет невзламываемым)

Шифр Цезаря

Для наглядной демонстрации шифра простой замены достаточно выписать под заданным алфавитом тот же алфавит, но в другом порядке или например, со смещением. Записанный таким образом алфавит называют алфавитом замены.

Максимальное количество ключей для любого шифра этого вида не превышает N!, где N – количество символов в алфавите.

Для математического описания криптографического преобразования предполагаем, что зашифрованная буква ay (ay ∈ Сi), соответствующая символу aх (aх ∈ Мi), находится на позиции (1)

где x, y – индекс (порядковый номер, начиная с 0) символа в используемом алфавите; k – ключ. Для расшифрования сообщения Сi необходимо произвести расчеты, обратные выражению (1), т. е. (2)

Пример. Имеем открытый текст Мi = «cba». На основе шифра Цезаря Сi = «fed». Здесь k = 3, N = 26. Первый символ открытого текста (с) имеет индекс 2 (помним, что начальный символ алфавита (а) имеет нулевой индекс). Значит, первый символ шифртекста (с) будет иметь индекс 2 + k = 5. А такой индекс в алфавите принадлежит символу f и т. д.

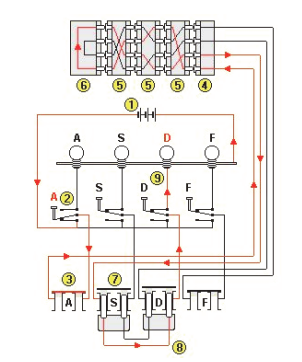
# 20. Особенности реализации шифровальной машины Энигма.

**Машина «Энигма»** – это электромеханическое устройство. Как и другие роторные машины, «Энигма» состоит из комбинации механических и электрических подсистем.

Механическая часть включает в себя клавиатуру, набор вращающихся дисков – роторов, которые расположены вдоль вала и прилегают к нему, и ступенчатого механизма, двигающего один или несколько роторов при каждом нажатии на клавишу.

Электрическая часть, в свою очередь, состоит из электрической схемы, соединяющей между собой клавиатуру, коммутационную панель, лампочки и роторы (для соединения роторов использовались скользящие контакты).

Замкнутую цепь составляют: батарея 1 (это могут быть и иные источники питания), нажатая двунаправленная буквенная клавиша 2, разъем коммутационной панели 3 (как видим, в одном случае – буква «а» – коммутационного перехода на другую букву нет), входной разъем (входное колесо) роторного модуля 4, роторный модуль 5 (состоит из трех роторов, как в версии «Энигмы» для Вермахта, Wehrmacht Enigma М3, или четырех – в версии «Энигмы» для военно-морского флота, Kriegsmarine Enigma M4), рефлектор 6. Последний возвращает ток (цепь) по другому пути через те же узлы, «зажигая» на ламповой панели букву «D», к другому полюсу батареи. Обратим внимание, что обратная часть цепи уже проходит с учетом выполненной коммутации (7 и 8).



*(ЕСЛИ БУДЕТ ВРЕМЯ ОТКРЫТЬ 28 ВОПРОС ТАМ ТОЖЕ ПРО ЭНИГМУ НО НЕМНОГО ДРУГОЕ)*

# 21. Шифр на основе аффинной системы подстановок Цезаря.

**Аффинный шифр** — это частный случай более общего моноалфавитного [шифра подстановки](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%BF%D0%BE%D0%B4%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B8). Поскольку аффинный [шифр](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80) легко дешифровать, он обладает слабыми [криптографическими](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F) свойствами

Применяя одновременно операции сложения и умножения по модулю n над элементами множества (индексами букв алфавита), можно получить систему подстановок, которую называют **аффинной системой подстановок Цезаря**. Определим процедуру зашифрования в такой системе:

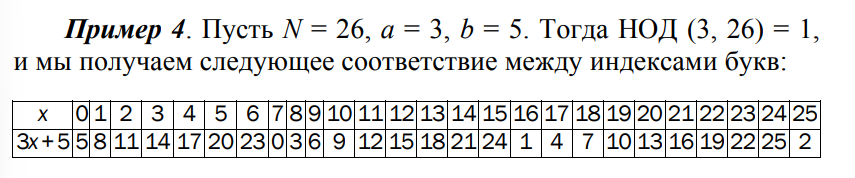
y ≡ ax + b mod N

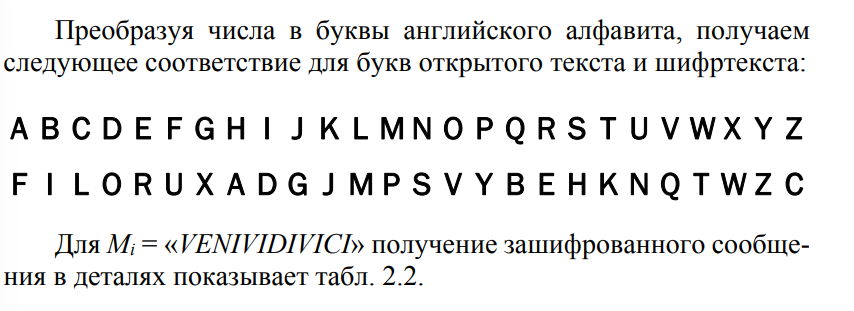
где a и b – целые числа.

При этом взаимно однозначные соответствия между открытым текстом и шифртекстом будут иметь место только при выполнении следующих условий: 0 ≤ a, b < N, наибольший общий делитель (НОД) чисел a, N равен 1, т. е. эти числа являются взаимно простыми.

Расшифрование основано на использовании соотношения x ≡ a^(–1) (y + N – b) mod N,

где a^(–1) – обратное к a число по модулю N, т. е. оно удовлетворяет уравнению аa^(–1) ≡ 1 mod N.







# 22. Система шифрования Цезаря с ключевым словом.

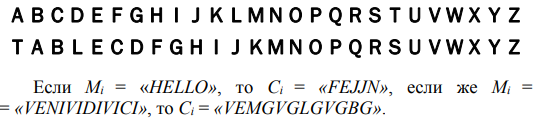
***Система шифрования Цезаря с ключевым словом (лозунгом).***

В качестве ключевого слова необходимо выбирать слово или короткую фразу (не более длины алфавита). Все буквы ключевого слова должны быть различными.

Является одноалфавитной системой подстановки. Особенностью этой системы является использование ключевого слова (лозунга) для смещения и изменения порядка символов в алфавите подстановки (желательно, чтобы все буквы ключевого слова были различными). Ключевое слово пишется в начале алфавита подстановки.

Пример:

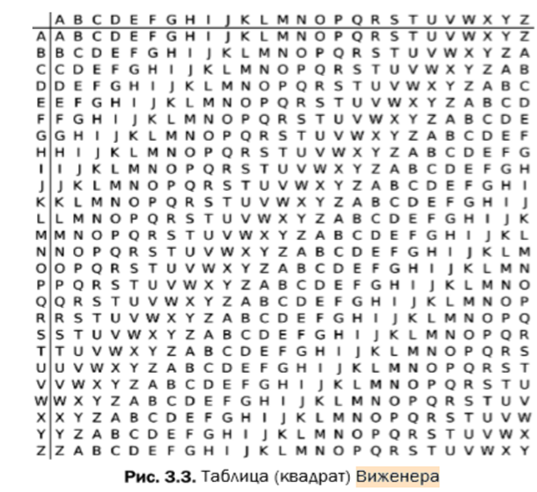
Для шифра с использованием кодового слова «TABLE» исходный алфавит (первая строка) и алфавит подстановки (вторая строка) выглядят следующим образом:



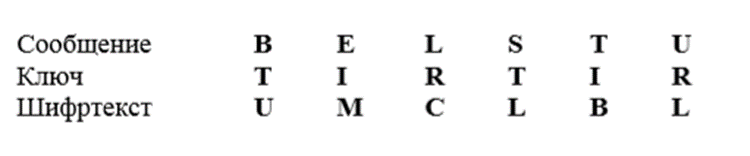
Метод можно видоизменить, если ключевое слово записывать начиная не с первого символа (нулевой индекс) во второй строке, а в соответствии с некоторым числом а: 0 ≤ a < N. Рассмотрим систему на примере.

# 23. Шифр Виженера.

В этом шифре мы имеем дело с последовательностью сдвигов, циклически повторяющейся. Основная идея заключается в следующем. Создается таблица (таблица Виженера) размером N×N (N – число знаков в используемом алфавите). Эти знаки могут включать не только буквы, но и, например, пробел или иные знаки. В первой строке таблицы записывается весь используемый алфавит. Каждая последующая строка получается из предыдущего циклического сдвига последней на 1 символ влево. Таким образом, при мощности алфавита (английского языка), равной 26, необходимо выполнить последовательно 25 сдвигов для формирования всей таблицы.



В этом шифре мы имеем дело с последовательностью сдвигов, циклически повторяющейся. Основная идея заключается в следующем. Создается таблица (таблица Виженера) размером N×N (N – число знаков в используемом алфавите). Эти знаки могут включать не только буквы, но и, например, пробел или иные знаки. В первой строке таблицы записывается весь используемый алфавит. Каждая последующая строка получается из предыдущего циклического сдвига последней на 1 символ влево. Таким образом, при мощности алфавита (английского языка), равной 26, необходимо выполнить последовательно 25 сдвигов для формирования всей таблицы.



# 24. Перестановочные шифры.

Сущность перестановочного шифрования состоит в том, что исходный текст (М) и зашифрованный текст (С) основаны на использовании одного и того же алфавита, а тайной или ключевой информацией является алгоритм перестановки. Шифры перестановки относятся к классу симметричных. Элементами текста могут быть отдельные символы (самый распространенный случай), пары, тройки букв и т.д.

Классическими примерами перестановочных шифров являются анаграммы. Анаграмма (от греч. ana – снова и gramma – запись) – литературный прием, состоящий в перестановке букв (или звуков), что в результате дает другое слово или словосочетание, например: проездной – подрезной, листовка – вокалист, апельсин – спаниель

В классической криптографии шифры перестановки делятся на два подкласса:

* **шифры простой**, или одинарной, перестановки – при зашифровании символы открытого текста Мi перемещаются с исходных позиций в новые (в шифртексте Сi) один раз;
* **шифры сложной**, или множественной, перестановки – при зашифровании символы открытого текста Мi перемещаются с исходных позиций в новые (в шифртексте Сi) несколько раз.

Среди шифров рассматриваемого подкласса иногда выделяют **шифры простой перестановки** (или перестановки без ключа). Символы открытого текста Мi перемешиваются по каким-либо правилам. Формально каждое из таких правил может рассматриваться в качестве ключа.

Пример 1. Простейшим примером является запись открытого текста в обратной последовательности.

В общем случае для использования шифров одинарной перестановки используется таблица, состоящая из двух строк: в первой

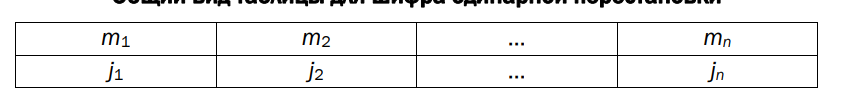
строке записываются буквы, во второй – цифры J. Строки состоят из

n столбцов. Буквы составляют шифруемое сообщение. Цифры J = j1,

j2, …, jn, где j1 – номер позиции в зашифрованном сообщении первого символа открытого текста, j2 – номер позиции в зашифрованном сообщении второго символа открытого текста и т. д. Таким образом, порядок следования цифр определяется используемым

правилом (ключом) перестановки символов открытого текста для

получения шифрограммы. Если предположить, что некоторое сообщение Мi состоит из букв от m1 до mn, то рассматриваемую таблицу можно представить как



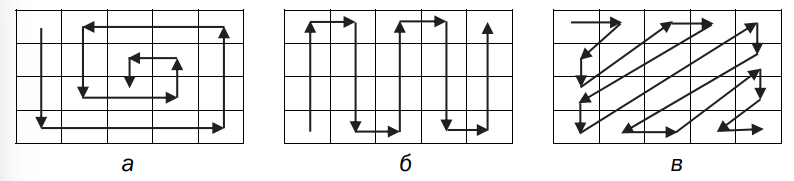
при отсутствии повторяющихся букв в шифруемом сообщении длиной n символов всего существует n! неповторяющихся ключей.

Для расшифрования сообщения, следуя логике рассмотренных процедур зашифрования, нам нужно также составить таблицу, первой строкой которой будет зашифрованный текст.

***Блочные шифры перестановки*** строятся по тем же правилам, что и шифры простой перестановки. Блок должен состоять из 2 или более символов. Если общее число таких символов в сообщении не кратно длине сообщения, то последний блок можно дополнить произвольными знаками.

***Шифры маршрутной перестановки***

Основой современных шифров рассматриваемого типа является геометрическая фигура, обычно прямоугольник или прямоугольная матрица. В ячейки этой фигуры по определенному маршруту (слево направо, сверху вниз или каким-либо иным образом) записывается открытый текст. Для получения шифрограммы нужно записать символы этого сообщения в иной последовательности, т. е. по иному маршруту. Записываем сообщение по строкам, считываем – по столбцам матрицы можно усложнить и считывать не по столбцам, а по спирали (рис. 3.2, а), зигзагом (рис. 3.2, б), змейкой (рис. 3.2, в) или каким-то другим способом. Такие способы шифрования несколько усложняют процесс, однако усиливают криптостойкость шифра.



Маршруты могут быть значительно более изощренными. Например, обход конем шахматной доски таким образом, чтобы в каждой клетке конь побывал один раз.

***Шифр вертикальной перестановки***

Данный шифр является разновидностью шифра маршрутной перестановки. К особенностям вертикального шифра можно отнести следующие:

• количество столбцов в таблице фиксируется и определяется длиной ключа;

• маршрут вписывания: слево направо, сверху вниз;

• шифрограмма выписывается по столбцам в соответствии с их нумерацией (ключом).

***Шифры множественной перестановки***

Особенностью шифров данного подкласса является минимум двукратная перестановка символов шифруемого сообщения. В простейшем случае это может задаваться перемешиванием не только столбцов (как в примере 4), но и строк. Таким образом, этот случай соответствует использованию двух основных ключей: длина одного из них равна числу столбцов, другого – числу строк. К ключевой информацию мы можем относить также способы вписывания сообщения и считывания отдельных символов из текущего столбца матрицы.

# 25. Методы симметричного криптопреобразования. Стандарт DES. Общая характеристика.

В симметричных системах отправитель и получатель используют один и тот же ключ, который должен быть известен только им.

Симметричные алгоритмы подразделяются на два подкласса. Одни алгоритмы обрабатывают открытый текст побитово (иногда побайтово). Такие алгоритмы называют потоковыми. Другие алгоритмы обрабатывают группы (блоки) битов открытого текста. Эти алгоритмы называют блочными. Примеры симметричных алгоритмов: DES, ГОСТ 28147-89, Lucifer, Blowfish.

Алгоритм DES (Data Encryption Standard), как и другие симметричные (и асимметричные) алгоритмы, использует множественные арифметическо-логические преобразования исходного текста.

Стандарт шифрования DES был разработан в 1970-х годах Национальным институтом стандартизации США (ANSI) и называется алгоритмом DEA (Data Encryption Algorithm).

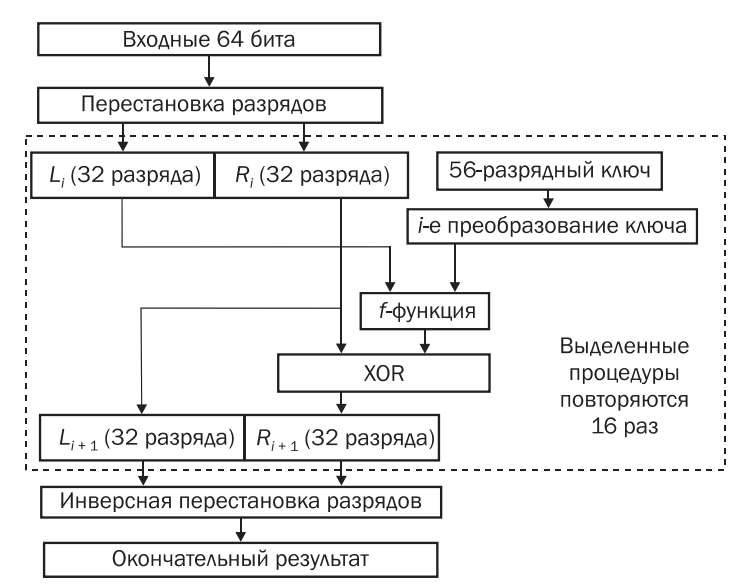
DEA является блочным алгоритмом и оперирует с блоками данных размером 64 бита. При этом используется ключ длиной 56 битов; дополнительно к ним вычисляются 8 битов четности: 8-й бит в каждом из 8 байтов ключа. Такая длина ключа соответствует 1017 комбинаций, что обеспечивало до недавнего времени достаточный уровень безопасности

Входной блок состоит из 64 бит, преобразуется в выходной блок идентичной длины.

В алгоритме применяется подстановка и перестановка битов текста. Комбинация этих методов образует раунд или цикл.

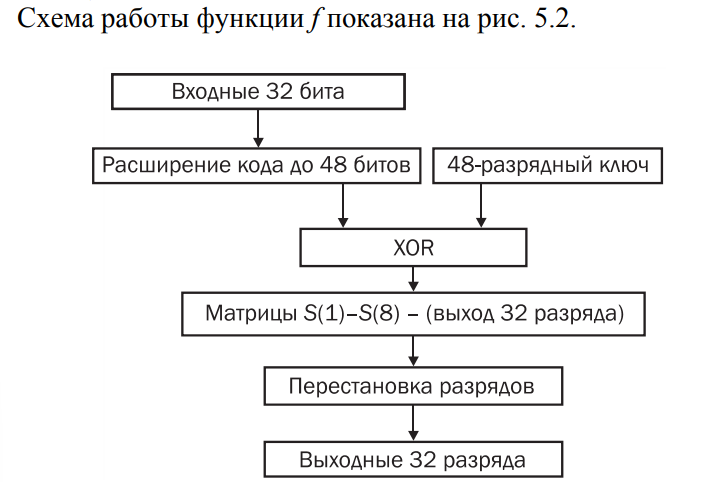
Один блок данных подвергается преобразованию в течении 16 раундов

После первоначальной перестановки и разделения 64-битного блока данных на правый(R) и левый(L) подблоки длиной по 32-бита выполняются 16 раундов одинаквых действий



*Начальная и конечная перестановка не влияет на криптостойкость алгоритма, поэтому часто в программной реализации ее не используют.*

Функция f оперирует с 32-разрядными словами исходного текста и использует в качестве параметра 48-разрядный ключ (в каждом из фиксированного множества преобразований биты ключа сдвигаются и затем из 56 битов ключа выбираются 48)



Расширяющая перестановка решает две задачи: 1) приведение размера правой половины блока данных в соответствие с размером ключа; 2) получение более длинного результата (длину позднее можно изменить).

Основной криптографический смысл такого преобразования (удлинения) состоит в том, что увеличивается влияние одного бита сообщения на общий результат. Это явление называется лавинным эффектом. Лавинный эффект повышает криптостойкость системы. Алгоритм задуман так, чтобы добиться большего влияния каждого бита открытого текста на шифртекст.

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

Всего используется 8 S-матриц (или S-блоков). Каждый такой блок имеет 6 входов и 4 выхода, т. е. на его выходе формируется 4-битное слово, значение которого определяется входным 6-битным словом

(S-блок размер 4 строки X 16 столбцов)

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

Стандарт шифрования DES был разработан в 1970-х годах Национальным институтом стандартизации США (ANSI) и называется алгоритмом DEA (Data Encryption Algorithm).

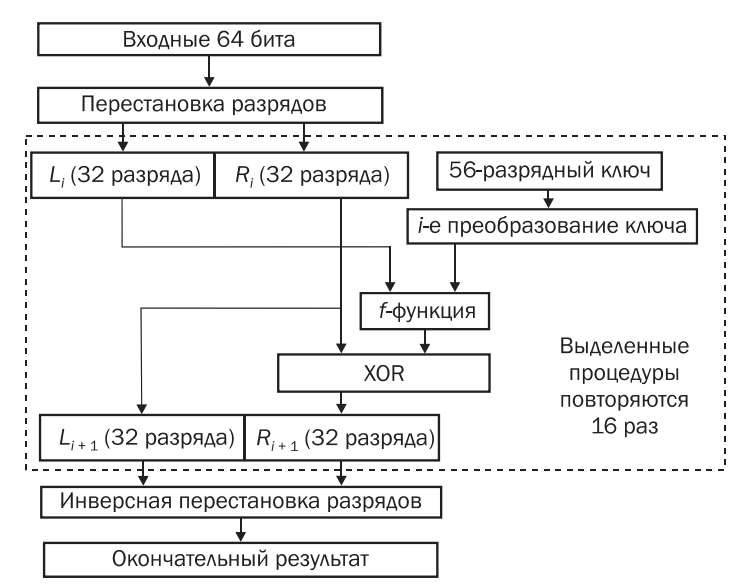
DEA является блочным алгоритмом и оперирует с блоками данных размером 64 бита. При этом используется ключ длиной 56 битов; дополнительно к ним вычисляются 8 битов четности: 8-й бит в каждом из 8 байтов ключа. Такая длина ключа соответствует 1017 комбинаций, что обеспечивало до недавнего времени достаточный уровень безопасности

Входной блок состоит из 64 бит, преобразуется в выходной блок идентичной длины.

В алгоритме применяется подстановка и перестановка битов текста. Комбинация этих методов образует раунд или цикл.

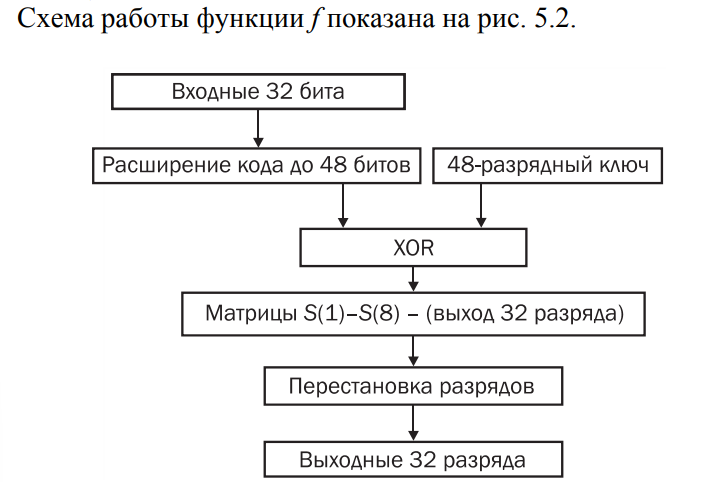
Один блок данных подвергается преобразованию в течении 16 раундов

После первоначальной перестановки и разделения 64-битного блока данных на правый(R) и левый(L) подблоки длиной по 32-бита выполняются 16 раундов одинаковых действий



*Начальная и конечная перестановка не влияет на криптостойкость алгоритма, поэтому часто в программной реализации ее не используют.*

Функция f оперирует с 32-разрядными словами исходного текста и использует в качестве параметра 48-разрядный ключ (в каждом из фиксированного множества преобразований биты ключа сдвигаются и затем из 56 битов ключа выбираются 48)



Расширяющая перестановка решает две задачи: 1) приведение размера правой половины блока данных в соответствие с размером ключа; 2) получение более длинного результата (длину позднее можно изменить).

Основной криптографический смысл такого преобразования (удлинения) состоит в том, что увеличивается влияние одного бита сообщения на общий результат. Это явление называется лавинным эффектом. Лавинный эффект повышает криптостойкость системы. Алгоритм задуман так, чтобы добиться большего влияния каждого бита открытого текста на шифртекст.

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

Всего используется 8 S-матриц (или S-блоков). Каждый такой блок имеет 6 входов и 4 выхода, т. е. на его выходе формируется 4-битное слово, значение которого определяется входным 6-битным словом

(S-блок размер 4 строки X 16 столбцов)

Преобразование в S-блоке:

* на вход поступает 6-битное “слово” - например 110010 - крайние биты указывают на строку в матрице 4х16 - центральные на столбец - на пересечении столбца и строки стоит число - этим числом и заменяется то что мы получили на входе таким образом получаем “слово” длинной 4 бита и в результате получим блок длиной 32-бита, а не 48 как до преобразования.

***Преобразования ключей.***

Первоначальный 64-битный ключ сокращается до 56-битного путем отбрасывания битов четности.

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 2 бита.

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.

Криптостойкость:

Этот важнейший параметр алгоритма определяется не только ***ключом***, но и ***преобразованием блока открытого текста или шифртекста с помощью S-матриц***.

Что касается ключа и его преобразований, алгоритму свойственна так называемая проблема слабых и полуслабых ключей. Т.к. 56-битный ключ делится пополам. Если все биты каждой половины равны 0 или 1, то во всех раундах будет использоваться одинаковый ключ - слабый ключ. Существуют пары различных ключей, которые при зашифровании превращают открытые тексты в одинаковый шифртекст - полуслабые ключи.

***Дифференциальный криптоанализ***

Криптоаналитик работает с парами шифртекстов, открытые тексты которых имеют некоторые отличия (разности; под разностью здесь понимается операция XOR). Метод предполагает анализ эволюции этой разности в процессе прохождения открытых текстов через раунды алгоритма с использованием одного и того же ключа.

***Линейный криптоанализ***

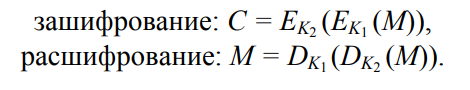
Если выполнить операцию XOR над некоторой частью открытого текста, затем над некоторой частью шифртекста, а затем над полученными результатами, то получается некая последовательность, которая представляет собой XOR некоторых битов ключа. Используя фрагменты открытых текстов и соответствующих им шифртектов, можно найти ключ.

К началу 1990-х годов алгоритм был отнесен к числу обеспечивающих низкую криптостойкость.

# 27. Методы симметричного криптопреобразования. Стандарты 3DES. Реализация и криптостойкость.

Одним из очевидных достоинств алгоритма DES является сравнительно высокая скорость реализации. Это связано, в основном, с малым размером ключа, т. е. преимущество переходит в недостаток и наоборот.

***Одно из направлений совершенствования алгоритма*** – использование базового DES в качестве компонента другого алгоритма. Таким другим алгоритмом может быть тот же DES. Наиболее простой метод объединения или наращивания DES состоит в использовании двух алгоритмов с различными ключами:



Криптостойкость такого преобразования значительно возрастает. Если для 56-битного ключа нужно выполнить максимально 256 попыток лобовой атаки и при этом требуется хранить в памяти компьютера 256 64-битных блоков, или 1017 байт, то при двойном DES число попыток взлома возрастает до 256 , а объем требуемой памяти – до 2115, или примерно до 5 ⋅ 1033 байт.

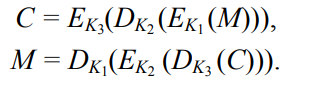
***Следующий вариант*** объединения предложен У. Тачменом. Он предусматривает выполнение формально трех DES (тройной DES или 3DES). Один из вариантов такого подхода предполагает следующие операции: отправитель сначала шифрует сообщение первым ключом, зачем расшифровывает результат вторым ключом и, наконец, опять шифрует первым ключом:

****

получатель соответственно расшифровывает сообщение первым ключом, зашифровывает вторым, расшифровывает первым:



Еще большей криптостойкостью обладает 3DES при использовании трех различных ключей:



Существует 3 типа алгоритма 3DES:

1. DES-EEE3 - шифрование 3 раза с 3 разными ключами(шифрование-шифрование-шифрование)
2. DES-EDE3 - операция шифрование-расшифрование-шифрование с разными ключами
3. DES-EEE2 и DES-EDE2 - шифрование-расшифрование-шифрование на первом и третьем этапах применяется один и тот же ключ

Стандарт 3DES длина ключа 192(64\*3) бита; т.к. в каждом байте используется только 7 битов, на самом деле длина ключа 168(56\*3) бит

Криптостойкость:

* Из-за атак “встреча по середине” (известно M и C нужно найти К) - эффективная криптостойкость составляет только 112 бит
* В DES-EDE в котором K1=K3 - эффективный ключ имеет длину 80 бит
* для успешной атаки на 3DES потребуется около 2^32 бит известного открытого текста, 2^113 шагов, 2^90 циклов DES-шифрования и 2^88 бит памяти

# 28. Шифровальная машина Энигма. Устройство, функционирование, криптостойкость.

**Машина «Энигма»** – это электромеханическое устройство. Как и другие роторные машины, «Энигма» состоит из комбинации механических и электрических подсистем.

Чтобы оценить криптостойкость шифра, нужно учитывать все возможные настройки машины. Для этого необходимо рассмотреть следующие свойства «Энигмы»:

• выбор и порядок роторов;

• разводку (коммутацию) роторов;

• настройку колец на каждом из роторов;

• начальное положение роторов в начале сообщения; •

отражатель;

• настройки коммутационной панели.

Чтобы выбрать 3 ротора из возможных 5, существует 60 комбинаций (5 · 4 · 3). Каждый ротор (его внутренняя проводка) может быть установлен в любом из 26 положений. Следовательно, с 3 роторами имеется 17 576 различных положений ротора (26 · 26 · 26). Кольцо на каждом роторе содержит маркировку ротора (что здесь неважно) и выемку, которая влияет на шаг перемещения расположенного левее ротора. Каждое кольцо может быть установлено в любом из 26 положений. Поскольку слева от третьего (наиболее левого) ротора нет ротора, на расчет влияют только кольца самого правого и среднего ротора. Это дает 676 комбинаций колец (26 · 26).

Коммутационная панель обеспечивает самый большой набор возможных настроек. Для первого кабеля одна сторона может иметь любое из 26 положений, а другая сторона – любое из 25 оставшихся положений, мы должны игнорировать все двойные числа во всех возможных комбинациях для одного кабеля, предоставляя (26 · 25) / (1! · 21 ), или 325 уникальных способов коммутаций одним кабелем. С использованием 10 кабелей на коммутационной панели получаются 150 738 274 937 250 различных комбинаций. Формула, где n равно количеству кабелей, равна 26! / (26 – 2n)! · n! · 2n . Численно это дает: 60 · 17 576 · 676 · 150 738 · 274 · 937 · 250 = = 107 458 687 327 250 619 360 000, или 1,07 · 1023.

Таким образом, практически рассматриваемая версия «Энигмы» (3 ротора с выбором из 5 роторов, отражатель В и 10 штекерных кабелей для коммутационной панели) может быть настроена на 1,07 · 1023 различных состояний, что сопоставимо с 77-битным криптографическим ключом.

**Зашифрование сообщения.**

1. Установить начальную стартовую позицию роторов (предположим, их 3) согласно текущей кодовой таблице (коду дня), например WZA.

2. Выбрать случайный ключ сообщения, например SXT. Затем оператор устанавливал роторы в стартовую позицию WZA.

3. Зашифровать ключ сообщения SXT. Предположим, что в результате зашифрования ключа получится UHL.

4. Далее оператор ставил ключ сообщения (SXT) как начальную позицию роторов и зашифровывал собственно сообщение. После этого он отправлял стартовую позицию (WZA) и зашифрованный ключ (UHL) вместе с сообщением.

Расшифрование сообщения.

1. Установить стартовые позиции роторов в соответствии с первой трехграммой (WZA).

2. Расшифровать вторую треграмму (UHL) и извлечь исходный ключ (SXT).

3. Далее получатель использовал этот ключ как стартовую позицию для расшифрования шифртекста. Обычно срок действия ключей составлял одни сутки.

«Энигма» строится на основе подстановочных шифров, подобных шифру Цезаря, в котором, как известно, ключ сообщения, который должен знать получатель, – это просто смещение между двумя алфавитами. Принято считать, что в основе шифра «Энигмы» лежит динамический шифр Цезаря.

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

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

**Алгоритм Lucifer**.

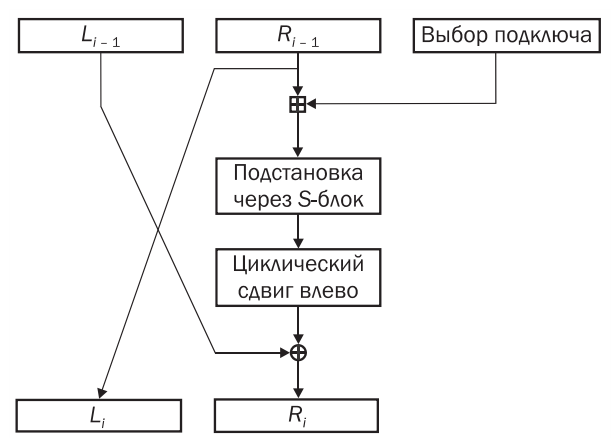
Алгоритм представляет собой последовательность перестановок и подстановок. Основные блоки напоминают алгоритм DES. S-блоки алгоритма Lucifer имеют 4-битные входы и выходы. Причем для выбора одного из двух возможных блоков используется бит ключа.

**Алгоритм IDEA**

Работает с 64-битными блоками данных и 128-битными ключами. Является подстановочно-перестановочным алгоритмом. Блок данных делится на четыре 16-битных блока, являющиеся входами первого раунда. Весь алгоритм состоит из 8 раундов. В каждом раунде 4 блока подвергаются операциям XOR, сложениям и умножениям друг с другом и шестью 16-битными подключами. Между раундами второй и третий блоки данных меняются местами. В конечной операции четыре 16-битных блока данных объединяются с 16-битными подключами. Программная реализация IDEA выполняется примерно вдвое быстрее, чем реализация DES.

**Алгоритм ГОСТ 28147-89**

Разработан в Советском Союзе. Работает с 64-битными блоками данных и 256-битными ключами. Алгоритм состоит из 32 раундов. Как и в DES, блок данных разбивается на левую (L) и правую (R) части. Один раунд алгоритма показан на рисунке

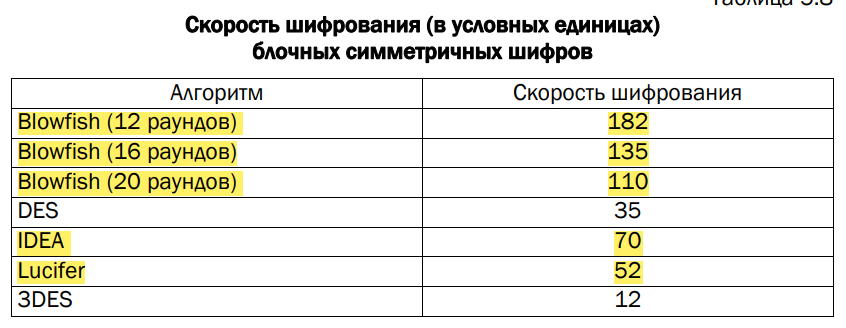


Алгоритм предусматривает использование 8 S-блоков. 256-битный ключ разбивается на восемь 32-битных подключей, каждый из которых используется в одном раунде. Расшифрование выполняется в обратной последовательности. S-блоки, как и в алгоритме Lucifer, имеют по 4 входа и 4 выхода.

В алгоритме отсутствует расширяющая перестановка в сравнении с DES. Это уменьшает лавинный эффект и, соответственно, снижает криптостойкость.

**Алгоритм Blowfish**

Алгоритм не запатентован и его программная реализация общедоступна. Алгоритм создан для линий связи, где не предусмотрена частая смена ключа. Работает с 64-битными блоками данных. Предусмотрена возможность расширения ключа до 448 битов. Алгоритм состоит обычно из 16 раундов (может быть 12 или 20), в каждом из которых выполняются операции подстановки и перестановки, зависящие от ключа. Используются только сложения и XOR над 32-битными словами. Каждый из 4 S-блоков хранит по 256 чисел.



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

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

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

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

Алгоритмы с открытым ключом можно использовать для решения:

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

По мнению Диффи-Хеллмана алгоритм шифрования должен:

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

***Криптоалгоритм на основе задачи об укладке ранца***

Ранцевый вектор S = (s1, …., sz) - это упорядоченный набор из z, z>=3, различных натуральных чисел si. Входом задачи называют пару (S’, S), где S’ - рюкзачный вектор, а S - натуральное число.

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

В наиболее известном варианте задачи о ранце требуется выяснить, обладает или нет данный вход (S’, S) решением. В варианте, используемом в криптографии, нужно для данного входа (S’, S) построить решение, зная, что такое решение существует. Оба эти варианта являются NP-полным.

Дано множество предметов общим числом z различного веса. Спрашивается, можно ли положить некоторые из этих предметов в ранец так, чтобы его вес стал равен определенному значению S? Более формально задача формулируется так: дан набор значений k1, k2, …., kx и суммарное значение S. Требуется вычислить значения bz такие что S = b1k1 + b2k2 + … + bzkz (bi может быть либо нулем, либо единицей, 1 - значит предмет кладут в рюкзак, 0 - не кладут)

Суть метода для шифрования состоит в том, что существуют две различные задачи укладки ранца: одна из них решается легко и характеризуется линейным ростом трудоемкости, а другая решается трудно. Легкий для укладки ранец можно трансформировать в трудный.

**Трудный** для укладки ранец применяется в качестве ***открытого ключа***, который легко использовать для ***зашифрования***, но ***невозможно*** – для ***расшифрования***.

В качестве ***закрытого ключа*** применяется **легкий** для укладки ранец, который предоставляет простой способ ***расшифрования*** сообщения.

В качестве закрытого ключа ***d***(легкого для укладки ранца) используется сверх возрастающая последовательность, состоящая из z элементов: d1, d2, …, dz **d**={di}, i=1, …, z

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

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

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

*Пример*

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

S = 270

270 = 210 + 52 + 6 + 2

тайный ключ 10100101

***Открытый ключ формируется на основе тайного:***

ei = di\*a mod n

a - некоторое число

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

*Продолжение примера*

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

e1 = 2\*31 mod 420 = 62 и т.д.

***Зашифрование сообщения***

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

*Продолжение пример*

Сообщение из 7 букв представляем в бинарном виде

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

Первый символ в бинарном виде 11010000

Укладка ранца 62+93+403

Вес ранца 558

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

***Расшифрование сообщения***

Вычисляем a^-1

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

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

Si = cia^-1 mod n

По итогу этих вычислений полученные значения будут считаться весом ранца

*Продолжение примера*

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

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

n =420, a =31

С = 155 365 558 155 924 1239 470

S1 = 155 \* 271 mod 420 = 5

используя d получаем 11000000

и этой последовательности соответствует символ алфавита в используемой таблице кодировки.

***Безопасность криптоалгоритма на основе задачи об укладке ранца***

Криптостойкость во многом определяется скоростью поиска нужного варианта укладки ранца.

Ранцевые криптосистемы не являются криптостойкими.

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

# 31. Управление криптографическими ключами. Алгоритм рукопожатия.

**Управление ключами** – информационный процесс, включающий реализацию следующих основных функций:

* генерация ключей
* хранение ключей
* распределение ключей

Безопасность любого криптографического алгоритма определяется используемым криптографическим ключом. Криптографические ключи должны иметь достаточную длину и случайные значения битов.

**Генерация ключей**. В реальных системах используются специальные аппаратные и программные методы генерации случайных ключей. Как правило используют датчики случайных чисел. Однако степень случайности их генерации должна быть достаточно высокой. Идеальными генераторами являются устройства на основе “натуральных” случайных процессов (радиоактивный распад, белый шум)

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

**Распределение ключей** – обмен ключами между сторонами

***Протокол рукопожатия***

SSL клиент и сервер договариваются об установлении связи с помощью процедуры рукопожатия. Во время рукопожатия клиент и сервер договариваются о различных параметрах, которые будут использованы, чтобы обеспечить безопасность соединения:

1. **Клиент посылает серверу номер версии SSL клиента**, поддерживаемые алгоритмы шифрования и сжатия, специфичные данные для сеанса и другую информацию, которая нужна серверу, чтобы общаться с клиентом, используя SSL.
2. **Сервер посылает клиенту номер версии SSL** **сервера**, алгоритм сжатия и шифрования (выбранные из посланных ранее клиентом), специфичные данные для сеанса и другую информацию, которая нужна серверу, чтобы общаться с клиентом по протоколу SSL. Сервер также посылает свой сертификат, который требует проверки подлинности клиента. После идентификации сервер запрашивает сертификат клиента.
3. **Клиент использует информацию**, переданную сервером для проверки подлинности. Если сервер не может быть проверен, пользователь получает предупреждение о проблеме и о том, что шифрование и аутентификация соединения не может быть установлена. Если сервер успешно прошел проверку, то клиент переходит к следующему шагу.
4. Используя все данные, полученные до сих пор от процедуры рукопожатие, **клиент** (в сотрудничестве с сервером) **создаёт предварительный секрет сессии**, в зависимости от используемого шифра от сервера, шифрует его с помощью открытого ключа сервера (полученного из сертификата сервера, отправленного на 2-м шаге), а затем отправляет его на сервер.
5. **Если сервер запросил аутентификацию клиента** (необязательный шаг рукопожатия), клиент также подписывает еще один кусок данных, который является уникальным для этого рукопожатия и известным как для клиента, так и сервера. В этом случае, клиент отправляет все подписанные данные и собственный сертификат клиента на сервер вместе с предварительно зашифрованным секретом.
6. **Сервер пытается аутентифицировать клиента**. Если клиент не может пройти проверку подлинности, сеанс заканчивается. Если клиент может быть успешно аутентифицирован, сервер использует свой закрытый ключ для расшифровки предварительного секрета, а затем выполняет ряд шагов (которые клиент также выполняет), чтобы создать главный секрет.
7. **И клиент, и сервер используют секрет для генерации ключей сеансов**, которые являются симметричными ключами, использующиеся для шифрования и расшифрования информации, которой обмениваются во время SSL сессии. Происходит проверка целостности (то есть, для обнаружения любых изменений в данных между временем когда он был послан, и временем его получения на SSL-соединении).
8. **Клиент посылает сообщение серверу**, информируя его, что будущие сообщения от клиента будут зашифрованы с помощью ключа сеанса. Затем он отправляет отдельное, зашифрованное сообщение о том, что часть рукопожатие закончена.
9. И в заключение, **сервер посылает сообщение клиенту**, информируя его, что будущие сообщения от сервера будут зашифрованы с помощью ключа сеанса. Затем он отправляет отдельное, зашифрованное сообщение о том, что часть рукопожатие закончена.

На этом рукопожатие завершается, и начинается защищенное соединение, которое зашифровывается и расшифровывается с помощью ключевых данных. Если любое из перечисленных выше действий не удается, то рукопожатие SSL не удалось, и соединение не создается.

# 32. Распределение ключей на основе симметричных систем.

***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 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 передаем по сети сообщение зашифрованное и человек посередине 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.

# 33. Алгоритм передачи ключа по Диффи-Хеллману.

Алгоритм Диффи-Хеллмана для распределения ключей: абоненты А и В могут воспользоваться этим алгоритмом для обмена ключевой информацией по открытым каналам. Предварительно стороны выбирают большие простые числа n и g.

Протокол обмена:

1. А выбирает случайное большое число чб вычисляет X = g^x mod n и результат вычисления отсылает В
2. В выбирает случайное большое число y, вычисляет Y = g^y mod n и результат вычисления отсылает А
3. А вычисляет k1 = Y^x mod n
4. B вычисляет k2 = X^y mod n

Т. о., k1=k2=g^xy mod n = k

Это число сторонами может использоваться как совместный ключ(секретный). Для того чтобы третья сторона смогла вычислить значение k, она должна вычислить значение дискретного логарифма.

Протоко Д-Х является уязвимым для атаки, называемой “человек в середине”: злоумышленник С может перехватить открытое значение, посылаемое А к В, и послать вместо него свое значение. Затем он может перехватить открытое значение, посылаемое от В к А, и также передать вместо него свое открытое значение. Тем самым С получит общие секретные ключи с А и В и сможет читать и/или модифицировать сообщения, передаваемые от одной стороны к другой.

# 34. Алгоритм шифрования RSA. Реализация и криптостойкость.

***Безопасность алгоритма*** основана на трудоемкости разложения на множители (факторизации) больших чисел. Открытый и закрытый ключи являются функциями двух больших простых чисел разрядностью 100–200 десятичных цифр или даже больше. Предполагается, что восстановление открытого текста по шифртексту и открытому ключу равносильно разложению числа на два больших простых множителя.

***Предварительный этап - генерация ключа***

1. Случайным образом выбирается два больших простых числа p и q. Рассчитывается произведение n=p\*q. Это первое число входящее в ключ
2. Вычисляется функция Эйлера: φ(n) = (p-1)(q-1)
3. Случайным образом выбирается простое число e - вторая часть ключа, которое удовлетворяет условиям: e<φ(n); e и φ(n) должны быть взаимно простыми числами
4. Вычисляется число d - третья часть ключа, которое является обратным числу e: e\*d = 1 mod (φ(n))

Пара чисел (e, n) делается открытым ключом, числа p и q можно забыть(но это тоже тайная информация)

Пара чисел (d, n) - секретный ключ, эти числа тоже взаимно простые.

Первая пара используется для зашифрования вторая для расшифрования.

***Использование ключа***

*Зашифрование*

Если шифруется сообщение М, состоящее из r блоков: m1, m2, …, mi, …, mr, то шифр текст С будет состоять из такого же числа (r) блоков, представляемых числами: ci = (mi)^e mod n

*Расшифрование*

Для расшифрования каждого зашифрованного блока производиться выисление вида mi = (ci)^d mod n

**Криптостойкость алгоритма**. Зависит от трудоемкости решения проблемы разложения на множители больших чисел. «Лобовой метод» вскрытия системы RSA заключается в нахождении числа d, обратного е по модулю φ(n). Это легко сделать, если известны числа p и q. Отметим, что математически не доказано, что для восстановления сообщения по шифртексту и по значению открытого ключа нужно разложить n на множители. Тем не менее именно этот подход является наиболее очевидным. Известно, что факторизованые числа длиной более 512 битов. Значит, нужно выбирать n больше этого значения. Кроме того, числа p и q не должны быть слишком близкими друг к другу.

# 35. Алгоритм шифрования Эль-Гамаля. Реализация и криптостойкость.

Данный алгоритм является альтернативой алгоритму RSA и, при равном значении ключа, обеспечивает ту же криптостойкость.

Стойкость алгоритма Эль-Гамаля основана на трудности вычисления дискретных логарифмов.

***Предварительный этап – генерация ключа, состоящего из четырех чисел:***

1. Выбирается простое число p и два случайных числа, меньших чем p: числа x и g
2. Далее вычисляется y = g^x mod p

Открытый ключ: y, g и p;

Тайный ключ: x

***Использование ключа***

В отличии от RSA в алгоритме Э-Г используется параметр k. Использование этого параметра снижает вероятность взлома шифра. Это число является секретным и должно быть взаимно простым с р-1

*Зашифрование*

Если шифруется сообщение М, состоящее из r блоков: m1, m2, …, mi, …, mr, то шифртекст С будет состоять из такого же числа блоков но состоящего из пары чисел: ai = g^k mod p, bi = (y^k \* mi) mod p

*Расшифрование*

Для расшифрования производится вычисление вида: mi = bi/(ai)^x mod p

Т.к. (ai)^x в силу формулы(которую мы используем для зашифрования) можно заменить на g^kx mod p и учитывая то, что y = g^x mod p, можем получить подтверждение справедливости выражения: 

Справедливо также



# 36. Потоковое шифрование. Типы. Гаммирование в потоковом шифровании.

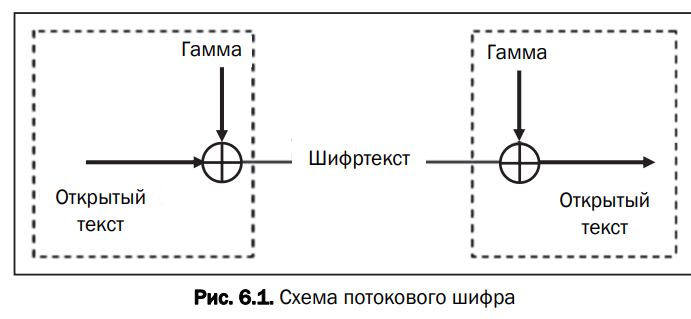
**Потоковый шифр** – это, по сути, симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифртекста С в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста М.

Потоковый шифр реализует другой подход к симметричному шифрованию, нежели блочные шифры.

В качестве символов могут выступать как отдельные биты, так и символы (байты). Таким образом, потоковые шифры подходят для шифрования непрерывных потоков данных – голоса, видео и т. д.

Все потоковые шифры деляться на два класса: синхронные и ассинхронные(или самосинхронизирующиеся).

Основной задачей потоковых шифров является выработка некоторой последовательности(гаммы) для зашифрования, т.е. выходная гамма является ключевым потоком(ключом) для сообщения



***Синхронные потоковые шифры(СПШ)*** - поток ключей генерируется независимо от открытого текста и шифртекста. Главное свойство - нераспространение ошибок.

Ошибки отсутствуют, пока работают синхронно шифровальное и дешифровальное устройства отправителя и получателя информации. Один из методов борьбы с рассинхронизацией - разбить открытый текст на отрезке, начало и конец которых выделить вставкой контрольных меток.

Синхронные потоковые шифры уязвимы к атакам на основе изменения отдельных битов шифртекста.

*Свойства*:

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

В ***самосинхронизирующихся потоковых шифрах*** символы ключевой гаммы зависят от исходного секретного ключа шифра и от конечного числа последних знаков зашифрованного текста. Внутреннее состояние генератора потока ключей является функцией фиксированного числа предыдущих битов шифртекста. Поэтому генератор потока ключей на приемной стороне, приняв фиксированное число битов, автоматически синхронизируется с генератором гаммы.

Недостаток этих шифров - распространение ошибок, так как искажение одного бита в процессе передачи шифртекста приведет к искажению нескольких битов гаммы и, соответственно, расшифрованного сообщения

*Свойства*:

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

**Гаммирование** – метод последовательного симметричного шифрования, суть которого состоит в том, что символы шифруемого текста последовательно складываются с символами некоторой специальной последовательности, которая называется гаммой.

***Генератор гаммы*** выдает ключевой поток(гамму): K=k1, k2, …., kL

Поток битов открытого текста: M = m1, m2, … , mL

Поток битов шифртекста: ci = mi + ki

Расшифрование производится операцией XOR между той же самой гаммой и зашифрованным текстом: mi = ci + ki

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

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

**Потоковый шифр** – это, по сути, симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифртекста С в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста М.

Потоковый шифр максимально должен имитировать одноразовый блокнот. В соответствии с этим ключ должен по своим свойствам максимально походить на случайную числовую последовательность.

Ключевые последовательности (случайные последовательности (СП), либо псевдослучайные последовательности (ПСП)) вырабатываются специальными блоками систем потокового шифрования – генераторами.

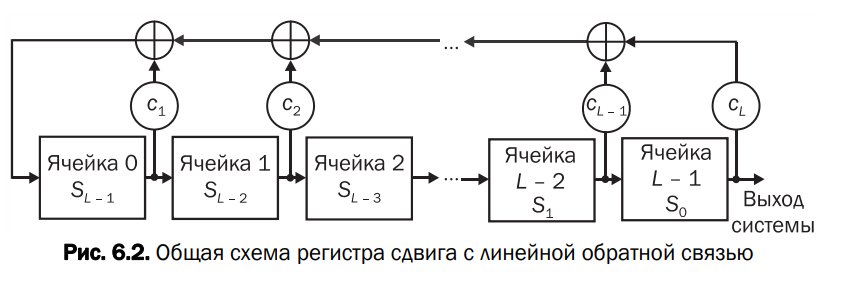
В Беларуси в настоящее время действует стандарт СТБ 34.101.47–2017 «Информационные технологии и безопасность. Алгоритмы генерации псевдослучайных чисел».

Стандарт устанавливает криптографические алгоритмы генерации псевдослучайных чисел. Алгоритмы стандарта могут применяться для построения ключей, синхропосылок, одноразовых паролей, других непредсказуемых или уникальных параметров криптографических алгоритмов и протоколов. Стандарт применяется при разработке, испытаниях и эксплуатации средств криптографической защиты информации.

Указанный стандарт определяет базовые понятия в рассматриваемой предметной области:

* **случайные числа** (последовательности) – последовательность элементов, каждый из которых не может быть предсказан (вычислен) только на основе знания предшествующих ему элементов данной последовательности;
* **псевдослучайные числа** – последовательность элементов, полученная в результате выполнения некоторого алгоритма и используемая в конкретном случае вместо последовательности случайных чисел.

***Регистры сдвига (РС)*** используются в качестве генераторов ПСП в силу простоты реализации на основе цифровой логики. РС с линейной обратной связью (РСЛОС) состоит из двух частей: собственно РС и функции обратной связи. На рис. представлена общая схема РС с линейной обратной связью. Функция обратной связи реализуется с помощью сумматоров сложения по модулю 2 (элементы XOR; на рис. обозначены в виде кружочков со знаком сложения).



Последний разряд РС в каждом такте формирует очередной символ псевдослучайной последовательности (ПСП). Такие последовательности называют также М-последовательностями (или последовательностями максимальной длины).

Количество ячеек L называют длиной регистра. Биты (ячейки) i обычно нумеруются числами от 0 до L – 1. Содержимое i-й ячейки обозначается через SL – 1 – i.

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

* содержимое ячейки L – 1 формирует часть выходной последовательности;
* содержимое i-й ячейки перемещается в (i + 1)-ю ячейку, новое содержимое ячейки 0 определяется битом обратной связи, который вычисляется сложением по модулю с определенными коэффициентами ci битов ячеек.

Помимо этого, выходная последовательность зависит, главным образом, от начального состояния каждой ячейки регистра.

РСЛОС строятся на основе примитивных порождающих полиномов

Важным свойством многочлена C(x) является 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. Иными словами: так как существует 2L – 1 разных ненулевых состояний регистра, то период последовательности, генерируемой РСЛОС при любом ненулевом начальном состоянии, не превышает 2L – 1.

# 38. Особенность шифра Вернама.

**Потоковый шифр** – это, по сути, симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифртекста С в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста М.

Классический пример потокового шифра – **шифр Вернама**, или одноразовый блокнот.

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

Так, например, при применении ключа (1 1 1 0 1) на букву «А» (1 1 0 0 0) получаем зашифрованное сообщение (0 0 1 0 1)

Зная, что для принимаемого сообщения имеем ключ (1 1 1 0 1), легко получить исходное сообщение той же операцией: (00101)XOR (11101)=(11000)

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

• быть истинно случайным (последовательность, полученная с использованием любого алгоритма, является не истинно случайной, а псевдослучайной);

• совпадать по размеру с заданным открытым текстом;

• применяться только один раз.

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

Если для гаммы последовательность битов выбирается случайно и длина гаммы равна, по крайней мере, длине сообщения, то взломать шифр невозможно. Но у данного режима шифрования есть и отрицательная особенность – проблемы с передачей и хранением ключей, ведь ключи, сравнимые по длине с передаваемыми сообщениями, трудно использовать на практике. Поэтому основная идея современных потоковых шифров – реализовать концепцию одноразового блокнота, используя секретный ключ меньшей длины, из которого для гаммы генерируется псевдослучайная числовая последовательность, похожая на случайную.

# 39. Стеганографические методы защиты информации. Классификация и области использования. Метод наименее значащих бит.

***Стеганографическая система*** - совокупность средств и методов, которые используются для формирования скрытого канала передачи(или хранения информации)

Абстрактно ***стеганографическая система*** обычно определяется как некоторое множество отображений одного пространства(множество сообщений) в другое пространство(множество стеганосообщений)

## Основные компоненты:

* контейнер С(файл-контейнер) - размещается тайное сообщение
* тайное сообщение - осаждаемое в контейнер для передачи или хранения
* ключи - выполняют ту же функцию что и в криптографии
* контейнер с встроенным сообщением(стеганоконтейнер) - передается по открытому каналу
* субъекты системы: отправитель и получатель

В зависимости ***от формата документа-контейнера*** стеганогравия делится на:

* аудио-
* видео-
* графичкская-
* текстова стеганография и др.

**Стеганографической системой ∑** будем называть совокупность сообщений M, контейнеров C, ключей K, стеганосообщений (заполненных контейнеров) S и преобразований (прямого F и обратного F–1), которые их связывают: 

Сущностью системы является тайное хранение или передача одной информации в другой информации, которая является открытой.

При построении стеганографической системы нужно учитывать:

* свойства контейнера должны быть модифицированы так, чтобы изменение невозможно было выявить при визуальном контроле.
* противник имеет полное представление о стеганографической системе и деталях ее реализации; единственной информацией, которая остается ему неизвестной, является ключ, с помощью которого только его держатель может установить факт присутствия и содержание скрытого сообщения
* если противник каким-то образом узнает о факте существования скрытого сообщения, это не должно позволить ему извлечь подобные сообщения до тех пор, пока ключ хранится в тайне;
* потенциальный противник должен быть лишен каких-либо технических и иных преимуществ в распознавании или раскрытии содержания тайных сообщений.

**Метод НЗБ:**

Большинство исследований в предметной области посвящено использованию в качестве стеганоконтейнеров изображений.

Это обусловлено:

* относительно большим объемом цифрового представления изображений, что позволяет внедрять большой объем данных;
* заранее известным размером контейнера, отсутствием ограничений, накладываемых требованиями реального времени;
* наличием в большинстве реальных изображений текстурных областей, имеющих шумовую структуру и хорошо подходящих для встраивания информации;
* слабой чувствительностью человеческого глаза к незначительным изменениям цветов изображения, его яркости, контрастности, содержанию в нем шума, искажениям вблизи контуров;
* хорошо разработанными в последнее время методами цифровой обработки изображений.

Метод НЗБ основывается на ограниченных способностях зрения или слуха человека, вследствии чего людям тяжело различить незначительные вариации цвета или звука.

Каждая точка кодируется тремя байтами. Каждый байт определяет интенсивность красного (Red), зеленого (Green) и синего (Blue) цветов. Совокупность интенсивностей цвета в каждом из трех каналов определяет оттенок пикселя.

Младшие биты дают незначительный «вклад» в изображение по сравнению со старшими. Замена одного или даже нескольких младших битов для человеческого глаза будет почти незаметна, поскольку реально человек может различать около полторы сотни цветовых оттенков.

Одним из простейших и понятных для решения наших задач является формат BMP (BitMaP) – одна из форм представления растровой графики. Изображение представляется в виде матрицы пикселей, где каждая точка характеризуется тремя параметрами: x-координатой, y-координатой и цветовым кодом на основе RGBмодели. Все операции графического ввода-вывода на экран монитора в конечном итоге осуществляются в этом формате.

Контейнеры на основе BMP-формата разделяют на два класса: чистые и зашумленные. В первых прослеживается связь между младшими и остальными битами элементов цвета, а также видна зависимость самих младших битов между собой. Осаждение сообщения в такой контейнер нарушает такие зависимости, что легко выявляется аналитиком. Если же картинка зашумлена то определить встроенное сообщение сложнее. Таким образом, в качестве файлов-контейнеров для метода НЗБ рекомендуется использовать файлы, которые не были созданы на компьютере изначально.

Другим из растровых форматов используемых в стеганографии контейнеров является формат PNG. По качеству цветового отображения данный формат превосходит JPEG и GIF, но размер файла будет на 30–40% больше.

При модификации даже 3– 4 младших разрядов состояние графического стеганоконтейнера у экспертов подозрений не вызывает при визуальном его контроле. Исходя из такой оценки, следует соотносить объем встроенного сообщения VM с объемом VC используемого контейнера.

Если размер изображения 500×500 = 250 000 пикселей, то с учетом используемой 3-цветовой модели имеем 750 000 единиц цветовых координат. Если мы планируем модифицировать только самые младшие биты всех цветовых каналов матрицы, то максимальный объем осаждаемого сообщения (VMmax) не должен превышать 750000 битов.

# 40. Текстовая стеганография. Многоключевая модель стеганографической системы.

***Стеганографическая система*** - совокупность средств и методов, которые используются для формирования скрытого канала передачи(или хранения информации)

Абстрактно ***стеганографическая система*** обычно определяется как некоторое множество отображений одного пространства(множество сообщений) в другое пространство(множество стеганосообщений)

## Основные компоненты:

* контейнер С(файл-контейнер) - размещается тайное сообщение
* тайное сообщение - осаждаемое в контейнер для передачи или хранения
* ключи - выполняют ту же функцию что и в криптографии
* контейнер с встроенным сообщением(стеганоконтейнер) - передается по открытому каналу
* субъекты системы: отправитель и получатель

В зависимости ***от формата документа-контейнера*** стеганогравия делится на:

* аудио-
* видео-
* графичкская-
* текстова стеганография и др.

К текстовой стеганографии относятся методы, предусматривающие использование в качестве контейнера файла-документа текстового типа.

Многообразие методов текстовой стеганографии подразделяется на ***синтаксические*** методы, которые не затрагивают семантику текстового сообщения, и ***лингвистические***, которые основаны на эквивалентной трансформации текстовых файлов-контейнеров, сохраняющей смысловое содержание текста, его семантику.

К ***синтаксическим*** методам компьютерной стеганографии, которые характеризуются сравнительно невысокой эффективностью (с точки зрения объема встраиваемой информации), относятся следующие:

* изменение расстояния между строками электронного текста - суть метода используется текстс различными межстрочными расстояниями: выделяется максимальное и минимальное расстояния между строками, позволяющее кодировать соответственно символы «1» и «0» осаждаемого сообщения
* изменение расстояния между словами - осаждение информации основано на модификации расстояния между словами текста-контейнера
* изменение количества пробелов между словами - основан на том, что, например, чередование одинарного пробела и двойного кодирует «1», переход же с двойного пробела на одинарный кодирует «0»
* на основе внесения специфических изменений в шрифты, т. е.начертания отдельных букв
* изменение интервалов табуляции
* Null Chipper(нулевой лепет) - размещение тайной информации на установленных позициях слов или в определенных словах текста-контейнера, который, как правило, лишен логического смысла
* увеличение длины строки(искусственное увеличение каждой строки за счет пробелов)
* использование регистра букв - ноль верхний регистр, единица - нижний
* использование невидимых символов - знак «пробел» кодируется символом с кодом 32, но в тексте его можно заменить также символом, имеющим код 255 (или 0), который является «невидимым» и отображается как пробел.

К ***лингвистическим*** методам относятся:

* метод синонимов - например подмножество синонимов: {«тайный», «секретный», «конфиденциальный», «доверительный»}. В приведенном подмножестве каждое слово имеет единственное одинаковое смысловое значение, что позволяет закодировать каждое слово своим уникальным кодом подмножество синонимов: {«тайный», «секретный», «конфиденциальный», «доверительный»}. В приведенном подмножестве каждое слово имеет единственное одинаковое смысловое значение, что позволяет закодировать каждое слово своим уникальным кодом в зависимости от двух битов секретного сообщения.
* метод переменной длины слова - длина слов в сообщении зависит от содержания секретного сообщения и способа кодирования слов: обычно одно слово текста-контейнера определенной длины кодирует два бита информации из стеганосообщения
* метод первой буквы - программа-помощник в этом методе накладывает ограничение уже не на длину слова, а на первую (можно на вторую) букву
* мимикрия - генерирует осмысленный текст, используя синтаксис, описанный в Context Free Grammar (CFG), и встраивает информацию, выбирая из CFG определенные фразы и слова;

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

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

Использовать эллиптические кривые для создания криптосистем с открытым ключом.

Их безопасность основана на трудности решения задачи дискретного логарифмирования в группе точек эллиптической кривой над конечным полем. Этим и обусловлена их высокая криптостойкость по сравнению с другими алгоритмами.

**Эллиптические кривые** – математический объект, который может быть определен над любым полем. В криптографии обычно используются конечные поля. Для точек на эллиптической кривой вводится операция сложения, которая играет ту же роль, что и операция умножения в криптосистемах RSA и Эль-Гамаля.

Конечным полем называется алгебраическая система, которая состоит из конечного множества F и двух бинарных операций (сложения и умножения). Порядком поля называется количество элементов в поле (во множестве F).

Еще одним преимуществом криптосистем на эллиптических кривых является высокая скорость обработки информации. Например, уровень стойкости, который достигается, скажем, в RSA при использовании 1024-битных ключей, в системах на эллиптических кривых реализуется при том же параметре длиной 160 битов. Криптосистемы на эллиптических кривых нецелесообразно применять для шифрования больших объемов данных. Но зато их можно эффективно использовать для систем цифровой подписи и ключевого обмена. С 1998 года использование эллиптических кривых для решения криптографических задач, таких как цифровая подпись, было закреплено в стандартах США ANSI X9.62 и FIPS 186–2, а в 2001 году аналогичный стандарт был принят в России – ГОСТ Р 34.10–2001.

Процесс формирования кривой:

1. Выбираем случайное простое число p. Его длина в битах (t = log p + 1; понятно, что t должно быть целочисленным с округлением в большую сторону) должна быть такой, чтобы сделать невозможным применение общих методов нахождения логарифмов на кривой. Величина t в настоящее время принимается не менее 256 битов.
2. Выбираем случайные числа a и b такие, что a, b (mod p) != 0 и (4\*a^3 + 27\* b^2) mod p !=0

b - выбирается случайным образом, a - небольшое целое число

1. Определяем число точек N на кривой, равное Ep(a, b). N - должно иметь большой простой делитель q, а лучще всего чтобы было простым числом N = q
2. Проверяем выполняется ли неравенство (p^k - 1) mod q != 0 для всех k, 0<k<32. Если нет, то возвращаемся к шагу 2.
3. Проверяем, выполняется ли неравенство q!=p. Если нет возвращаемся к шагу 2.
4. Необходимая для криптографических приложений кривая получена.

Задача, которую решает криптоаналитик при использовании криптосистемы на базе эллиптических уравнений, относится к задачам дискретного логарифмирования на эллиптической кривой и формулируется следующим образом: даны точки Р и Q на эллиптической кривой порядка N (N – число точек на кривой). Необходимо найти единственное число х, такое что Р = х ⋅ Q. Величина х и является дискретным логарифмом от Q по основанию Р.

Вычисление точки P как результата скалярного произведения целого положительного х на точку Q является достаточно простой задачей. С другой стороны, вычисление значения х на основании точек P и Q является вычислительно трудной задачей. Для больших значений х эта задача практически неразрешима. На этом факте и основана рассматриваемая криптографическая система.

# 42. Представление и описание эллиптической кривой на основе алгебраической геометрии

***Эллиптическая кривая*** (ЭК, ЕС – Elliptic Сurve) – это не эллипс. Она называется так потому, что описывается кубическим уравнением, подобным тем, которые используются для вычисления кривой эллипса.

***Эллиптические кривые*** – математический объект, который может быть определен над любым полем.

***Эллиптическая кривая над вещественными числами*** – это множество точек, описываемых уравнением (1) при этом константы (а и b – вещественные числа) должны удовлетворять условию (2)

Формула (1) называется уравнением Вейерштрасса, а условие (2) исключает из рассмотрения кривые с особыми точками или особые кривые

Частью ЭК является ***бесконечно удаленная точка*** (также известная как идеальная точка), которую мы обозначают символом О.

***Группа*** – непустое множество с определенной на нем бинарной операцией, называемой сложением и удовлетворяющей нескольким аксиомам.

***Группа для ЭК*** есть непустое множество, элементы которого являются точками ЭК, обладающими следующими свойствами:

* единичный элемент – это бесконечно удаленная точка О;
* обратная величина точки R – это точка, симметричная относительно оси Х;
* сложение задается следующим правилом: сумма трех ненулевых точек P, Q и –R, лежащих на одной прямой, будет равна P + Q + (–R) = О.

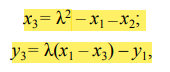
Закон сложения точек ЭК:

* прямая проходящая через точки R и -R, является вертикальной прямой, которая не пересекает ЭК ни в какой третьей точке; если R = (x, -y), то R + (x, y) = O. Точка (x, y) является отрицательным значением точки R и обозначается -R. Таким образом, по определению R + (-R) = O
* P + Q = R: пусть P и Q - две различные точки ЭКи Р не равно Q; если проведем через P и Q прямую, то она пересечет ЭК еще только в одной точке, называемой -R; точка -R отображается относительно оси X в точку R, равную сумме точке Р и Q: P + Q = R

Если P=Q то говорим о удвоении точки: Р + Р = 2Р

***Скалярное умножение*** осуществляется посредством нескольких комбинаций сложения и удвоения точек эллиптической кривой.

Например, точка 25P может быть представлена как 25P = = 2(2(2(2P)) + 2(2(2P))) + P.

Если Р=(x1, y1) и Q = (x2, y2) P+Q=(x3, y3) 

где  при 

λ – угловой коэффициент секущей, проведенной через точки Р = (х1, у1) и Q = (х2, у2). При Р = Q секущая превращается в касательную, чем и объясняется наличие двух формул для вычисления λ.

# 43. Арифметические операции в эллиптической криптографии

***Эллиптические кривые*** – математический объект, который может быть определен над любым полем.

Арифметические операции в эллиптической криптографии производятся над точками кривой.

Основной операцией является сложение. Сложение двух точек.

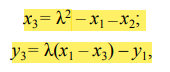
Закон сложения точек ЭК:

* прямая проходящая через точки R и -R, является вертикальной прямой, которая не пересекает ЭК ни в какой третьей точке; если R = (x, -y), то R + (x, y) = O. Точка (x, y) является отрицательным значением точки R и обозначается -R. Таким образом, по определению R + (-R) = O
* P + Q = R: пусть P и Q - две различные точки ЭКи Р не равно Q; если проведем через P и Q прямую, то она пересечет ЭК еще только в одной точке, называемой -R; точка -R отображается относительно оси X в точку R, равную сумме точке Р и Q: P + Q = R

Если P=Q то говорим о удвоении точки: Р + Р = 2Р

***Скалярное умножение*** осуществляется посредством нескольких комбинаций сложения и удвоения точек эллиптической кривой.

Например, точка 25P может быть представлена как 25P = = 2(2(2(2P)) + 2(2(2P))) + P.

Если Р=(x1, y1) и Q = (x2, y2) P+Q=(x3, y3) 

где  при 

λ – угловой коэффициент секущей, проведенной через точки Р = (х1, у1) и Q = (х2, у2). При Р = Q секущая превращается в касательную, чем и объясняется наличие двух формул для вычисления λ.

По определению, эллиптическая кривая обладает следующим свойством: если три ее точки лежат на одной прямой, то их сумма равна О.

Это свойство позволяет описать основные правила сложения, а также умножения точек эллиптической кривой:

* пусть P и Q – две различные точки эллиптической кривой и Р не равно Q. Проведем через P и Q прямую. Она пересечет эллиптическую кривую только в одной точке, называемой –R. Точка –R отображается относительно оси Х в точку R, равную сумме точек P и Q; закон сложения точек эллиптической кривой: P + Q = R;
* прямая, проходящая через точки R и –R, является вертикальной прямой, которая не пересекает эллиптическую кривую ни в какой третьей точке; если R = (х, –у), то R + (х, у) = О. Точка (х, у) является отрицательным значением точки R и обозначается –R; таким образом, по определению R + (–R) = О;
* если O – нулевой элемент, то справедливо равенство O = –O, а для любой точки P эллиптической кривой имеем P + O = P;
* чтобы сложить точку Р с ней самой, нужно провести касательную к кривой в точке Р; закон удвоения точки Р: P + P = 2 ⋅ P;
* умножение точки Р на целое положительное число k определяется как сумма k точек Р: k ⋅ P = P + P + P + … + P;
* • скалярное умножение осуществляется посредством нескольких комбинаций сложения и удвоения точек эллиптической кривой. Например, точка 25 ⋅ P может быть представлена как 25 ⋅ P = (2 ⋅ (2 ⋅ (2 ⋅ (2 ⋅ P)))) + (2 ⋅ (2 ⋅ (2 ⋅ P))) + P.

# 44. Система согласования криптографических ключей на основе эллиптической кривой

***Эллиптические кривые*** – математический объект, который может быть определен над любым полем.

Система похожа на алгоритм Диффи-Хеллмана, т.е. предназначена для передачи закрытой информации по открытому каналу.

Стойкость рассматриваемого алгоритма определяется сложностью вычисления дискретного логарифма над эллиптической кривой Eр(а, b). Действительно, вычисление точки Q как результата скалярного произведения целого положительного х на точку G является достаточно простой задачей. С другой стороны, вычисление значения х на основании точек Q и G является вычислительно трудной задачей. Для больших значений х эта задача практически неразрешима. На этом факте и основана рассматриваемая криптографическая система.

Обмен ключами с использованием эллиптических кривых может быть выполнен следующим образом.

1. Выбирается простое число р и параметры a и b для эллиптической кривой. Это задает эллиптическую группу точек Ер(a, b).
2. В Ер(a, b) выбираем генерирующую точку G = (х, у). Важно чтобы наименьшее значение q, при котором выполняется условие q\*G = O, оказалось очень большим простым числом.

Параметры Ер(a, b) и G криптосистемы являются параметрами, известными всем участникам. Обмен ключами между участники информационного процесса (А и В) можно провести по следующему алгоритму

1. Сторона А выбирает целое число ka < q. Это число тайный ключ стороны А
2. А генерирует открытый ключ: Ya = ka \* G и отсылает стороне В
3. Сторона В выбирает себе тайный ключ kb и вычисляется Yb = kb \* G, отсылает стороне А
4. Стороны А и В получив открытые ключи, генерируют секретный ключ Ka = ka\* Yb = k, Kb = kb\*Ya = K

Обе стороны получили один и тот же результат поскольку: ka\* Yb = ka \* (kb\*G) = kb \* (ka\*G) = kb\*Ya

# 45. ЭЦП. Назначение и свойства.

***Электронная цифровая подпись*** – контрольная характеристика сообщения, которая вырабатывается с использованием личного ключа, проверяется с использованием открытого ключа, служит для контроля целостности и подлинности сообщения и обеспечивает невозможность отказа от авторства.

ЭЦП позволяет выполнять те же функции, что и собственноручная (поставленная «от руки») подпись:

* **аутентифицировать лицо**, подписавшее сообщение; ЭЦП получается в результате криптографического преобразования электронных данных документа с использованием личного ключа ЭЦП;
* **контролировать целостность сообщения**; ЭЦП вычислена на основании исходного состояния документа и соответствует лишь ему, поэтому при любом случайном или преднамеренном изменении документа подпись станет недействительной;
* **защищать сообщение от подделок**; любая подделка должна быть выявлена путем операций сравнения соответствующих атрибутов подписанного и полученного адресатом сообщений;
* **доказать авторство лица**, подписавшего сообщение; создать корректную ЭЦП можно, лишь зная закрытый ключ, известный только его владельцу (лицу, подписавшему документ).

Важнейшие отличительные особенности ЭЦП:

* 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.

Важное свойство цифровой подписи заключается в том, что ее может проверить каждый, кто имеет доступ к открытому ключу ее автора (здесь речь идет об ЭЦП на основе алгоритмов асимметричного шифрования). Один из участников обмена сообщениями после проверки подлинности цифровой подписи может передать подписанное сообщение еще кому-то, кто тоже в состоянии проверить эту подпись.

И еще одна особенность. Обычно подписываемые с помощью ЭЦП сообщения не шифруются. Они пересылаются в исходном виде. Их содержимое не защищено от нарушения конфиденциальности. Путем совместного применения известных алгоритмов шифрования и ЭЦП можно создавать сообщения, которые будут и зашифрованы, и подписаны. Для этого автор сначала должен добавить к сообщению свою сгенерированную ЭЦП, а затем – зашифровать получившуюся в результате пару (состоящую из самого сообщения и подписи к нему) соответствующим ключом. Получатель расшифровывает полученное сообщение тем же (при симметричной системе) или публичным ключом отправителя (при использовании асимметричной системы). Если проводить аналогию с пересылкой обычных бумажных документов, то этот процесс похож на то, как если бы автор документа поставил под ним свою печать, а затем положил его в бумажный конверт и запечатал, с тем чтобы конверт был распечатан только тем человеком, кому адресовано сообщение.

# 46. ЭЦП. Основные методы генерации. Атаки на ЭЦП

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

Основным компонентом в технологии ЭЦП является ключ. Принадлежность ключа, в предположении, что он известен только законным пользователям, позволяет решать все «возложенные на ЭЦП» задачи. В соответствии с этим обстоятельством функции ЭЦП могут быть реализованы на основе классических методов зашифрования/расшифрования:

ЭЦП может быть реализована на основе:

* **симметричных систем**( с тайным ключом) - не отличается от DES
* **симметричных систем и посредника** - создаются две симметричные системы: между отправителем и посредником, посредником и получателем. Посредник выдает двум сторонам совершенно разные тайные ключи
* **асимметричных систем**(с открытым ключом) - сообщение шифруется тайным ключом отправителя, отправитель верифицирует подпись(устанавливает авторство - для расшифрования публичный ключ отправителя, гарантия надежности если после расшифрования формат и содержимое документа имеют логическую стойкость) с помощью своего открытого ключа.

Целью атак на цифровые подписи является, в конечном итоге, возможность их подделки или фальсификации.

***Атака с использованием открытого ключа***. Имеется в виду ключ субъекта, подписывающего документ. Криптоаналитик обладает только открытым ключом.

***Атака на основе известных сообщений***. В распоряжении криптоаналитика имеются некоторые ЭЦП и соответствующие им документы.

***Адаптивная атака на основе выбранных сообщений***. Криптоаналитик может получить подписи электронных документов, которые он выбирает сам. Основные результаты или цели описанных атак можно классифицировать следующим образом

***Полный взлом ЭЦП***. Получение закрытого ключа отправителя подписанного сообщения. Это означает полный взлом алгоритма.

***Универсальная подделка подписи***. Криптоаналитик находит алгоритм, позволяющий подделывать подписи для любого электронного документа.

***Выборочная подделка подписи***. Дает возможность подделывать подписи для документов, выбранных криптоаналитиком.

***Экзистенциальная подделка подписи***. Дает возможность получения допустимой подписи для некоторого документа, не выбираемого криптоаналитиком.

Самой опасной атакой является адаптивная атака на основе выбранных сообщений. При анализе алгоритмов ЭЦП на криптостойкость принято принимать во внимание и детально анализировать именно ее.

Современные алгоритмы генерации и использования ЭЦП оставляют криптоаналитику мало шансов на получение закрытого ключа алгоритма из-за вычислительной сложности задач, на основе которых ЭЦП построена

Более вероятен поиск коллизий в хеш-функциях. Коллизия первого рода эквивалентна экзистенциальной подделке, а коллизия второго рода – выборочной.

***Подделка документа*** (коллизия первого рода). Криптоаналитик подбирает документ М′ к данной подписи. Однако в подавляющем большинстве случаев такой документ может быть только один. Это оригинальный документ М, для которого и создавалась данная ЭЦП. Причина здесь кроется в следующем. Во-первых, документ представляет из себя осмысленный, логически стройный текст. Во-вторых, текст документа оформлен по установленной форме. Это значит, если для фальшивого набора байт (M′) и произойдет коллизия с хешем исходного документа (М), то должны выполниться 3 следующих условия:

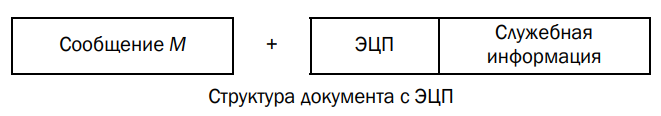
* случайный документ должен соответствовать сложно структурированному формату документа;
* случайный документ должен представлять собой текст, оформленный по установленной форме;
* текст документа M′ должен быть осмысленным и грамотным (по определению). Вероятность такой коллизии ничтожно мала.

Получение двух документов с одинаковой подписью. Вероятность такого события является более высокой в сравнении с вероятностью наступления коллизии первого рода. В этом случае злоумышленник «фабрикует» два документа с одинаковой ЭЦП и в нужный момент подменяет один документ другим. При использовании криптостойкой хэш-функции такая атака должна быть также вычислительно сложной. Однако эти угрозы могут реализоваться из-за слабостей конкретных алгоритмов хэширования, подписи или ошибок в их реализация.

# 47. ЭЦП на основе симметричной криптографии

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

Общая структура подписанного электронного документа М представляет собой, как правило, конкатенацию этого документа и ЭЦП. Кроме этих двух элементов, интегральный документ может содержать некоторую служебную информацию.



***Симметричная схема цифровой подписи*** использует один и тот же ключ для генерации ЭЦП и ее проверки.

Основные функции ЭЦП, с формальной точки зрения, реализуются примитивной процедурой шифрования/расшифрования. Ведь аутентичнось, целостность, защиту от подделок и доказательство авторства обеспечивает собственно симметричный ключ, используемый двумя сторонами в процессе обмена сообщениями. Этот ключ согласован сторонами, известен (должен быть известен) только этим абонентам. Хотя в дополнение к обычным образом зашифрованному сообщению не формируется ЭЦП как самостоятельный элемент послания или просто шифртекста, как видим, формально другая сторона получает вместе с сообщением все необходимые вышеперечисленные доказательства.

Первыми, кто обратил внимание на возможность симметричной схемы ЭЦП, были основоположники самого понятия: У. Диффи и М. Хеллман, которые опубликовали описание алгоритма подписи одного бита с помощью блочного шифра. Из этого следует, что подписывать (шифровать) следует каждый бит сообщения М, т. е. по размеру ЭЦП может превосходить подписываемый документ на несколько порядков. В силу этого серьезного недостатка идея не нашла практического применения.

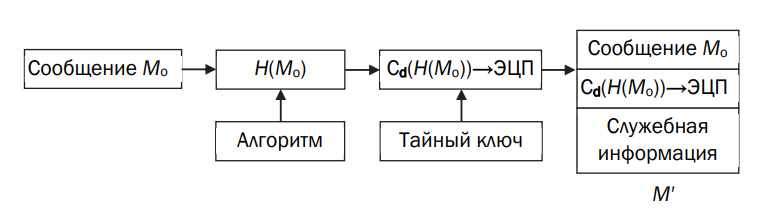
Однако если предусмотреть наличие в системе третьего лица – арбитра или посредника (П), пользующегося доверием обеих сторон, то можно избежать указанного недостатка

# 48. ЭЦП на основе алгоритма RSA

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

Могут существовать две ситуации:

* сообщение Мo подписывается и передается в открытом (незашифрованном) виде;
* сообщение Мo подписывается и передается в зашифрованном виде.

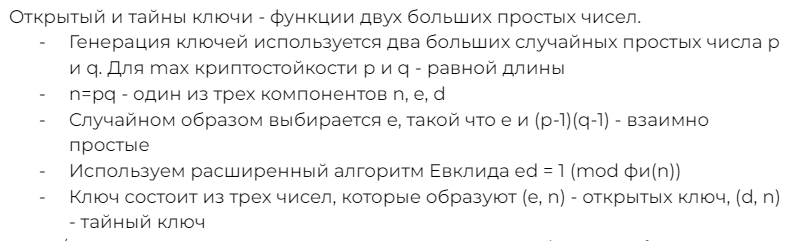


***Формирования ЭЦП.***

1 Вычисляется хеш-образ сообщения

2 Вычисляется ЭЦП по хешу с использованием закрытого ключа отправителя. Подпись S вычисляется на основе алгоритма RSA. 

d0, n0 - элементы тайного ключа отправителя



3 Присоединяем ЭЦП к сообщению

4 Отправляем сообщение получателю

***Верификация подписи:***

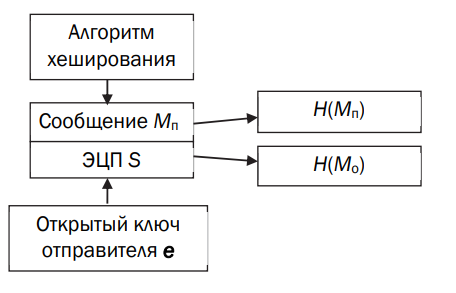
1 Получатель отделяет цифровую подпись от сообщения.

2 Применяет к сообщению операцию хеширования(используя ту же функцию что и отправитель, если мы формировали подпись на основе хеша сообщения), в итоге получаем хеш образ сообщения.

3 Используя открытый ключ отправителя расшифровываем подпись используя формулу

И извлекаем из ЭЦП хеш-образ отправленного сообщения.

4 Далее проверяем соответствие обоих хеш-образов, и если они совпадают то отправитель действительно является тем за кого себя выдает и сообщение при передаче не подверглось искажению.



Если подписываемое сообщение М(М') также ***должно передаваться в зашифрованном*** виде, то обычно М' шифруется на стороне отправителя стандартным образом: с помощью открытого ключа получателя (eп и nп), который перед основным процессом верификации подписи расшифровывает послание своим тайным ключом: dп и nп. Далее осуществляются вычисления и анализ, как и в первом случае.

# 49. ЭЦП на основе симметричной криптосистемы и посредника

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

Создаются две симметричные системы: между отправителем и посредником, посредником и получателем. Посредник выдает двум сторонам совершенно разные тайные ключи

Основной алгоритм состоит в следующем.

1. Посредник П вырабатывает для А и В разные (сеансовые, например) ключи: KА и KВ.
2. Абонент А шифрует свое сообщение М ключом KА и отсылает его посреднику: С = ЕKА (М).
3. П расшифровывает С ключом KА (М = DKА (С)), тем самым извлекает сообщение М. Присоединяет к этому сообщению подтверждение того, что автором его является абонент А (обозначим эту часть нового сообщения МД). Таким образом сформирован конкатенированный документ М′ = М || МД. Посредник шифрует М′ ключом KВ: С1 = ЕKВ (М′). Зашифрованное сообщение С1 отсылается абоненту В.
4. Абонент В расшифровывает С1 ключом KВ (М′ = DKВ (С1)), тем самым извлекает сообщение М′. Из этого сообщения он получает оригинал сообщения М и подтверждение тому, что автором этого сообщения является абонент А

Таким образом, здесь выполнены все функциональные требования, присущие ЭЦП:

1. подпись достоверна (П – гарант);
2. подпись неподдельна, так как только А и В знают ключи, использовавшиеся в процедурах (к П – абсолютное доверие);
3. подписанный документ нельзя изменить или подделать;
4. подпись нельзя отрицать. Если ключи действительно были сеансовыми, то подпись невозможно использовать повторно

# 50. ЭЦП DSS.

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

Алгоритм DSA (Digital Signature Algorithm – алгоритм цифровой подписи), или DSS (Digital Signature Standard – стандарт цифровой подписи), является одним из известных, нередко и сейчас применяемых. В алгоритме используются следующие параметры:

p – простое число длиной от 64 до 1024 битов (число должно быть кратно 64);

q – 160-битный простой множитель (р – 1).

Далее вычисляется число g



где v – любое число, меньшее (р – 1), для которого выполняется условие:



То есть значение v - не взаимно простое с p

Числа p, q, v могут использоваться группой лиц.

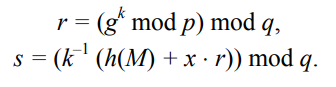
**Открытый ключ** у вычисляется в соответствии с выражением 

где x < q; х – закрытый ключ.

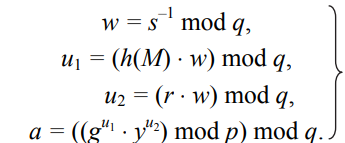
H(m) – хеш подписываемого сообщения.

ЭЦП состоит из двух чисел: r и s. Число k здесь играет такую же роль как в алгоритме шифрования Эль-Гамаля

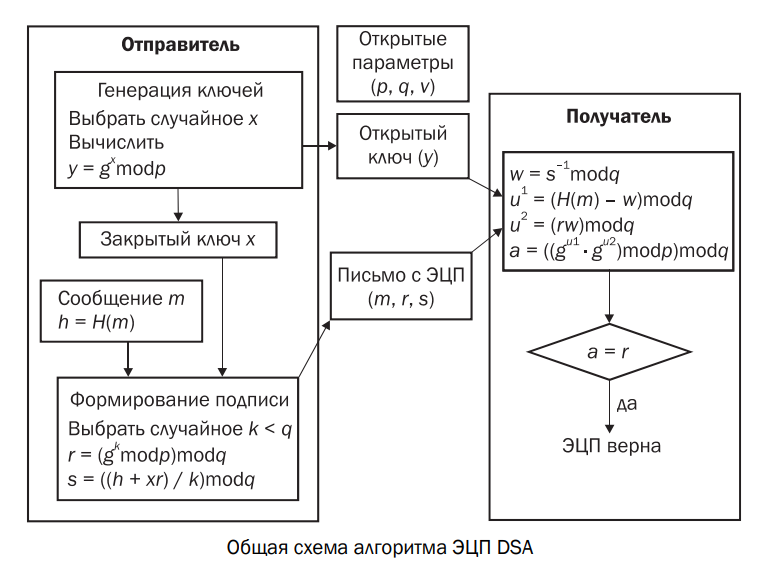
***Генерация ЭЦП:***



***Верификация подписи:***



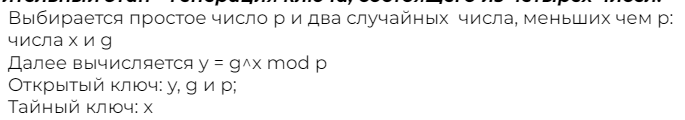
Подпись считается достоверной, если a = r



# 51. ЭЦП на основе алгоритма Эль-Гамаля

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

Ключевая информация отправителя для ЭЦП создается по алгоритму Эль-Гамаля



Чтобы подписать сообщение Мо, обладатель используемых для ЭЦП ключей должен выбрать случайное число k, взаимно простое с (р – 1). Затем вычисляется числа а и b, являющиеся цифровой подписью (S = {a, b}): 

для вычисления b с помощью расширенного алгоритма Евклида решается уравнение 

*Результатом зашифрования является одна пара чисел, а не пара для каждого блока исходного сообщения*

Получателю отправляется сообщение.

Для верификации подписи вычисляется хеш полученного сообщения H(Mп) = h. Далее нужно убедиться что выполняется равенство: 

Если равенство выполняется, подпись верифицируется.

# 52. ЭЦП на основе эллиптической кривой.

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

В криптографии на основе ЭК тайный ключ – это случайное целое d, выбранное из множества {1, 2, ..., q – 1}, где q – порядок подгруппы; открытый ключ – это точка Q, такая, что Q = dG, где G – базовая точка подгруппы.

Криптостойкость алгоритмов на основе ЭК определяется, например, для алгоритма ЭЦП в стандарте Республики Беларусь параметром l, называемым уровнем стойкости и принимающим значения (рекомендуется) из {128, 192, 256}. При этом для взлома ключа злоумышленнику нужно выполнить 2^l операций

***Первый этап: выбор (генерация) ЭК***

Обычно он основан на выполнении следующих условий и операций:

1. Входными параметрами являются: число l, число p, удовлетворяющее условию 

Можно использовать некоторое простое число , где с - небольшое натуральное число

1. Выбирается число b такое, что 0 < b < p

Т.о. задана ЭК: Ep(a, b)

1. Выбираются порядок q (простое число) и генерирующая точка G, которая задается двумя координатами, например G = (0, yG)

***Второй этап: генерация ключевой информации***

1. Входными параметрами являются: p, a, b, q и G
2. Генерируется тайный ключ - число d, выбранное из множества {1, 2, …, q-1}
3. Вычисляется открытый ключ - точка Q: Q=dG, к открытому ключу также относятся p, a, b, q

***Генерация ЭЦП***

1. Выбираем число k (1 < k < q), q - порядок точки G
2. Вычислить точку kG = (x, y), вычислить r = x mod q; при r = изменить k и повторить шаг 2
3. Вычислить t = k^(-1) mod q
4. Вычислить s = (t (H(M) + dr)) mod q, при s = 0 изменить k и повторить алгоритм
5. Стороне В отсылаются сообщение М и ЭЦП (числа r и s)

***Верификация ЭЦП***

Получатель знает алгоритм хеширования, который использовался отправителем, открытый ключ отправителя

1. Проверить выполнение условия: 1<r, s<q; если условие не выполняется, то легитимность подписи не подтверждается, в противном случае - выполняются дальнейшие шаги
2. Вычисляются H(M) и w = s^(-1) mod q
3. Вычисляются u1 = wH(M) (mod q), u2 = wr (mod q)
4. Вычисляются Gu1 + Qu2 = (x’, y’), v = x’ mod q
5. Сравниваются v и r; если равенство выполняется, подтверждается легитимность подписи и целостность полученного сообщения.

# 53. Алгоритм К. Шнорра. Стандарт ЭЦП в РБ.

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

Cхема является основой стандарта ЭЦП в Беларуси. Существующий стандарт ЭЦП в Республике Беларусь (СТБ 34.101.45-2013) основан на схеме Шнорра (а также на эллиптических кривых) 26.

Алгоритм ЭЦП К. Шнорра (К. Schnorr) является вариантом алгоритма ЭЦП Эль-Гамаля.

Одной из особенностей ЭЦП Эль-Гамаля является то, что число p должно быть очень большим, чтобы сделать действительно трудной проблему дискретного логарифма. Рекомендуемая длина p должна составлять по крайней мере 1024 бита. Чтобы уменьшить размер подписи, Шнорр предложил новую схему, но с уменьшенным размером подписи.

***Ключевая информация***:

p – простое число в диапазоне от 512 до 1024 битов;

q –160-битное простое число, делитель (p – 1);

любое число g (g ≠ 1) такое, что



Числа p, g, q являются открытыми и могут применяться группой пользователей.

Выбирается число х < q (х является тайным ключом) и вычисляется последний элемент открытого ключа: 

Секретный ключ имеет длину не менее 160 битов. Для подписи сообщения Мо выбирается случайное число k (1 < k < q) и вычисляет параметр а:



Далее вычисляется хеш от канкатенации сообщения Мо и числа а: h = H(Mo||a). Обратим внимание, что хэш-функция непосредственно не применяется к сообщению. Создается хеш-образ подписываемого сообщения, спереди присоединенного к числу а. Далее вычисляется значение b: 

Получателю отправляются М' = Мо||S; S = {h, b}.

Для проверки подписи получатель вычисляет 

Затем он проверяет выполнение равенства: h = Н(Mп||Х). Подпись достоверна, если равенство выполняется. Порядок величин х и h – около 140 двоичных разрядов, порядок числа k – около 70–72 разрядов.

# 54.Протокол Kerberos.

**Назначение** - для пересылки зашифрованного сообщения ( А В) по открытым каналам на платформе ОС Windows при взаимод с Т;

Опирается на протокол Нидхэма-Шрёдера (R. NeedhamM. Schröder) и базируется на симметричном шифровании данных

***Протокол Нидхэма-Шрёдера***

A, B, Т – имена участников, ЕА - ключ, общий для А и Т, ЕВ – ключ, общий для В и Т

A -> T: A, B, RA; RA - случайное число, сгенерированное А

Т генерирует случайный сеансовый ключ К; Затем шифрует: C = EA(RA, B, K; EB(K,A)); T: C -> A

А извлекает из С: К и убеждается, что RA равное RA для 1-го этапа; Извлекает EB(K,A) = C3; A: C3 -> B

B, используя EB, извлекает K из С3; B - генерирует случайное число RB, создает шифртекст C4 = K(RB) и B: C4 -> A

A расшифровывает C4 ключом Кб создает шифртекст С5 = K(RB - 1); A: C5 -> B

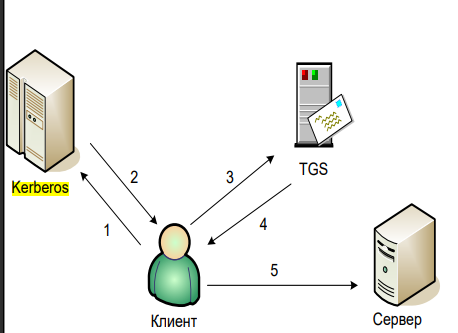
B расшифровывает C5 ключом К и убеждается, что известное ему RB уменьшено на 1

Т.о. создан секретный сеансовый ключ К для А и В

# 

Установленная в сети TCP/IP служба Kerberos, является доверенной стороной (T)

* Основой Kerberos является БД Клиентов и их секретных ключей
* Сетевые службы, которые требуют аутентификацию, должны зарегистрировать в Kerberos свои секретные ключи
* Так как Kerberos знает все секретные ключи, он может убеждать одни объекты в подлинности других. Керберос создает сеансовые ключи, которые выдаются Клиенту и Серверу, и никому больше
* Для шифрования используется алгоритм DES
* Для организации канала связи Клиент запрашивает у Kerberos разрешение на обращение к службе организации таких сообщений, эта служба называется Ticket Granting Service (TGS) — служба выделения мандата



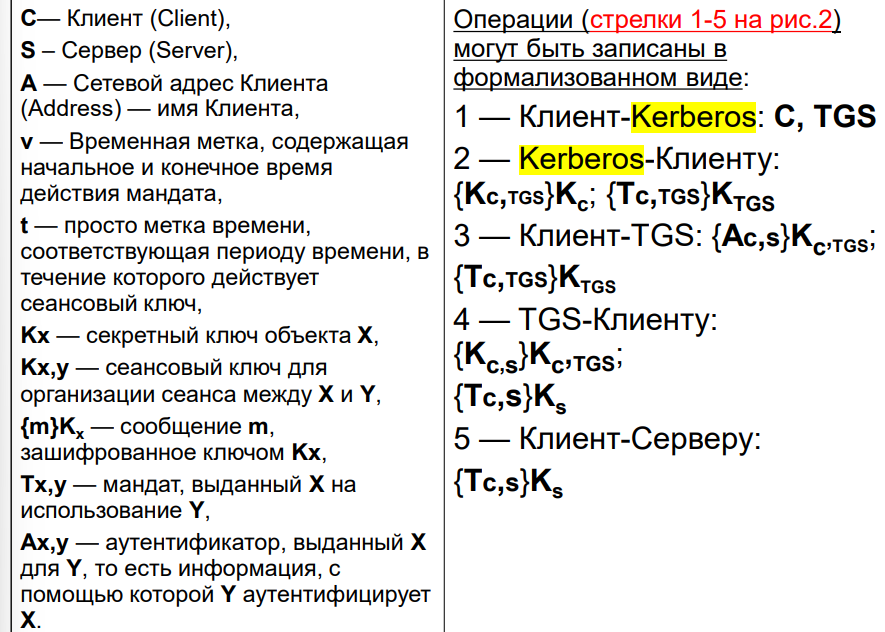
1 — Клиент запрашивает Керберос разрешение на обращение к службе TGS.

2 — После анализа предоставленных документов о возможности организации сообщения между Кл и Серв Керберос выдает Кл-ту соответствующее разрешение.

3 — Пользуясь разрешением службы Керберос, Кл запрашивает TGS о выделении ему мандата на организацию канала между Клиентом и Сервером.

4 — Получение такого мандата.

5 — Клиент пересылает соответствующее сообщение серверу



Kerberos использует 2 типа удостоверений:

* Мандаты (для безопасной передачи Серверу данных о личности Клиента):



Клиент не может расшифровать мандат, поскольку он не знает секретный ключ Ks, но он может предъявить его Серверу, как докозательство его аутентичности, т.е. прочитать либо изменить мандат ни Клиент, ни кто-либо иной не может.

* Аутентификаторы (это дополнительная информация, предъявляемая вместе с мандатом)



Клиент создает аутентификатор на каждый сеанс, Ключ - является просто ключом и необязательным дополнительным элементом сеанса и все эти данные шифруются общим ключом, известным Клиенту и Серверу: Kc,s . В отличие от мандата, аутентификатор используется только один ра

# 55. Деструктивные программы. Классификация и методы нейтрализации.

***Вирусы*** — это саморазмножающиеся программы путем дописывания собств-х кодов к исполняемым файлам. Вирусы могут содержать, а могут не содержать деструктивные функции.

***Макровирусы*** - это файловые вирусы. Макровирусы заражают различные документы и электронные таблицы, такие, как, например, файлы редакторов Word и Excel. Код этих вирусов создается на макроязыках, отсюда и их название. Большинство макровирусов обладают свойствами резидентов и действуют только во время работы с инфицированным документом.

***Черви*** — это программы, которые самостоятельно размножаются по сети и, в отличие от вирусов, не дописывают себя (как правило) к исполняемым файлам. Все черви съедают ресурсы компьютера, “нагоняют” интернет-трафик и могут привести к утечке данных с вашего компьютера.

***Кейлоггеры*** — программы, которые регистрируют нажатия клавиш, делают снимки рабочего стола, способом отслеживают действия пользователя во время работы за компьютером и сохраняют эти данные в скрытый файл на диске, затем этот файл попадает к злоумышленнику

***Трояны*** — собирают конфиденциальную информацию с компьютера пользователя (пароли, базы данных и пр.) и тайно по сети высылают их злоумышленнику. Существует разновидность троянов под названием Trojan-Downloader, которая, осуществляет несанкционированную загрузку на компьютер пользователя программного обеспечения (обычно зловредного).

***Боты*** — распространенный в наше время вид зловредного ПО, который устанавливается на компьютерах пользователей и используется для атак на другие компьютеры

***Снифферы*** — это анализаторы сетевого трафика. Могут использоваться в составе зловредного ПО, скрытно устанавливаться на компьютере пользователя и отслеживать данные, которые отправляет или получает пользователь по сети.

***Руткиты***— сами по себе не являются зловредным ПО. Назначение — скрывать работу других зловредных программ (кейлоггеров, троянов, червей и т.д.) как от пользователя, так и от программ безопасности (антивирусов, файерволов, систем обнаружения атак и пр.)

***Звонилка*** —может просто изменять настройки уже существующих соединений удаленного доступа на компьютере пользователя или создавать новое соединение

***Эксплоиты*** — это программы, которые через ошибку в программном обеспечении компьютера могут предоставить несанкционированный доступ машине или просто вывести ее из строя (завесить, перезагрузить).

***AdWare*** (приставка "Ad" является сокращением от английского слова "advertisement" — реклама, а слово "Ware" переводится как "продукт") — это приложение, которое показывает рекламу, доставляемую через интернет.

***SpyWare*** (англ. Spy — шпион, Ware — продукт) — программа-шпион, которая собирает и передает посторонним лицам информацию о пользователе без его согласия. В основном, SpyWare используется для маркетинговых исследований, поэтому собранная информация обычно передается на серверы рекламных фирм

***Эксплойты*** (эксплоит, сплоит, англ. exploit, эксплуатировать) — компьютерная программа, фрагмент программного кода или последовательность команд, использующие уязвимости в программном обеспечении и применяемые для проведения атаки на вычислительную систему;

это подвид вредоносных программ, которые содержат данные или исполняемый код, способный воспользоваться одной или несколькими уязвимостями в программном обеспечении на локальном или удаленном компьютере.

Особая проблема - эксплойты неизвестных уязвимостей, обнаруженных и использованных преступниками, — так называемые уязвимости нулевого дня

***Angler*** — один из самых сложных наборов эксплойтов на черном рынке. Обнаруживает антивирусы и виртуальные машины (часто используемые экспертами по безопасности как приманки) и задействует шифрованные файлы для затруднения исследования.

***Nuclear Pack*** — поражает жертв эксплойтами Java и Adobe PDF.

***Neutrino*** — набор эксплойтов от русскоязычных разработчиков для Java.

***Blackhole Kit*** — наиболее распространенная веб-угроза (в 2012 году), нацеленная на уязвимости в старых версиях браузеров Firefox, Chrome, Internet Explorer и Safari, а также многих популярных плагинов, таких как Adobe Flash, Adobe Acrobat и Java

***Меры борьбы с вредоносным ПО***

* регулярное обновление операционной системы;
* использование безопасного браузера (Opera, Mozilla Firefox) и почтового клиента (The Bat!, Mozilla Thunderbird , Sylpheed) ; Файервол (англ. firewall — огненная стена) — это браузер типа программный фильтр (существуют и программно-аппаратные файерволы), который отслеживает входящий и исходящий сетевой трафик компьютера и блокирует потенциально опасные соединения.
* установка надежного файервола (Outpost Firewall Pro, ZoneAlarm Free Firewall, Gomodo Firewall );

***Файервол*** (англ. firewall — огненная стена) — это браузер типа программный фильтр (существуют и программно-аппаратные файерволы), который отслеживает входящий и исходящий сетевой трафик компьютера и блокирует потенциально опасные соединения.

* установка антивируса и антишпионского ПО

Cуществует два режима работы антивируса:

***Режим сканера*** — это основной режим.

***Режим резидентного монитора*** — в этом режиме антивирус работает постоянно в оперативной памяти, пока работает операционная система, и налету проверяет все файлы, с которыми осуществляются какие-нибудь действия в систем

* Частое обновление антивирусных баз
* Проверка файлов в архивах

Наиболее популярные в нашей стране антивирусы: Dr.Web, антивирус Касперского. Avira AntiVir PersonalEdition Classic , avast! 4 Home Edition.

***Превентивные меры борьбы с вредоносным ПО***

Проверка на вредоносное ПО с помощью

* антишпионов (AVG Anti-Spyware Free Edition),
* антируткитов (Rootkit Unhooker),
* антикейлоггеров (Advanced Anti Keylogger)
* отключение неиспользуемых служб

***Службы (Services)*** — это приложения, которые запускаются в фоновом режиме и обеспечивают многие важные функции ОС. При стандартной установке Windows XP Professional в систему устанавливается около 80-и системных служб.

* ручная диагностика системы
  + Проверка автозагрузки файлов
  + Просмотр списка процессов

# 56. Оценка безопасности парольной защиты.

Эффективность использования пароля:

A = {ai} - алфавит, состоящий из фиксированного набора символов

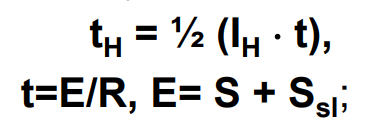
i ∊ [1, N], N - мощность алфавита

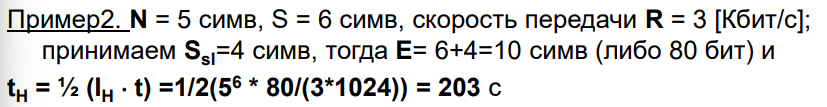
s - длина пароля H, при H = ‘12AAa!!\*’ s=8

Колличество комбинаций пароля при фиксированном N: IH = N^s



Безопасное время использования пароля:





Принимае Р - это вероятность того, что пароль будет взломан за М месяцев

Р0 - нижняя граница Р

**Р0 = n1/n2**; где n1 - число попыток взлома пароля за M месяцев, n2 - число всех возможных паролей при определенных N и s

**n1 = n11/n12**; n11 - число символов, которые можно передать по сети за М месяцев, n12 - число символов, передаваемых в одной попытке

n1 = (R\*M\*24(ч/д) \* 60(сек/мин) \* 30(д/мес)) / Е

n2 = N^s

Тогда P0 = (R\*M\*24\*60\*60\*30)/(E\*N^s)

Т.к. P>P0 , P> (R\*M\*24\*60\*60\*30)/(E\*N^s) или иначе N^s >= (4.32 \* 10^4 \*R \* M)/(E\*P) - формула Андерсена

N^s >= (2.59\*10^6\*R\*M)/(E\*P)

