Оглавление

[1. Сущность проблемы информационной безопасности систем. Характеристики и параметры ИС и ИВС. 4](#_Toc167998211)

[2. Характеристика угроз безопасности современным ИС и ИВС. 7](#_Toc167998212)

[3. Политика безопасности современных ИС и ИВС 9](#_Toc167998213)

[5. Энтропия источника сообщения. Энтропия Хартли. 11](#_Toc167998214)

[6. Двоичный симметричный канал передачи информации. 12](#_Toc167998215)

[7. Энтропия двоичного алфавита. 13](#_Toc167998216)

[8. Условная энтропия. Энтропийная оценка потерь при передаче информации. 14](#_Toc167998217)

[9. Базовые понятия криптографии. Основы теории больших чисел. Проблема дискретного логарифма. 16](#_Toc167998218)

[10. Основная теорема арифметики. Алгоритм Евклида нахождения НОД 18](#_Toc167998219)

[11. Основы модулярной арифметики. Вычеты. 20](#_Toc167998220)

[12. Обратные вычисления по модулю в криптографии. Расширенный алгоритм Евклида. 20](#_Toc167998221)

[13. Функция Эйлера в криптографии. 22](#_Toc167998222)

[14. Хеш-функция и ее свойства. Области использования хеш-функций. 23](#_Toc167998223)

[15. Общая характеристика алгоритмов хеширования классов MD и SHA. 25](#_Toc167998224)

[16. Алгоритмы хеширования класса MD. Области использования. 27](#_Toc167998225)

[17. Алгоритмы хеширования класса SHA. Области использования. 29](#_Toc167998226)

[18. Общая классификация криптографических методов защиты информации. 31](#_Toc167998227)

[20. Особенности реализации шифровальной машины Энигма. 32](#_Toc167998228)

[21. Шифр на основе аффинной системы подстановок Цезаря. 33](#_Toc167998229)

[22. Система шифрования Цезаря с ключевым словом. 34](#_Toc167998230)

[23. Шифр Виженера. 34](#_Toc167998231)

[24. Перестановочные шифры. 36](#_Toc167998232)

[25. Методы симметричного криптопреобразования. Стандарт DES. Общая характеристика. 37](#_Toc167998233)

[28. Шифровальная машина Энигма. Устройство, функционирование, криптостойкость. 43](#_Toc167998234)

[29. Сравнительная характеристика алгоритмов Lucifer, IDEA, ГОСТ 28147-89, Blowfish. 43](#_Toc167998235)

[30. Криптографические системы с открытым (публичным) ключом. Задача об укладке ранца. 45](#_Toc167998236)

[31. Управление криптографическими ключами. Алгоритм рукопожатия. 45](#_Toc167998237)

[32. Распределение ключей на основе симметричных систем. 45](#_Toc167998238)

[33. Алгоритм передачи ключа по Диффи-Хеллману. 45](#_Toc167998239)

[34. Алгоритм шифрования RSA. Реализация и криптостойкость. 46](#_Toc167998240)

[35. Алгоритм шифрования Эль-Гамаля. Реализация и криптостойкость. 47](#_Toc167998241)

[36. Потоковое шифрование. Типы. Гаммирование в потоковом шифровании. 48](#_Toc167998242)

[37. Генерация ключевой информации для потокового шифрования. Генераторы ПСП на основе регистров сдвига. 50](#_Toc167998243)

[38. Особенность шифра Вернама. 53](#_Toc167998244)

[39. Стеганографические методы защиты информации. Классификация и области использования. Метод наименее значащих бит. 54](#_Toc167998245)

[40. Понятие эллиптической кривой. Принципы построения криптосистемы на эллиптических кривых 55](#_Toc167998246)

[41. Представление и описание эллиптической кривой на основе алгебраической геометрии 56](#_Toc167998247)

[42. Арифметические операции в эллиптической криптографии 57](#_Toc167998248)

[43. ЭЦП. Назначение и свойства. 59](#_Toc167998249)

[44. ЭЦП. Основные методы генерации. Атаки на ЭЦП 60](#_Toc167998250)

[45. ЭЦП на основе симметричной криптографии 61](#_Toc167998251)

[46. ЭЦП на основе алгоритма RSA 62](#_Toc167998252)

[47. ЭЦП на основе симметричной криптосистемы и посредника 64](#_Toc167998253)

[48. ЭЦП DSS. 65](#_Toc167998254)

[49. ЭЦП на основе алгоритма Эль-Гамаля 66](#_Toc167998255)

[51. Алгоритм К. Шнорра. Стандарт ЭЦП в РБ. 67](#_Toc167998256)

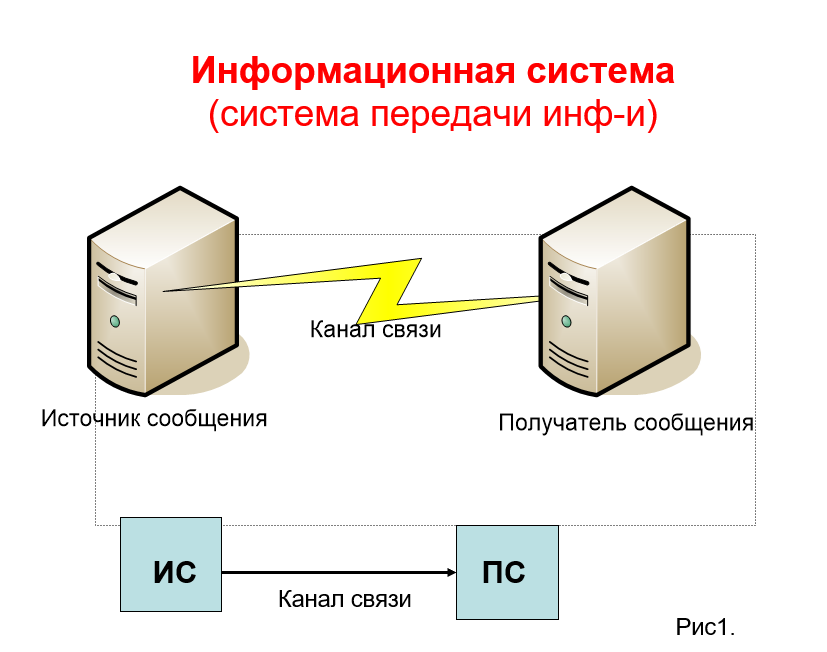
[53. Деструктивные программы. Классификация и методы нейтрализации. 69](#_Toc167998257)

[54. Оценка безопасности парольной защиты. 71](#_Toc167998258)

Вопросы

по дисциплине «Информационная безопасность», 2023-24 уч. год, летний семестр

1. Сущность проблемы информационной безопасности систем. Характеристики и параметры ИС и ИВС.
2. Характеристика угроз безопасности современным ИС и ИВС.
3. Политика безопасности современных ИС и ИВС
4. Энтропия источника сообщения. Энтропия Шеннона.
5. Энтропия источника сообщения. Энтропия Хартли.
6. Двоичный симметричный канал передачи информации.
7. Энтропия двоичного алфавита.
8. Условная энтропия. Энтропийная оценка потерь при передаче информации.
9. Базовые понятия криптографии. Основы теории больших чисел. Проблема дискретного логарифма.
10. Основная теорема арифметики. Алгоритм Евклида нахождения НОД
11. Основы модулярной арифметики. Вычеты.
12. Обратные вычисления по модулю в криптографии. Расширенный алгоритм Евклида.
13. Функция Эйлера в криптографии.
14. Хеш-функция и ее свойства. Области использования хеш-функций.
15. Общая характеристика алгоритмов хеширования классов MD и SHA.
16. Алгоритмы хеширования класса MD. Области использования.
17. Алгоритмы хеширования класса SHA. Области использования.
18. Общая классификация криптографических методов защиты информации.
19. Подстановочные шифры. Шифр Цезаря.
20. Особенности реализации шифровальной машины Энигма.
21. Шифр на основе аффинной системы подстановок Цезаря.
22. Система шифрования Цезаря с ключевым словом.
23. Шифр Виженера.
24. Перестановочные шифры.
25. Методы симметричного криптопреобразования. Стандарт DES. Общая характеристика.
26. Методы симметричного криптопреобразования. Стандарт DES. Структура одного цикла. Криптостойкость алгоритма.
27. Методы симметричного криптопреобразования. Стандарты 3DES. Реализация и криптостойкость.
28. Шифровальная машина Энигма. Устройство, функционирование, криптостойкость.
29. Сравнительная характеристика алгоритмов Lucifer, IDEA, ГОСТ 28147-89, Blowfish.
30. Криптографические системы с открытым (публичным) ключом. Задача об укладке ранца.
31. Управление криптографическими ключами. Алгоритм рукопожатия.
32. Распределение ключей на основе симметричных систем.
33. Алгоритм передачи ключа по Диффи-Хеллману.
34. Алгоритм шифрования RSA. Реализация и криптостойкость.
35. Алгоритм шифрования Эль-Гамаля. Реализация и криптостойкость.
36. Потоковое шифрование. Типы. Гаммирование в потоковом шифровании.
37. Генерация ключевой информации для потокового шифрования. Генераторы ПСП на основе регистров сдвига.
38. Особенность шифра Вернама.
39. Стеганографические методы защиты информации. Классификация и области использования. Метод наименее значащих бит.
40. Понятие эллиптической кривой. Принципы построения криптосистемы на эллиптических кривых
41. Представление и описание эллиптической кривой на основе алгебраической геометрии
42. Арифметические операции в эллиптической криптографии
43. ЭЦП. Назначение и свойства.
44. ЭЦП. Основные методы генерации. Атаки на ЭЦП
45. ЭЦП на основе симметричной криптографии
46. ЭЦП на основе алгоритма RSA
47. ЭЦП на основе симметричной криптосистемы и посредника
48. ЭЦП DSS.
49. ЭЦП на основе алгоритма Эль-Гамаля
50. ЭЦП на основе эллиптической кривой.
51. Алгоритм К. Шнорра. Стандарт ЭЦП в РБ.
52. Протокол Kerberos.
53. Деструктивные программы. Классификация и методы нейтрализации.
54. Оценка безопасности парольной защиты.
55. Сущность проблемы информационной безопасности систем. Характеристики и параметры ИС и ИВС.



Из книги

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

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

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

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

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

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

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

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

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

Другое определение: *информационная безопасность* – все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности информации или средств ее обработки:

1) конфиденциальность (confidentiality) – состояние информации, при котором доступ к ней осуществляют только субъекты, имеющие на нее право;

2) целостность (integrity) – избежание несанкционированной модификации информации;

3) доступность (availability) – избежание временного или постоянного сокрытия информации от пользователей, получивших права доступа.

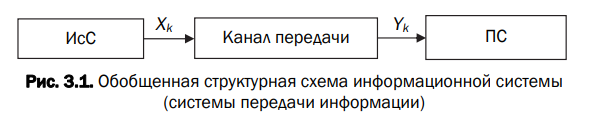
**Безотказность** – это свойство технического объекта непрерывно сохранять работоспособное состояние в течение некоторого времени.

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

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

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

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. Таким образом, простейшая информационная система состоит из трех элементов: источника сообщения, канала передачи сообщения и получателя сообщения.



Кодирование в широком смысле – преобразование сообщения в сигнал.

Кодирование в узком смысле – представление исходных знаков, называемых символами, в другом алфавите. 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 преобразование.

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

1) обеспечение конфиденциальности (секретности) – относится к защите информации от несанкционированного доступа по секретному ключу;

2) обеспечение анонимности (неотслеживаемости);

3) обеспечение аутентификации информации и источника сообщения.

***Основные характеристики и параметры двоичных систем.***

1. Алфавит – конечная совокупность символов (знаков), с помощью которых можно представить любое сообщение в ИС: А{ai} Символы алфавита будем обозначать через {аi}, где 1 ≤ i ≤ N; N – мощность алфавита. Минимальное число элементов алфавита Nmin = 2, А = {0, 1} – двоичный код

2. Мощность алфавита – количество символов, составляющих алфавит: N(A)

3. Вероятность появления определенного символа. Или Вероятность того, что произвольный символ ξ произвольного документа (текст, база данных, текст программы) будет буквой «аi»: P(ξ = аi) = p(аi)

1. Характеристика угроз безопасности современным ИС и ИВС.

Страница 12 практикум 1

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*.

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

• *преднамеренные*:

− 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).

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

Основные факторы (угрозы):

1) действия злоумышленника;

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) несанкционированное распространение материальных носителей за пределами организации;

6) разглашение информации компетентными людьми;

7) утеря носителей информации;

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

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

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

11) отказы системы электроснабжения;

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

Несанкционированный доступ с помощью деструктивных программных средств осуществляется, как правило, через компьютерные сети.

Что-то из ответов прошлых лет

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

Одним из важнейших дестабилизирующих работу ИВС факторов считают электромагнитные и ионизирующие излучения. Ионизирующие излучения также могут иметь естественную (солнечная радиация) и искусственную (изотопы урана и тория излучают даже пластмассы) природу.

Классификацию вредоносного ПО можно представить следующим образом: 1) вирусы (viruses); 2) черви (worms); 3) кейлоггеры (keyloggers); 4) трояны (trojans); 5) боты (bots); 6) снифферы (sniffers); 7) руткиты (rootkits).

Выделяют несколько основных угроз безопасности, возникающих при использовании бесплатных точек доступа Wi-Fi: 1) сети, организованные хакерами, могут выдавать себя за вполне легальные бесплатные точки доступа; 2) атаки с помощью вредоносного ПО компьютера, подключенного к этой точке доступа; 3) сниффинг; 4) хищение персональной информации методом «человек посередине» (man in the middle); 5) Фишинг.

1. Политика безопасности современных ИС и ИВС

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

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: а) международные договоры Республики Беларусь; б) Конституция Республики Беларусь; в) законы Республики Беларусь, например Закон Республики Беларусь от 10 ноября 2008 г. № 455-З «Об информации, информатизации и защите информации»; г) указы Президента Республики Беларусь; д) постановления Правительства Республики Беларусь; е) нормативные правовые акты министерств и ведомств; ж) нормативные правовые акты субъектов, органов местного самоуправления и т. д.

2. Международные стандарты, например: а) BS 7799-1:2005 – Британский стандарт BS 7799 Part 1 – Code of Practice for Information Security Management; б) BS 7799-2:2005 – Британский стандарт BS 7799 Part 2 – Information Security Management – Specification for Information Security Management Systems; в) ISO/IEC 17799:2005; г) ISO/IEC 27001:2005; д) ISO/IEC 27002; е) ISO/IEC 27005; ж) SO/IEC 27040:2015.

2)организационно-технические и режимные меры и методы (политика информационной безопасности);

− организацию охраны, режима, работу с кадрами, с документами;

− 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: а) средства авторизации; б) аудит;

2) системы мониторинга: а) системы мониторинга сетей; б) анализаторы протоколов;

3) антивирусные средства: а) антивирусные программы; б) программные и иные антиспамовые средства; в) межсетевые экраны;

4) криптографические средства: а) шифрование данных; б) электронная цифровая подпись;

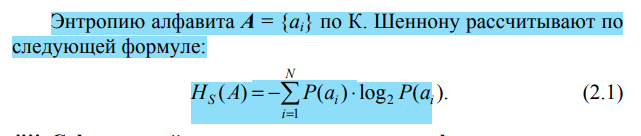
5) системы бесперебойного питания;

6) системы аутентификации: а) пароль; б) ключ доступа (физический или электронный); в) биометрия (анализаторы отпечатков пальцев, анализаторы сетчатки глаза, анализаторы голоса, анализаторы геометрии ладони и др.).

1. Энтропия источника сообщения. Энтропия Шеннона.

Лекция: 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.

Понятие энтропии применительно к ИС ввел К. Шеннон



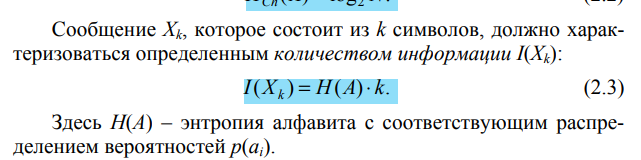
Чтобы вычислить энтропию по Шеннону нужно знать мощность и вероятность появления символа.

Энтропия по Шеннону предполагает наличие разных вероятностей появления символов.

Знак минус в формуле Шеннона не означает, что количество информации в сообщении – отрицательная величина. Объясняется это тем, что вероятность р, согласно определению, меньше единицы, но больше нуля. Так как логарифм числа, меньшего единицы, т.е. log pi – величина отрицательная, то произведение вероятности на логарифм числа будет положительным.

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

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

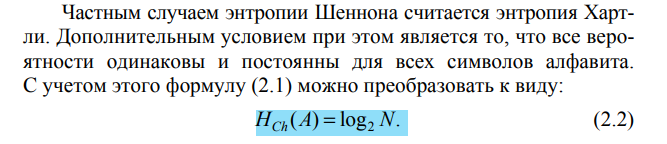


Условная энтропия – это кол-во потерянной информации, приходящейся на 1 символ сообщения

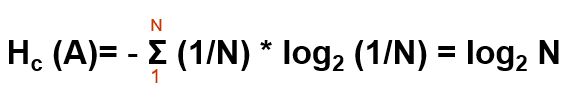
Эффективная энтропия – 1 - условная энтропия (показывает кол-во реально переданных данных)

Чем больше мощность алфавита, тем информативнее каждый символ этого алфавита, поскольку **энтропия** — это количество информации, соответствующей одному символу алфавита.

1. Энтропия источника сообщения. Энтропия Хартли.



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

Вывод формулы из лекции 

1. Двоичный симметричный канал передачи информации.

Чатик

***Двоичный симметричный канал (ДСК)*** – это простейший канал связи, на вход которого подаются двоичные символы (0 или 1), где каждый передаваемый бит может быть искажен с фиксированной вероятностью.

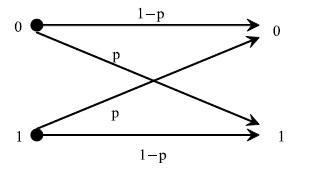
Основные характеристики:

**Двоичный канал**: Канал передает только два символа - 0 и 1

**Симметричность**: Вероятность ошибки одинакова для каждого бита. Это означает, что вероятность того, что переданный "0" станет "1" (и наоборот), одинакова и равна *p*, где 0≤*p*≤0.5

**Фиксированная вероятность ошибки**: Вероятность *p* называется вероятностью ошибки канала. Вероятность правильной передачи бита равна 1−*p*

*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).*

При передаче сообщения по ДСК в каждом бите сообщения с вероятностью p может произойти ошибка, независимо от наличия ошибок в других битах. Ошибка заключается в замене знака 0 на 1 или 1 на 0.

Если вероятность ошибки в ДСК отлична от 0 (р > 0), переданное сообщение может содержать ошибки: Хk ≠ Yk. Количество информации в таком сообщении при его передаче по ДСК будет определяться не энтропией двоичного алфавита (в соответствии с выражением (2.3)), а эффективной энтропией Hе(A) алфавита или пропускной способностью канала:



1. Энтропия двоичного алфавита.

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

А{0, 1} - алфавит, N=2 -мощность алфавита

Пусть вероятность встречи 0 будет:

P(ξ = 0) = p(0), а вероятность 1 будет P(ξ = 1) = p(1)

Энтропии Шеннона:

Энтропия двоичного алфавита(просто подставили в формулу вероятности):

H(A2) = -р(0)\*log2(р(0)) - р(1)\*log2(р(1) )

Сумма вероятностей должна быть равна 1

р(0) + р(1) = 1

Значит можем заменить р(0) = 1 – р(1)

Подставляем в уравнение вместо p(0):

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

Если я правильно помню, то для решения уравнения нужно решить дифференциальное уравнение:

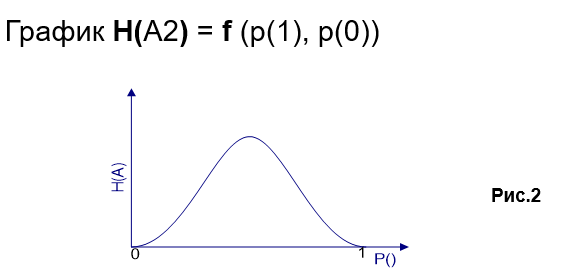
d H(A2)/ d (р(1)) = 0

В итоге(с помощью сложнейших математических вычислений калькулятором) получаем( вместо р(1) будет х):

р(1) = 0.5

р(0) = 0.5

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



По формуле Хартли:

Hc (A)= log2 N

H(A2)= log2 2 = 1.

В общем и целом энтропия двоичного алфавита равна 1

\*Энтропия двоичного алфавита может принимать значения в диапазоне от 0 до 1 бита. Этот диапазон определяется вероятностями появления каждого символа алфавита.

**Вопрос**: каким условиям соответствует максимальное значение энтропии двоичного алфавита? – Энтропия *H*(*X*) максимальна, когда вероятности обоих символов равны, то есть *p*=0.5

1. Условная энтропия. Энтропийная оценка потерь при передаче информации.

**Определение***.* *Условной энтропией Источника дискретного сообщения* **X** называем величину

**H(X|Y) = P(Y=0)H(X|Y=0) + P(Y=1)H(X|Y=1) = - р log2 р - q** **log2 q**

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

**Пример 5**. Пусть известно, что Р(*Х=0*) = Р(*Х=1*) =0.5 и р=0.01.

Из (14) 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 случая из примера 5 **Не** = 0.919 *бит*

Если вероятность ошибки в ДСК отлична от 0 (р > 0) (изначально передавали один символ, а при получении символ заменился на другой), переданное сообщение может содержать ошибки: Хk ≠ Yk. Количество информации в таком сообщении при его передаче по ДСК будет определяться не энтропией двоичного алфавита (в соответствии с выражением (2.3)), а эффективной энтропией Hе(A) алфавита или пропускной способностью канала:



p — вероятность того, что символ будет передан с ошибкой

q — вероятность того, что произвольный символ будет передан правильно

**Математическая основа** — теорема Байеса. Совместная вероятность случайных событий А и В:

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

P(A|B)=

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

Условная энтропия показывает потери информации, приходящуюся на 1 бит, а эффективная энтропия показывает, сколько реально информации передает каждый бит с учётом потерь.

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

**Задача??** Если скорость передачи 100Мбит/с и мы считаем, что это хороший канал, то при вероятности ошибки 10-5, сколько ошибочных бит в секунду будет передано?

100Мбит — это 100\*106, за одну секунду будет передано 108 бит. Чтобы получилась вероятность ошибки 10-5, на нужно число ошибочных бит поделить и мы должны получить величину (дома досчитать).

Число ошибочных битов в секунду *N* можно рассчитать по формуле:

*N*=скорость передачи данных×вероятность ошибки

108 \*10-5 = 103 =1000

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

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

Эффективная энтропия: *He=0,919 бит.*

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

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

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

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

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

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

Из лекции

**Криптография** — наука (раздел математики) о методах обеспечения конфиденциальности или безопасности информации, связанной с тайной, интегральностью данных и идентификацией.

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

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

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

М – открытый текст (сообщение)

С – шифрограмма

К – множество ключей

***Основы теории чисел***

В основе современной криптографии лежит теория чисел.

Определение 1. Множество всех целых чисел (обозначим буквой Z) есть набор всех ***действительных чисел*** без дробной части: {..., –3, –2, –1, 0, 1, 2, 3, ...}.

Определение 2. ***Натуральные числа*** являются подмножеством целых чисел и образуют множество N: {1, 2, 3, ...}.

Определение 3. ***Делимость*** – одно из основных понятий теории чисел. Если для некоторого целого числа a и натурального числа b существует целое число q, такое, что bq=a, то говорят, что число a делится на b. В этом случае b называется делителем числа a, а a называется кратным числа b. При этом используются следующие обозначения:

Определение 4. Делитель a называется *собственным* *делителем* числа b, если 1 < |a| < |b|, и *несобственным* – в противном случае.

*Свойство 1 собственного делителя*: положительный наименьший собственный делитель составного числа n не превосходит √n.

*Свойство 2 собственного делителя*. Положительный наименьший собственный делитель составного числа n есть простое число

Определение 5. 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.

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

Определение 6. Натуральное число n называется простым, если n > 1 и не имеет положительных делителей, отличных от 1 и n

*Перечислим несколько важных свойств простых чисел.*

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

Порядок записи сомножителей после последнего знака равенства соответствует канонической форме.

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**

Определение 8. Если два простых числа отличаются на 2, то их называют *числами-близнецами*. Таких чисел не очень много. Например, ими являются 5 и 7, 29 и 31, 149 и 151.

«решето Эратосфена»

***Проблема дискретного логарифма***

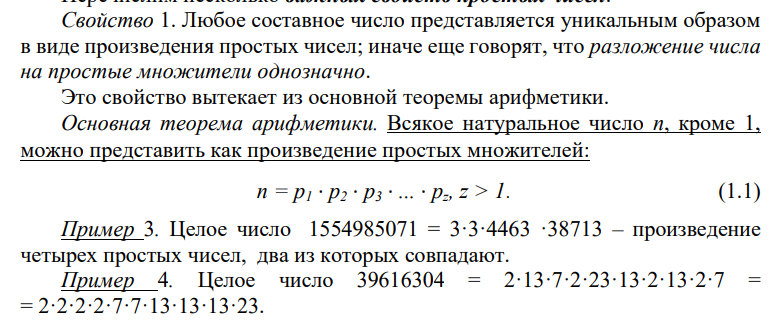
Если известны три некоторых числа (*а, х, n*), то достаточно легко можно вычислить число y: *y = ax mod n.*

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*.

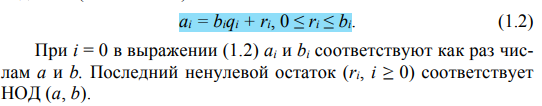
Решения существуют не для всех дискретных логарифмов (напомним, что речь идет только о целочисленных решениях).

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.

1. Основная теорема арифметики. Алгоритм Евклида нахождения НОД



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

Определение 10**. Взаимно простыми** являются целые числа, наибольший общий делитель которых равен 1.

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

*аu + bv = 1.*

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

*аu + bv = d.*

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

1234 = 54 \* 22 + 46

54 = 46 \* 1 + 8

46 = 8 \* 5 + 6

8 = 6 \* 1 + 2

6 = 2 \* 3 + 0

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

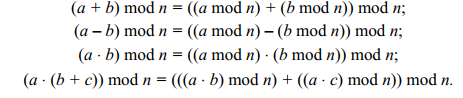
1. Основы модулярной арифметики. Вычеты.

Понятие «модулярная арифметика» ввел немецкий ученый К. Ф. Гаусс. В этой арифметике мы интересуемся остатком от деления числа а на число n (n – натуральное число и n > 1). Если таким остатком является число b, то можно записать: a ≡ b (mod n), или a ≡ b mod n. Такая формальная запись читается как «a сравнимо с b по модулю n».

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

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

**Правила модулярной арифметики**



Существуют *методы ускорения таких вычислений*.   
***Пример.*** Нужно вычислить модуль ***n*** некоторого числа ***а***в 8 степени:

*а*8 mod *n.*  
Понятно, что семь операций умножения числа ***а***могут дать огромное число. Порядок чисел, которыми оперирует вычислитель, можно значительно уменьшить, если воспользоваться промежуточными вычислениями по модулю: ((*а*2 mod *n*)2 mod *n*)2 mod *n.*  
***Пример.*** Предположим, что показатель степени не является степенью 2. Пусть это будет, например, 25, т. е. необходимо вычислить *а*25 mod *n.*  
После понятных рассуждений последовательность операций можем представить в виде следующей *аддитивной цепочки*:  
 *а*25 mod *n =* (*а* ⋅ *а*24) mod *n =* ((((*а*2 ⋅ *а*)2)2)2 *а*) mod *n =*  
 *=*(((((((*а*2 mod *n*) *а*) mod *n*)2 mod *n*)2 mod *n*)2 mod *n*)*а*) mod *n*.

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

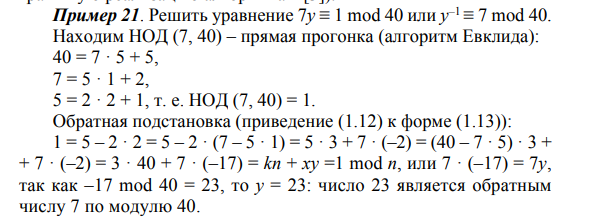


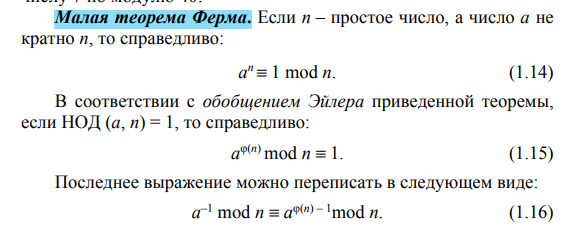
Нахождение чисел, обратных по модулю, легко реализуется с помощью расширенного алгоритма Евклида

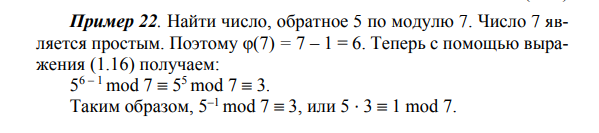
Теорема 2. Если НОД (a, b) = d, то справедливо следующее соотношение (соотношение Безу):

*аu + bv = d. (1.4)*

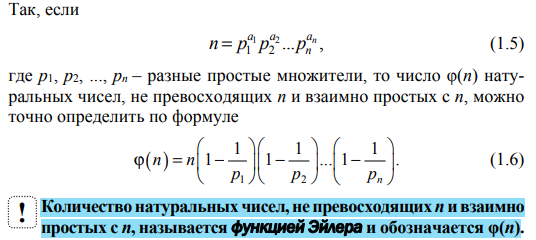
Формула (1.4) называется также реализацией «***расширенного алгоритма Евклида***». Этот алгоритм состоит из двух этапов: собственно алгоритма Евклида и вычислений на основе обратных подстановок или последовательного выражения остатков в каждом из шагов предыдущего этапа с соответствующим приведением подобных на каждом шаге.

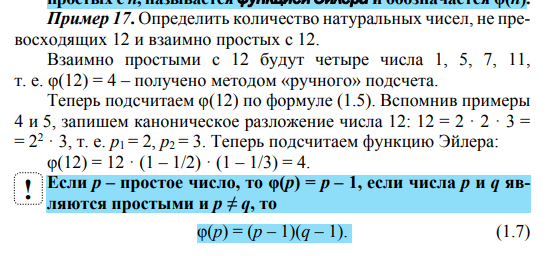


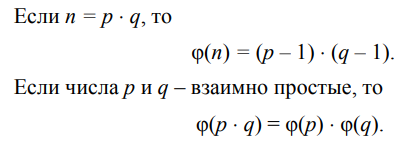


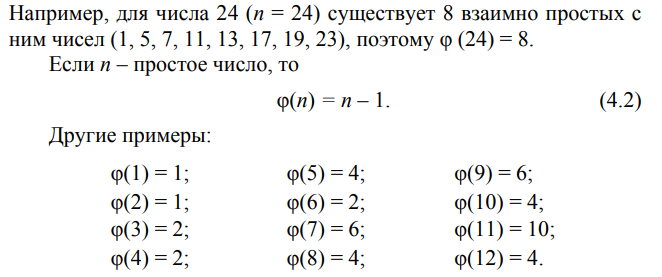


1. Функция Эйлера в криптографии.





Книга Урбановича стр 51



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 криптографии

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

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

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

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

Все существующие функции хеширования можно разделить на два больших класса:

• бесключевые хеш-функции, зависящие только от сообщения;

• хеш-функции с секретным ключом, зависящие как от сообщения, так и от секретного ключа

Криптографическая хеш-функция – это специальный класс хеш-функций, который имеет различные свойства, необходимые для решения задач в области криптографии

***Основные задачи, решаемые с помощью хеш-функций:***

• аутентификация (хранение паролей);

• проверка целостности данных;

• защита файлов;

• обнаружение зловредного ПО;

• криптовалютные технологии

К основным свойствам хеш-функций можно отнести следующие.

***Свойство 1.*** ***Детерминированность***: независимо от того, сколько раз вычисляется H(M), M – const, при использовании одинакового алгоритма код хеш-преобразования h всегда должен быть одинаковым.

***Свойство 2. Скорость вычисления хеша h***

***Свойство 3. Сложность обратного вычисления:*** для известного *H(М)* невозможно (практически) определить *М*. Это важнейшее свойство хеш-функции для криптографических применений – свойство *односторонности преобразования.*

***Свойство 4.*** Даже минимальные изменения в хешируемых данных *(М ≠ М')* должны изменять хеш: *Н(M) ≠ Н(М')*.

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

***Коллизией хеш-функции Н*** называют ситуацию, при которой различным входам (в общем случае – х и у или *М ≠ М'*) соответствует одинаковый хеш-код: *H(x) = H(y)* или *H(М) = = H(М').*

Зная М, трудно найти такое М' (М ≠ М'), для которого H(М) = H(М'). Если последнее равенство выполняется, то говорят о *коллизии 1-го рода*.

Если случайным образом выбраны два сообщения (М и М'), для которых H(М) = H(М'), говорят *о коллизии 2-го рода*.

Из книжки Урбановича

***Коллизией 1-го рода*** считаем ситуацию, при которой для данного сообщения М и для иного произвольного сообщения М′ (М ≠ М′) имеем h(М) = h(М′), вычисленные с использование одной и той же хеш-функции (или алгоритма хеширования).

Приведем для лучшего понимания простую аналогию. Из закрытой емкости с разноцветными шарами мы выбрали шар определенного цвета, например красного (М). Далее наугад достаем другой шар (М′). Если он будет окрашен в тот же цвет, что и первый (в красный), то мы будет считать такое совпадение коллизией 1-го рода.

***Коллизией 2-го рода*** считаем ситуацию, при которой для двух произвольных сообщений М и М′ (М ≠ М′) имеем h(М) = h(М′), вычисленные с использование одной и той же хеш-функции (или алгоритма хеширования).

ледуя вышеописанной аналогии, в данном случае одновременно извлекаются два шара. Коллизия наступает при их одинаковой окраске.

***Мерой криптостойкости хеш-функции считается вычислительная сложность нахождения коллизии.***

***Для хеш-функций одним из основных средств поиска коллизий является метод, основанный на известной статистической задаче – «парадоксе дня рождения».***

*Немного подробнее про парадокс «дней рождений»*

Основной постулат **парадокса «дней рождения»** гласит: в группе минимум из 23 человек с вероятностью более 0,5 день рождения одинаков хотя бы у двух членов группы. При этом предполагается, что:

• в этой группе нет близнецов;

• люди рождаются независимо друг от друга, т. е. дата (день) рождения любого человека не влияет на дату рождения другого;

• люди рождаются равномерно и случайно, т. е. люди с равной вероятностью могут рождаться в любой день года;

В атаке «дней рождения» m соответствует количеству календарных дней в году, а М – множеству людей, составляющих группу. Люди «хешируются» в их дни рождения, которые могут быть одним из значений m.

При атаке «дней рождения» злоумышленник будет случайным образом подбирать Мi и Мj и сохранять пары их хешей, пока не найдет двух значений, при которых h(Мi) = h(Мj).

Иначе говоря, стоит задача отыскания наименьшего n, при котором хеши двух значений m будут «одним днем рождения».

1. Общая характеристика алгоритмов хеширования классов MD и SHA.

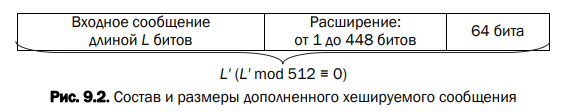
Алгоритмы семейства SHA (SHA – Secure Hash Algorithm) являются в настоящее время широко распространенными. По существу, во многих случаях завершился переход от SHA-1 к стандартам версии SHA-2. SHA-2 – собирательное название алгоритмов SHA-224, SHA-256, SHA-384 и SHA-512. SHA-224 и SHA-384 являются, по сути, аналогами SHA-256 и SHA-512 соответственно

Базовые алгоритмы обоих рассматриваемых семейств (MD и SHA) условно можно разделить на 5 стадий:

***1. расширение входного сообщения;***

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.

**Пример:** пусть исходное сообщение, или первообразная, будет словом «CRYPTO» (М = «CRYPTO»), или в кодах ASCII – это будут десятичные (67-82-89-80-84-79) и соответствующие двоичные (01000011-01010010-01011001-01010000-01010100-01001111; здесь для отделения чисел используются дефисы) числа. Как видим, длина L хешируемого сообщения равна 48 битам. Эту длину мы должны расширить до 448 битов, добавив одну «1» и 399 «0». В последнюю часть из 64 битов полученного 512-битного модуля (L' = 512) мы запишем справа двоичное представление числа L = 48: 110000. В остальные 58 разрядов (из 64) мы впишем «0». После этого полученный расширенный блок делим на 16 32-разрядных подблоков:

***2. разбивка расширенного сообщения на блоки;*** На этой стадии каждый 512-битный блок разделяется на 16 32-разрядных слов (16 ⋅ 32 = 512). Именно 32-разрядное слово является основной информационной и структурной единицей рассматриваемого алгоритма.

***3. инициализация начальных констант;***

***4. обработка сообщения поблочно (основная процедура алгоритма хеширования);***

основная операция заключается в циклической (пораундовой или поэтапной) обработке 512-битных блоков. Таких циклов может быть 3 (как в MD-4), или 4 (как в MD-5), или более. В каждом цикле используется своя нелинейная функция (обычно обозначаемая по порядку F, G, H, …), зависящая от текущего состояния 4 (в MD), 5 (в SHA-1), 8 (SHA-256) и т. д. переменных, начальные состояния которых известны, а текущие – зависят от выполненных операций над хешируемым сообщением

Каждый регистр содержит 32-разрядную беззнаковую величину, поэтому размер хеша будет равен 128 (32 ⋅ 4 = 128) битам.

***5. вывод результата.*** результат вывода содержимого регистров A, B, C, D, точнее говоря, конкатенация

|  |  |  |
| --- | --- | --- |
|  | MD5 | SHA1 |
| Max длина входного сообщения | 264 – 1 битов | 264 – 1 битов |
| Длина хеша | 128 битов | 160 битов |
| Пораундовая обработка 512-битных блоков | 4 цикла по 16 операций в каждом | 4 цикла по 20 операций в каждом |
| Константы | *a, b, c, d* | *a, b, c, d, e* |
|  | Быстрее | Медленнее |
|  |  | Надежнее |

1. Алгоритмы хеширования класса MD. Области использования.

Базовые алгоритмы обоих рассматриваемых семейств (MD и SHA) условно можно разделить на 5 стадий:

• расширение входного сообщения;

• разбивка расширенного сообщения на блоки;

• инициализация начальных констант;

• обработка сообщения поблочно (основная процедура алгоритма хеширования);

• вывод результата.

Из книжки Урбановича

Основные ***отличия между алгоритмами MD4 и MD5*** заключаются в следующем.

1. MD5 имеет на один раунд больше – 4 против 3 у MD4.

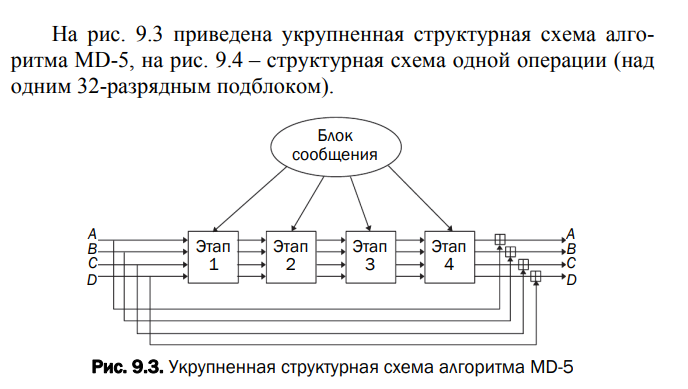
2. Чтобы уменьшить влияние входного текста, была введена уникальная константа для каждого раунда: t[i].

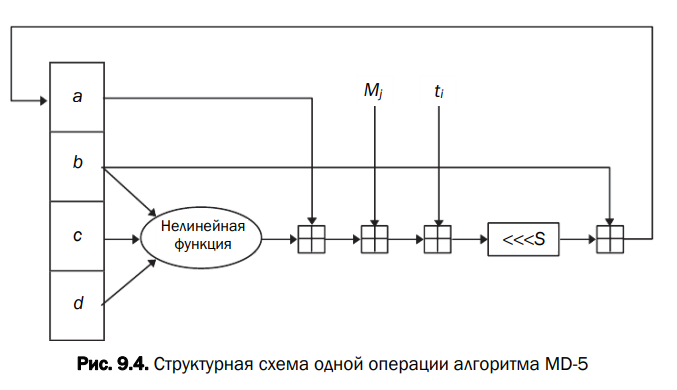
3. Во втором раунде функция G(x, y, z): (x AND y) OR (x AND z) OR (y AND z) заменена на (x AND z) OR ((y AND (NOT z)) для того, чтобы сделать G(x, y, z) менее симметричной.

4. На каждом шаге используется значение, полученное на предыдущем шаге. Это дает более быстрое изменение результата при изменении входных данных.

5. Для ускорения лавинного эффекта значения циклического сдвига в каждом раунде оптимизированы. Для этих же целей количество сдвигов различается от раунда к раунду и выбрано так, чтобы еще более увеличить этот эффект.

6. Изменен порядок, в котором обрабатываются слова в раундах 2 и 3 для того, чтобы сделать их менее похожими друг на друга.





Здесь знак «+» в квадратной фигуре соответствует операции сложения по модулю 232.

Главный модуль (рис. 9.3) состоит из четырех похожих этапов (у MD-4 было только три этапа). На каждом этапе 16 раз используются различные операции. Каждая операция представляет собой нелинейную функцию над тремя из a, b, c и d. Затем она добавляет этот результат к четвертой переменной, подблоку текста Мj и константе ti. Далее результат циклически сдвигается вправо на переменное число s битов и добавляет результат к одной из переменных a, b, c и d. Наконец, результат заменяет одну из этих переменных

Результатом хеширования h является конкатенация последних значений указанных переменных, т. е. 32 · 4 = 128 битов

***Применяется для:***

1. Проверка целостности скачанных файлов: некоторые программы идут вместе со значением хеша, чтобы пользователи могли убедиться, что файл не был изменен или поврежден.

2. Цифровые подписи и сертификаты: MD5 применялся в криптографических протоколах, таких как SSL и TLS, для хеширования сообщений. Однако сейчас эти применения устарели из-за безопасности.

3. Хранение паролей

4. Идентификация и поиск дубликатов файлов: MD5 хеши создавались для уникальной идентификации файлов, документов и других объектов.

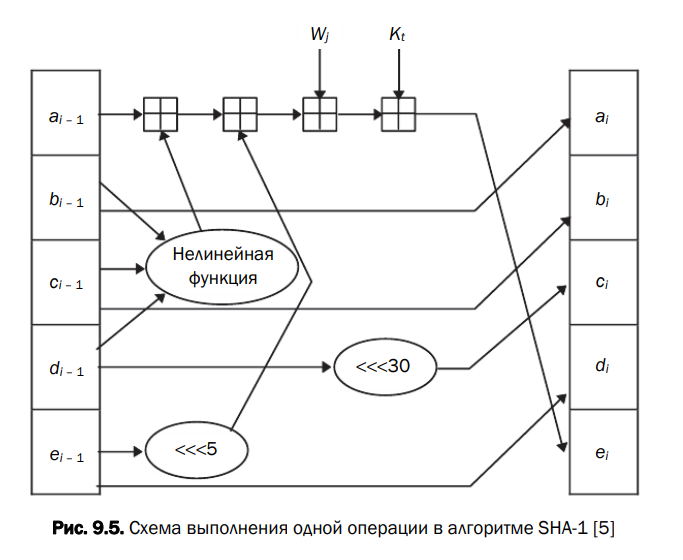
5. Системы контроля версий: раньше применялись

1. Алгоритмы хеширования класса SHA. Области использования.

Как и MD5, SHA являются, по сути, улучшенными продолжениями MD4 (в наибольшей степени это относится к SHA-1)

Главный цикл состоит из четырех раундов, каждый из которых включает по 20 операций (вспомним – в алгоритмах MD таких операций 16). Каждая такая операция предусматривает вычисление нелинейной функции над тремя переменными из набора а, b, с, d, e. После этого производятся операции сдвига и сложения, аналогичные вышерассмотренным, с использованием константы t

Алгоритм SHA, по сути, совпадает с алгоритмом MD4, отличаясь наличием расширяющего преобразования, дополнительным циклом обработки и улучшенным лавинным эффектом. Алгоритм MD5 – это улучшенный MD4.



Блок сообщения трансформируется из 16 32-битных слов (от M0 по M15) в 80 32-битных слов (W0, …, W79) с помощью следующего алгоритма:

После обработки всех 512-битных блоков выходом является 160-битный дайджест сообщения в виде конкатенации последних значений переменных a, b, c, d, e.

в алгоритмах MD-5 SHA-1 результат текущего действия прибавляется к результату предыдущего. Это направлено на усиление лавинного эффекта

***Области использования алгоритмов хеширования класса SHA***

1. Проверка целостности данных

2. Цифровые подписи и сертификаты: SSL/TLS сертификаты: Хеши SHA-256 и выше используются в цифровых сертификатах для безопасного обмена ключами и установления защищенных соединений.

3. для хеширования паролей перед их хранением в базах данных, Аутентификация пользователей.

4. Блокчейн и криптовалюты

5. Системы контроля версий: Алгоритмы SHA применяются для создания уникальных идентификаторов коммитов и версий файлов, что позволяет точно отслеживать изменения в репозиториях кода

6. Сетевая безопасность: VPN и IPsec: В протоколах виртуальных частных сетей и IPsec используются хеши SHA для аутентификации и обеспечения целостности передаваемых данных.

7. Аутентификация и идентификация

1. Общая классификация криптографических методов защиты информации.

**Классификация систем шифрования (шифров)(лекция)**

1. На основе процедуры шифрования:

* Ш. подстановочные;
* Ш. перестановочные.

1. На основе генерирования и использования ключа:

* Ш. блочные;
* Ш. поточные.

1. На основе типа ключа:

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

**Подстановочный шифр** - замена символов открытого текста соответственно символами того же или иного алфавита - подстановка.

**Перестановочный шифр** – основан на перестановке символов открытого текста.

Наибольшая уязвимость подстаночных и перестановочных шифров – сохранение в зашифрованных документах вероятностных свойств символов используемого алфавита Пример. Частотные свойства английского алфавита

1. 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

Приведенные утверждения справедливы для следующих типов

подстановочных шифров:

• *моноалфавитных* (шифры однозначной замены или простые

подстановочные) – Цезаря;

• *полиграммных* (одна подстановка соответствует сразу нескольким символам исходного текста) – шифр Порты;

• *омофонических* (однозвучные шифры или шифры многозначной замены);

• *полиалфавитных* (шифры состоят из нескольких шифров однозначной замены) – Таблица Трисемуса, Шифр Виженера

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

Для зашифрования

y ≡ x + k mod N,

где x, y – индекс (порядковый номер, начиная с 0) символа в используемом алфавите; k – ключ.

Для расшифрования сообщения Сi необходимо произвести расчеты, обратные выражению (2.1), т. е.

х ≡ у – k mod N.

Существуют различные модификации шифра Цезаря, в частности, Атбаш и лозунговый шифр.

Атбаш. Этот шифр состоит в замене каждой буквы другой буквой, которая находится в алфавите на таком же расстоянии от конца алфавита, как оригинальная буква – от начала.

***Одним из существенных недостатков моноалфавитных шифров является их низкая криптостойкость. Зачастую метод криптоанализа базируется на частоте встречаемости букв исходного текста.***

1. Особенности реализации шифровальной машины Энигма.

Из практикума хз

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).

«Энигма» состоит из 5 основных блоков:

• панели механических клавиш 1 (дают сигнал поворота роторных дисков);

• трех (или более) роторных дисков 2, каждый имеет контакты по сторонам, по 26 на каждую, которые коммутируют в случайном порядке; по окружности нанесены буквы латинского алфавита либо числа;

• рефлектора 3 (имеет контакты с крайним слева ротором);

• коммутационной панели 4 (служит для того, чтобы дополнительно менять местами электрические соединения (контакты) двух букв);

• панели в виде электрических лампочек 5; индикационная панель с лампочками служит индикатором выходной буквы в процессе шифрования.

Конкретный механизм мог быть разным, но общий принцип был таков: при каждом нажатии на клавишу самый правый ротор сдвигался на одну позицию, а при определенных условиях сдвигались и другие роторы. Движение роторов приводило к различным криптографическим преобразованиям при каждом следующем нажатии на клавишу на клавиатуре, т. е. зашифрование/расшифрование сообщений основано на выполнении ряда замен (подстановок) одного символа другим

***Как мы отмечали выше, «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.***

Оценка криптостойкости «Энигмы»

1. Шифр на основе аффинной системы подстановок Цезаря.

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.

Аффинный шифр на основе системы подстановок Цезаря является более сложным и безопасным, чем классический шифр Цезаря, благодаря использованию линейного преобразования. Однако, несмотря на эту усложненность, он остается уязвимым к простым криптоанализам, таким как анализ частот.

1. Система шифрования Цезаря с ключевым словом.

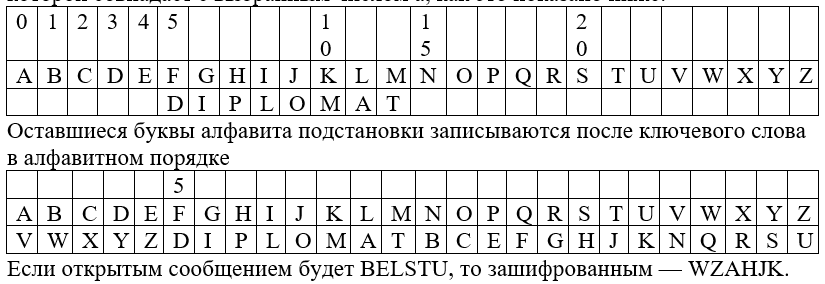
Система шифрования Цезаря с ключевым словом (лозунгом). Также является одноалфавитной системой подстановки. Особенностью этой системы является использование ключевого слова (лозунга) для смещения и изменения порядка символов в алфавите подстановки (желательно, чтобы все буквы ключевого слова были различными).

Ключевое слово пишется в начале алфавита подстановки. 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.

Пример

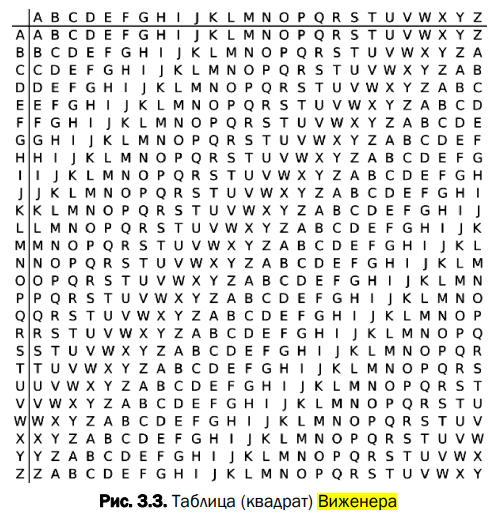
Выберем некоторое число a и слово или короткую фразу в качестве *ключевого слова.* Пусть выбраны слово DIPLOMAT в качестве ключевого и число а=5.

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



1. Шифр Виженера.

В шифре Виженера 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 предыдущей циклическим сдвигом последней на 1 символ влево. Таким образом, при мощности алфавита (английского языка), равной 26, необходимо выполнить последовательно 25 сдвигов для формирования всей таблицы.



То есть ключ мы смотрим по вертикали(до черты), а сообщение по горизонтали и на их пересечении будет буква шифртекста. Например, если первым символом сообщения будет символ В, а первым символом ключа будет символ Т, то первым символом шифртекста будет символ U.

Шифрование происходит на основе формулы (3.3).

*y = (x + k) mod N,*

где x-порядковый номер начиная с 0 в алфавите исходного сообщения, а k--порядковый номер начиная с 0 в алфавите для ключа

то есть B=1, T=19

*y(B)=(1+19)mod26=20 = U*

При расшифровании следует использовать формулу (3.4).

*x = (y – k) mod N,*

*x(U)=(20-19)mod26=1=B*

Криптостойкость

1. Анализ частот: Поликалфавитная природа шифра Виженера затрудняет анализ частот, поскольку один и тот же символ открытого текста может шифроваться по-разному в зависимости от позиции ключа. Это существенно усложняет криптоанализ по сравнению с моноалфавитными шифрами.

Его криптостойкость сильно зависит от длины и случайности ключа. При использовании достаточно длинного и случайного ключа шифр Виженера может обеспечить более высокий уровень безопасности

1. 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

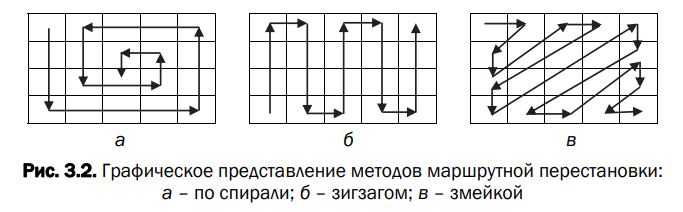
В классической криптографии шифры перестановки делятся на два подкласса:

• 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.

***1. Шифры одинарной перестановки***

* *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. Для использования на практике рассмотренный метод зашифрования/расшифрования не очень удобен. При больших значениях n приходится работать с таблицами, состоящими из большого числа столбцов.
* *Шифры простой блочной перестановки* – Указанные шифры 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* – В ячейки таблицы 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



* *Шифр вертикальной перестановки* – является разновидностью шифра маршрутной перестановки. При этом 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)

***2. 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.



Криптостойкость перестановочных шифров зависит от длины ключа и сложности перестановок. [Увеличение размера ключа повышает криптостойкость, а соответственно, удлинение блоков перестановки также способствует улучшению стойкости](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80)ы

1. Методы симметричного криптопреобразования. Стандарт DES. Общая характеристика.

Симметричная криптография Безопасность определяется:

•стойкостью (криптостойкостью) алгоритма —, в малой степени,

•длиной ключа.

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

В симметричных системах Отправитель и Получатель используют один и тот же ключ, К.

**Симметричной криптосистемой** называется криптосистема, в которой для шифрования, и для дешифрования используется один и тот же ключ.

Симметричные алгоритмы подразделяются на два подкласса.

1. Одни алгоритмы обрабатывают открытый текст побитово (иногда побайтово). Такие алгоритмы называют ***потоковыми***.

2. Другие алгоритмы обрабатывают группы (блоки) битов открытого текста. Эти алгоритмы называют ***блочными***.

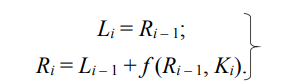
Примеры симметричных алгоритмов: DES, ГОСТ 28147-89, Lucifer, Blowfish.

Все перечисленные стандарты и алгоритмы блочных шифров (БШ) строятся на основе подстановочных и перестановочных шифров, т. е. являются комбинационными. БШ относятся также к классу симметричных.

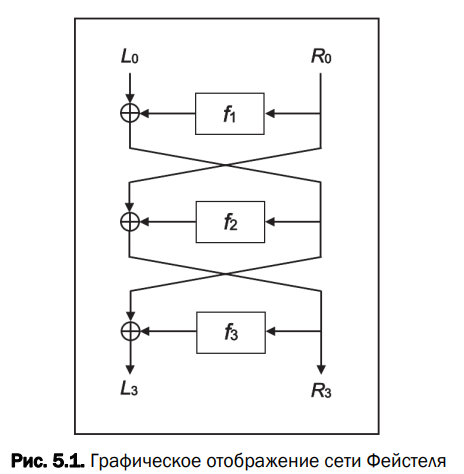
Блочное зашифрование (расшифрование) предполагает разбиение исходного открытого (зашифрованного) текста на равные блоки, к которым применяется однотипная процедура зашифрования (расшифрования). Указанная однотипность характеризуется прежде всего тем, что процедура зашифрования (расшифрования) состоит из совокупности повторяющихся наборов преобразований, называемых раундами.

Сеть Фейстеля, практикум 2 стр 70

При зашифровании сообщение разбивается на блоки одинаковой (фиксированной) длины (как правило – 64 или 128 битов). Каждый входной блок шифруемого сообщения изначально делится на два подблока одинакового размера: левый (L0) и правый (R0). Далее в каждом i-м раунде выполняются преобразования в соответствии с формальным представлением ячейки сети Фейстеля



По какому-либо математическому правилу вычисляется раундовый ключ Ki. В приведенном выражение знак «+» соответствует поразрядному суммированию на основе «XOR»

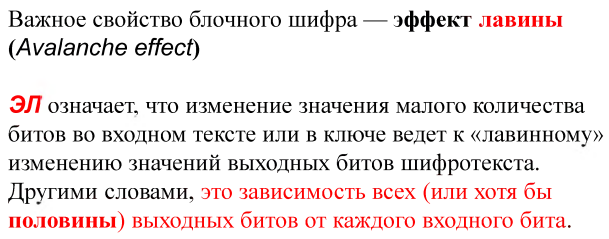


Расшифрование происходит так же, как и зашифрование, с той лишь разницей, что раундовые ключи будут использоваться в обратном порядке по отношению к зашифрованию.

Фейстель описывает два блока преобразований с использованием функции f (Ri – 1, Ki):

• блок подстановок (S-блок, англ. S-box);

• блок перестановок (P-блок, англ. P-box).

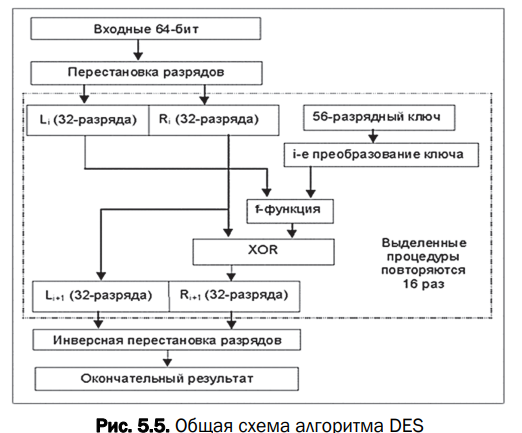


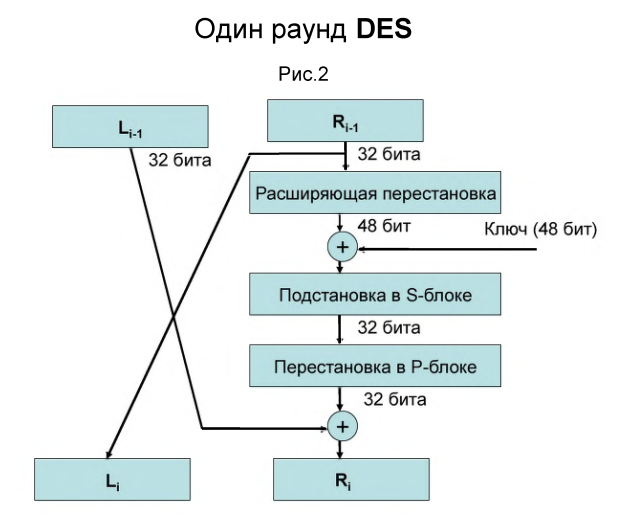
**В основе криптостойкости блочных шифров лежит идея К. Шеннона о представлении составного шифра таким образом, чтобы он обладал двумя важными свойствами: рассеиванием и перемешиванием.**

***Рассеивание* должно скрыть отношения между зашифрованным текстом и исходным текстом. Рассеивание подразумевает, что каждый символ (или бит) в зашифрованном тексте зависит от одного или всех символов в исходном тексте. Другими словами, если единственный символ в исходном тексте изменен, несколько или все символы в зашифрованном тексте будут также изменены.**

**Идея относительно *перемешивания* заключается в том, что оно должно скрыть отношения между зашифрованным текстом и ключом*.***

Стандарт DES (Data Encryption Standard – стандарт шифрования данных). Алгоритм строится на основе сети Фейстеля.





Входной блок данных, состоящий из 64 битов, преобразуется в выходной блок идентичной длины.

Один блок данных подвергается преобразованию (и при зашифровании, и при расшифровании) в течение 16 раундов.

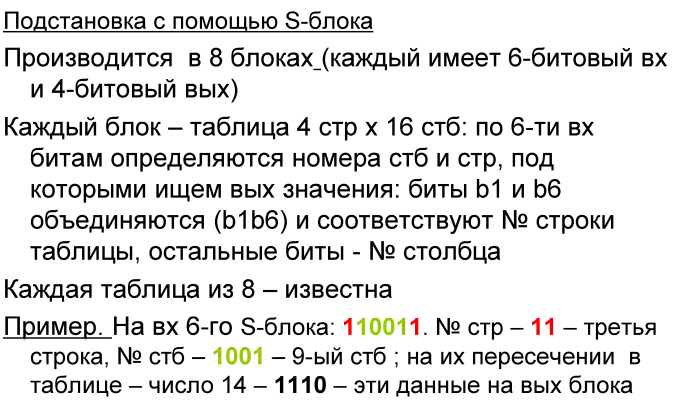
После первоначальной перестановки и разделения 64-битного блока данных на правую (R0) и левую (L0) половины длиной по 32 бита выполняются 16 раундов одинаковых действий

Левая и правая ветви каждого промежуточного значения обрабатываются как отдельные 32-битные значения, обозначенные Li и Ri.

Вначале правая часть блока Ri расширяется до 48 битов с использованием таблицы, которая определяет перестановку плюс расширение на 16 битов. Эта операция приводит размер правой половины в соответствие с размером ключа для выполнения операции XOR

После выполнения перестановки с расширением для полученного 48-битного значения выполняется операция XOR с 48-битным подключом Ki.

Затем полученное 48-битное значение подается на вход блока подстановки S, результатом которой является 32-битное значение. Подстановка выполняется в восьми блоках подстановки или восьми S-блоках (S-boxes). При выполнении этой операции 48 битов данных делятся на восемь 6-битных подблоков, каждый из которых по соответствующей таблице замен замещается четырьмя битами. В результате выполнения этого этапа получаются восемь 4-битных блоков, которые вновь объединяются в единое 32-битное значение.



Далее полученное 32-битное значение обрабатывается с помощью перестановки Р, которая не зависит от используемого ключа.

И наконец, результат перестановки объединяется с помощью операции XOR с левой половиной первоначального 64-битного блока данных. Затем левая и правая половины меняются местами, и начинается следующий раунд.

После выполнения 16-раундового зашифрования 64-битного блока данных осуществляется конечная перестановка (IP−1 ). Она является обратной к перестановке IP.

При расшифровании на вход алгоритма подается зашифрованный текст. Единственное отличие состоит в обратном порядке использования частичных ключей Ki. Ключ K16 используется в первом раунде, K1 – в последнем.

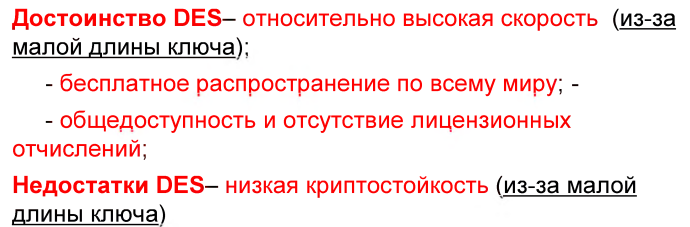
***Ключи***

Каждый 8-й бит исходного 64-битного ключа отбрасывается. Эти 8 битов, находящих в позициях 8, 16, 24, 32, 40, 48, 56, 64, изначально добавляются в исходный ключ таким образом, чтобы каждый байт содержал четное число единиц. Первоначальное значение разделяется на две половины, каждая из которых сдвигается независимо.

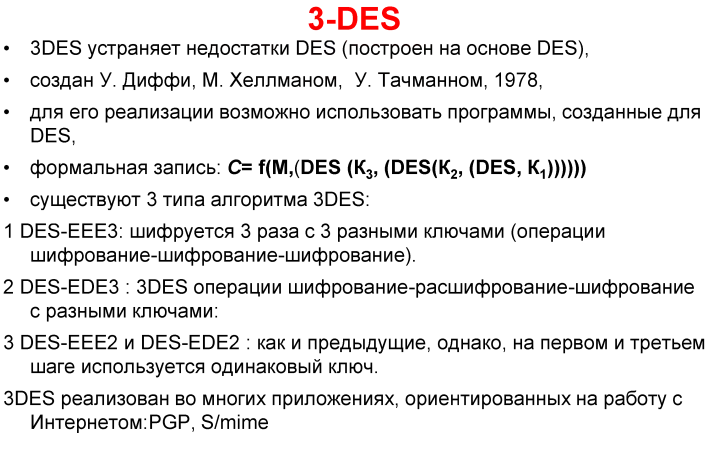
Если все биты каждой половины равны 0 или 1, то для всех раундов алгоритма используется один и тот же ключ (Слабый ключ)

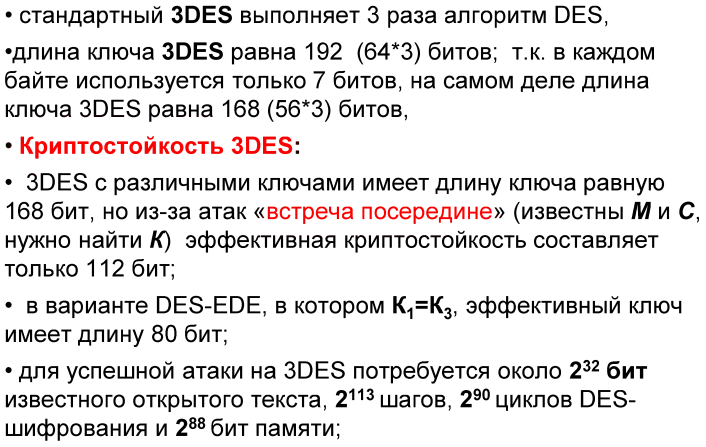
вместо 16 различных подключей эти ключи генерируют только два различных подключа (это Полуслабый ключ)





1. Методы симметричного криптопреобразования. Стандарты 3DES. Реализация и криптостойкость.





Применение 3-DES

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

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

Не известны успешные атаки на 3-DES.

Современная альтернатива 3-DES – AES Rijndael.

1. Шифровальная машина Энигма. Устройство, функционирование, криптостойкость.
2. Сравнительная характеристика алгоритмов Lucifer, IDEA, ГОСТ 28147-89, Blowfish.

|  |
| --- |
| ***Алгоритм Lucifer***  Первая версия: Использовала блоки и ключи длиной по 48 бит и основывалась на SP-сетях. (подстановочно-перестановочная сеть)  Вторая версия: Была запатентована и использовала сеть Фейстеля с 64-разрядными ключами и 32-битными блоками.  Третья версия: Предложена в 1973 году, оперировала с 128-битными блоками и ключами.  Развитие алгоритма Lucifer привело к созданию стандарта DES (Data Encryption Standard), который стал широко использоваться в криптографии  **Стандарт ГОСТ 28147–89**  Шифр ГОСТ 28147–89 построен по тем же принципам, что и американский DES, однако по сравнению с DES первый более удобен для программной реализации.  Размер блока составляет 64 бита, размер ключа – 256 битов, количество раундов – 32. Исходный 256-битный ключ делится на восемь 32-битных подключей. Они используются в 32 тактах в следующем порядке.  Алгоритм представляет собой классическую сеть Фейстеля. Шифруемый блок данных разбивается на две одинаковые части, правую R и левую L. Правая часть складывается по модулю 232 с подключом раунда и посредством принятого алгоритма шифрует левую часть. Перед следующим раундом левая и правая части меняются местами.  Сначала правый блок складывается по модулю 232 с подключом. Полученное 32-битное сообщение делится на восемь 4-битных чисел. Каждое из этих 4-битных чисел преобразуется соответствующим S-блоком в другое 4-битное число  ***Алгоритм Blowfish***  Основой алгоритма является сеть Фейстеля с 16 раундами. Длина блока равна 64 битам, ключ может иметь любую длину в пределах 448 битов  **Из чата**  **1. Lucifer**  История и происхождение: Разработан в начале 1970-х годов корпорацией IBM. Один из первых блочных шифров, ставший основой для DES.  Тип шифра: Блочный шифр.  Размер блока: 128 бит.  Размер ключа: Первоначально 48 бит, позднее увеличен до 128 бит.  Структура: Состоит из сети Фейстеля.  Безопасность: Устарел и не используется в современных системах из-за уязвимости к криптоанализу. Его наследник, DES, также стал небезопасен для современных применений.  Применение: Историческое значение в развитии криптографии, использовался до появления более надежных алгоритмов.  **2. IDEA (International Data Encryption Algorithm)**  Тип шифра: Блочный шифр.  Размер блока: 64 бита.  Размер ключа: 128 бит.  Структура: Не является сетью Фейстеля, использует смешанную конфигурацию с 8 раундами шифрования.  Безопасность: Считается достаточно безопасным. Не было обнаружено эффективных методов криптоанализа, которые могли бы полностью сломать алгоритм.  Применение: Используется в PGP (Pretty Good Privacy) и других системах шифрования для обеспечения конфиденциальности.  **3. ГОСТ 28147-89**  Тип шифра: Блочный шифр.  Размер блока: 64 бита.  Размер ключа: 256 бит.  Структура: Сеть Фейстеля с 32 раундами.  Безопасность: В целом, считается достаточно надежным. Устойчив к линейному и дифференциальному криптоанализу. Однако, есть критики, утверждающие, что безопасность может зависеть от использования конкретных S-блоков.  Применение: Широко использовался и продолжает использоваться в России и странах СНГ для защиты государственной и коммерческой информации.  **4. Blowfish**  Тип шифра: Блочный шифр.  Размер блока: 64 бита.  Размер ключа: Варьируется от 32 до 448 бит.  Структура: Сеть Фейстеля с 16 раундами.  Безопасность: Считается надежным для многих применений.Уязвимости были найдены в определённых условиях (ключи определённой длины). В целом, является одним из хорошоисследованных и широко применяемых алгоритмов.  Применение: Используется в широком спектре приложений, включая шифрование данных в сетевых протоколах и хранении данных. |

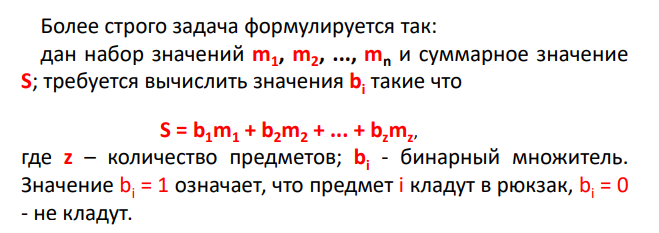
1. Криптографические системы с открытым (публичным) ключом. Задача об укладке ранца.

**В основу асимметричной криптографии положена идея использовать ключи парами: один – для зашифрования (открытый, или публичный, ключ), другой – для расшифрования (тайный ключ). Отметим, что указанная пара ключей принадлежит получателю зашифрованного сообщения. Все алгоритмы шифрования с открытым ключом основаны на использовании односторонних функций, к числу которых, как известно, относится вычисление дискретного логарифма.**

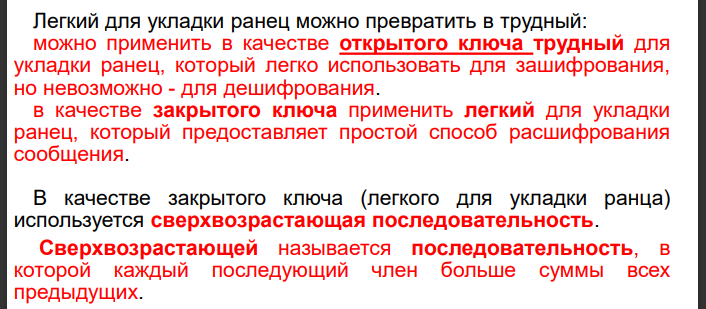
**Односторонней функцией** (one-way function) называется математическая функция, которую относительно легко вычислить, но трудно найти по значению функции соответствующее значение аргумента, т. е. зная х, легко вычислить f(x), но по известному f(x) трудно найти подходящее значение x.

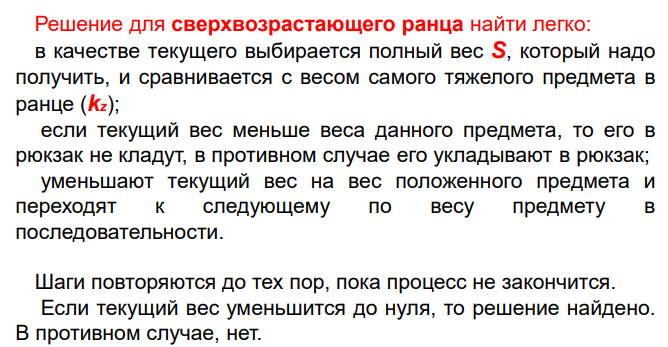
**Рюкзак**

В общем виде задачу можно сформулировать так: из заданного множества предметов mi общим числом z со свойствами «стоимость» и «вес» S, требуется отобрать некоторое число предметов таким образом, чтобы получить максимальную суммарную стоимость при одновременном соблюдении ограничения на суммарный вес



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 - нет.



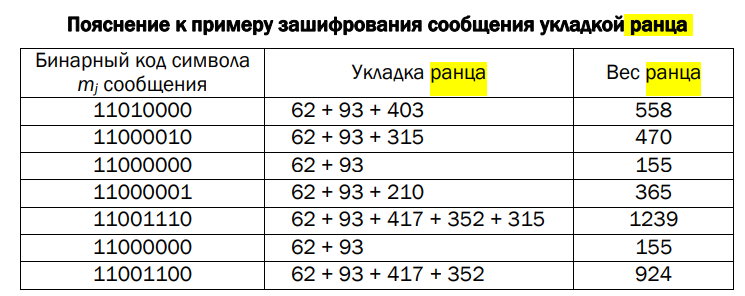


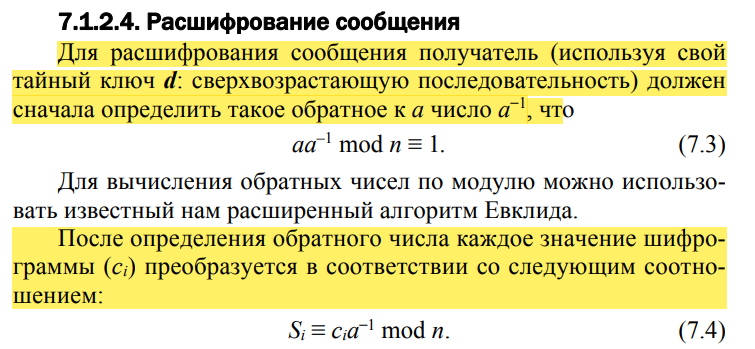
Открытый ключ e представляет собой нормальную (не сверхвозрастающую) последовательность. Он формируется на основе закрытого ключа и не позволяет легко решить задачу об укладке ранца. Для получения открытого ключа e (e = {ei}, i = 1, …, z) все значения закрытого ключа умножаются на некоторое число a по модулю n:



Значение модуля n должно быть больше суммы всех чисел последовательности; кроме того, НОД (а, n) = 1.

**Для зашифрования** сообщения (М) оно сначала разбивается на блоки, по размерам равные числу (z). 1 указывает на присутствие элемента последовательности в ранце, а 0 – на его отсутствие,





1. Управление криптографическими ключами. Алгоритм рукопожатия.

**Управление криптографическими ключами**

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

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

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

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

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

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

**Распределение ключей**

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

К нему предъявляются два требования:

* оперативность и точность распределения;
* скрытность распределяемых ключей.

Введем обозначения:

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

**Протокол рукопожатия (основа протокола SSL)**

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

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

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

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

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

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

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

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

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

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

1. Распределение ключей на основе симметричных систем.

**РК на основе симметричных систем.**

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

Проблемы:

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

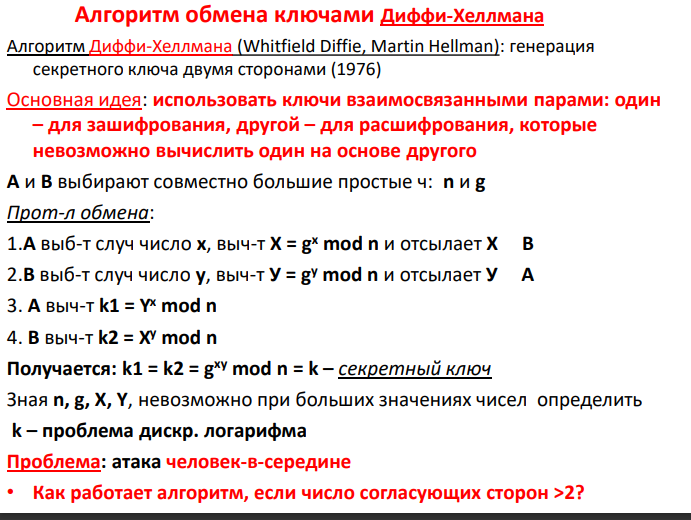
*Возможное решение:* иерархическая (древовидная) система с пользователями, находящимися на листьях, и ЦРК в промежуточных узлах.

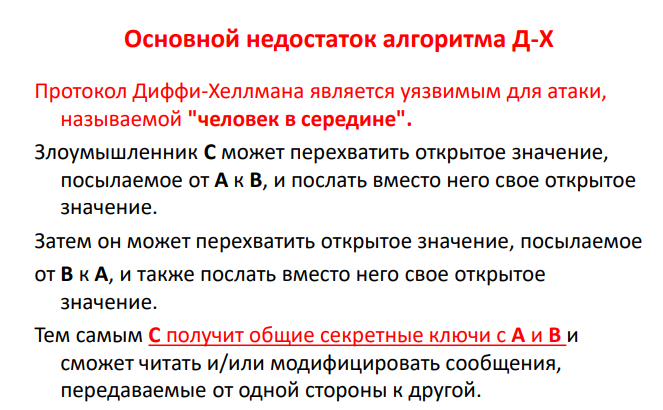
1. Используя открытые сети.

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

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

1. Алгоритм передачи ключа по Диффи-Хеллману.





1. Алгоритм шифрования RSA. Реализация и криптостойкость.

В основу асимметричной криптографии положена идея использовать ключи парами: один – для зашифрования (открытый, или публичный, ключ), другой – для расшифрования (тайный ключ). Отметим, что указанная пара ключей принадлежит получателю зашифрованного сообщения. Все алгоритмы шифрования с открытым ключом основаны на использовании односторонних функций, к числу которых, как известно, относится вычисление дискретного логарифма.

асимметричная криптография основана на сложности решения некоторых математических задач. По существу, таких задач две:

• разложение больших чисел на простые сомножители (задача факторизации);

• вычисление дискретного логарифма в конечном поле, а также вычислительные операции над точками эллиптической кривой.

*Для генерации двух ключей*: тайного и открытого (а по сути – двух взаимосвязанных частей одного ключа, т. е. ключа, принадлежащего одному физическому лицу (или группе лиц), либо одному юридическому лицу), используются два больших случайных простых числа p и q. Для максимальной большей криптостойкости нужно выбирать p и q равной длины. Рассчитывается произведение: *n = pq*. Это есть один из трех компонент ключа, состоящего из чисел *n, e, d*

Затем случайным образом выбирается второй компонент ключа (открытый ключ или ключ зашифрования, *e*, такой что *e* и *(p – 1)(q – 1)* являются взаимно простыми числами; вспомним, что *(p – 1)(q – 1) = φ(n)* – функция Эйлера)

Наконец, расширенный алгоритм Евклида используется для вычисления третьего компонента ключа: ключа расшифрования *d* такого, что выполняется условие:

*ed ≡ 1 (mod φ(n))*

Другими словами:

*d–1 ≡ e (mod φ(n)).*

открытый (публичный) ключ (e, n) и тайный ключ (d, n)

***Для зашифрования/расшифрования используется ключ получателя: отправитель шифрует сообщение открытым ключом, а получатель расшифровывает шифртекст своим тайным ключом.***

*Зашифрование*. Если шифруется сообщение М, состоящее из r блоков: m1, m2, …, mi, …, mr, то шифртекст С будет состоять из такого же числа (r) блоков, представляемых числами:

*ci ≡ (mi) e mod n*

*Расшифрование*. Для расшифрования каждого зашифрованного блока производится вычисление вида:

*mi ≡ (ci) d mod n.*

**Основой криптостойкости RSA** является сложность разложения большого числа на простые множители, известного как задача факторизации. Вот основные аспекты, влияющие на криптостойкость алгоритма RSA:

**Размер ключа**: Чем больше 𝑛, тем сложнее его факторизовать. Современные стандарты: Рекомендуется использовать длину ключа не менее 2048

**Выбор простых чисел 𝑝 и 𝑞**: Рекомендуется выбирать простые числа 𝑝*p* и 𝑞*q*, которые не слишком близки по значению и имеют примерно одинаковую длину

1. Алгоритм шифрования Эль-Гамаля. Реализация и криптостойкость.

Он может быть использован для решения трех основных криптографических задач: для зашифрования/расшифрования данных, для формирования цифровой подписи и для согласования общего ключа. Кроме того, возможны модификации алгоритма для схем проверки пароля, доказательства идентичности сообщения и другие варианты

***Как подчеркивалось выше, безопасность алгоритма Эль-Гамаля, как и безопасность алгоритма Диффи – Хеллмана, основана на трудности вычисления дискретных логарифмов.***

Алгоритм Эль-Гамаля фактически использует схему Диффи – Хеллмана, чтобы сформировать общий секретный ключ для абонентов, передающих друг другу сообщение, и затем сообщение шифруется путем умножения его на этот ключ.

И в случае шифрования, и в случае формирования цифровой подписи каждому пользователю необходимо сгенерировать пару ключей.

Рассматриваемый алгоритм отличается от алгоритма RSA несколькими параметрами и особенностями:

1) генерацией ключевой информации и числом компонент, составляющих ключ;

2) каждому блоку (символу) открытого сообщения в шифртексте на основе алгоритма Эль-Гамаля соответствуют 2 блока (в RSA – один-один);

3) в алгоритме Эль-Гамаля при зашифровании используется число (обозначим его k), которое практически никак не связано с ключевой информацией получателя и которое принимает (по определению) различные значения при зашифровании различных блоков сообщения

***Генерация ключевой информации.***

1. Выбирается простое число *р*.

2. Выбирается число (*g, g < p*), являющееся *первообразным корнем числа р* – очень важный элемент с точки зрения безопасности алгоритма (см. ниже).

**Первообразный корень** (primary (residual) root) по модулю р является таким числом, что его степени (*gi* , 1 ≤ i ≤ p – 1) дают все возможные по модулю *р* вычеты (остатки), которые взаимно просты с *p*.

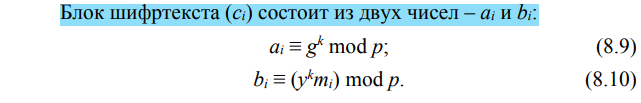
3. Выбирается число *х (х < p)*

4. Вычисляется последний компонент ключевой информации:

*y ≡ gх mod р.*

Владельцу сформированной ключевой информации, состоящей из 4 чисел, может посылаться некоторый шифртекст, созданный с использованием открытого ключа получателя: *p, g, y*. Расшифрование шифртекста получатель производит своим тайным ключом: *p, g, х*.

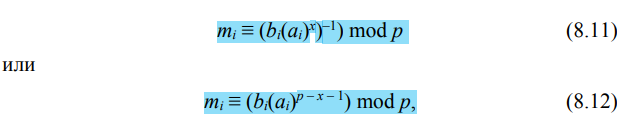
*Зашифрование* отправителем (каждого отдельного блока mi исходного сообщения) предусматривает использование, как это особо подчеркивалось выше, некоторого случайного числа *k (1 < k < p – 1).*



Здесь стал очевидным упомянутый недостаток алгоритма шифрования Эль-Гамаля: удвоение (реально – примерно в 1,5 раза) длины зашифрованного текста по сравнению с начальным текстом.

Случайное число k должно сразу после вычисления уничтожаться

Расшифрование ci. Выполняется по следующей формуле: (Считать про ще по 2 формуле)



Алгоритм Эль-Гамаля является асимметричным криптографическим алгоритмом, основанным на сложности вычисления дискретного логарифма в конечных полях. Его **криптостойкость** зависит от математической сложности базовых проблем и выбранных параметров. Вот основные аспекты, влияющие на криптостойкость алгоритма Эль-Гамаля:

1. **Размер ключа**:

2. **Выбор параметров**: Число *p* должно быть достаточно большим и случайным, чтобы избежать возможности предсказуемого анализа. Плохо выбранные *p* и *g* могут ослабить безопасность системы.

3. **Дискретный логарифм**: Основная криптостойкость алгоритма Эль-Гамаля основана на сложности вычисления дискретного логарифма, то есть нахождения такого числа *x*, что 𝑔𝑥≡ℎ mod  𝑝

1. Потоковое шифрование. Типы. Гаммирование в потоковом шифровании.

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

Особенности.

1.Операции зашифрования и расшифр. вып-ся *поразрядно*.

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

3.Поточный шифратор и деш-р требует задания начального значения ключа

4.Пот.шифры исп-ся в специальных приложениях и редко обсуждаются

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

Классический ПШ – Шифр Вернама

***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.



***Типы поточных шифров:***

1. *Синхронные* –

• поток гаммы генерируется независимо от открытого текста и шифротекста;

• для успешного расшиф-я необходимо синхрон-ть ключ с шифротекстом; Свойства:

1. Искажение одного символа в шифротексте искажает только один символ в расшифр-м тексте (+),

2. Защита от любых вставок и удалений шифротекста, так как они приведут к потере синхронизации и будут обнаружены (+)

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

2. *Самосинхронизирующиеся* (асинхронные) – В самосинхронизирующихся потоковых шифрах символы ключевой гаммы зависят от исходного секретного ключа шифра и от конечного числа последних знаков зашифрованного текста.

• значение ключа зависит либо от исх текста, либо от шифротекста;

• поток ключей создается функцией ключа и фиксированного числа знаков шифртекста (N): внутреннее состояние генератора является функцией предыдущих N битов шифртекста - генератор потока ключей (при расшифровании), приняв N битов, автоматически синхронизируется с шифрующим генератором

Свойства:

• Так как каждый знак открытого текста влияет на следующий шифртекст, статистические свойства открытого текста распространяются на весь шифртекст (+),

• ошибочно удаленный или добавленный символ (бит) вызывает только ограниченное кол-во ошибочных символов в дешифрованном тексте, после чего правильный текст восстанавливается (+)

• каждому неправильному биту шифртекста соответствуют N ошибок в открытом тексте) (-)

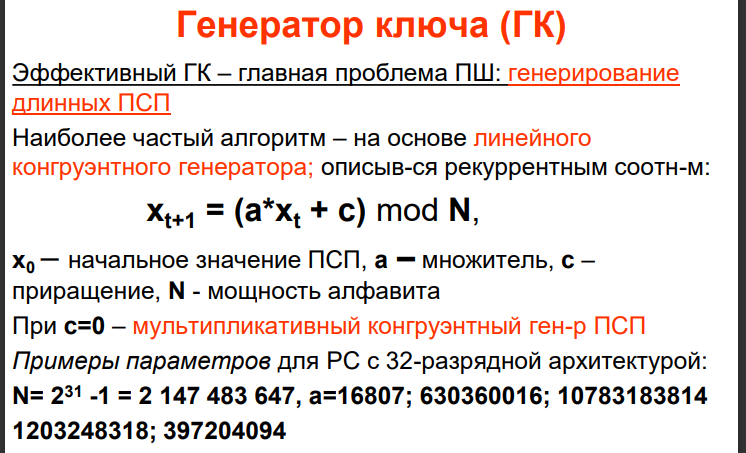
Потоковые шифры преобразуют открытый текст в шифртекст *по одному биту (символу) за операцию*. Генератор потока ключей выдает поток битов: k1, k2, …, ki. Этот поток битов (иногда называемый бегущим ключом) и поток битов открытого текст m1, m2, …, mi, …, mz подвергаются операции *сложения по модулю два* (XOR), и в результате получается поток битов шифртекста

При расшифровании, для восстановления битов открытого текста, операция XOR выполняется над битами шифртекста и тем же самым потоком ключей:

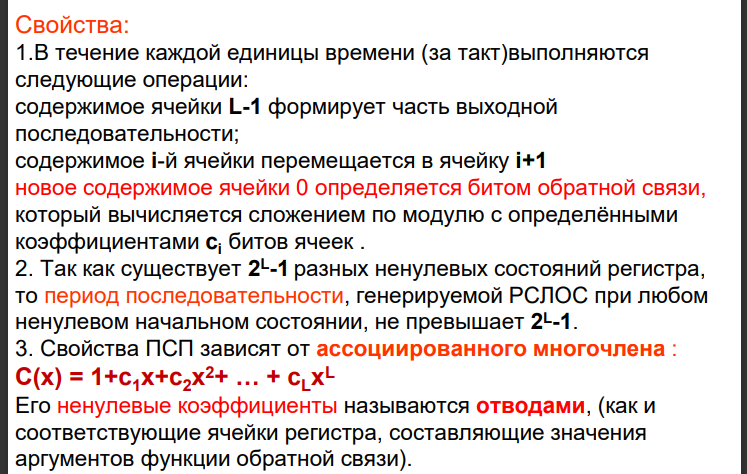
mi = ci ⊕ ki.

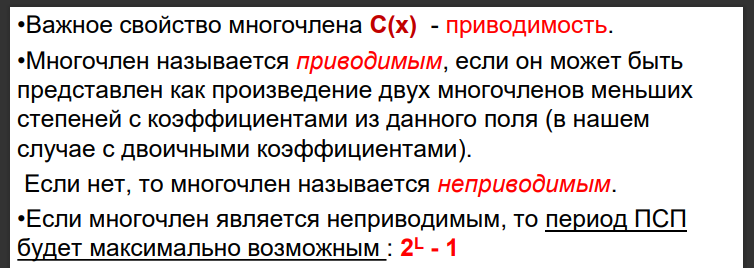
Безопасность системы полностью зависит от свойств генератора потока ключей.

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









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

Часто используемый алгоритм генерирования (программно или аппаратно) ПСП реализуется на основе так называемого линейного конгруэнтного генератора, описываемого следующим рекуррентным соотношением:



где xt и xt + 1 – соответственно t-й (предыдущий) и (t + 1)-й (текущий, вычисляемый) члены числовой последовательности; а, с и n – константы. Период такого генератора (период ПСП) не превышает n.

Если параметры a, b и c выбраны правильно, то генератор будет порождать случайные числа с максимальным периодом, равным c. При программной реализации значение с обычно устанавливается равным 2b – 1 или 2b , где b – длина слова в битах.

Генератор практически не используется в криптографии в силу низкой криптостойкости.

**Генератор ПСП на основе регистров сдвига**

РС(регистр сдвига) с линейной обратной связью (РСЛОС) состоит из двух частей: собственно РС и функции обратной связи

РСЛОС строятся на основе примитивных порождающих полиномов (многочленов),

Если многочлен является неприводимым, то период ПСП при ненулевом начальном условии (ненулевом состоянии) регистра будет максимально возможным: 2L – 1.

1. Особенность шифра Вернама.

**Особенности:**

1.Операции зашифрования и расшифрования выполняются поразрядно.

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

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

4.Потоковые шифры используются в специальных приложениях и редко обсуждаются

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

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

Если взять сообщение и ключ (должен быть случайной последовательностью)

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

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

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

При соблюдении данных свойств расшифровать сообщение нельзя.

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

1. 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. При этом скрытый канал организуется на базе и внутри открытого канала.

«Скрытость» канала передачи тайной информации отличает стеганографию от криптографии: в первом случае тайной является сам факт наличия канала (передачи информации).

*Основные компоненты стеганосистемы:*

*• контейнер С* - файл, в котором размещается тайное сообщение М; именно контейнер является скрытым каналом;

*• тайное сообщение М*, осаждаемое в контейнер для передачи

или хранения (например, с целью доказательства или защиты авторских прав на документ-контейнер;

• *ключи, K* системы, выполняющие ту же функцию, что и криптографические ключи; ключей может быть несколько, в соответствии с этим современные стеганосистемы характеризуют как *многоключевые*: один ключ используется встраивания/извлечения тайной информации, другой – с выбором элементов (например, битов) контейнера для его модификации, третий – для предварительного преобразования тайной информации

*• контейнер со встроенным сообщением*, или стеганоконтей нер, S, который передается по открытому каналу,

*• субъекты системы: отправитель и получатель.*

В зависимости от формата документа-контейнера цифровую (или компьютерную) стеганографию подразделяют на классы:

• аудиостеганография;

• видеостеганография;

• графическая стеганография;

• текстовая стеганография;

Большинство исследований в предметной области посвящено использованию в качестве стеганоконтейнеров изображений (текст также можно рассматривать как изображение). Это обусловлено следующими причинами: см практикум 2 стр 170

Метод НЗБ основывается на ограниченных способностях зрения или слуха человека, вследствие чего людям тяжело различать незначительные вариации цвета или звука. Младшие биты (выделены бледным, справа) дают незначительный «вклад» в изображение по сравнению со старшими. Замена одного или даже нескольких младших битов для человеческого глаза будет почти незаметна.

Процесс внедрения (встраивания):

- Сообщение переводится в двоичный формат.

- В каждом байте пикселя (или семпла звука) наименее значимый бит заменяется на бит из сообщения.

- Для цветного изображения это может означать изменение LSB в каждом из каналов RGB.

- Для серого изображения это изменение LSB в каждом пикселе

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

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

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

**Эллиптическая кривая** над вещественными числами – это множество точек, описываемых уравнением

*у2 = х3 + aх + b – уравнение Вейерштрасса,*

при этом константы (а и b – вещественные числа) должны удовлетворять условию

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

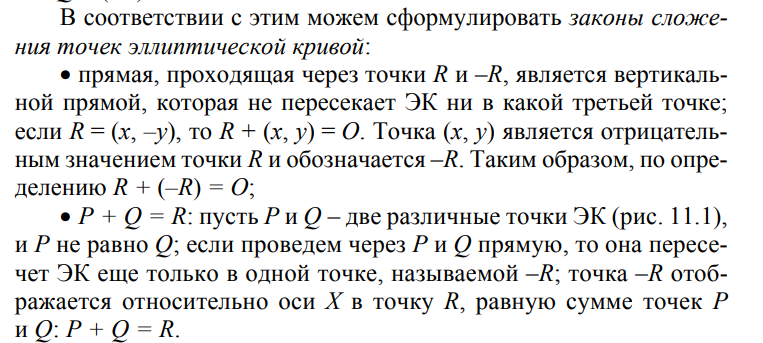
**Группа** – непустое множество с определенной на нем бинарной операцией, называемой сложением и удовлетворяющей нескольким аксиомам

**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.



**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**

**Конечное поле** – это множество конечного числа элементов. Примером конечного поля является множество целых чисел по модулю p, где p – простое число.

**Эллиптическая кривая над полем Fp** задается теми же уравнениями, что и ЭК над действительными числами, только все вычисления производятся по модулю р (mod p):



1. Представление и описание эллиптической кривой на основе алгебраической геометрии

**Эллиптическая кривая** над вещественными числами – это множество точек, описываемых уравнением

*у2 = х3 + aх + b – уравнение Вейерштрасса,*

при этом константы (а и b – вещественные числа) должны удовлетворять условию

1. Арифметические операции в эллиптической криптографии

с операцией умножения точки на целое число напрямую связана идея, надежность и криптостойкость эллиптической криптографии

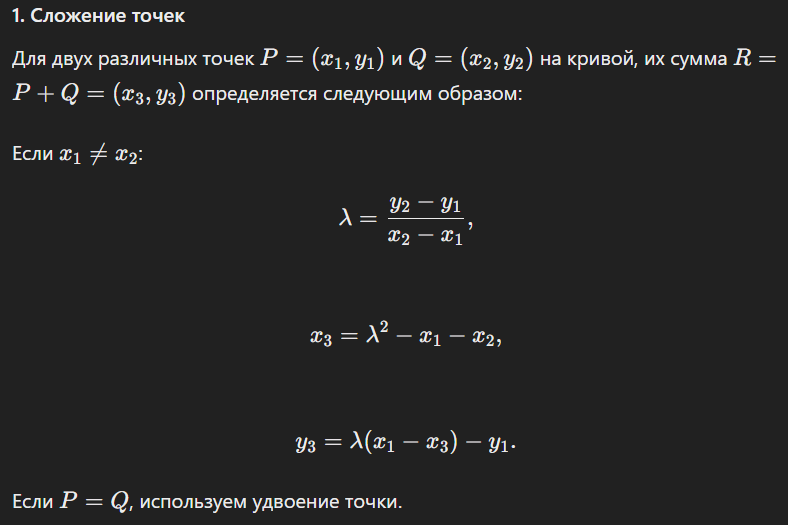
**Основные арифметические операции**

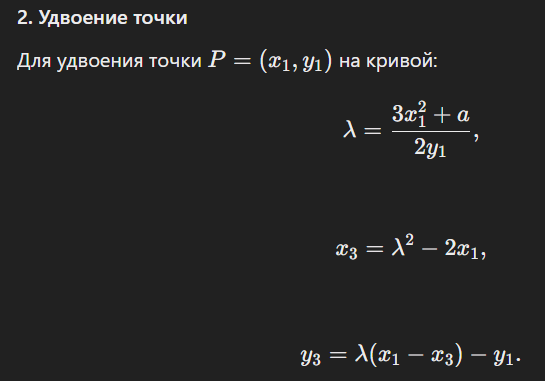
1. Сложение точек (Point Addition)

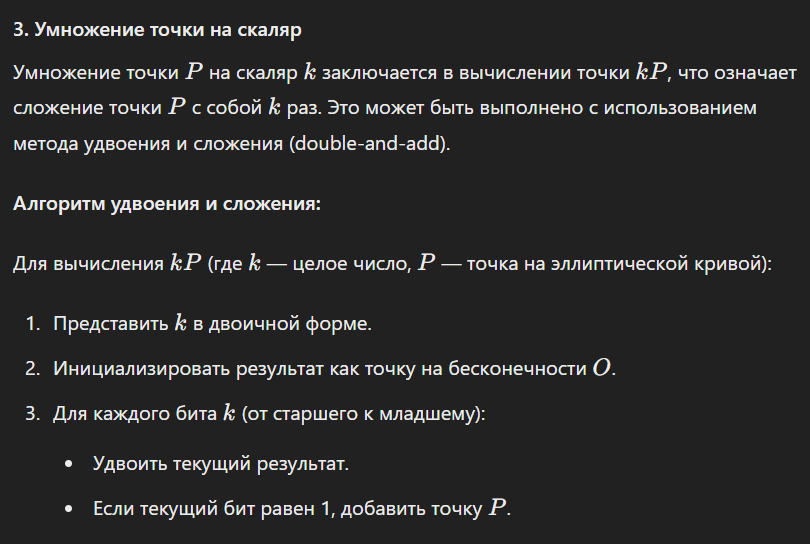
2. Удвоение точки (Point Doubling)

3. Умножение точки на скаляр (Scalar Multiplication)

Из чата







**Применение арифметических операций в криптографии**

1. **Ключевое соглашение (ECDH):** Использует умножение точки на скаляр для обмена секретными ключами.
2. **Цифровая подпись (ECDSA):** Включает операции сложения и умножения точек для генерации и проверки подписей.
3. **Шифрование (ECIES):** Применяет эллиптические кривые для шифрования данных, обеспечивая высокий уровень безопасности.
4. ЭЦП. Назначение и свойства.

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

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

• *аутентифицирование* лица, подписавшего сообщение;

• *контроль целостности* подписанного сообщения;

• *защита сообщения* от подделок: ЭЦП вычислена на основании исходного состояния документа и соответствует лишь ему, поэтому при любом случайном или преднамеренном изменении документа подпись станет недействительной;

• *доказательство авторства* лица, подписавшего сообщение, если это лицо отрицает свое авторство.

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.

Основные типы ЭЦП

1. На основе симметричных криптосистем(с тайным ключом); авторизацией документа является сам факт зашифрования его секретным ключом

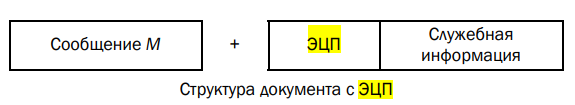
2. На основе симметричных криптосистем и посредника (создаются две симметричные системы: между отправителем и посредником и между посредником и получателем. Причем посредник выдает двум сторонам различный тайный (для иных субъектов системы) ключ)

3. На основе ассиметричных криптосистем; (с открытым ключом) ЭЦП на основе RSA: **сообщение, отправляемое получателю, шифруется тайным ключом отправителя.** **Отправитель же верифицирует подпись (в данном случае – устанавливает авторство, используя для расшифрования публичный ключ отправителя,** и получает гарантию в защищенности переданного сообщения от подделок, если после расшифрования формат и содержание документа имеют логическую стройность) с помощью открытого ключа отправителя

\*4. На основе ассиметричных криптосистем и однонаправленных хэш-функций. (можно отнести и к 3 пункту)

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

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 (дата, время отправки или различные данные об отправителе), как это схематично показано на рисунке.



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)

1. 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.

**Основные типы ЭЦП**

1. На основе симметричных криптосистем(с тайным ключом); авторизацией документа является сам факт зашифрования его секретным ключом

2. На основе симметричных криптосистем и посредника (создаются две симметричные системы: между отправителем и посредником и между посредником и получателем. Причем посредник выдает двум сторонам различный тайный (для иных субъектов системы) ключ)

3. На основе ассиметричных криптосистем/ На основе ассиметричных криптосистем и однонаправленных хэш-функций; (с открытым ключом) ЭЦП на основе RSA: **сообщение, отправляемое получателю, шифруется тайным ключом отправителя.** **Отправитель же верифицирует подпись (в данном случае – устанавливает авторство, используя для расшифрования публичный ключ отправителя,** и получает гарантию в защищенности переданного сообщения от подделок, если после расшифрования формат и содержание документа имеют логическую стройность) с помощью открытого ключа отправителя

***Атаки***

Атака с использованием открытого ключа.

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

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.

Подделка документа (коллизия первого рода). Коллизии: Если злоумышленник может найти два разных сообщения с одинаковым хешем, это позволяет ему подменить одно сообщение другим без изменения подписи.

1. ЭЦП на основе симметричной криптографии

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.

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

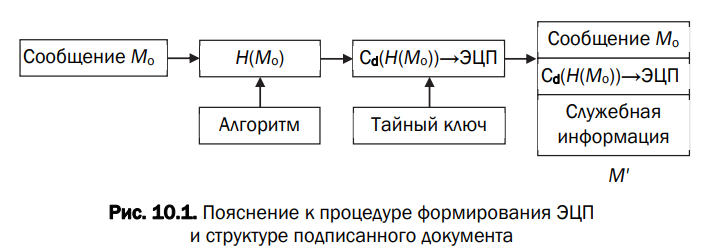
Однако если предусмотреть наличие в системе третьего лица – арбитра или посредника (П), пользующегося доверием обеих сторон, то можно избежать указанного недостатка.

1. ЭЦП на основе алгоритма RSA

Про ЭЦП на основе хешей подписываемых сообщений

Классическая технология использования ЭЦП предусматривает подписание не самого сообщения (обозначим его здесь Мо), а его хеша, H(Mо).

Общая структура подписанного электронного документа – Мо – М' – представляет собой, как правило, конкатенацию этого документа и ЭЦП S. Кроме этих двух элементов, интегральный документ может содержать некоторую служебную информацию (дата, время отправки или различные данные об отправителе)



Важное свойство цифровой подписи заключается в том, что ее может проверить (верифицировать) каждый, кто имеет доступ к открытому ключу ее автора.

**при генерации ЭЦП** (по классической схеме) для сообщения М отправитель последовательно выполняет следующие действия:

1. вычисляет хеш (хеш-образ) сообщения М: Н(М);

2. вычисляет содержание ЭЦП (собственно ЭЦП S) по хешу Н(М) с использованием своего закрытого ключа *d: S =Сd(Н(М));*

3. присоединяет (конкатенирует) ЭЦП к сообщению М и некоторой служебной информации, создавая таким образом итоговое сообщение М';

4. посылает сообщение М' получателю;

**При верификации** – получив сообщение М', другая сторона последовательно выполняет следующие действия:

1. отделяет цифровую подпись S от сообщения М (для общего случая применим одинаковые символьные обозначения);

2. применяет к сообщению М операцию хеширования, используя ту же функцию, что и отправитель, и получает хеш-образ полученного сообщения;

3. используя открытый ключ отправителя, расшифровывает S, т. е. извлекает из ЭЦП хеш-образ отправленного сообщения;

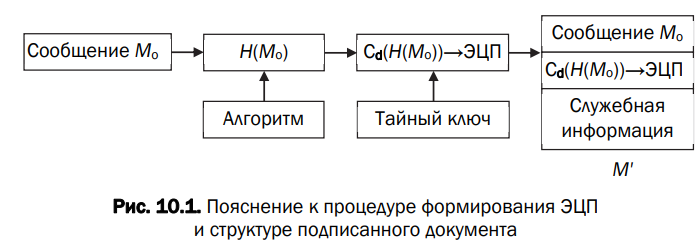
4. проверяет соответствие (равенство) обоих хеш-образов, и если они совпадают, то отправитель действительно является тем, за кого себя выдает, а сообщение при передаче не подверглось искажению.

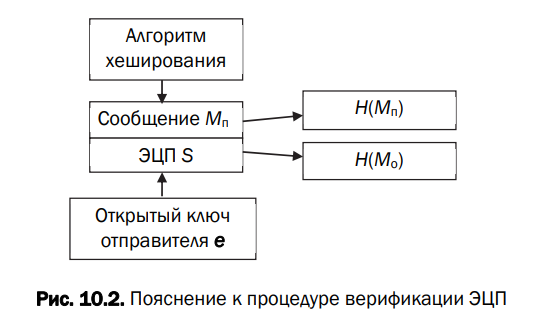
Здесь можно рассматривать две ситуации:

• сообщение Мo подписывается и передается в открытом (незашифрованном) виде;

• сообщение Мo подписывается и передается в зашифрованном виде.

Первый случай соответствует схеме и операциям, представленным на рис. 10.1 и рис. 10.2





*Генерация ЭЦП*

**

*dо* и *no* – элементы тайного ключа отправителя.

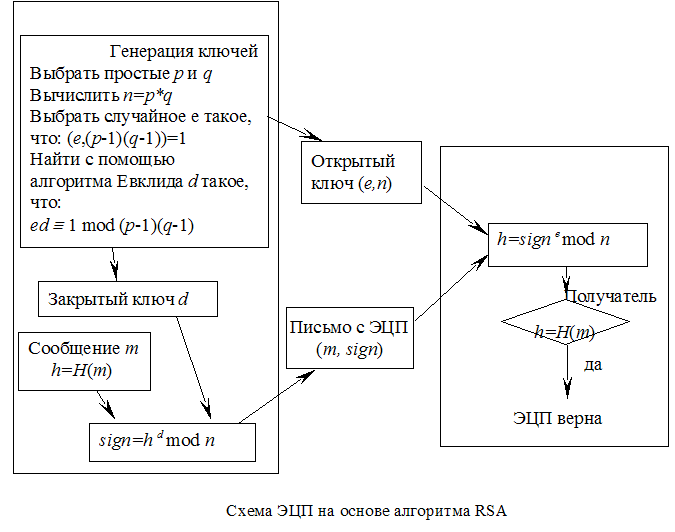
Передаваемое сообщение М' = Мo||S.

Соответственно, операция *расшифрования* на приемной стороне (получатель анализирует Мп||S) будет производиться в соответствии с формулой (8.6) с известной модификацией ключей:

**

Далее вычисляется *Н(Mп).* Если *H(Mo) = H(Mп),* подпись верифицирована.

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 (еп 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.



1. ЭЦП на основе симметричной криптосистемы и посредника

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);

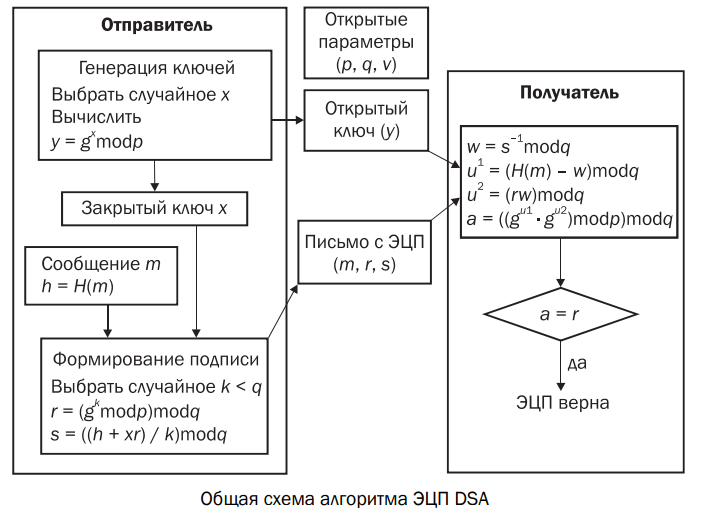
3) подписанный документ нельзя изменить или подделать;

4) подпись нельзя отрицать.

Если ключи действительно были сеансовыми, то подпись невозможно использовать повторно.

1. ЭЦП DSS.

Стандарт DSS (Digital Signature Standard – стандарт цифровой подписи) для использования в алгоритме ЭЦП DSA (Digital Signature Algorithm – алгоритм цифровой подписи).

****

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…

*Открытый ключ* у вычисляется в соответствии с выражением

*y = gx mod p,*

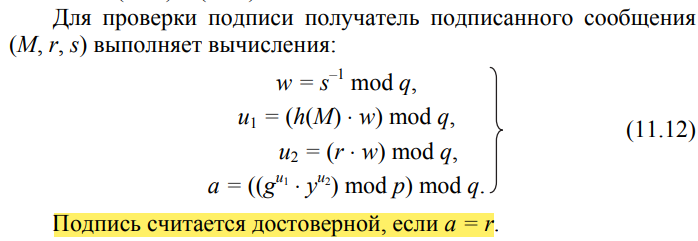
где *x < q; х* – закрытый ключ.

В алгоритме подписывается не само сообщение М, а его хеш: h(М). Для хеширования сообщений используется алгоритм SHA. После генерации ключей их обладатель может подписывать свои сообщения.

**Генерация ЭЦП:**



**Верификация подписи**



1. ЭЦП на основе алгоритма Эль-Гамаля

Схему Эль-Гамаля можно использовать как для шифрования, так и для цифровых подписей. В сравнении, например, с ЭЦП на основе RSA рассматриваемая схема обеспечивает более высокое быстродействие

ключ состоит из четырех чисел.

**Генерация ключей:**

1. Выбирается простое число *р* и два случайных числа, меньших, чем р: числа х и g.

2. Далее вычисляется *y = gx mod p.*

*Открытый ключ: y, g, р;*

*тайный ключ: х.*

Чтобы **подписать сообщение** М, обладатель используемых для ЭЦП ключей должен

1. выбрать случайное число k, взаимно простое с р – 1.

2. Затем вычисляются числа а и b, являющиеся цифровой подписью:

*a = gk mod p;*

для вычисления b с помощью расширенного алгоритма Евклида решается уравнение *Н(Mо) ≡ (xa + kb) mod (p – 1).*

3. Получателю отправляется сообщение М' = Мо||S

**Для верификации подписи** вычисляется хеш полученного сообщения *Н(Мп) = h.* Далее нужно убедиться, что выполняется равенство

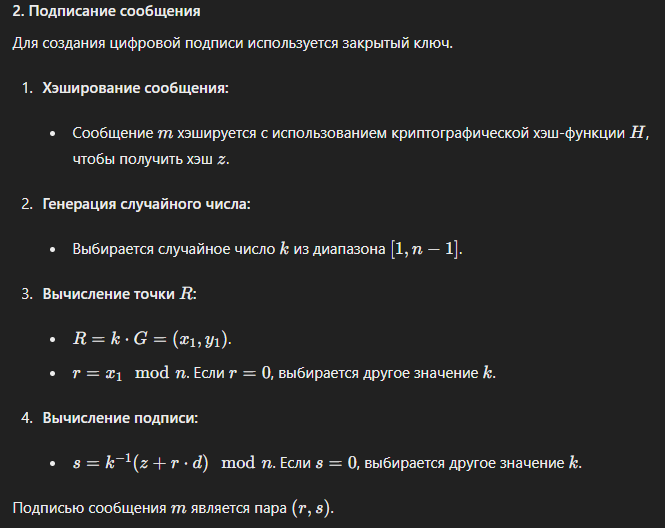
*ya ab ≡ gh mod p*

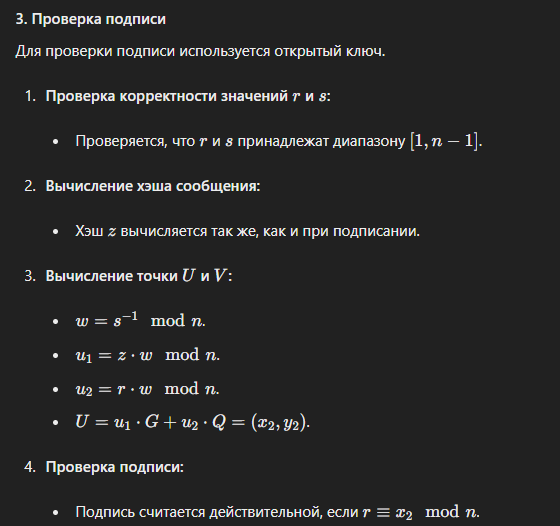
Если равенство выполняется, подпись верифицируется.

1. ЭЦП на основе эллиптической кривой.

Электронная цифровая подпись (ЭЦП) на основе эллиптических кривых (ECDSA - Elliptic Curve Digital Signature Algorithm) является одним из наиболее популярных методов цифровой подписи. Этот метод обеспечивает высокий уровень безопасности при меньших длинах ключей по сравнению с другими алгоритмами, такими как RSA.







1. Алгоритм К. Шнорра. Стандарт ЭЦП в РБ.

Рассматриваемая схема является основой стандарта ЭЦП в Беларуси. 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 *gq ≡ 1 mod p.*

Выбирается число *х < q* (*х* является тайным ключом) и вычисляется последний элемент открытого ключа: *y ≡ g–х mod p*.

**Для подписи сообщения**

1. Мо выбирается случайное число *k (1 < k < q)*

2. вычисляет параметр а:

*а ≡ gk mod p*.

3. Далее вычисляется хеш от канкатенации сообщения Мо и числа а:

*h = H(Mo||a).*

4. Вычисляется значение *b: b ≡ (k + xh) mod q*. (10.11)

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

**Для проверки подписи** получатель вычисляет *Х ≡ gb yh (mod p)*

Затем он проверяет выполнение равенства: h = Н(Mп||Х).

Подпись достоверна, если равенство выполняется.

1. Протокол Kerberos.

Из чата

Протокол Kerberos — это сетевой протокол аутентификации, разработанный для безопасной идентификации пользователей и служб в недоверенных сетях. Он был разработан Массачусетским технологическим институтом (MIT) и является частью проекта Athena. Kerberos используется для предотвращения перехвата и повторного использования учетных данных.

**Основные принципы работы**

1. **Централизованное управление аутентификацией**
   * Kerberos использует централизованный сервер аутентификации, называемый Центром распространения ключей (Key Distribution Center, KDC).
   * KDC состоит из двух основных компонентов: Сервер аутентификации (Authentication Server, AS) и Сервер выдачи тикетов (Ticket Granting Server, TGS).
2. **Симметричное шифрование**
   * Kerberos использует симметричное шифрование для обеспечения безопасности передачи данных.
   * Ключи шифрования известны только пользователю и KDC.
3. **Использование тикетов**
   * Аутентификация в Kerberos основана на использовании тикетов, которые представляют собой временные сертификаты, удостоверяющие личность пользователя.

**Основные этапы аутентификации**

1. **Аутентификация пользователя**
   * Пользователь отправляет запрос на аутентификацию (Authentication Request) в AS, содержащий его имя и метку времени.
   * AS проверяет учетные данные пользователя и отправляет ему тикет предоставления тикетов (Ticket Granting Ticket, TGT), зашифрованный с использованием ключа, известного только пользователю и AS.
2. **Получение тикета для доступа к сервису**
   * Пользователь отправляет запрос в TGS с TGT и именем сервиса, к которому он хочет получить доступ.
   * TGS проверяет TGT и выдает тикет сервиса (Service Ticket), зашифрованный с использованием ключа, известного только TGS и сервису.
3. **Доступ к сервису**
   * Пользователь отправляет сервису запрос на доступ с использованием Service Ticket.
   * Сервис проверяет тикет и, если он действителен, предоставляет пользователю доступ.

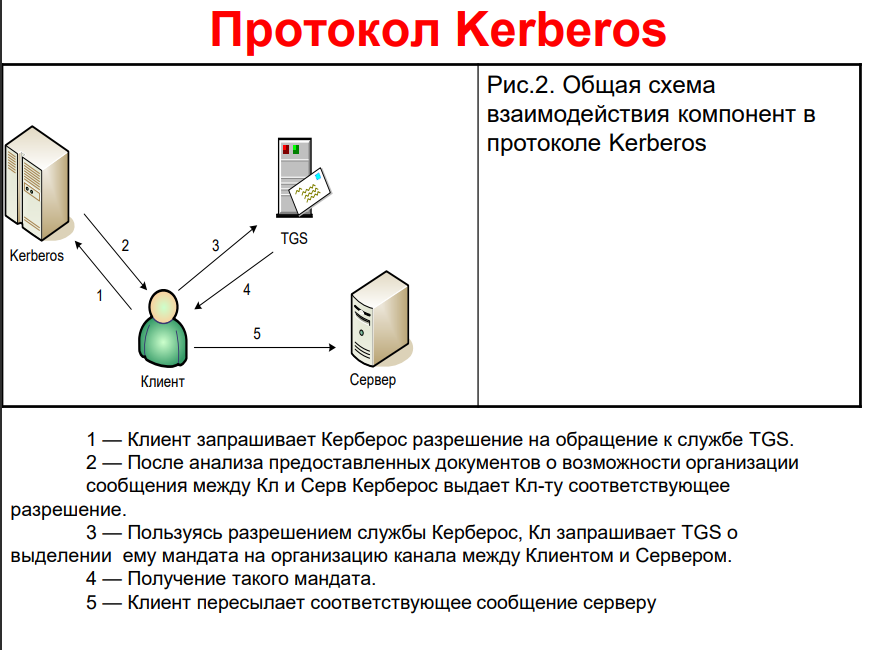
Из лк

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. Деструктивные программы. Классификация и методы нейтрализации.

**Классификация вредоносного ПО**

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

2. **Черви** (worms) — 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

3.**Кейлоггеры** (keyloggers) — 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

4. **Трояны** (trojans), троянские кони — 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)

5. **Боты** (bots) — распространенный в наше время вид зловредного ПО, который устанавливается на компьютерах пользователей и используется для атак на другие компьютеры (сети bothet)

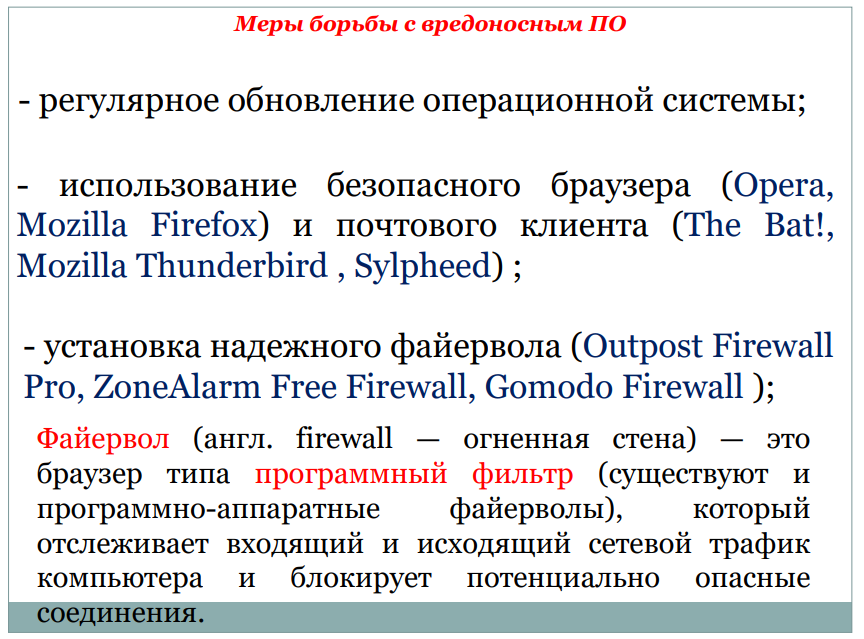
6. **Снифферы** (sniffers - to sniff — нюхать) — это анализаторы сетевого трафика. Могут использоваться в составе зловредного ПО, скрытно устанавливаться на компьютере пользователя и отслеживать данные, которые отправляет или получает пользователь по сети.

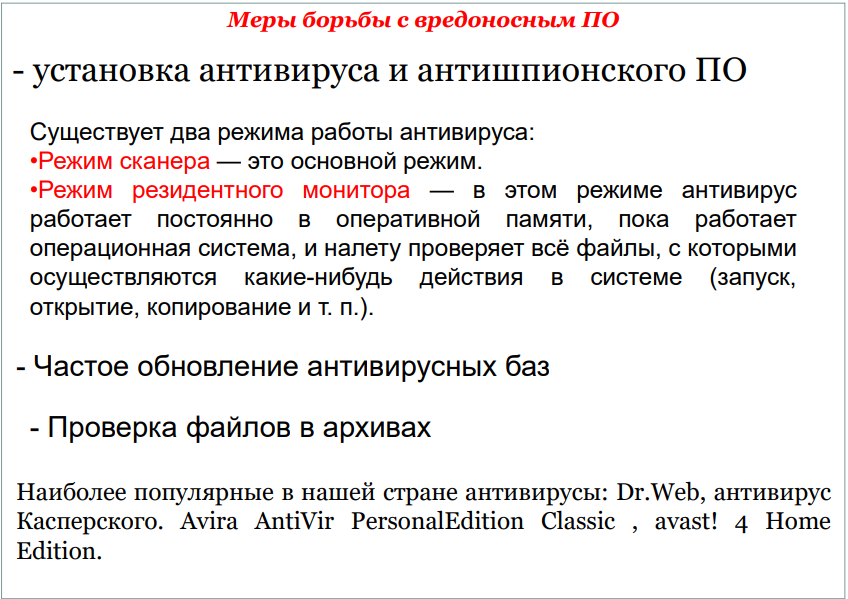
7. **Руткиты** (rootkits - наборы root'а) — сами по себе не являются зловредным ПО. Назначение — скрывать работу других зловредных программ (кейлоггеров, троянов, червей и т.д.) как от пользователя, так и от программ безопасности (антивирусов, файерволов, систем обнаружения атак и пр.).

8. “**Звонилка**” (Dialer или Porn-Dialer) —может просто изменять настройки уже существующих соединений удаленного доступа на компьютере пользователя или создавать новое соединение.

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

**Меры борьбы с вредоносным ПО**





1. Оценка безопасности парольной защиты.

Из чата

**Оценка безопасности парольной защиты**

Безопасность паролей играет критически важную роль в защите информации и систем от несанкционированного доступа. В оценке безопасности парольной защиты учитываются несколько ключевых факторов, таких как сложность паролей, методы их хранения, и применяемые механизмы аутентификации. Рассмотрим основные аспекты безопасности парольной защиты:

**1. Длина и сложность паролей**

* **Длина**: Чем длиннее пароль, тем сложнее его угадать или взломать методом перебора. Рекомендуется использовать пароли длиной не менее 12-16 символов.
* **Сложность**: Пароли должны содержать комбинации букв верхнего и нижнего регистра, цифры и специальные символы. Это увеличивает пространство поиска для атаки методом перебора.

**2. Политика управления паролями**

* **Регулярная смена паролей**: Рекомендуется регулярно менять пароли (например, каждые 3-6 месяцев).
* **Запрет на повторное использование**: Пользователи не должны использовать ранее использованные пароли.
* **Многофакторная аутентификация (MFA)**: Дополнение пароля вторым фактором аутентификации, таким как SMS-код, аппаратный токен или биометрические данные.

**3. Защита при хранении паролей**

* **Хеширование**: Пароли должны храниться в хешированном виде с использованием современных криптографических хеш-функций, таких как SHA-256, bcrypt или Argon2.
* **Соление**: Добавление случайного значения (соли) к каждому паролю перед его хешированием предотвращает атаки с использованием предвычисленных таблиц (rainbow tables).

**4. Обнаружение и предотвращение атак**

* **Брутфорс атаки**: Ограничение количества неудачных попыток ввода пароля и применение временной блокировки учетной записи после нескольких неудачных попыток.
* **Фишинг**: Обучение пользователей распознавать фишинговые атаки и использование фильтров для блокировки фишинговых писем.
* **Социальная инженерия**: Обучение пользователей основам информационной безопасности и методам защиты от атак социальной инженерии.

**5. Использование современных методов и технологий**

* **Менеджеры паролей**: Рекомендуется использовать менеджеры паролей для генерации и хранения сложных и уникальных паролей для каждой учетной записи.
* **Обновление ПО**: Регулярное обновление программного обеспечения для устранения уязвимостей, которые могут быть использованы для обхода парольной защиты.