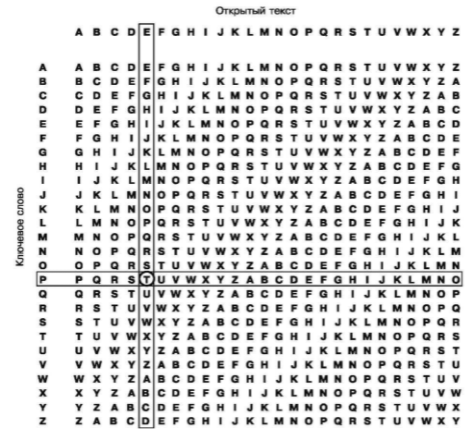
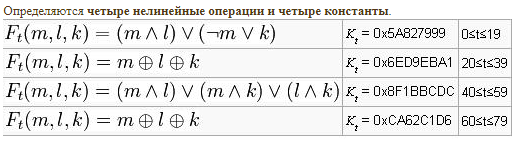
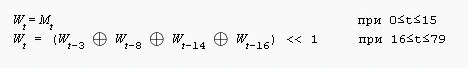
1. **Разница кодирования и шифрования**  
   кодирование – для удобства передачи по сокетам и прочей хуйне, шифрование – с целью безопасности шоб ничего не спиздили.
2. **Многоалфавитные шифры** – состоит из нескольких шифров простой замены.
3. **Моноалфавитные шифры** подстановки (шифр простой замены) – шифр, при котором каждый символ открытого текста меняется на определенный, фиксированный при этом ключе символ того же алфавита.
4. **Энтропия** – мера хаотичности информации, неопределенность появления какого-либо символа первичного алгоритма. При отсутствии информационных потерь численно равна количеству информации на символ передаваемого сообщения.
5. **Формула энтропии**:   
   Информационная энтропия для независимых случайных событий x с n возможными состояниями (от 1 до n) рассчитывается по формуле:   
   p(i) -–вероятность появления символа = количество вхождений символа поделенное на количество символов в тексте.
6. **Энтропия для слова ААААААА** = 7/7\*log2(7/7) = 0
7. **Смысл энтропии:**   
   На сколько мы по предыдущему сообщению можем предугадать следующий символ.
8. **Что лучше, энтропия = 3.2 или энтропия = 4.8 и почему?**   
   Чем больше энтропия – тем сложнее предугадать текст, идеальный шифр дает 8, который дает полную хаотичность. Значит лучше 4.8 в контексте безопасности.
9. **Как работает шифр Цезаря?**   
   Шифр подстановки, у которого каждый символ в тексте заменяется символом, которое на какое-то число позиций правее него в алфавите. Если число 3, то тогда А -> Г, Б -> Д.
10. **Rot – 13** – частный случай шифра Цезаря, сдвиг на 13 символов.
11. **Как работает шифр Виженера?**   
    Шифр Виженера – полиалфавитный подстановочный шифр. Тоесть для подстановки используется несколько алфавитов.   
    Таблица шифрования:   
      
    Тогда попробуем пошифровать:   
    Текст: I WANNA DIE  
    Ключ: EGOR  
    Синхронизируем текст с ключом:   
    I WANNA DIE  
    E GOREG ORE  
    Далее смотрим пересечение в строчке I, в столбике E и получаем M, аналогично с другими.   
    Получаем:   
    M COERG RZI
12. **Атака Беббиджа основана** на факте того, что повторяющиеся части открытого текста, зашифрованные одним и тем же ключевым словом, приводят к идентичным сегментам шифрованного текста.
13. **Как ломаем шифр Цезаря (статистически)?**   
    Можно взломать шифр цезаря с помощью частотного анализа, т.к. шифрование не меняет частоту появления символа.
14. **Как ломаем шифр Виженера (статистически для каждого полуалфавита)**Так же, как и шифр Цезаря, только для каждой n-ной буковки.
15. **Как определяем длину ключа в атаке Бебиджа (Касиски)?**В сообщении берем считаем все расстояния между одинаковыми подстроками в строке, например строка: HLVTHMLMHLVIEYUWYHLV  
    красным выделены одинаковые подстроки. Далее считаем количество букв от буквы H в первом слове HLV до буквы H во втором слове HLV и (8: делители – 2, 4, 8) от буквы H во втором слове HLV до буквы H в третьем слове HLV. (9: делители – 3, 9)   
    Дальше находим все делители (2, 4, 8, 3, 9) этих расстояний (не общие), не берем в учет единицу, т.к. если единица – это шифр Цезаря.   
    Дальше мы статистически считаем каждую n-ую букву для каждой возможной длины, это значит, что мы пытаемся расшифровать каждую n-ую букву с каждым возможным смещением алфавита. Та пара длины ключа и смещения алфавита, которая дала наилучший скор (наиболее похоже на английский язык статистически) и есть искомая длина ключа и смещение алфавита.
16. **Как дальше используем длину ключа?** Допустим мы нашли ключ, и он равен 3.   
    KEYKEYKEYKEYKEY  
    HELLOWORLDHELLO - сообщение  
    HLODL – ключ буква K  
    EORHL - ключ буква E  
    LWLEO - ключ буква Y  
    И те буквы, которые одинакового цвета в нижней строке (сообщение) имеют одинаковый сдвиг и каждую отдельную подстроку нужно анализировать как английский текст (тоесть взламывать как шифр Цезаря).
17. **Как ломаем дальше с помощью длины ключа?   
    То же шо 16**
18. **Как по итогу получаем значение ключа?   
    То же что и 15**
19. **Рандомные и псевдорандомные числа, как генерируются и чем отличаются?**Псевдорандомные числа – предсказуемый набор случайных чисел, который каждый раз будет одним и тем же, обладают свойствами случайных чисел, и генерируются с какой-то начальной точки (например время или дата etc) + сдвиг. Их не нужно использовать в криптографии, поскольку если взломщик узнает initial state, то может вполне подобрать сдвиг.   
    Рандомные числа – набор гарантировано случайных чисел, который генерируется с использованием каких-то физических данных (температура процессора, частота волны, скорость движение морской волны, температура воздуха, звуковой диапазон крика чайки).
20. **Алгоритм шифрования RSA**  
    Начнем с того, что это *асиметрическое шифрование*!!!!!!!!!!!!!!!!!!!  
      
    Шаг первый. Подготовка ключей.   
    Нужно сделать предварительные действия: сгенерировать публичный и приватный ключ.   
    - выбираем два простых числа, например p = 3, q = 7.   
    - вычисляем модуль – произведение наших чисел: n = p \* q = 3 \* 7 = 21  
    - вычисляем функцию Эйлера: ф = (p – 1) \* (q – 1) = 2 \* 6 = 12  
    - выбираем число е, которое должно соответствовать критериям:   
     1. Оно должно быть простое   
     2. Оно должно быть меньше ф. Имеем варианты: 3, 5, 7, 11  
     3. Оно должно быть взаимно простое с ф.   
    таким образом у нас остаются варианты: 5, 7, 11. Выбираем е = 5, это так называемая открытая экспонента.   
      
    Теперь пара чисел {e, n} = {5, 21} – наш открытый ключ. Он отправляется собеседнику, чтоб зашифровать сообщение, но это еще не все, нужно получить закрытый ключ.   
      
    Мне нужно получить число d, обратное e по модулю ф. Тоесть остаток от деления по модулю ф произведения d \* e, должен быть равен единице. Исходя из этого, d может быть равен 5, но чтоб не путать с e, возьмем d = 17.   
    Пара {d, n} = {17, 21} – наш секретный ключ, который мы не передаем никому. Он нужен для расшифровки зашифрованного открытым ключом сообщения.   
      
    Шаг второй. Шифрование.   
    Допустим сообщение – число 19. У нас имеется открытый ключ {e, n} = {5, 21}. Шифрование выполняется по следующему алгоритму:   
    - возводите ваше вообщение в степень e и выполняете деление по модулю n. Тоесть, возводим 19 в степень 5 = 2476099 и берем остаток от деления на 21, получаем 10.   
    10 – наше зашифрованное сообщение.   
    *Важное уточнение!!!!!!! Число в сообщении не должно быть больше n.*   
      
    Использование на практике:   
    Допустим есть слово КРОТ. Нужно перевести его в числовое представление, например код букв в системе ASCII. Для каждой буквы пишем число, и шифруем эти числа с помощью алгоритма, приведенного выше.   
    *Важно!!! Числа для генерации ключа должны быть достаточно большими.*
21. **Алгоритм дешифрования RSA**  
    Мы получили зашифрованные данные (10) и у нас есть приватный ключ {d, n} = {17, 21}.   
    Стоит обратить внимание на то, что открытый ключ не может расшифровать сообщение. А закрытый ключ мы никому не говорили, в этом и прелесть асиметрического шифрования.   
    Начинаем раскодировать:   
    - делаем операцию, очень похожую на кодирование, но вместо e используем d. Возводим сообщение в степень d: 10 в 17 степень, вычисляем остаток от деления на 21 и получаем число 19. Число 19 и есть нашим оригинальным сообщением.
22. **Расширенный алгоритм Эвклида**  
    1. Выберем начальную единичную матрицу E = {{1, 0}, {0, 1}}.   
    2. Вычислим остаток от деления r = a mod b.   
    3. Если r = 0, то решение – второй столбец матрицы E – решение задачи (вектор {x, y}).   
    4. С уравнения a = b\*q + r выведем q = (a - r) / b  
    5. Заменим E = E \* {{0, 1}, {1, -q}}. Переходим к пункту 2.
23. **Криптостойкость RSA**  
    До сих пор математически не доказано, что для восстановления m по e и n обязательно нужно раскладывать число n на множители. Но если будет открыт метод криптоанализа RSA, позволяющий получить d, то этот метод можно будет также использовать для разложения на простые множители числа n. Такие алгоритмы в настоящее время не найдены. Существует несколько достаточно эффективных способов взлома, но они работают только при выборе определенных параметров n, e и d, а также при неосторожном обращении с секретными ключами. Грамотным выбором параметров и четкой системой использования секретных ключей эффективность подобного рода атак сводится к нулю. Таким образом, считается, что стойкость рассматриваемого алгоритма полностью зависит от трудоемкости разложения на множители больших чисел.
24. **Использование RSA**  
    Используют для цифровой подписи или шифровки другого более простого ключа, поскольку достаточно большие вычислительные объемы.
25. **Для чего используется криптографическое хеширование? Какие алгоритмы для него используются?**   
    Хеширование генерирует из сообщения произвольной длины хеш-значение заданной длины. Поскольку одинаковое сообщение генерирует одинаковый хеш и необратимо, то хеширование необходимо для проверки целостности/подлинности или для секьюрного хранения данных.   
    Используемые алгоритмы:   
    SHA-1/SHA-256/SHA-512.
26. **Описать принципы алгоритма SHA-1, его преимущества и недостатки.**  
    Secure Hash Algorithm 1 — алгоритм криптографического хеширования. Для входного сообщения произвольной длины алгоритм генерирует 160-битное хеш-значение, называемое также дайджестом сообщения. Используется во многих криптографических приложениях и протоколах. Также рекомендован в качестве основного для государственных учреждений в США.

Исходное сообщение разбивается на блоки по 512 бит в каждом. Последний блок дополняется до длины, кратной 512 бит. Сначала добавляется 1, а потом нули, чтобы длина блока стала равной 512 — 64 бит. В оставшиеся 64 бита записывается длина исходного сообщения в битах. Дополнение последнего блока осуществляется всегда, даже если сообщение уже имеет нужную длину. Таким образом, число добавляемых битов находится в диапазоне от 1 до 512. Инициализируются пять 32-битовых переменных   
A = a = 0x67452301

B = b = 0xEFCDAB89

C = c = 0x98BADCFE

D = d = 0x10325476

E = e = 0xC3D2E1F0  
  
Главный цикл итеративно обрабатывает каждый 512-битный блок. Итерация состоит из четырех этапов по двадцать операций в каждом. Блок сообщения преобразуется из 16 32-битовых слов Mi в 80 32-битовых слов Wj по следующему правилу:  


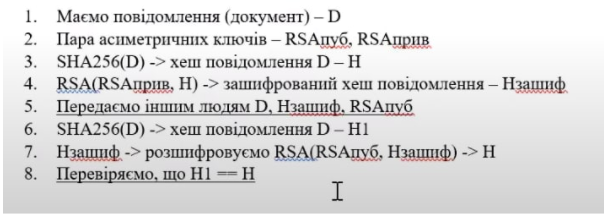
1. **С какой целью используется электронная цифровая подпись (ЭЦП)?**   
   Электронная цифровая подпись (ЭЦП) – мощное средство контроля подлинности информации в электронном виде, обеспечения целостности электронных данных, подтверждения их авторства и актуальности. Электронная цифровая подпись – это информационный объект, создаваемый для подписываемых данных, позволяющий удостовериться в целостности и аутентичности этих данных.
2. **Описать принцип работы ЭЦП**  
   Широко применяемая в настоящее время технология электронной подписи основана на асимметричном шифровании с открытым ключом и опирается на следующие принципы:

Можно сгенерировать пару очень больших чисел (открытый ключ и закрытый ключ) так, чтобы, зная открытый ключ, нельзя было вычислить закрытый ключ за разумный срок. Механизм генерации ключей строго определён и является общеизвестным. При этом каждому открытому ключу соответствует определённый закрытый ключ. Если, например, Иван Иванов публикует свой открытый ключ, то можно быть уверенным, что соответствующий закрытый ключ есть только у него.

Имеются надёжные методы шифрования, позволяющие зашифровать сообщение закрытым ключом так, чтобы расшифровать его можно было только открытым ключом. Механизм шифрования является общеизвестным.

Если электронный документ поддается расшифровке с помощью открытого ключа, то можно быть уверенным, что он был зашифрован с помощью уникального закрытого ключа. Если документ расшифрован с помощью открытого ключа Ивана Иванова, то это подтверждает его авторство: зашифровать данный документ мог только Иванов, т.к. он является единственным обладателем закрытого ключа.

Однако шифровать весь документ было бы неудобно, поэтому шифруется только его хеш — небольшой объём данных, жёстко привязанный к документу с помощью математических преобразований и идентифицирующий его. Шифрованный хеш и является электронной подписью.



1. **Какие существуют стандарты ЭЦП? Чем они отличаются? Где целесообразнее их использовать?**Американские стандарты электронной цифровой подписи: DSA, ECDSA

Российские стандарты электронной цифровой подписи: ГОСТ Р 34.10-94 (в настоящее время не действует), ГОСТ Р 34.10-2001

Украинский стандарт электронной цифровой подписи: ДСТУ 4145-2002

Стандарт PKCS#1 описывает, в частности, схему электронной цифровой подписи на основе алгоритма RSA

схема Шнорра

ElGamal

Вероятностная схема подписи Рабина

Отличаются они используемыми алгоритмами.

1. **На основе какой особенности хеширования может быть осуществлена атака на данные, защищенные с помощью ЭЦП? Какое название носит парадокс, на котором эта атака базируется? В чем его математическая подоплека? Як повысить защищенность ЭЦП от такой атаки?**   
   Атака “дней рождения” полагается на высокую вероятность нахождения коллизий между случайными попытками и порядком перестановок

Коллизии:

1 — Для N почти невозможно подобрать такое сообщение М, для которого H(N) = H(M)

2 — почти невозможно найти пару сообщений, для которых H(m1) = H(m2)

Отличия в том, что идут две разных атаки

1. **Стеганографія. Основні поняття - стеганографічна система, повідомлення, контейнер, ключ.**  
   Стеганографія - це наука про приховану передачу інформації шляхом збереження в таємниці самого факту передачі. На відміну від криптографії, яка приховує вміст секретного повідомлення, стеганографія приховує саме його існування.  
   Стеганографічна система (стегосистема) - об'єднання методів і засобів, що використовуються, для створення прихованого каналу для передачі інформації. При побудові такої системи домовилися про те, що:
   1. ворог уявляє роботу стеганографічної системи. Невідомим для противника є ключ за допомогою якого можна дізнатися про факт існування і змісту таємного повідомлення.
   2. При виявленні супротивником наявності прихованого повідомлення, він не повинен мати змогу отримати повідомлення до тих пір поки він не буде володіти ключем.
   3. Противник не має технічних та інших переваг.

Повідомлення - це термін, що використовуються, для загальної назви переданої прихованої інформації, буде це текст написаний молоком на білому папері або цифровий файл, в якому захована інформація.  
Контейнер - так називається будь-яка інформація, яка використовується для приховування таємного повідомлення. Порожній контейнер - контейнер, що не містить таємного послання. Заповнений контейнер (стегоконтейнер) - контейнер, що містить таємне послання.  
Ключ (стегоключ) - таємний ключ, потрібний для приховання стегоконтейнера. Ключі в стегосистемі бувають двох типів: приватні та публічні. Якщо стегосистеми використовує приватний ключ, то він повинен бути створений або до початку обміну повідомленнями, або переданий по захищеному каналу. Стегосистема, що використовує публічний ключ, повинна бути влаштована таким чином, щоб було неможливо отримати з нього приватний ключ. В цьому випадку публічний ключ ми можемо передавати по незахищеному каналу.

1. **Комп'ютерна стеганографія. Приклади.**  
   Комп'ютерна стеганографія - напрям класичної стеганографії, заснований на особливостях комп'ютерних платформ. Приклади - стеганографічна файлова система StegFS для Linux, приховування даних в невикористовуваних областях форматів файлів, підміна символів в назвах файлів, текстова стеганографія і т.д.  
   Приклади:
   1. Використання зарезервованих полів комп'ютерних форматів файлів - суть методу полягає в тому, що частина поля розширень, що не заповнена інформацією про розширення, за замовчуванням заповнюється нулями. Відповідно ми можемо використовувати цю «нульову» частину для запису своїх даних. Недоліком цього методу є низький ступінь прихованості(скритності) і малий обсяг інформації, що передається.
   2. Метод приховування інформації в місцях гнучких дисків, що не використовуються - при використанні цього методу інформація записується в частини диска, що не використовуються, наприклад, на нульову доріжку. Недоліки: низька продуктивність, передача невеликих за обсягом повідомлень.
   3. Метод використання особливих властивостей полів форматів, які не відображаються на екрані - цей метод заснований на спеціальних «невидимих» полях для отримання зносок, покажчиків. Наприклад, написання чорним шрифтом на чорному тлі. Недоліки: низька продуктивність, невеликий обсяг інформації, що передається.
   4. Використання особливостей файлових систем - при зберіганні на жорсткому диску файл завжди займає ціле число кластерів (мінімальних обсягів інформації, що адресуються). Наприклад, в файловій системі FAT32 стандартний розмір кластера- 4 КБ. Відповідно для зберігання 1 КБ інформації на диску виділяється 4 КБ, з яких 1КБ потрібен для зберігання файлу, а решта 3 ні на що не використовуються - відповідно їх можна використовувати для зберігання інформації. Недолік даного методу: легкість виявлення.
2. **Цифрова стеганографія. Алгоритми.**  
   Цифрова стеганографія - напрям класичної стеганографії, заснований на приховуванні або впровадженні додаткової інформації в цифрові об'єкти, викликаючи при цьому деякі спотворення цих об'єктів. Але, як правило, дані об'єкти є мультимедіа-об'єктами (зображення, відео, аудіо, текстури 3D-об'єктів) та внесення спотворень, які знаходяться нижче порога чутливості середньостатистичної людини, не приводить до помітних змін цих об'єктів. Крім того, у відцифрованих об'єктах, що спочатку мали аналогову природу, завжди присутній шум квантування; далі, при відтворенні цих об'єктів з'являється додатковий аналоговий шум і нелінійні спотворення апаратури, все це сприяє кращому прихованню інформації.  
   Всі алгоритми вбудовування прихованої інформації можна розділити на кілька підгруп:
   1. Працюючі з самим цифровим сигналом. Наприклад, метод LSB.
   2. «Накладення» інформації, що приховується. В даному випадку відбувається накладення зображення (звуку, іноді тексту), що приховується за оригіналом. Часто використовується для вбудовування цифрових водяних знаків (ЦВЗ).
   3. Використання особливостей форматів файлів. Сюди можна віднести запис інформації в метадані або в різні інші зарезервовані поля файлу, що не використовуються.
3. **Метод LSB.**  
   LSB (Least Significant Bit, найменший значущий біт) - суть цього методу полягає в заміні останніх значущих бітів в контейнері (зображення, аудіо або відеозапису) на біти повідомлення, що приховується. Різниця між порожнім і заповненим контейнерами повинна бути не відчутна для органів сприйняття людини.  
   Приклад

Припустимо, є 8-бітове зображення в градаціях сірого. 00h (00000000b) позначає чорний колір, FFh (11111111b) - білий. Усього є 256 градацій (28). Також припустимо, що повідомлення складається з 1 байта - наприклад, 01101011b. При використанні 2 молодших біт в описах пікселів, нам буде потрібно 4 пікселя. Припустимо, вони чорного кольору. Тоді пікселі, що містять приховане повідомлення, будуть виглядати наступним чином: 00000001 00000010 00000010 00000011. Отже колір пікселів зміниться: першого-на 1/255, другого і третього-на 2/255 і четвертого- на 3/255. Такі градації, мало того що непомітні для людини, можуть взагалі не відображатися при використанні низькоякісних пристроїв виведення зображення.

Методи LSB є нестійкими до всіх видів атак і можуть бути використані тільки при відсутності шуму в каналі передачі даних. Виявлення LSB-закодованого стего здійснюється за допомогою аномальних характеристик розподілу значень діапазону молодших бітів відліку цифрового сигналу.

Інші методи приховування інформації в графічних файлах орієнтовані на формати файлів з втратою, наприклад, JPEG. На відміну від LSB вони більш стійкі до геометричних перетворень. Це можливо за рахунок змін у широкому діапазоні якості зображення, що призводить до неможливості визначення джерела зображення.

1. **Стегоаналіз.**   
   Стегоаналіз - це протидія стеганографії, як криптоаналіз - це протидія криптографії. Основна мета стеганографії - приховати факт передачі даних. Отже, основна мета стегоаналіза - виявити факт приховування передачі даних.  
   Розглянемо метод виявлення послідовного вбудовування в LSB одного біта на прикладі зображення у форматі BMP.

Існує думка, що в зображеннях LSB є випадковими. Насправді це не так! Хоча людське око і не помітить змін зображення при зміні останнього біту, статистичні параметри зображення будуть змінені. Перед приховуванням дані зазвичай архівуються (для зменшення об’єму) або шифруються (для забезпечення додаткової стійкості повідомлення при потраплянні у чужі руки). Це робить біти даних дуже близькими до випадкових. Послідовне вбудовування такої інформації замінить LSB зображення випадковими бітами. І це можна виявити!

Для прикладу візьмемо одну кольорову компоненту повнокольорового зображення BMP і на ній покажемо процес знаходження вбудовування. Яскравість кольорової компоненти може приймати значення від 0 до 255. У бінарній системі - від 0000 0000 до 1111 1111.

Розглянемо пари:

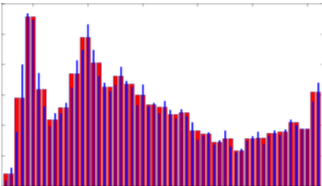
0000 00**00** <-> 0000 00**01**;

0000 00**10** <-> 0000 00**11**;

...

1111 11**00** <-> 1111 11**01**;

1111 11**10** <-> 1111 11**11**.

Дані числа відрізняються між собою тільки в LSB (виділені жирним). Таких пар для кольорової компоненти BMP зображення: 256/2 = 128. У разі стеганографічного вбудовування тобто заміни LSB на випадкову послідовність, кількість пікселів в парах вирівняється. Гістограма приймає «сходинковий» ​​вид (два сусідніх значення яскравості зустрічаються з однаковою частотою).  
  
На малюнку синім кольором позначена гістограма зображення без вбудовування, а червоним - гістограма того ж зображення після вбудовування даних замість. Порівняння двох гістограм дає можливість стегоаналізу послідовно приховуваних біт.