**Лаба 5: перестановочные шифры**

1. В чем заключается основная идея криптографических преобразований на основе шифров перестановки?

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.

2. Привести классификационные признаки и дать сравнительную характеристику разновидностям подстановочных шифров.

Относится к классу симметричных. В класс криптографии шифры перестановки делятся на 2 подкласса:

1 - простой(ординарной) перестановки. При замене символы М перемещаются с исходной позиции на новые один раз.

2 - сложный (множественной) перестановки, те символы перемещаются с исходной позиции несколько раз

3. Сколько разновидностей шифров, подобных шифру Цезаря, можно составить для алфавитов русского и белорусского языков?

N - 1, где N - число букв в алфавите

4. Охарактеризовать криптостойкость перестановочных и подстановочных шифров.

Подстановочный - N!

Перестановочный - при отсутствии повторяющихся букв в шифруемом сообщении длиной N символов всего существует N! неповт ключей

5. Привести примеры дать характеристику перестановочным шифрам, не рассмотренным в материалах к данной лабораторной работе.

Поворотная решетка?

Шифр с применением специальной решетки с дырками, которая накладывается на зашифрованный текст и по буквам в этих дырках читается шифрованное сообщение.

6. Имеются ли предпочтения в выборе размеров используемой таблицы для перестановочных шифров?

Таблица ближе к квадратному виду

7. Охарактеризовать основные методы взлома перестановочных шифров.

Можно попробовать статический перебор/ угадать маршрут

8. Классификация шифров

* На основе процедуры шифрования (подстановочные, перестановочные)
* На основе генерирования использования ключа (блочные, поточные)
* На основе типа ключа (симм. / асимм.)

**Лаба 6: Энигма**

Ротор

ротор производил очень простой тип шифрования: элементарный шифр замены. Например, контакт, отвечающий за букву E, мог быть соединён с контактом буквы T на другой стороне ротора. Но при использовании нескольких роторов в связке (обычно трёх или четырёх) за счет их постоянного движения получается более надежный шифр.

Рефлектор

За последним ротором следовал рефлектор. Рефлектор соединял контакты последнего ротора попарно, коммутируя ток через роторы в обратном направлении, но по другому маршруту. Наличие рефлектора гарантировало, что преобразование, осуществляемое «Энигмой», есть инволюция, то есть расшифрование представляет собой то же самое, что и шифрование. Однако наличие рефлектора делает невозможным шифрование какой-либо буквы через саму себя. Это было серьёзным концептуальным недостатком, впоследствии пригодившимся дешифровщикам.

Были различные варианты Энигмы, где рефлектор был неподвижен в процессе шифрования и где рефлектора двигался во время шифрования, как и остальные диски.

Для начала шифрования нужно указать положения всех трех роторов:

(правого, центрального, левого). I II III B 0-2-2

Основной недостаток немецкой Энигмы было то, что одну и ту же букву зашифровало одной и той же другой буквой

1. Дать пояснение к структуре шифровальных машин Энигма.

2. На основе каких шифров строится машина Энигма?

Перестановочные и подстановочные

3. Дать пояснение к принципам зашифрования сообщений.

4. Дать характеристику криптостойкости шифровальной машины

Энигма.

26\*26\*26 = 17576: кол-во букв можем установить на роторе

10\*9\*8=720: всего 10 панелей было, мы можем на 1ую 10 панелей, на 2ую 9 панелей и на 3ью 8 панелей

26!/2!\*13! = 32 382 376 266 240 000: так же у нас был рефлектор, которым мы могли менять наши буквы (те комбинаций, которых мы можем получить)

32 382 376 266 240 000 \* 720 \* 175<76 = 409 789 904 583 912 652 800 000: ну и конечное число - это сколько всего комбинаций мы можем установить, получается достаточно большое число и можно сделать вывод, что Энигма достаточно криптостойкая в этом плане

5. Дать характеристику (с численными оценками) криптостойкости ма-

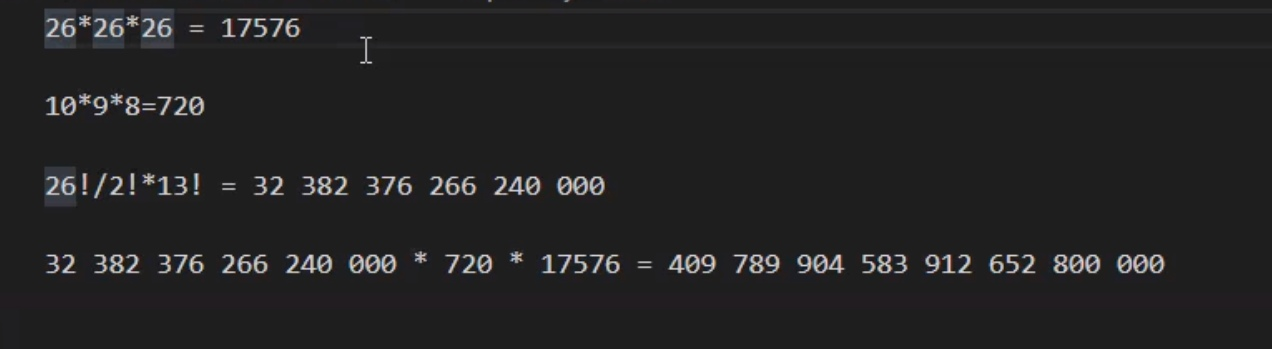
шины-симулятора на основе разработанного приложения.

(смотри 4)

6. Пояснить основные принципы расшифрования сообщений Энигмы.

Утсанавливаем роторы в исходные позиции и тогда мы получим тот текст, который был зашифрован

7. Ваши предложения по модификации известных аналогов Энигмы?



**Лаба 7: Исследование блочных шифров (DES)**

**Алгоритм Des:**

Есть начальное сообщение которое делим на блоки по 64 бит, если 64 нет до докидываем наборами битов которые установили до нас.Блоки делятся на блоки по 32 бит и расширяется до 48 для взаимодействия с ключом. Боковые биты выставляются в обратном порядке.(Заложено до нас). Дальше генерим сам ключ, который состоит из 64 бит и на каждой 8 позиции ставим тестовые биты которые позволяют сохранять информацию и проверять ключ на цельность. После их удаления ключ становится 56. Ключ делится на два боковых ключа Ci и Di по 28 битов. Потом мы на позициях 1,2,9,16 сдвигаем на единицу, а остальные на 2 бита. После перестановки мы снова получаем ключи Ci+1 и Di+1 и они организуются в один ключ и начинают взаимодействовать с нашим сообщением. Производим операцию xor (если 1 и 1 это 0, 0 и 0 это 0,а 1 и 0 это 1) после преобразования xor получим таблицу 6 на 8. Два крайних мы складываем в одну таблицу которые отвечают за строи который мы используем, центральный номер столбца который используем. И выставляем отсюда числа. Так же можно сказать что мы могли делать перестановки которые перемешивают биты это конечно не увеличивает криптостойкость но придумано до нас.

***Ключи:***  
Сначала 64

Каждый 8 бит удаляется

Получаем 56 бит

Потом есть специальный алгоритм

По которому расставляются биты

Там есть табличка

Эти биты расставляются

Получается 56 расставленных битов

Теперь у тебя берется 28 первых битов

Это блок C0

И оставшиеся 28 битов

Это блок D0

И делаются сдвиги

Причем каким образом

У тебя есть 16 раундов

1, 2, 9 и 16 раунды делается циклический сдвиг влево на 1

В остальных - на 2

Это для шифрования

После того, как в раунде произошел сдвиг

C0 и D0 объединяются

И ты снова получаешься 56-битный ключ

И теперь тебе нужен подключ раунда

Ты, скажем, создаешь копию полученного 56-битного ключа

И по таблице убираешь еще 8 бит

Получается 48-битный

Он только для этого раунда

Алгоритм блочный, симметричный.

Слабый ключ - состоит только из 0 и единиц.

Полуслабый ключ - при котором разные ключи дают одинаковый результат при шифровании.

Вопросы:

2. В чем отличие блочных алгоритмов шифрования от потоковых?

Главным различием блочных и поточных шифров является то что в блочных шифрах для шифровании всех порций используется один и тот же ключ, а для поточных ключи разной длины

3. Что понимается под "раундом" алгоритма шифрования?

Раундом называют один из последовательных шагов обработки данных в алгоритме блочного шифрования.

4. Охарактеризовать и привести формальное описание сети Фейстеля.

Даже само название сети фейстеля означает что она имеет ячеистую топологию.

Можно сказать что одна ячейка соответствует одному раунду зашифрования или расшифрования. Каждый входной блок шифруемого сообщения делится на два одинаковых по размеру это на левый и правый. Далее в каждом раунде выполняется преобразовании в соответствии с представлением ячейки Фейстеля

5. Какие стандартные операции используются в блочных алгоритмах шиф-

Рования?

Простейшие операции: Лежит операция xor : A+B(mod n). Помимо этой операции в ГОСТ и Blowfish предусмотрена операция сложения более по модулю 2^n

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

Сложение по модулю 2^n увеличивает криптостойкость для блочных шифров.

9. Дать пояснение принципам реализации «лавинного» эффекта.

Лавинный эффект это понятие которое часто применяется к блочным шифрам . Это свойство которое означает, что изменение малого количества бит во входном текста или ключе приводит к значительному изменению значений выходных битов

12. Сколько можно реализовать (теоретически) разновидностей алгоритма 3DES?Существует несколько алгоритмов для тройного деса.

DES-EEE3 Шифруется 3 раза с тремя разными ключами

DES-EDE3 операции шифрования расшифрования с различными ключами

DES-EEE2 используется для того что бы зашифровывать только 2 ключа совпадают

DES-EDE2 используется два одинаковых числа

13. Какие факторы влияют на стойкость блочного алгоритма шифрования?

На стойкость блочного шифра оказывают влияние размер самого блока, размер ключа и количество раундов шифрования.

14. В чем состоит сущность дифференциального криптоанализа?

Метод дифференциального криптоанализа - метод криптоанализа в блочных шифрах который основан на изучении преобразования разностей между шифрованными значениями на разных раундах шифрования

15. В чем состоит сущность линейного криптоанализа?

Линейный криптоанализ - называется методом криптоанализа который соотносит известные пары открытого текста и шифротекста для получения ключа.

17. Где применяются блочные криптоалгоритмы?

Ну тот же des используется для защиты финансовой информации, тройной дес используется при обработки кредитных карт

**Лаба 8 Поточные шифры:**

p и q простые числа которые генерируют n

n в совместимости с таким числом как e которое так же является простым ключом

d генерируется при помощи е и фи называется секретным ключом.

Пример дешифрования к нам пришли {5,3,3} . Для того что бы дешифровать мы эту 5 возводили в степень d и искали бы модуль от числа 3337 . Если бы мы шифровали то мы 199 возводили бы в степень е и искали бы по модулю 3337. Все веселье в том, что у нас есть два ключа и только при помощи закрытого ключа мы можем расшифровать текст зашифрованный при помощи открытого ключа.

Ну как говорили люди уже сдававшие до меня у нас есть такие так называемые Боб, Алиса, Ева . Предположим что Боб хочет послать Алисе сообщение m. Алгоритм шифрования в котором мы сначала берем текст и зашифровываем его при помощи открытого ключа Алисы . Алиса принимает зашифрованное сообщение берет свой закрытый и с помощью его расшифровывает сообщение. Плюс в том что если Ева захочет преобразовать наше сообщение обратно она не сможет это сделать даже имея все открытые ключи и зашифрованное сообщение. Это плюс

Минус что при изменении даже 1 символа в битовой последовательности все будет нарушено и мы не получим наше сообщение.

Схема на практике не юзается так как при одних и тех же входных значениях мы получим один результат

ПСП на основе регистров сдвига

Есть число 84320 которое делало генератор 1...8 нулевой биты нулевой соединятся с ближайшим xor а так же бит 2 3 4 соединяются с xor. Как делается ну рассказал а как генерируется ну есть приводимы и неприводимые 84320 является приводимым так как в конце вычисления мы получаем 255 а это 256-1 равняется 255 тоесть порядок его 255 это находится 2 в степени количество все регистров минус еденица . Есть так же неприводимые 420 к примеру 420 так как его порядок 6 а два в шестой степени это куда больше чем у нас получается. Если они неприводимы то мы проверяем битовую последовательность пока не получим значение в нулевом

Беру значения возле xor и последнее значение суммирую беру по модулю двойки и в зависимости от того что получилось но или еденица ставлю это на первое место а остальные биты сдвигаю вправо

Алгоритм BBS

Начальное значение x^2mod(n) где n как и в rsa считается произведением чисел p и q но в это случаем при делении чисел p и q на 4 мы должны получать остаток 3; число x должно быть взаимно простым с n;

Алгоритм RC4

Алгоритм RC4 формируется на основе ключа ну к примеру у нас формируется ключ он состоит из 6 битов понятно чти это битовая последовательность из нулей и едениц. Есть какие то 16 символов от 0 до 15 в этом случае для них генериуется ключ так как он по 6 битов тоесть каждые 6 битов вставляются и формируется таблица и проходим весь алгоритм который представлен ниже

Ответы на вопросы:

1. Особенность потокового в том что каждый символ сообщения М превращается в зашифрованный С при этом алгоритм по которому происходит шифрование зависит от какого-то ключа и от местоположения в тексте.

2.

В синхронных ключевая последовательность генерируется независимо от открытого текста. Главное свойство СПШ – нераспространение ошибок, т.е. Если произойдет ошибка в открытом тексте, то это никак не повлияет на ключ. Синхронные потоковые шифры уязвимы к атакам на основе изменения отдельных бит шифртекста.

А в асинхронных (самосинхронизирующиеся) есть зависимость от текста, т.е. У нас уже будет распространение ошибок. Но при этом он устойчив к атакам отдельных бит.

3.

случайные числа (последовательности) – последовательность элементов, каждый из которых не может быть предсказан (вычислен) только на основе знания предшествующих ему элементов данной последовательности;

• псевдослучайные числа – последовательность элементов, полученная в результате выполнения некоторого алгоритма и используемая в конкретном случае вместо последовательности случайных чисел.

Он должен обладать свойством что на основании каких то элементов этой последовательности мы не можем вычислить абсолютно все что нам нужно ведь это должны быть какие то случайные числа быть.

4.

Достоинством линейных конгруэнтных генераторов псевдослучайных чисел является их простота и высокая скорость получения псевдослучайных значений. Линейные конгруэнтные генераторы находят применение при решении задач моделирования и математической статистики, однако в криптографических целях их нельзя рекомендовать к использованию, так как специалисты по

криптоанализу научились восстанавливать всю последовательность ПСЧ по нескольким ее значениям.

Часто реализуемые алгоритмы псп реализуются на основе конгруэнтного генератора. Генераторы не используются для криптографии так как у их низкая криптостойкость и но они используются при моделировании

Числа а (период) и с (от b- 1 до 2b)

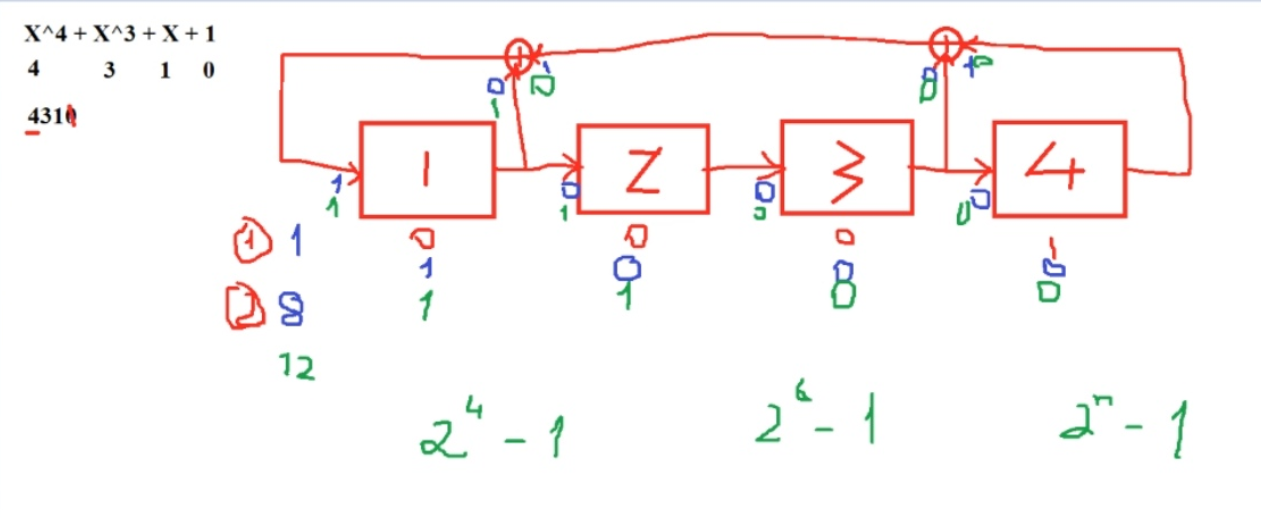
1

Зависит от ключа и места символа в тексте

5

-----

6



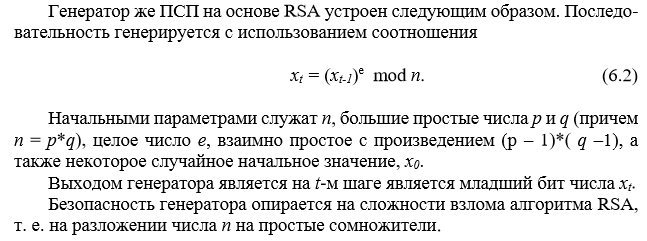
Кол-во послед 2^(кол-во ячеек - 1)

***Приводимый***( регистра будет максимально возможным: 2L – 1). и **не приводиый** (кодга повторение происходит меньше чем через 2^n-1)

7

8

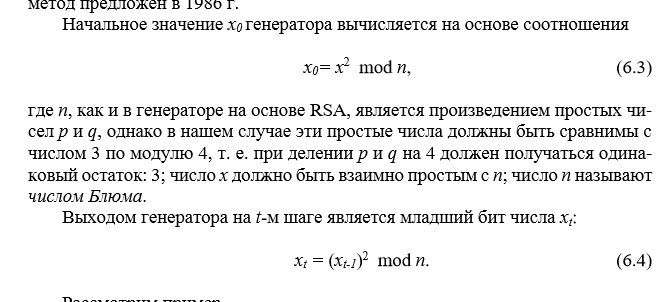
9

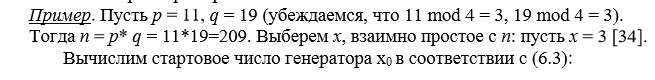


6.1.2.3

P и Q

10





х0 = х^2 mod n = 3^2 mod 209 = 9 mod 209 = 9.

х1 = 9^2 mod 209 = 81 mod 209 = 81

х5 = 42^2 mod 209 = 1764 mod 209 = 92

х9 = 196^2 mod 209 = 38416 mod 209 = 169

11 RC4

12

Random.org вычисляет на основе давления и темп воздуха (с помощью физ явл которые мы не способны предсказать)

**Лаба 9: Исследование асимметричных шифров**

**Зашифрование сообщения**

**Для зашифрования сообщения (М) оно сначала разбивается на блоки, по размерам равные числу (z) элементов последовательности в ранце. Затем, считая, что 1 указывает на присутствие элемента последовательности в ранце, а 0 – на его отсутствие, вычисляются полные веса рюкзаков (Si, i = 1, . . ., z): по одному ранцу для каждого блока сообщения с использованием открытого ключа получателя, e.**

**Открытый ключ, е: {62, 93, 186, 403, 417, 352, 315, 210} .**

**Результат зашифрования (упаковки ранца) каждого блока (буквы) сообщения с помощью открытого ключа представлен в правом столбце таблицы 7.1. Пояснение к примеру зашифрования сообщения укладкой ранца Бинарный код символа mj сообщения Укладка ранца Вес ранца 11010000 62+93+403 558.**

**Таким образом, зашифрованное сообщение С = 558 470 155 365 1239 155 924: с1 = 558, с2 = 470 и т. д.**

**Расшифрование сообщения**

**Для расшифрования сообщения получатель (использует свой тайный ключ, d: сверхвозрастающую последовательность) должен сначала определить обратное к а число: а-1 , такое что a\*а-1(mod n) = 1.**

**Si = ci \* а-1 mod n**

**В нашем примере значение а-1 = 271: 31\*271 mod 420 = 1. Вспомним, что сверхвозрастающая последовательность равна d= {2, 3, 6, 13, 27, 52, 105, 210}, а также n = 420, a = 31; шифртекст: 155 365 558 155 924 1239 470 Расшифрование первого блока шифртекста. Сначала вычисляем, используя (7.4), вес первого ранца (при c1=155): S1 = c1\*а -1 (mod n) = 155\*271 (mod 420) =5.**

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

**Криптостойкость алгоритма во многом определяется скоростью (временем) поиска нужного варианта укладки ранца. Понятно, что для последовательности из шести-десяти или немногим более того элементов нетрудно решить задачу укладки ранца, даже если последовательность не является сверхвозрастающей. При практической же реализации алгоритма ранец должен содержать не менее нескольких сотен элементов. Длина каждого члена сверхвозрастающей последовательности должна быть несколько сотен бит, а длина числа n – от 100 до 200 бит. Для получения этих значений практические реализации алгоритма используют генераторы ПСП.**

**Самый быстрый алгоритм поиска решения имеет экспоненциальную зависимость от числа возможных предметов. Если добавить к последовательности весов еще один член, то найти решение станет вдвое труднее. Это намного труднее сверхвозрастающего ранца, где, при добавлении к последовательности одного элемента, поиск решения увеличивается на одну операцию.**

**Ранцевые криптосистемы не являются криптостойкими. А. Шамир и Р. Циппел обнаружили, что, зная числа а, a-1 и n («секретную лазейку»), можно восстановить сверхвозрастающую последовательность по нормальной последовательности. Важно то, что числа а и n («секретная пара») не обязательно должны быть теми же, что использовались при создании системы легальным пользователем.**

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

**Ранцевый (рюкзачный) вектор S = (s1, . . ., sz) – это упорядоченный набор из z, z ≥ 3, различных натуральных чисел si.**

**2. Сформулировать задачу укладки ранца.**

**В наиболее известном варианте задачи о ранце требуется выяснить, обладает или нет данный вход (S, S) решением. В варианте, используемом в криптографии, нужно для данного входа (S, S) построить решение, зная, что такое решение существует. Оба эти варианта являются NP-полными. Имеются также варианты этой задачи, которые не лежат даже в классе NP.**

**8. Можно ли, с Вашей точки зрения, одновременно зашифровывать (и, соответственно, – одновременно расшифровывать) более, чем по одному символу текста. Обосновать решение. Привести примеры решений.**

**Да,**

**9. Что такое «секретная лазейка»?**

**зная числа а, a-1 и n («секретную лазейку»), можно восстановить сверхвозрастающую последовательность по нормальной последовательности**

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

**Криптостойкость алгоритма во многом определяется скоростью (временем) поиска нужного варианта укладки ранца. Ранцевые криптосистемы не являются криптостойкими.**

**Сейчас не используется.**

**Лаба 10:**

**Исследование асимметричных шифров RSA и Эль-Гамаля**

Алгоритм RSA

Алгоритм с открытым ключом

Основан на трудности разложения на множители больших чисел. Открытый и закрытый ключи являются функциями двух больших простых чисел.

Выбираются два больших случайных простых числа, p и q. Для максимальной большей криптостойкости нужно выбирать p и q равной длины.

n = pq

Ключ состоит из чисел n, e, d.

случайным образом выбирается e, такой что e взаимнопростой с функцией эйлера (p-1)(q-1).

Вычисляется d с помощью расширенного алгоритма евклида à d-1 = e(mod φ(n)).

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

зашифрование ci = (mi) e mod n.

расшифрование mi = (ci) d mod n.

Криптостойкость

В 1999 году 512-битный ключ был вскрыт за семь месяцев. Это означает, что 512-битные ключи уже не обеспечивают достаточную криптостойкость. Сейчас в критических системах применяются ключи длиной 1024 и 2048 бит.

Алгоритм Эль-Гамаля

Используется для зашифрования/расшифрования данных, для формирования цифровой подписи и для согласования общего ключа

Основан на трудности вычисления дискретных логарифмов.

Выбирается простое число, р. Выбирается число (g, g < p), являющееся первообразным корнем числа р.

Выбирается число х (х < p) и вычисляется y è y = g х mod р

Шифрование происходит с использованием открытого ключа получателя: p, g, y. Расшифрование шифртекста получатель производит своим тайным ключом: p, g, х.

Тайным является лишь одно число (как и в RSA)

Первообразный корень по модулю р является таким числом, что его степени (gi , 1 ≤i≤p-1 ) дают все возможные по модулю р остатки, которые взаимно просты с p

Блок шифртекста (ci) состоит из двух чисел: аi и bi

ai = gk mod p

bi = (yk \*mi) mod p

Недостаток алгоритма шифрования Эль-Гамаля: удвоение в 1,5 раза длины зашифрованного текста по сравнению с начальным текстом.

Расшифрование ci выполняется по следующей формуле: mi = (bi \*(ai)x)-1 ) mod p

Криптостойкость

Если для зашифрования двух разных блоков (m1 и m2) некоторого сообщения использовать одинаковые k, то для соответствующих шифртекстов c1 = (a1, b1) и c2 = (a2, b2) выполняется соотношение b1(b2)-1 = m1(m2)-1 . Из этого выражения можно легко вычислить m2, если известно m1.

При примерно одинаковой размерности ключей рассмотренные алгоритмы обеспечивают примерно одинаковый уровень криптостойкости.

**Лаба 11:**

**Исследование криптографических 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 длины L

Хеширование – это преобразование входного массива данных определенного типа и произвольной длины (практически) в выходную битовую строку фиксированной длины.

Результаты называют хешем, хеш-кодом, хеш-таблицей или дайджестом сообщения

Все 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.

Основные задачи, решаемые с помощью хеш-функций:

• аутентификация (хранение паролей),

• проверка целостности данных,

• защита файлов,

• обнаружение зловредного ПО,

• криптовалютные технологии.

Основные свойства хеш-функций:

Ø Детерминированность

Ø Скорость вычисления хеша h – система неэффективна если скорость низкая

Ø Сложность обратного вычисления - односторонность преобразования

Типы хеш-функций:

Ø Специализированные (MD)

Ø На основе блочных шифров – MDC

Ø На основе модулярной арифметики

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').

В коллизии первого рода надо подобрать текст, хэш которого совпадает с имеющимся хэшем. То есть при лобовой атаке перебираем тексты, пока однажды хэш не совпадёт с заданным. В коллизии второго рода надо подобрать 2 разных текста с одинаковыми хэшами. То есть перебираем тексты, пока хэш не совпадёт с одним из ранее полученных хэшей. Вероятность этого существенно выше. Приближённо, чтобы с вероятностью 0.5 найти коллизию первого рода надо перебрать 2n-1 текстов, а для 2го – корень из 2n.

Постулат Парадокса «Дней Рождения» гласит: в группе минимум из 23 человек с вероятностью более 0,5 день рождения у 2-х лиц одинаков. Парадоксом является высокая (как кажется на первый взгляд) вероятность наступления указанного события. При этом предполагается, что

1. в этой группе нет близнецов;

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

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

4. с формальной точки зрения это означает, что вероятность р1 рождения отдельно выбранного члена группы (как и любого человека) в любой выбранный день равна р1 =1/365 (хотя известно, что в реальности рождение людей не совсем соответствует такому предположению).

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).

С 5 августа 2015 г. утвержден и опубликован в качестве действующего стандарта (FIPS 202) алгоритм SHA-3, одной из отличительных особенностей которого является использование конструкции «криптографической губки». В этой конструкции реализован итеративный подход для создания функции с произвольной длиной на входе и произвольной длиной на выходе на основе определенного преобразования. На основе технологии «губки» построен ныне действующий в РБ стандарт хеширования.

Алгоритмы семейства MD входные сообщения максимальной длины 264-1 бит преобразуют в хеш длиной l = 128 бит. Исключением является последняя – 6 – из версий алгоритма, где длина результирующего хеша может изменяться от 1 до 512 бит

Максимальный объем хешируемых сообщений для алгоритмов SHA-1, SHA-256, SHA-224 такой же, как и для алгоритмов MD. Однако длина хешей разная: в SHA-1 – 160 бит; во алгоритмах, относящихся к семейству SHA-2 – соответствует числу, дополняющему через дефис название алгоритма. Максимальная же длина входных сообщений в алгоритмах SHA-512, SHA-384, SHA-512/256, SHA-512/224 составляет 2 128-1 бит.

Алгоритм

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

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 бит.

2) разбивка расширенного сообщения на блоки;

3) инициализация начальных констант;

4) обработка сообщения поблочно (основная процедура алгоритма· хеширования);

5) вывод результата

**Лаба 12:**

**Исследование алгоритмов генерации и верификации электронной цифровой подписи**

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

Функции ЭЦП:

1. аутентификация

2. контроль целостности

3. защита от подделок

4. подтверждение авторства

ЭЦП – бинарная последовательность, зависящая от содержания сообщения.

Виды ЭЦП:

1. на основе симметричных систем (с тайным ключом)

2. на основе симметричных систем и посредника

3. на основе асимметричных систем (с открытым ключом).

ЭЦП на основе RSA

1.4.1 Берем 2 простых числа p и q, например p=1301 q=1321

1.4.2 Вычисляем модуль n = p \* q = 1718621

1.4.3 Вычисляем функцию эйлера ф = (p-1)\*(q-1) = 1716000

1.4.4 Выбираем е меньше ф и взамнопростое с ф, например число 510331

{ e, n } - открытый ключ

{ d, n } - закрытый

1.4.5 Находим d, обратное е по модулю ф → d = 3571

1.4.6 Возводим сообщение в степень d по модулю n

(986770) ^ 3571 mod 1718621 = 1242146

1.4.7 На стороне клиента с помощью { e, n } находим прообраз

(1242146) ^ 510331 mod 1718621 = 986770

ЭЦП Эль-Гамаль:

1.5.1 Выбираем простое число p – например p=23

1.5.2 Выбираем произвольное g, являющееся первообразным корнем по модулю p – g=5;

1.5.3 Выбираем случайное x, 1 < x < p; => x=3

1.5.4 Вычисляем y, y = g^x mod p = 10

1.5.5 Выбираем случайное число k, например k=7

1.5.6 Вычисляем a, a = (g^k) mod p = 17

1.5.7 H(M0) = (xa + kb) mod (p – 1)

1.5.8 b = (k^-1)(H(M0) – x \* a) = 19\*(986770 – 3\*17) mod 22 = 9

Получилась подпись { 17; 9 }

1.5.9 Верификация:

1.5.9.1 (10^17 \* 17^9) mod 23 = 4

1.5.9.2 5 ^ 986770 mod 23 = 4

1.5.10 4 = 4 => подпись верифицирована.

ЭЦП на основе DSA

В алгоритме используются следующие параметры: p – простое число длиной от 64 до 1024 бит (число должно быть кратно 64); q – 160-битный простой множитель р-1.

ЭЦП Шнорр

1.6 В алгоритме используются следующие открытые параметры: p – простое число в диапазоне от 512 до 1024 бит; q –160-битное простое число, делитель p-1;

1.6.1 Пускай p=1811

1.6.2 q=181

1.6.3 gq = 1 mod 1811;

g = 1219

1.6.4 Пусть x = 20

1.6.5 Для возведения g-x надо найти обратное число к g по модулю 1811;

g-1 = 621;

1.6.6 y = 621-20 mod 1811 = 857;

1.6.7 Пусть k = 11, тогда

a = 62111 mod 1811= 1755;

Конкатенация – 9867701755

1.6.8 b = (11 + 20 \* 9867701755) mod 181 = 26 Подпись S={9867701755, 26}

1.6.9 Проверяем подпись:

X = 62126 \* 8579867701755 mod 1811 = 1534

**Лаба 13:**

**Исследование криптографических алгоритмов на основе 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, причем 4a3 +27b2 ≠ 0.

Формула называется уравнением Вейерштрасса, а условие исключает из рассмотрения кривые с особыми точками или особые кривые.

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

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

1. единичный элемент – это бесконечно удалённая точка О;

2. обратная величина точки R – это точка, симметричная относительно оси Х;

3. сложение задается следующим правилом: сумма трех ненулевых точек P, Q и -R, лежащих на одной прямой, будет равна P + Q + (-R) = О.

Законы сложения точек эллиптической кривой:

1. Прямая, проходящая через точки R и –R, является вертикальной прямой, которая не пересекает ЭК ни в какой третьей точке; если R = (х, – у), то R + (х,у) = О. Точка (х,у) является отрицательным значением точки R и обозначается –R. Таким образом, по определению R + (–R) = О;

2. P + Q = R: пусть P и Q – две различные точки ЭК (рис. 11.1), и Р не равно Q; если проведем через P и Q прямую, то она пересечет ЭК еще только в одной точке, называемой –R; точка –R отображается относительно оси Х в точку R, равную сумме точек P и Q: P + Q = R;

Если Р = (х1 , у1 ) и Q = (х2 , у2 ), то Р + Q = (х3 , у3 ) определяется в соответствии с правилами:

x3= λ2 – х1 – х2

у3= λ (х1–х3 ) – у1

l = ( у2 – у1)/( х2 – х1), если Р ≠ Q и l = 3(х1)2+а) / 2у1, если Р = Q.

Из этого следует, что число лямбда - угловой коэффициент секущей, проведенной через точки Р = (х1 , у1 ) и Q = (х2 , у2 ). При Р = Q секущая превращается в касательную, чем и объясняется наличие двух формул для вычисления.

Генерация ЭК

1.1 Выбор (генерация) ЭК - число l, число р, удовлетворяющее условию 22L-1 < р < 2 2L , р = 3 mod 4, 0 < a < p

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

Таким образом, задана ЭК: Ер(а, b)

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

Генерация ключевой информации.

2.1. Входными параметрами являются: р, а, b, q и G

2.2. Генерируется тайный ключ – число d, выбранное из множества {1, 2, …, q–1}.

2.3. 2.3. Вычисляется открытый ключ – точ

2.4. к открытому ключу также относятся р, а, b, q

Использование ЭК в криптографии

1. В алгоритмах согласования (передача) ключевой информации (на основе идеи Диффи-Хеллмана),

2. в алгоритмах асимметричного шифрования/дешифрования сообщений,

3. в алгоритмах генерации/верификации ЭЦП

**Лаба 14:**

**Исследование стеганографического метода на основе преобразования наименее значащих бит**

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

Абстрактно *стеганографическая система* обычно определяется, как некоторое множество отображений одного *пространства (множества возможных сообщений, М)* в другое *пространство (множество возможных стеганосообщений, S)*, и наоборот

*контейнер*, С, в котором размещается тайное сообщение, М; именно контейнер является упомянутым скрытым каналом

*тайное сообщение*, М, осаждаемое в контейнер для передачи или хранения (например, с целью доказательства или защиты авторских прав на документ-контейнер [2, 53-56]; здесь речь может идти о невидимых цифровых водяных знаках, ЦВЗ);

*ключи или ключевая информация*, 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 отождествляется с методом осаждения/извлечения тайной информации, другой – с выбором элементов (например, битов) контейнера для его модификации при осаждении тайной информации, третий (или третьи) – для предварительного (перед осаждением) преобразования тайной информации (например, на основе помехоустойчивого кодирования, сжатия или зашифрования) и т. д. [2, 57, 58];

*контейнер с осажденным сообщением* или *стеганоконтейнер*, S, который передается по открытому каналу, также являющемуся важным компонентом анализируемой системы; стеганоконтейнер будем именовать также стеганосообщением;

