## Введение

Криптография является одним из наиболее мощных средств обеспечения конфиденциальности и контроля целостности информации.

В криптографической терминологии исходное послание именуют открытым текстом (plaintext или cleartext). Чтобы скрыть смысл исходного текста применяются два типа преобразований: кодирование и шифрование.

Для кодирования (encode) используются кодировочные книги или таблицы, содержащие наборы часто используемых фраз. Каждой из этих фраз соответствует произвольно выбранное кодовое слово, которое чаще всего задается набором цифр. Для декодирования (decoding) требуется такая же книга или таблица.

Шифрование или зашифрование (encryption) представляет собой процедуру преобразования открытого текста в зашифрованное сообщение или шифротекст (ciphertext). Процесс, при котором из шифротекста извлекается открытый текст, называют расшифровкой (decryption). Обычно в процессе шифровки и дешифровки используется некий ключ и алгоритм обеспечивает, что расшифрование можно сделать, лишь зная этот ключ.

Ключ - конкретное секретное состояние некоторого параметра (параметров), обеспечивающее выбор одного преобразования из совокупности возможных для используемого метода шифрования.

Лабораторная работа №1. Криптоанализ классических шифров

*Шифры перестановки, шифры замены. Примеры шифрования и дешифрования.*

Шифр Цезаря

Шифр Цезаря относится к группе так называемых одноалфавитных шифров подстановки. При использовании шифров этой группы «каждый символ открытого текста заменяется на некоторый, фиксированный при данном ключе символ того же алфавита». Способы выбора ключей могут быть различны. В шифре Цезаря ключом служит произвольное число k, выбранное в интервале от 1 до 25. Каждая буква открытого текста заменяется буквой, стоящей на k знаков дальше нее в алфавите. К примеру, пусть ключом будет число 3. Тогда буква A английского алфавита будет заменена буквой D, буква B — буквой E и так далее.

Для наглядности зашифруем слово HABRAHABR шифром Цезаря с ключом k=7. Построим таблицу подстановок:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g |

И заменив каждую букву в тексте получим: C('HABRAHABR', 7) = 'OHIYHOHIY'.

При расшифровке каждая буква заменяется буквой, стоящей в алфавите на k знаков раньше: D('OHIYHOHIY', 7) = 'HABRAHABR'.

Криптоанализ шифра Цезаря Малое пространство ключей (всего 25 вариантов) делает простой перебор самым эффективным и простым вариантом атаки. Для вскрытия необходимо каждую букву шифртекста заменить буквой, стоящей на один знак левее в алфавите. Если в результате этого не удалось получить читаемое сообщение, то необходимо повторить действие, но уже сместив буквы на два знака левее. И так далее, пока в результате не получится читаемый текст.

Шифр простой замены

Шифр простой замены относится к группе одноалфавитных шифров подстановки. Ключом шифра служит перемешанный произвольным образом алфавит. Например, ключом может быть следующая последовательность букв: XFQABOLYWJGPMRVIHUSDZKNTEC.

При шифровании каждая буква в тексте заменяется по следующему правилу. Первая буква алфавита замещается первой буквой ключа, вторая буква алфавита — второй буквой ключа и так далее. В нашем примере буква A будет заменена на X, буква B на F.

При расшифровке буква сперва ищется в ключе и затем заменяется буквой стоящей в алфавите на той же позиции.

Криптоанализ шифра простой замены

Пространство ключей шифра простой замены огромно и равно количеству перестановок используемого алфавита. Так для английского языка это число составляет 26! = 288. Разумеется наивный перебор всех возможных ключей дело безнадежное.

Выбирается случайная последовательность букв — основной ключ. Шифртекст расшифровывается с помощью основного ключа. Для получившегося текста вычисляется коэффициент, характеризующий вероятность принадлежности к естественному языку.

Основной ключ подвергается небольшим изменениям (перестановка двух произвольно выбранных букв). Производится расшифровка и вычисляется коэффициент полученного текста.

Если коэффициент выше сохраненного значения, то основной ключ заменяется на модифицированный вариант.

Шаги 2-3 повторяются пока коэффициент не станет постоянным.

Для вычисления коэффициента используется еще одна характеристика естественного языка —частота встречаемости триграмм. Чем ближе текст к английскому языку тем чаще в нем будут встречаться такие триграммы как THE, AND, ING. Суммируя частоты появления в естественном языке всех триграмм, встреченных в тексте получим коэффициент, который с большой долей вероятности определит текст, написанный на естественном языке.

Задания

Ниже два шифртекста одного и того же сообщения, зашифрованные с помощью классических шифров: Цезарь, простой замены.

Шифртекст 1.

Srobdoskdehwlf vxevwlwxwlrq flskhuv

Шифртекст 2.

KjgyVgkcVWZqdX nsWnqdqsqdji XdkcZmn

Найдите соответствующий открытый текст, вскрыв шифр Цезаря, а затем найдите ключ шифра простой замены, используя для дешифрования известный открытый текст. Обе атаки должны быть полностью описаны.

Лабораторная работа №2. Маршрутные и подстановочные шифры

*Примеры шифрования и дешифрования, вскрытие ключа.*

Шифры маршрутной перестановки используют некоторую геометрическую фигуру (плоскую или объемную). Преобразования состоят в том, что в фигуру исходный текст вписывается по ходу одного маршрута, а выписывается по другому.

Шифр табличной маршрутной перестановки основанны на таблицах. При шифровании в такую таблицу вписывают исходное сообщение по определенному маршруту, а выписывают (получают шифрограмму) - по другому. Для данного шифра маршруты вписывания и выписывания, а также размеры таблицы являются ключом.

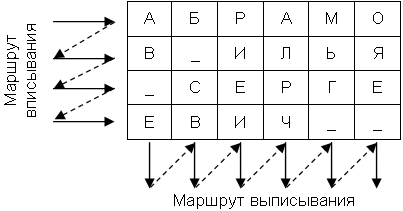


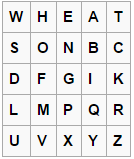
Рис.1 Пример использования шифра маршрутной перестановки

Задание.

Необходимо зашифровать свою фамилию шифром маршрутной перестановки

При дешифровании текста используют частотные характеристики открытого текста. Однако для получения устойчивой картины длина послания должна быть существенно больше ключа. Одной из наиболее устойчивых характеристик осмысленного текста является отсутствие запретных биграмм (пара соседних букв). Например, биграммы «Ъ + Ь», «гласная + Ь», «пробел + Ь». Знание и использование частотной диаграммы открытого текста значительно упростит дешифрование шифра перестановки

Шифр Плейфера

Шифр Плейфера — подстановочный шифр, реализующий замену биграмм. Для шифрования необходим ключ, представляющий собой таблицу букв размером 5\*5 (без буквы J).  
  
Процесс шифрования сводится к поиску биграммы в таблице и замене ее на пару букв, образующих с исходной биграммой прямоугольник.  
Рассмотрим, в качестве примера следующую таблицу, образующую ключ шифра Плейфера:  
  


Задание.

Открытый текст

Please note that spaces and punctuation characters have been removed before encryption

Шифротекст

LMBEUDOPUASIIYUNEDDUDOENSPARTEYOPODGCTEDUASTIDZBCBDPUCNPZBACBKMTDZDPGWZOYOPO

Найти ключ

Лабораторная работа №3. Шифрование, дешифрование информации с применением криптографических алгоритмов гаммирования

*Примеры шифрования и дешифрования.*

Гаммирование

Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел (ПСЧ) и наложении полученной гаммы на открытые данные обратимым образом, например, используя операцию "Исключающего ИЛИ" –

a ⊕ b = 1, если a ≠ b;

a ⊕ b = 0 если a = b .

Процесс дешифрования данных сводится к повторной генерации гаммы шифра при известном ключе и наложении такой гаммы на зашифрованные данные.

Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей.

Задания

1. Написать программу генерации шифра для заданных ***a*** и ***s*** по формуле:

*Ci = (a Pi + s) mod N* , где

*P* – порядковый номер символа открытого текста *(0 ≤ Pi ≤ N-1)*;

*C* – порядковый номер символа зашифрованного текста *(0 ≤ Ci ≤ N-1)*;

*N* – размер алфавита;

*a* – десятичный коэффициент;

*s* – коэффициент сдвига.

Напишите программы шифровки и расшифровки для метода моноалфавитной подстановки по заданному шифру. Язык русский. Шифр, открытый текст и зашифрованный текст должны быть в текстовых файлах с кодировкой ASCII.

Лабораторная работа №4. Исследование ассиметричных алгоритмов шифрования

*Создание ключей в системе PGP, передача подписанных и защищенных сообщений.*

PGP (Pretty Good Privacy **-** довольно хорошая секретность**)** - это криптографическая (шифровальная) программа с высокой степенью надежности, которая позволяет пользователям обмениваться информацией в электронном виде в режиме полной конфиденциальности.

В PGP применяется принцип использования двух взаимосвязанных ключей: открытого (public key) и закрытого (private key). Это очень большие числа, генерируемые случайным образом (1024 бита, 2048 бит и т.д.). К закрытому ключу имеет доступ только отправитель сообщения, а открытый ключ публикуется или распространяться через коммуникационные сети среди своих корреспондентов. При этом открытым ключом информация шифруется, закрытым расшифровывается.

Открытые ключи можно публиковать на сервере открытых ключей или распространять через коммуникационные сети среди корреспондентов. Они хранятся в компьютере в каталоге pubring.pkr в виде "сертификатов открытых ключей", которые включают в себя:

1. идентификатор пользователя владельца ключа (обычно это имя пользователя);
2. временную метку, которая указывает время генерации пары ключей;
3. собственно ключи.

Закрытые (секретные) ключи аналогично хранятся в виде «сертификатов секретных ключей» в каталоге secring.skr. При этом каждый секретный ключ шифруется отдельным паролем.

Основные функции PGP:

- генерация пары закрытый - открытый ключ;

- шифрование файла с помощью открытого ключа;

- расшифровка файла с помощью закрытого ключа;

- наложение цифровой подписи с помощью закрытого ключа;

- проверка электронной подписи с помощью открытого ключа.

Программа PGP имеет дружественный интерфейс и относительно высокую скорость шифрации-дешифрации сообщений. Ее последняя версия (PGP 8.0) русифицирована, что предопределило широкое распространение PGP среди пользователей.

Процесс шифрования сообщения с помощью PGP состоит из ряда шагов (рис. 1). Сначала программа сжимает текст. Это сокращает время на отправку сообщения через модем и увеличивает надежность шифрования.

*Примечание*: Большинство приемов криптоанализа (взлома зашифрованных сообщений) основаны на исследовании “рисунков”, присущих текстовым файлам, что помогает взломать ключ. Сжатие ликвидирует эти “рисунки”.

Для обеспечения установления подлинности сообщения его можно «подписать». Это делается добавлением к сообщению электронной (цифровой) подписи, которую получатель может проверять, используя открытый ключ отправителя для расшифровки.

Цифровая подпись - это блок данных, сгенерированный с использованием секретного ключа. Программа делает это следующим образом:

1. Из документа генерируется дайджест сообщения (это 160 или 128 – битная "выжимка" или контрольная сумма файла сообщения), к нему добавляется информация о том, кто подписывает документ и штамп времени.
2. Закрытый ключ отправителя используется для зашифровки дайджеста сообщения, таким образом, "подписывая" его.
3. Дайджест сообщения передается вместе с самим сообщением в зашифрованном виде. При идентификации подписи создается новый дайджест и сравнивается с переданным дайджестом, если они совпадают, то подпись считается подтвержденной. Если сообщение подвергнется какому-либо изменению, ему будет соответствовать другой дайджест, т.е. будет обнаружено, что сообщение было изменено.

Распознавание электронной подписи показывает, что отправителем был действительно создатель сообщения, и что сообщение впоследствии не изменялось.

Следующим шагом является генерирование так называемого сессионного (временного) ключа, который представляет собой случайное число значительно меньшего размера, чем открытый и закрытый ключ (128 бит, 168 бит), что обеспечивает высокое быстродействие шифрации-дешифрации. Временный ключ генерируется автоматически с использованием строго случайных событий, в качестве источника которых используются параметры нажатий клавиш и движений мыши.

Данным сессионным ключом шифруется сообщение, а сессионный ключ зашифровывается с помощью публичного ключа получателя сообщения и отправляется к получателю вместе с зашифрованным текстом (рис. 1.).

Открытый ключ получателя

Асимметричное

шифрование

100110101

Сессионный

ключ

Шифрованный сессионный ключ

Шифрованное сообщение

Сообщение

Сжатие

Симметричное

шифрование

# Рис.1.. Процесс шифрования сообщения

Расшифровка происходит в обратной последовательности. Программа PGP получателя сообщения использует закрытый ключ получателя для извлечения временного сессионного ключа, с помощью которого программа затем дешифрует зашифрованный текст (рис.2).

Закрытый ключ получателя

Сессионный

ключ

100110101

Сообщение

Восстано-

вление

Шифрованный сессионный ключ

Шифрованное сообщение

Асимметричное

дешифрование

Симметричное

дешифрование

# Рис.2. Процесс дешифрования сообщения

Задание.

* + 1. Изучите теоретическую часть.
    2. Переведите число  в двоичную систему счисления.
    3. Пусть каждая из 16 первых букв русского алфавита (абвгдежзийклмноп) имеет четырехразрядный двоичный код, соответствующий ее номеру от 0 до 15, т.е. а ‑ , б ‑ , …, п ‑ . Составьте из этих букв произвольное сообщение состоящее из 32 символов, затем разбейте полученное сообщение на блоки длиной 64 бита. Значения полученных блоков запишите в десятичной системе счисления.
    4. Найдите состояние 28‑разрядного двоичного регистра сдвига после циклического сдвига влево на 5, числа , предварительно записанного в регистр.
    5. Найдите сумму по модулю 2 двух чисел  и .

Лабораторная работа №5. Исследование методов идентификация и аутентификация пользователя. Протоколы «рукопожатия» и идентификации типа «запрос-ответ».

*Парольная идентификация и аутентификация, разграничение доступа к ресурсам*

Идентификация – это присвоение пользователям идентификаторов и проверка предъявляемых идентификаторов по списку присвоенных.

Аутентификация – это проверка принадлежности пользователю предъявленного им идентификатора. Часто аутентификацию также называют подтверждением или проверкой подлинности.

Парольные системы идентификации/аутентификации является наиболее распространенными методами пользовательской аутентификации. В данном случае, информацией, аутентифицирующей пользователя, является некоторый секретный пароль, известный только легальному пользователю. Методы парольной аутентификации пользователя являются наиболее простыми методами аутентификации и при несоблюдении определенных требований к выбору пароля являются достаточно уязвимыми. Цель злоумышленника – подобрать аутентифицирующую информацию (пароль) легального пользователя.

Оценка стойкости парольной защиты осуществляется следующим образом.

Пусть A – мощность алфавита паролей (количество символов, которые могут быть использованы при составлении пароля. Например, если пароль состоит только из малых английских букв, то A=26).

L – длина пароля.

 - число всевозможных паролей длины L, которые можно составить из символов алфавита A.

V – скорость перебора паролей злоумышленником.

T – максимальный срок действия пароля.

Тогда, вероятность P подбора пароля злоумышленником в течении срока его действия V определяется по следующей формуле.



Примеры.

Пример 1.

Определить время перебора всех паролей, состоящих из 6 цифр.

Алфавит составляют цифры n=10.

Длина пароля 6 символов k=6.

Таким образом, получаем количество вариантов: С=nk=106

Примем скорость перебора s=10 паролей в секунду. Получаем время перебора всех паролей t= C/s=105секунд1667минут28часов1,2 дня.

Примем, что после каждого из m=3 неправильно введенных паролей идет пауза в v=5 секунд. Получаем время перебора всех паролей T=t\*5/3=16667секунд2778минут46часов1,9 дня.

Титог = t+T = 1,2 + 1,9 = 3,1 дня

2. Пример 2.

Определить минимальную длину пароля, алфавит которого состоит из 10 символов, время перебора которого было не меньше 10 лет.

Алфавит составляют символы n=10.

Длина пароля рассчитывается: k=logn C= lg C.

Определим количество вариантов C= t \* s=10лет\*10 паролей в сек. = 10\*10\*365\*24\*60\*603,15\* 109 вариантов

Таким образом, получаем длину пароля: k=lg (3,15\*109) = 9,5

Очевидно, что длина пароля должна быть не менее 10 символов.

Задание

1. Определить время перебора всех паролей с параметрами.

Алфавит состоит из n символов.

Длина пароля символов k.

Скорость перебора s паролей в секунду.

После каждого из m неправильно введенных паролей идет пауза в v секунд

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| вариант | n | k | s | t | v |
| 1 | 33 | 10 | 100 | 0 | 0 |
| 2 | 26 | 12 | 13 | 3 | 2 |
| 3 | 52 | 6 | 30 | 5 | 10 |
| 4 | 66 | 7 | 20 | 10 | 3 |
| 5 | 59 | 5 | 200 | 0 | 0 |
| 6 | 118 | 9 | 50 | 7 | 12 |
| 7 | 128 | 10 | 500 | 0 | 0 |
| 8 | 150 | 3 | 200 | 5 | 3 |
| 9 | 250 | 8 | 600 | 7 | 3 |
| 10 | 500 | 5 | 1000 | 10 | 10 |

2. Определить минимальную длину пароля, алфавит которого состоит из n символов, время перебора которого было не меньше t лет.

Скорость перебора s паролей в секунду.

|  |  |  |  |
| --- | --- | --- | --- |
| вариант | n | m | s |
| 1 | 33 | 100 | 100 |
| 2 | 26 | 120 | 13 |
| 3 | 52 | 60 | 30 |
| 4 | 66 | 70 | 20 |
| 5 | 59 | 50 | 200 |
| 6 | 118 | 90 | 50 |
| 7 | 128 | 100 | 500 |
| 8 | 150 | 30 | 200 |
| 9 | 250 | 80 | 600 |
| 10 | 500 | 50 | 1000 |

Задание

1. Загрузите файловый менеджер Total Commander.

2. Создайте текстовый файл, содержащий Вашу фамилию (команда

Shift+F4) и сохраните его под именем Пароль1.txt.

3. Добавьте созданный файл в архив (команда Alt+F5) и установите

флажок Шифровать (только ZIP).

4. В качестве пароля выберите любую комбинацию из 4 цифр и под-

твердите пароль.

5. Загрузите программу Advanced ZIP Password Recovery (адрес

уточните у преподавателя).

6. Укажите тип атаки Перебор, выберите набор Все цифры, длину

пароля – от 1 до 4 символов.

7. Запустите перебор (команда Старт) и сохраните отчет (зафикси-

руйте общее число паролей и время поиска (рис. 3.2).

8. Выберите набор все печатаемые и выполните перебор. Сохрани-

те отчет.

9. Выберите перебор по маске, укажите маску ???? и выполните пе-

ребор. Сохраните отчет.

10. Выберите перебор по словарю и выполните перебор. Сохраните

отчет.

Лабораторная работа №6. Исследование методов идентификация и аутентификация пользователя. Протоколы «рукопожатия» и идентификации типа «запрос-ответ».

*Конфигурирование почтового клиента*

Шифрование сообщений в программе Outlook Express

Сообщение – это данные определенного формата, предназначенные для передачи по коммуникационным сетям. Оно создается в специальных почтовых программах и может включать в себя не только текст, но и файлы различных типов, в том числе графические и звуковые.

Для создания и пересылки зашифрованных сообщений можно использовать различные почтовые программы, которые поддерживаются программой PGP: Outlook Express, The Bat!, Exchange, Eudoraи др. Проще всего это делается в русифицированной программе Outlook Express**.**

Шифрованиеи подпись сообщений в этой программе осуществляется следующим образом**.**

1. Запустить программу Outlook Expressчерез главное меню путем выбора пунктаПрограммы/Outlook Express.

Задание

Создайте сообщение, содержащее приветствие своему лучшему другу (в качестве адреса электронной почты отправителя взять адрес компьютера, а в качестве электронного адреса получателя взять <фамилия>@mail.ru).

1. Выполнить команду Файл/Создать/Сообщениеили нажать кнопку Создать сообщение на панели инструментов. Окно развернуть на весь экран.
2. В появившемся окне Создать сообщение(рис.1) написать текст сообщения на рабочем поле. В строке Кому **-**  написать электронный адрес корреспондента, а в строке Тема **-** указать тему сообщения (можно не указывать). К сообщению можно присоединить любой файл, выполнив команду Вставка/Вложение файла, указав в окне обзора файл, который следует присоединить и нажать Вложить.

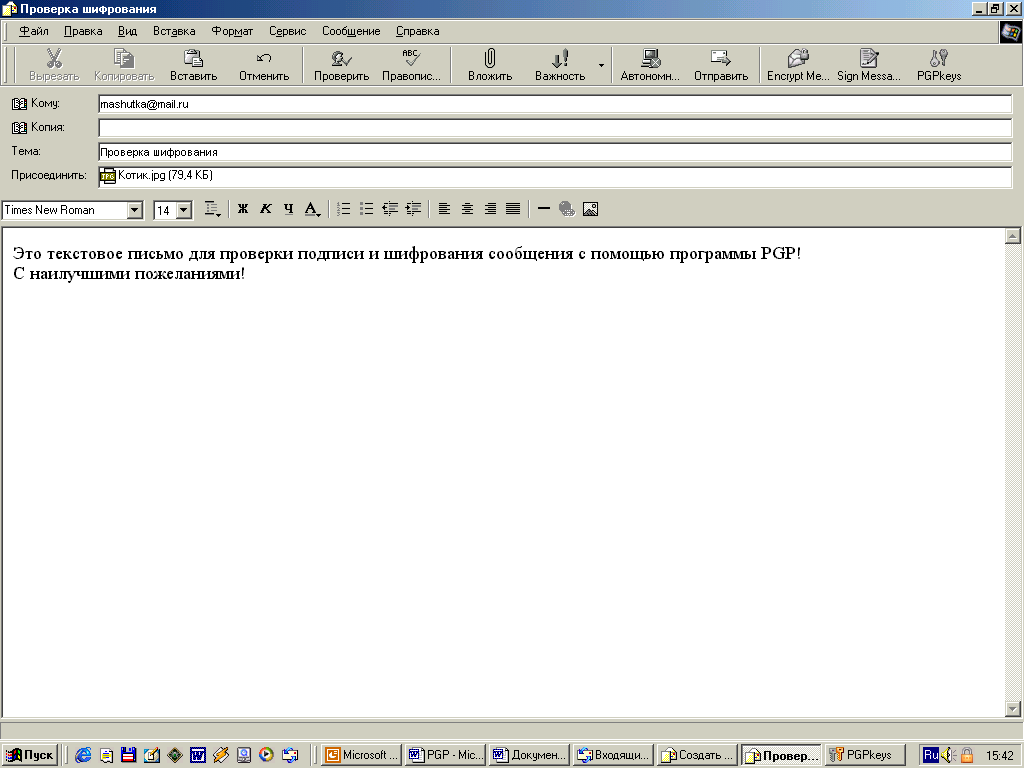


Рис. 1. Окно программы Outlook Express с созданным сообщением.

1. Созданное сообщение можно подписать, нажав на панели инструментов на кнопку *Sign Message (PGP)* или зашифровать, нажав кнопку *Encrypt Message (PGP)* (при этом кнопка просто вдавливается и больше ничего не происходит). Можно нажать сразу две кнопки для одновременной подписи и шифрования.

Для шифрования и подписи сообщения в меню **PGP**, раскрывающемся с Панели задач (знак – амбарный замок) выполнить команду *Current Windows/Encrypt & Sign*

Задание

Подпишите и зашифруйте созданное сообщение, следуя дальнейшим инструкциям.

1. Выполнить команду Файл/Отправитьили нажать кнопкуОтправить на панели инструментов**.**

7) В появившемся окне Recipient Selectionнеобходимо найти и выделить мышкой публичный ключ своего корреспондента (получателя сообщения, который обычно именуется именем получателя) и нажать на *OK*.

8) В появившемся окне PGP Enter Passphrase for Selection Keyвыделить свою ключевую пару для подписи сообщения и нажать OK.

1. Сразу же после этого программа автоматически зашифрует сообщение и поместит его в папку исходящих сообщений ***outbox.***

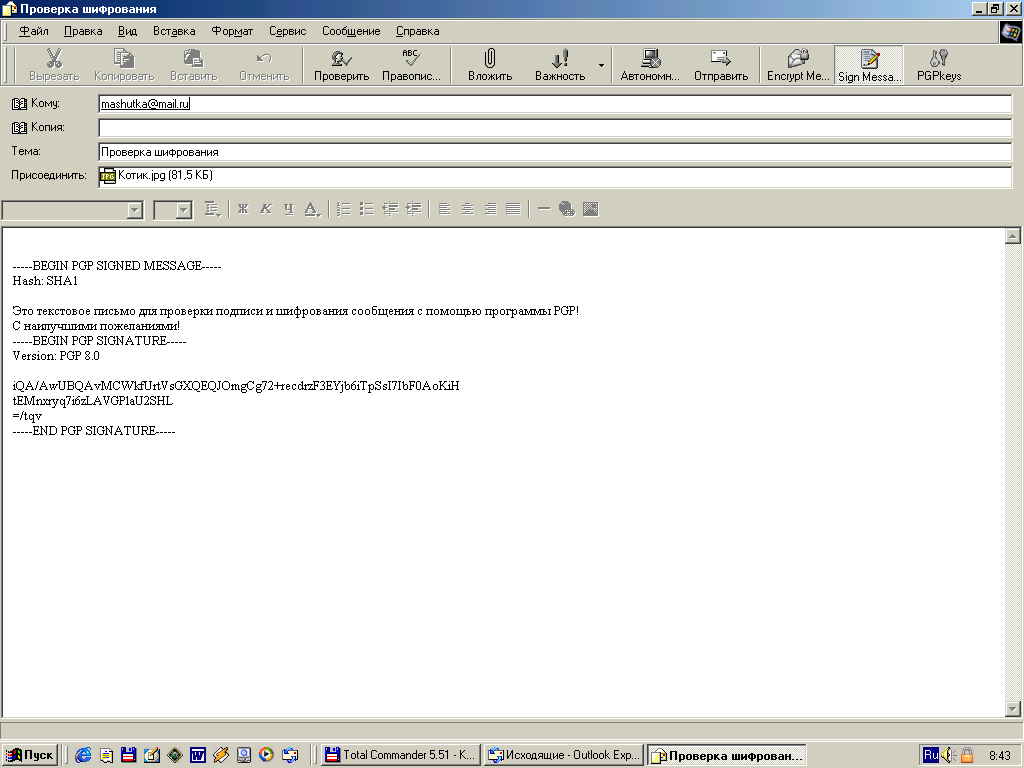


Рис. 2. Окно программы Outlook Expressс подписанным сообщением

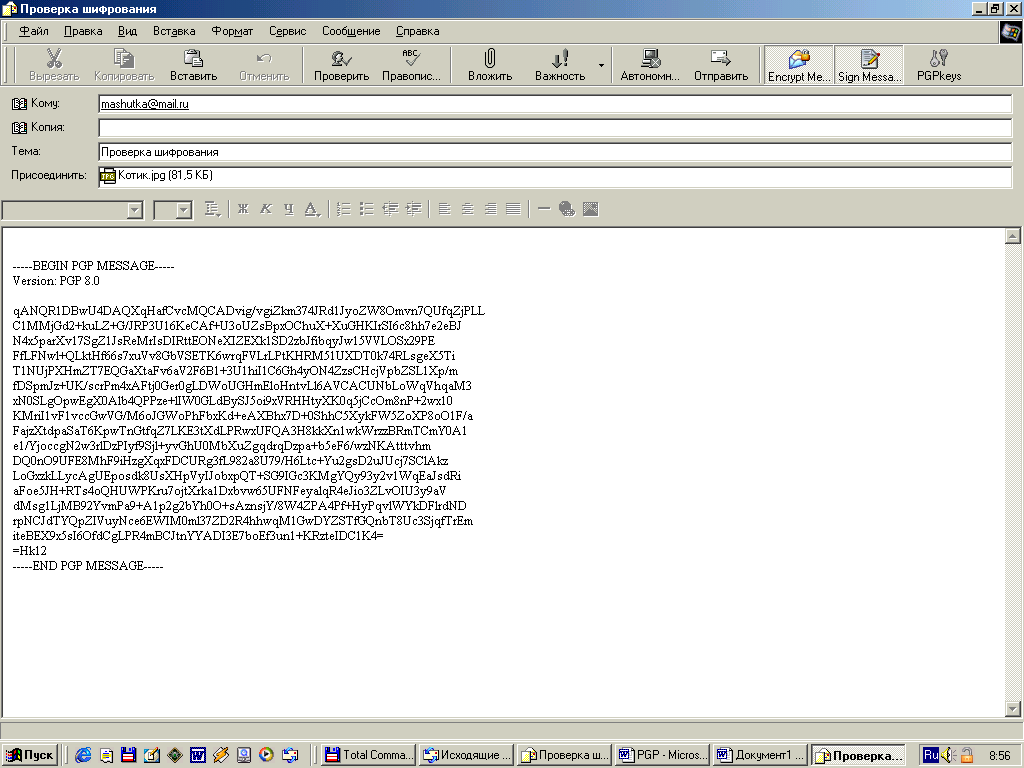


Рис. 3. Окно программы Outlook Expressс зашифрованным сообщением

Закройте программу Outlook Express.

Лабораторная работа №7. Сертификаты инфраструктуры открытых ключей и их структура. Функции удостоверяющего центра.

*Установка цифрового сертификата для почтового клиента*

На сегодняшний день проблема, затрудняющая использование ЭЦП, заключается в необходимости проверки подлинности открытых ключей. Прежде чем выполнять шифрование открытого текста или проверку подписи, абонент-отправитель/получатель должен сначала убедиться в подлинности открытого ключа, который он для этого использует. Уникальность и функциональная связанность ключевой пары гарантирует однозначность и обратимость криптографического преобразования, но не позволяет достоверно установить, кто именно воспользовался открытым/секретным ключом для шифрования или вычисления электронной цифровой подписи. Решение обозначенной проблемы заключается в организации специальных доверенных центров, отвечающих за аутентификацию открытых ключей пользователей. Аутентификация обеспечивается путем заверения открытого ключа и уникального идентификатора владельца этого ключа электронной цифровой подписью Удостоверяющего Центра. Совокупность владельцев (подписчиков) сертификатов, пользователей сертификатов, Удостоверяющих Центров и других вспомогательных структур образует инфраструктуру открытых ключей (Public Key Infrastructure - PKI).

Упрощенно, PKI представляет собой систему, основным компонентом которой является удостоверяющий центр и пользователи, взаимодействующие между собой используя сертификаты, выданные этим удостоверяющим центром. Деятельность инфраструктуры управления открытыми ключами осуществляется на основе регламента системы. Инфраструктура открытых ключей основывается на использовании принципов криптографической системы с открытым ключом. Инфраструктура управления открытыми ключами состоит из центра сертификации (удостоверяющего центра — УЦ), конечных пользователей и опциональных компонентов: центра регистрации и сетевого справочника.

Основные функции удостоверяющего центра:

1. проверка личности будущих пользователей сертификатов;
2. выдача пользователям сертификатов;
3. аннулирование сертификатов;
4. ведение и публикация списков отозванных сертификатов (Certificate Revocation List/CRL), которые используются клиентами инфраструктуры открытого ключа, когда они решают вопрос о доверии сертификату.
5. Дополнительные функции удостоверяющего центра:
6. УЦ может производить генерацию пар ключей, один из которых будет включен в сертификат.
7. По запросу, при разрешении конфликтов, УЦ может производить проверку подлинности электронной подписи владельца сертификата, выданного этим УЦ.

Сертификат – это электронный документ, который содержит электронный ключ пользователя (открытый ключ), информацию о пользователе, которому принадлежит сертификат, электронную подпись центра выдачи сертификатов (УЦ), информацию о сроке действия сертификата и другие атрибуты. Сертификат не может быть бессрочным, он всегда содержит дату и время начала и окончания своего действия.

Белорусские стандарты регламентирующие использование электронной цифровой подписи, официальное название которых «Процедура выработки и проверки ЭЦП» и «Функция хэширования», были разработаны группой белорусских специалистов в 1999 г. и официально приняты в 2000 г.

В данной лабораторной работе реализован стандарт «Процедура выработки и проверки ЭЦП», а стандарт «Функция хэширования» реализован в упрощенном виде.

В этих стандартах наряду с элементами классических процедур ЭЦП используются современные идеи, позволяющие увеличить криптостойкость и быстродействие. Так, открытый ключ и секретный ключ связаны известным соотношением

,

которое позволяет легко вычислить  по , но очень сложно решение обратной задачи ‑ вычисления  по . К подписываемому сообщению добавляется случайная компонента , что усложняет возможный подбор хэш‑значения злоумышленником по известному тексту сообщения.

Обозначения принятые в стандарте СТБ‑1176.02‑99

*  ‑ множество, состоящее из чисел ;
*  ‑ присвоение параметру  значения ;
*  ‑ остаток от деления  на , где  ‑ натуральное число или ноль,  ‑ натуральное число;
*  ‑ натуральное число  такое, что  и , где  и  ‑ взаимно простые числа;
*  ‑ наименьшее целое число, не меньшее чем ;
*  ‑ наибольшее целое число, не большее чем ;
*  ‑ разложение неотрицательного целого числа  по основанию , где  и  ‑ натуральные числа,
*  ‑ целое число, ;
*  ‑ бинарная операция, определенная на множестве неотрицательных целых чисел по формуле , где , , ;
*  ‑ операция  определяется для любых  и  по формуле ;
* ‑ степень числа на основе операции , определяется индуктивно по формуле , где  ‑ натуральное число;
*  ‑ функция хэширования, процедура вычисления значений которой соответствует СТБ.

Процедура выработки ЭЦП

1. Выбираются параметры  и , которые определяют уровень криптографической стойкости ЭЦП. Число  является длиной записи числа  в системе счисления по основанию 2,  является длиной записи числа  в системе счисления по основанию 2.
2. В соответствии с выбранными  и  генерируются простые числа  и  такие, что  делит  нацело.
3. Генерируется случайное число , .
4. Вычисляется . Если , то перейти к пункту 3.
5. Генерируется случайное число , , которое является секретным ключом.
6. Вычисляется число , которое является открытым ключом.
7. Генерируется случайное число , .
8. Вычисляется . Далее число  разлагается по основанию , т.е. . Таким образом, получаются коэффициенты .
9. Формируется последовательность , состоящая из коэффициентов  и блоков открытого текста .
10. Вычисляется значение хэш‑функции . Если , то перейти к пункту 6.
11. Вычисляется . Если , то перейти к пункту 6.
12. Вычисляется . ЭЦП последовательности  есть число .
13. Отправляется , .

Процедура проверки ЭЦП

1. Вычисляется .
2. Вычисляется .
3. Если хотя бы одно из условий  и  не выполнено, то ЭЦП считается недействительной и работа алгоритма завершается.
4. Вычисляется .
5. Число  разлагается по основанию , т.е. . Таким образом, получаются коэффициенты .
6. Формируется последовательность , состоящая из коэффициентов  и блоков открытого текста .
7. Вычисляется хэш‑функция .
8. Проверяется условие . При совпадении  и  принимается решение о том, что ЭЦП была создана при помощи личного ключа подписи , связанного с открытым ключом проверки подписи , а так же ЭЦП и последовательность  не были изменены с момента их создания. В противном случае подпись считается недействительной.

Стандарт «Процедура выработки и проверки ЭЦП» содержит алгоритмы и процедуры выработки и проверки электронной цифровой подписи, а также подробные инструкции по:

* выбору величин  и  (размер  и );
* генерации  и ;
* генерации .

Задание

1. Изучите теоретическую часть.
2. Определите число , где ,  и , .
3. Сколько существует различных пар простых чисел  и , находящихся в диапазоне  и таких, что  делит  нацело?
4. Определите с помощью расширенного алгоритма Евклида число , обратное числу  (т.е. ) по модулю .
5. Разложите число  по основанию 10, т.е. необходимо найти коэффициенты  ряда .
6. Разложите число  по основанию , т.е. необходимо найти коэффициенты  ряда .

Задание

Получение цифрового сертификата

1. Заведите почтовый ящик на сервере TUT.BY или любом другом

бесплатном сервере (или используйте уже существующий ящик).

2. Перейдите по адресу – http://www.thawte.com/.

3. Перейдите по ссылке free personal e-mail certificates в меню

products.

4. Нажмите кнопку join и заполните последовательно ряд анкет, нажимая в конце каждой страницы кнопку NEXT.

5. После заполнения всех анкет Вам будет выслано тестовое сообщение (Mail Ping), содержащее коды для продолжения регистрации (Probe и Ping) и адрес для дальнейшей регистрации.

6. Перейдите по указанному адресу и введите коды.

7. На следующей странице оставьте неотмеченным флажок I enrolled because an ESO requested I obtain Name Validation or Strong Extranet Certificates и нажмите next.

8. В появившемся окне введите Ваше имя (совпадает с адресом почты; например, alexged@mail.by) и пароль.

9. Выберите тип сертификата X.509 Format Certificates (нажмите кнопку Request).

10. Выберите сертификат Microsoft Internet Explorer, Outlook and Outlook Express и нажмите request.

11. Выберите No Employment Information Available и нажмите next.

12. Установите флажок напротив Вашего адреса и нажмите next.

13. В одном из последующих окон анкеты выберите опцию Advanced Users: Configure Certificate Extensions (нажмите кнопку Configure).

14. Установите флажки: Digital Signature, Data-Encipherment, Key-Encipherment, Key Agreement и нажмите accept.

15. Выберите в качестве криптографического провайдера Microsoft Enhanced Cryptographic Provider v1.0.

16. В появившемся окне нажмите Yes, затем ОК.

17. В окне браузера нажмите кнопку Finish.

Установка цифрового сертификата

1. Установка сертификата проводится на Вашем личном или рабочем компьютере.

2. Перейдите на страницу менеджера сертификатов. Ссылки, последовательно, начиная с главной станицы www.thawte.com: login –Personal E-Mail Certificates – Certificates – View certificate status– MSIE.

3. Для установки сертификата нажмите кнопку fetch, а затем – Install your cert.

Конфигурирование почтового клиента

1. В почтовом клиенте MS Outlook Express добавьте учетную запись Вашего почтового ящика (команда Tools – Accounts… – Mail –

Add – Mail) и сконфигурируйте ее.

 Для почтовых ящиков сервера TUT.BY адреса серверов входящей и исходящей почты совпадают – mail.tut.by.

 В качестве номера порта исходящей почты необходимо установить 2525.

2. После конфигурирования Вашей учетной записи проверьте ее работоспособность: отправьте и получите почту.

3. В свойствах учетной записи на закладке Security выберите в качестве сертификата для цифровой подписи и шифрования полученный Вами сертификат от THAWTE.COM.

4. В качестве алгоритма шифрования установите 3DES.

Лабораторная работа №8. Примеры прикладных протоколов (протоколы заключения сделок, платежных систем, сертифицированная электронная почта, голосования и др.).

*Распределение ключа при наличии доверенного центра. Распределение секретного ключа без участия доверенного центра*

Защищённым каналом можно назвать любой канал, при котором собеседники общаются с использованием шифрования (шифруя все данные одним и тем же ключом). Только они могут пересылать эти данные по каналу и считывать, потому что только им известен секретный ключ. Собеседникам необходимо выработать общий ключ. Предположим, что у собеседников есть ключи друг друга (рассматривается ситуация, когда собеседники используют не один ключ). Они могут использовать любой из ключей, либо сгенерировать третий ключ, основываясь на своих закрытых ключах. Им нужно понять, какой ключ использовать. Это нетрудно, пока получателей всего двое. Проблема возникает, если получателей становится больше.

Если количество абонентов порядка ста, то количество парных ключей будет равно:

Для ста абонентов нужно пять тысяч ключей. Для десяти тысяч абонентов нужно < 5-106 ключей. То есть количество ключей в системе намного превышает количество абонентов. Причём в надёжной защищённой системе нужно регулярно обновлять ключи. Даже если заниматься обновлением ключей раз в 3 года (что достаточно редко), то будет затрачиваться слишком много усилий: каждый ключ необходимо надёжно сгенерировать, отправить получателям, убедиться, что они их получили, не будучи перехваченными.

Нужно организовать такую систему, чтобы в большом количестве ключей не было необходимости. Для этого можно использовать так называемый доверенный центр Трент. Именно в этом доверенном центре хранятся все ключи всех абонентов. Этот доверенный центр управляется самой организацией или главным системным администратором. У него содержится информация, каким ключом нужно шифровать данные, чтобы передать их собеседникам. У каждого из абонентов тоже есть соответствующий парный ключ. То есть, если в системе *n* абонентов, у Трента есть *n* ключей, отвечающих за связи с этими абонентами, причём эти ключи взаимные. Ключ конкретного собеседника одновременно известен и Тренту, и самому собеседнику. Когда они будут обмениваться данными, Трент будет шифровать данные этим ключом, и собеседник тоже будет шифровать данные этим же ключом, общим ключом его и Трента.

Протокол Kerberos был разработан в Массачусетском технологическом институте в середине 1980-х годов и сейчас является фактическим стандартом системы централизованной аутентификации и распределения ключей симметричного шифрования. Поддерживается операционными системами семейства Unix, Windows (начиная с Windows'2000), есть реализации для Mac OS.

Централизованное распределение ключей симметричного шифрования подразумевает, что у каждого абонента сети есть только один основной ключ, который используется для взаимодействия с центром распределения ключей (сервером ключей). Чтобы получить ключ шифрования для защиты обмена данными с другим абонентом, пользователь обращается к серверу ключей, который назначает этому пользователю и соответствующему абоненту сеансовый симметричный ключ.

Протокол Kerberos обеспечивает распределение ключей симметричного шифрования и проверку подлинности пользователей, работающих в незащищенной сети. Реализация Kerberos - это программная система, построенная по архитектуре "клиент-сервер". Клиентская часть устанавливается на все компьютеры защищаемой сети, кроме тех, на которые устанавливаются компоненты сервера Kerberos – доверенный центр Трент. В роли клиентов Kerberos могут, в частности, выступать и сетевые серверы (файловые серверы, серверы печати и т.д.).

Серверная часть Kerberos называется центром распределения ключей ( Key Distribution Center – KDC) и состоит из двух компонент:

сервер аутентификации (Authentication Server – AS);

сервер выдачи разрешений (Ticket Granting Server – TGS).

Каждому субъекту сети сервер Kerberos назначает разделяемый с ним ключ симметричного шифрования и поддерживает базу данных субъектов и их секретных ключей. Схема функционирования протокола Kerberos представлена на [рис. 1](http://www.intuit.ru/studies/courses/531/387/lecture/8998?page=1#image.5.1).



Рис. 1. Протокол Kerberos

Клиент C собирается взаимодействует по защищенному каналу с сервером SS (Service Server – сервер, предоставляющий сетевые сервисы).

Протокол использует только симметричное шифрование и предполагает, что у каждого корреспондента (Алисы и Боба) есть общий секретный ключ с третьей доверенной стороной (Трентом).

Алиса направляет доверенной стороне (Тренту) свой идентификатор и Боба:

Alice \to \left\{ A, B \right\} \to Trent

Трент генерирует два сообщения. Первое включает метку времени T_T, время жизни ключа L, новый сеансовый ключ для Алисы и Боба K и идентификатор БобаB. Это сообщение шифруется общим ключом Алисы и Трента. Второе сообщение содержит то же самое, кроме идентификатора — он заменён на идентификатор Алисы A. Само сообщение шифруется общим ключом Трента и Боба:

Trent \to \left\{ E_A \left(T_T, L, K, B\right), E_B \left(T_T, L, K, A\right) \right\} \to Alice

Алиса генерирует сообщение из собственного идентификатора A и метки времени T_T, после чего шифрует сообщение сеансовым ключом K и посылает Бобу вместе со вторым сообщением от Трента:

Alice \to \left\{ E_K \left(A, T_T\right), E_B \left(T_T, L, K, A\right) \right\} \to Bob

В целях собственной аутентификации Боб шифрует модифицированную метку времени T_T + 1 общим сеансовым ключом K и посылает её Алисе:

Bob \to \left\{ E_K \left(T_T + 1\right) \right\} \to Alice

Важным предположением является синхронизированность часов всех участников протокола. Однако на практике используется синхронизация с точностью до нескольких минут с запоминанием истории передач (с целью обнаружения повтора) в течение некоторого времени.

В специализированной литературе зачастую сообщение EB (K, A, t) называется ticket. Каждый раз, когда условная Алиса доказывает свою личность условному Бобу (ведь число пользователей в сети может быть гораздо больше, чем 2), она полагается на Трента (доверенный центр), доверяя ему сгенерировать новый секретный ключ и безопасно вручить его обоим пользователям. Новый секретный ключ K называется сеансовым ключом (Session key), и как раз Kerberos Ticket используется, чтобы доставить его Бобу. Kerberos Ticket является сертификатом, выданным доверенным центром и зашифрованным с помощью EB — общего ключа Боба и Трента. Среди прочей информации, билет содержит случайный сеансовый ключ, который будет использоваться для аутентификации Алисы Бобом, имя участника (в данном случае Алисе), которому сеансовый ключ был выдан, и срок действия, по истечении которого сеансовый ключ больше не действителен. Данный сертификат (Kerberos Ticket) не отправляется непосредственно Бобу, но вместо этого отправляется клиенту (Алисе), который направляет его для проверки Бобу в составе запроса. Так как сертификат шифруется на сервере ключом, известным только Тренту и Бобу, то изменить сертификат (Kerberos Ticket) для клиента (Алисы) не представляется возможным.

Задание

Выполнить распределение ключей по протоколу Kerberos, используя шифр простой замены.