ВОРОНЕЖСКИЙ ИНСТИТУТ ВЫСОКИХ ТЕХНОЛОГИЙ – АНОО ВПО

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ

По дисциплине «Защита информации»

Лабораторная работа № 7

«РАБОТА СО СПЕЦИАИЗИРОВАННЫМИ ПРОГРАММАМИ ЗАЩИТЫ ДАННЫХ»

Цель работы: изучить программу Pretty Good Privacy (PGP), зашифровать и проверить правильность тестового сообщения.
Используемое программное обеспечение: PGP 2.6.3

С момента появления в 1991 году программа PGP стала стандартом де-факто для программного обеспечения персонального шифрования и шифрования в мелком бизнесе. Мощь PGP не в том, что никто не знает, как ее взломать иначе как используя "лобовую атаку" (это не сила, а условие существования хорошей программы для шифровки), а в превосходно продуманном и чрезвычайно мощном механизме обработки ключей, быстроте, удобстве и широте распространения. Существуют десятки не менее сильных алгоритмов шифровки, чем тот, который используется в PGP, но популярность и бесплатное распространение сделали PGP фактическим стандартом для электронной переписки во всем мире.

Обычные средства криптографии (с одним ключом для шифровки и дешифровки) предполагали, что стороны, вступающие в переписку, должны были в начале обменяться секретным ключом, или паролем, с использованием некоего секретного канала для того, чтобы начать обмен зашифрованными сообщениями. Получается замкнутый круг: чтобы передать секретный ключ, нужен секретный канал. Чтобы создать секретный канал, нужен ключ.

Разработанная Филипом Циммерманном программа PGP относится к классу систем с двумя ключами, публичным и секретным. Это означает, что вы можете сообщить о своем публичном ключе всему свету, при этом пользователи программы смогут отправлять вам зашифрованные сообщения, которые никто, кроме вас, расшифровать не сможет. Вы же их расшифровываете с помощью вашего второго, секретного ключа, который держится в тайне.

Публичный ключ выглядит примерно так :

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3
mQCNAzF1IgwAAAEEANOvroJEWEq6npGLZTqssS5EScVUPVaRu4ePLiDjUz6U7a
Qr
Wk45dIxg0797PFNvPcMRzQZeTxYl0ftyMHL/6ZF9wcx64jyLH40tE2DOG9yqwKAn
yUDFpgRmoL3pbxXZx9lO0uuzlkAz+xU6OwGx/EBKYOKPTTtDzSL0AQxLTyGZA
AUR
tClCb2IgU3dhbnNvbiA8cmpzd2FuQHNlYXR0bGUtd2Vid29ya3MuY29tPokAlQMF
EDF2lpI4h53aEsqJyQEB6JcD/RPxg6g7tfHFi0Qiaf5yaH0YGEVoxcdFyZXr/ITz
rgztNXRUi0qU2MDEmh2RoEcDsIfGVZHSRpkCg8iS+35sAz9c2S+q5vQxOsZJz72B
LZUFJ72fbC3fZZD9X9lMsJH+xxX9CDx92xm1IglMT25S0X2o/uBAd33KpEI6g6xv
-----END PGP PUBLIC KEY BLOCK-----
```

Вы можете опубликовать свой публичный ключ на вашей Web странице , или послать его электронной почтой своему другу. Ваш корреспондент зашифрует сообщение с использованием вашего публичного ключа и отправит его вам. Прочесть его сможете только вы с использованием секретного ключа. Даже сам отправитель не сможет расшифровать адресованное вам сообщение, хотя он сам написал его 5 минут назад. На сегодня даже самым мощным компьютерам требуются века, чтобы расшифровать сообщение, зашифрованное с помощью PGP!

Генерация ключей.

Первое, что необходимо сделать – сгенерировать собственный набор общих и секретных ключей. Эти ключи будут сохранены в ключнице (keyring), и потом их всегда можно будет там найти. Для запуска процедуры генерации ключей наберите:

PGP – kg

Переключатель –kg указывает PGP создать пару ключей. Потребуется ввести некоторые данные. Сначала PGP потребует определения желательного уровня шифрования. Имеются три уровня, пронумерованные 1,2,3. Наименее защищенный уровень использует ключ с размером 512 бит, в то время как наиболее безопасный уровень – 1024 бита. За дополнительную защиту придется расплачиваться быстродействием. Затем PGP запросит идентификатор пользователя. Этот идентификатор будет добавлен к вашему открытому ключу, т.е. в нем будет содержаться имеющая значение информация. Адрес электронной почты следует вводить в угловых <скобках>. На следующем шаге необходимо сгенерировать начальное число ключей. Для этого вы набираете на клавиатуре случайный текст, а программа измеряет паузы между нажатиями клавиш. Кроме того, PGP запросит ввести пользовательский пароль. Этот пароль требуется для работы с частной стороной ключей и работы PGP. Убедитесь, что пароль выбран достаточно сложный, но вы в состоянии его запомнить. Если он станет кому-либо известен, то злоумышленник сможет воспользоваться вашими секретными ключами, чтобы расшифровать сообщения, посланный вам конфиденциально. Создание ключа потребует некоторого времени. Как только ключи сгенерированы, они помещаются в ключницу пользователя. По умолчанию создаются две ключницы: PUBRING.PGP и SECRING.PGP. Можно создавать другие ключницы, генерируя новые наборы ключей, используя различные имена.

Чтобы подготовить блок ASCII-текста, который вы будете использовать для распространения своего открытого ключа, требуется набрать команду вида

PGP – kxa John Smith

В результате действия этой команды на диске создается файл, содержащий открытый ключ Джона Смита. –kxa означает «извлечь ключ ASCII». Эта команда должна создать файл с именем SMITH.ASC, который содержит открытый ключ в форме, которую можно добавлять в сообщениям электронной почты.

Всякий раз, когда кто-либо присылает вам свой открытый ключ, вы должны сохранить ключ в файле. После этого следует ввести команду, добавляющую этот ключ к вашей ключнице, например:

PGP – ka paul.pgp

PGP сообщит вам о том, что открытй ключ некоторого человека был добавлен к ключнице открытых ключей.

Следует обратить внимание на то, что операция заканчивается запросом о том, хотите ли вы заверить ключ. Под понятием «заверить» подразумевается «убедиться в том, что ключ действительно принадлежит человеку, с которым предстоит связываться».

Заверение ключей.

Если вы хотите заверить ключ, следует безопасным образом войти в контакт с его владельцем. Попросите его ввести команду *PGP -kvc* для своей ключницы и сравните «отпечаток пальца», выведенный у него, с отпечатком, выведенным на вашем дисплее. Отпечаток пальца получается при пользовании хэш-функции для создания резюме сообщения. При этом защита заключается в уверенности, что вы говорите с владельцем ключа. Обсуждение отпечатка пальца по телефону, например, не вносит ничего, что ставило бы под угрозу целостность шифрования, если только вы не получаете информацию от самозванца.

Если вы не можете заверить ключ в тот момент, когда его добавляете, то можно сделать это позже. Чтобы заверить ключ, уже находящийся в ключнице, напечатайте команду

PGP – ks paul

Т.е. вы просите PGP заверить ключ Пола в заданной по умолчанию ключнице.

Шифрование сообщений.

Чтобы зашифровать сообщение, поместите текст в файл и воспользуйтесь переключателем –e. Напечатайте

PGP –e Forjoe.txt

PGP попросит ввести идентификатор человека, для которого предназначено сообщение, затем шифрует сообщение, записывает на диск и сообщает имя созданного файла. Для того, чтобы в конечном файле содержались только печатаемые символы можно добавить опцию a. Т.е.

PGP –ea Forjoe.txt

Расшифровка сообщений.

Получив сообщение, зашифрованное с использованием вашего открытого ключа, следует сохранить его на диске и воспользоваться командой вида

PGP Forjoe.asc

Подписи.

Если вы хотите подписать сообщение, а не шифровать его, используйте переключатель –s.

PGP –sa sometext.txt

Результатом действия команды станет создание файла, содержащего, на первый взгляд, информационный мусор. Но любой человек, у которого есть ваш открытый ключ, сможет прочитать сообщение, пользуясь командой

PGP sometext.txt

Также можно подписать и шифрованное сообщение

PGP –sea sometext.txt

PGP запросит открытый ключ получателя и затем выведет зашифрованный текст. Подписание такого сообщения обеспечит уверенность в том, что сообщение не пытались исказить или взломать.

Рабочее задание

1.1. Выработать два ключа для схемы RSA на программе PGP (можно использовать ключи, выработанные во время выполнения лабораторной работы 5). Один ключ для абонента A и один для абонента B.

1.2. Организовать модель сети засекреченной связи между пунктами A и B.

1.3. Создать электронную подпись файла. Изменить подписанный файл и попробовать проверить целостность электронной подписи.

1.4. Выработать новый ключ и передать его из пункта A в пункт B по протоколу, обеспечивающему одновременно конфидециальность и аутентификацию отправителя.

1.5. Подготовить отчет в файле Result_6. Создать электронную подпись для файла с отчетом на выработанном в п.1.3. ключе.

Выполнение задания

2.1. Задайте на своем компьютере двух пользователей A и B. Выработайте ключи для пункта A. Для этого необходимо стартовать программу

```
PGP.EXE -kg
```

На запрос о длине ключа в битах ввести 384.

На запрос о идентификаторе ключа (user ID) введите idA.

На запрос о пароле (pass phrase) введите свою фамилию только заглавными буквами, например, IVANOV_A.

Далее потребуется ввести случайный набор символов с клавиатуры до звукового сигнала, после чего начинается процесс выработки ключа.

Аналогично выработайте ключ для пункта B.

2.2. Если пункты A и B созданы на разных компьютерах, то потребуется обменяться файлами с открытыми ключами. Для того, чтобы перенести открытый ключ из пункта A в пункт B, скопируйте PUBRING.PGP в файл PUBRING.A на другом компьютере и добавте его к списку открытых ключей пункта B командой

```
PGP  -ka PUBRING.A
```

2.3. Подготовьте файл M.txt с посланием из пункта A в пункт B и зашифруйте его:

```
PGP -e M.TXT idB
```

Программа создаст шифрованный файл M.PGP

Скопируйте его в пункт B и расшифруйте:

```
PGP M.PGP -u idB
```

2.4. Создайте цифровую подпись файла M.TXT, т.е.

```
PGP -s M.TXT
```

Попробуйте проверить подпись

```
PGP M.TXT
```

Измените файл и снова проверьте целостность подписи.

2.5. Создайте на пункте A новый ключ с идентификатором newA. Выделите вновь созданный ключ в отдельный файл KEYA командой

```
PGP -kx newA KEYA
```

Зашифруйте файл KEYA для пункта B, одновременно подписав его на своем секретном ключе командой

```
PGP -es KEYA idB -u idA
```

Созданный файл KEYA.PGP передайте на пункт B.

Примечание. Если пункты A и B созданы на одном компьютере, то удалите новый ключ из списка открытых ключей командой

```
PGP -kr newA
```

2.6. В пункте B расшифруйте файл KEYA.PGP командой

```
PGP KEYA.PGP
```

Если подпись верна, добавьте новый ключ к списку открытых ключей командой

```
PGP -ka KEYA.PGP
```

Контрольные вопросы

1. Каково основное отличие асимметричной криптосхемы от криптосхемы с секретным ключом?
2. Почему в схеме шифрования с открытым распределением ключа для целей аутентификации применяется секретное преобразование?
3. Какими свойствами должна обладать хэш-функция?
4. Как средствами криптографии осуществлять контроль за достоверностью передачи информации?

Оформление отчета

Отчет должен быть представлен в виде файла с именем RESULT_6.TXT (образец прилагается).

Файл RESULT_6.TXT должен содержать электронную подпись, выполненную с помощью программы PGP на выработанном в п.2.5 ключе:

```
PGP -s RESULT_6.TXT newA
```

В качестве отчета преподавателю передаются следующие файлы:

RESULT_6.PGP - файл отчета с электронной подписью.

PUBRING.PGP - список выработанных открытых ключей.

В отчёте должны содержаться ответы на контрольные вопросы. Содержимое файлов RESULT_6.PGP и PUBRING.PGP должно быть отпечатано на бумажном носителе.