ВОЛЖСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛГОГРАДСКИ Й ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ»

Д.Н. Лясин, О.Ф. Абрамова

Асимметричная криптография и электронная цифровая подпись на примере системы GnuPG

Методические указания

Волгоград 2016

УДК 004.056

Рецензент:

канд. тех. наук доцент В. И. Капля

Издается по решению редакционно-издательского совета
Волгоградского государственного технического университета

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ: Асимметричная криптография и электронная цифровая подпись на примере системы GnuPG. Сост. Лясин Д.Н., Абрамова О.Ф...; Волгоград. гос. техн. ун-т. - Волгоград, 2016, – 27 с.

Содержатся сведения, необходимые для изучения принципов работы асимметричных алгоритмов шифрования и алгоритмов верификации электронной подписи. Представлены обобщенные структурные схемы работы алгоритмов, рассмотрены особенности их функционирования. Дано подробное описание процесса криптографической защиты информации с использованием свободно распространяемой системы Gnu Privacy Guard, начиная с процесса генерации и распространения ключей и заканчивая непосредственно шифрованием-дешифрацией файлов и проверкой электронной подписи. Выполнение настоящей лабораторной работы поможет студентам лучше понять порядок работы с асимметричными криптоалгоритмами, назначение систем сертификации открытых ключей, поможет освоить полезный на практике инструмент криптографической защиты данных. Приведен порядок выполнения работы, варианты заданий к лабораторным работам

Предназначены для студентов, обучающихся по направлениям 09.03.01 «Информатика и вычислительная техника" и 09.03.04 "Программная инженерия» всех форм обучения в рамках курса «Защита информации», а также по направлению 15.04.04 «Автоматизация технологических процессов и производств» в рамках курса «Хранение и защита компьютерной информации».

Ил. 16. Библиогр.: 5 назв.

Лабораторная работа N3
Асимметричная криптография и электронная цифровая подпись на примере системы GnuPG

Цель работы: знакомство с принципами криптографической защиты информации с использованием алгоритмов асимметричного шифрования и электронной цифровой подписи, приобретение навыков практического применения указанных методов защиты информации на основе системы GnuPG.

1. Основные принципы асимметричной криптографии

Долгое время шифрование как способ преобразования сообщения в форму, недоступную для восприятия неавторизованным пользователям, существовало только в форме симметричной криптографии, когда и отправитель, и получатель должны знать секретный ключ, используемый для шифрации/дешифрации сообщений.

Симметричное шифрование имеет недостатки, которые ограничивают возможности его применения в ряде конкретных случаев [2]. В частности, зачастую невозможно организовать секретный канал для обмена ключами шифрования между участниками взаимодействия. Еще одним недостатком симметричных шифров является необходимость хранения большого количества ключей. Для того чтобы в вычислительной сети могли конфиденциально попарно взаимодействовать N участников, необходимо наличие в системе N*(N-1)/2 ключей. Эти недостатки можно устранить, используя алгоритмы асимметричного шифрования. Например, для асимметричной системы достаточно иметь 2*N пар открытый/закрытый ключ, чтобы можно было организовать секретный канал между каждой парой участников.

Асимметричная система шифрования работает по схеме, представленной на рис. 1. Отличительной особенностью асимметричных алгоритмов является наличие пары ключей шифрования: открытого (публичного) $k_{от}$, который передается второй стороне по незащищенному каналу связи и поэтому может быть известен криптоаналитику, а также закрытого (частного) $k_{зак}$, который известен лишь одному человеку (получателю сообщения) и держится в секрете [1]. Пара ключей обладает тем свойством, что сообщение, зашифрованное на одном из ключей, может быть расшифровано только на другом ключе. Фактически это означает, что секретным каналом передачи информации на схеме рис. 1 является направление “A-B”, поскольку сообщение, зашифрованное на открытом ключе отправителем, может дешифровать своим закрытым ключом только получатель. Если необходимо организовать двунаправленный безопасный обмен сообщениями, необходимо использовать две пары ключей и пользователь B должен шифровать информацию с использованием открытого ключа пользователя A.

Асимметричная система решает указанные выше проблемы симметричного шифрования – здесь нет необходимости передавать секретный ключ на противоположную сторону, публичный ключ можно передавать по открытому каналу связи. Единственное требование к каналу распространению открытого ключа – он

должен быть аутентичным, т.е. всякий, получивший этот ключ, должен иметь возможность убедиться в его принадлежности лицу, заявленному как владелец соответствующей ключевой пары. На практике это требование реализуется путем сертификации открытого ключа – третья сторон, которой доверяют обе стороны взаимодействия, заверяет открытый ключ и выдает на него сертификат, подписанный электронной цифровой подписью. Открытый ключ в этом случае распространяется вместе с сертификатом, что дает возможность всегда удостоверится в его принадлежности. К асимметричным относятся такие алгоритмы шифрования как RSA, El Gemal, Рабина, Месси-Омуры.

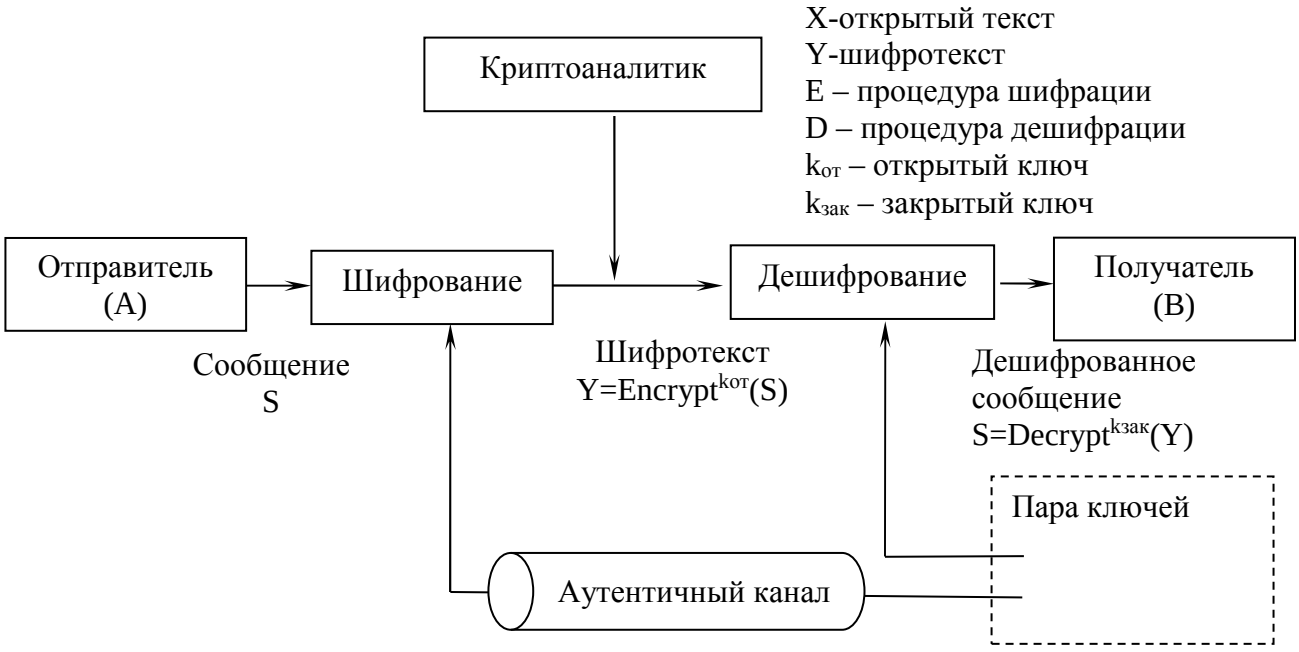


Рисунок 1. Обобщенная структурная схема асимметричной криптосистемы

Асимметричная криптография не вытесняет полностью симметричные алгоритмы с поля шифрования данных, поскольку сама обладает рядом недостатков, ключевой из которых – низкая скорость преобразования: асимметричный шифр в среднем на 3 порядка медленнее шифрует сообщение по сравнению с симметричным при аналогичной криптостойкости. В связи с этим сегодня приоритетной является гибридная схема шифрации, когда сообщение кодируется симметричным шифром с использованием сеансового ключа, а по асимметричной схеме шифруется и передается на противоположную сторону лишь сам сеансовый ключ.

Глядя на схему на рис.1 можно сделать вывод, что шифрация информации с использованием закрытого ключа не имеет смысла. Действительно, зачем кодировать информацию, если всякий, кто знает открытый ключ (читай – любой желающий) может ее декодировать. Однако, шифрование информации закрытым ключом имеет смысл. Тут необходимо вспомнить, что защита информации обеспечивает не только конфиденциальность информации, но и ее аутентичность, целостность, апеллируемость. Действительно, если мы зашифруем информацию закры-

тым ключом, а затем сможем дешифровать открытым, то сможем сделать вывод о том, что именно владелец пары ключей и никто другой шифровал информацию (закрытый ключ известен только ему), то есть она становится для нас аутентичной, а в случае использования сертифицированных ключей еще и апеллируемой. Это свойство ключевой пары лежит в основе формирования и верификации электронной цифровой подписи (ЭЦП).

ЭЦП – это набор методов, которые позволяют перенести свойства рукописной подписи под документом в область электронного документооборота [4]. Она обеспечивает аутентичность автора сообщения, уникальность подписи, контроль целостности передаваемого сообщения, невозможность переноса подписи под другой документ. Все эти свойства достигаются за счет синтеза асимметричной криптографии и хеширования документов.

Хеширование – процесс получения уникального дайджеста сообщения, который уникально идентифицирует сообщение: вычислительно трудно подобрать как сообщение с заранее известным хеш-дайджестом, так и два разных сообщения с одинаковым дайджестом. Формируют подобные дайджесты специальные хеш-функции. Примерами таких функция являются MD-5, SHA-1, SHA-2, SHA-3, RIPEMD-160, ГОСТ 34.11–12. На вход подобной функции подается сообщение произвольной длины, а на выходе формируется блок данных фиксированной длины (от 128 до 512 бит у различных алгоритмов), который и будет уже упоминавшимся дайджестом сообщения. Хеш-функция чувствительна ко входу и изменение даже одного бита во входном сообщении приводит к существенному (до 50% инверсии бит) изменению результирующего дайджеста на выходе. Поэтому очень трудно подобрать сообщение, которое будет иметь заданный дайджест. Рассмотрим, как выглядит схема формирования и верификации ЭЦП (рис. 2).

В схеме на рис. 2 владелец пары ключей выступает отправителем письма с подписью. Чтобы сформировать подпись, отправитель получает дайджест текста сообщения и шифрует его своим закрытым ключом. Результат шифрования можно использовать как электронную подпись, ее вместе с самим сообщением отправляют получателю. На стороне получателя подпись дешифруется открытым ключом отправителя и результат дешифрации сравнивается с результатом хеширования пришедшего сообщения. Если дайджест сообщения на стороне получателя совпадает с расшифрованным дайджестом стороны отправителя, то можно считать подпись верной. Подобную схему реализуют алгоритмы ЭЦП DSA, Шнорра, ГОСТ 34.10–12.

Для того, чтобы подменить письмо, злоумышленнику необходимо либо подобрать другое, устраивающее его по содержимому, сообщение, которое имеет дайджест, аналогичный дайджеста исходного сообщения, либо по известному открытому ключу подобрать закрытый ключ и подменить в посылке как само письмо, так и подпись к нему. Обе эти задачи при условии использовании перечисленных выше алгоритмов ЭЦП является вычислительно неразрешимыми на современном уровне технологий. Открытый и закрытый ключи генерируются одновременно, и между ними существует определенная математическая связь. Основная задача проектировщика асимметричного алгоритма заключается в том, чтобы по

известному открытому ключу было бы невозможно (очень трудоемко) получить секретный ключ шифрования. Для этого в основу асимметричных

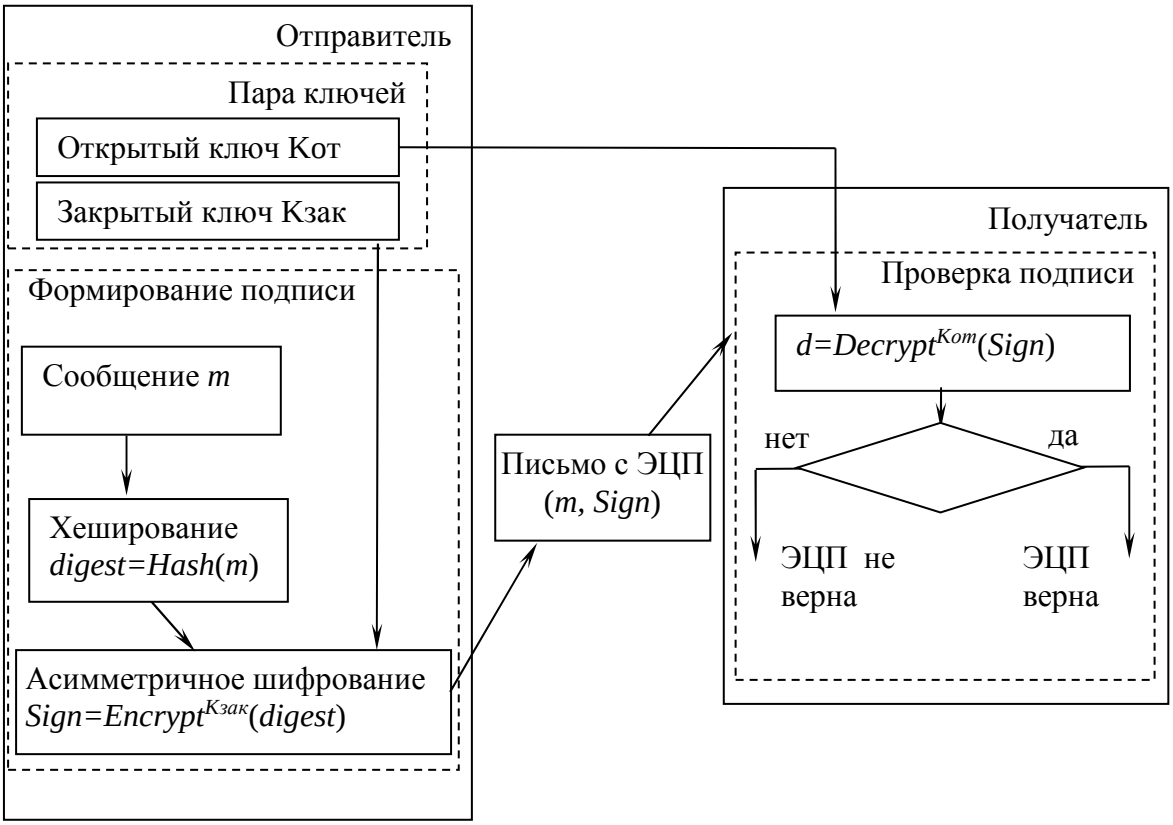


Рисунок 2. Обобщенная схема процесса формирования/проверки ЭЦП

алгоритмов закладываются вычислительно трудные задачи факторизации, дискретного логарифмирования, проецирования точек на эллиптической кривой и т.д. Практические аспекты использования асимметричной криптографии и электронной цифровой подписи рассмотрим на примере использования программной системы GnuPG.

2.Общие сведения о GnuPG.

2.1. **GnuPG** – первое знакомство

GnuPG (англ. *GNU Privacy Guard*) — свободная распространяемая компьютерная система (распространяется под лицензией GNU General Public License), позволяющая выполнять операции шифрования (кодирования) и цифровой подписи почтовых сообщений, файлов и другой информации, представленной в электронном виде. Является идеологическим наследником разработанной Филиппом Циммерманном в 1991 году системы PGP, ставшей в середине 2000-х годов пропиетарной. GPG предоставляет своим пользователям невскрывемые на современном уровне развития криптологии криптоалгоритмы. Перечислим основные возможности системы:

- Полностью реализует стандарт OpenPGP.

- Поддерживает электронную подпись с помощью алгоритмов ElGamal, DSA, RSA и хеш-функций MD5, SHA-1, SHA-2, RIPE-MD-160 и TIGER.
- Поддерживает асимметричное шифрование с использованием алгоритмов ElGamal и RSA и длиной ключа от 1024 до 4096 бит.
- Позволяет осуществлять симметричное шифрование с использованием блочных алгоритмов AES, CAST5, 3DES, Twofish, Blowfish, Camellia.
- Поддерживает алгоритмы сжатия: ZIP, ZLIB, BZIP2.
- Имеет модульную архитектуру, позволяющую устанавливать плагины с дополнительной функциональностью.
- Интегрированная поддержка серверов ключей.
- Может работать в консольном и графическом (для Window-платформ) режиме.

GPG имеет множество реализаций, совместимых между собой и рядом других программ (например, PGP) благодаря стандарту OpenPGP (RFC 4880), но имеющих разный набор функциональных возможностей. Начиная с версии 2.0 GPG поддерживает стандарт S/MIME (IETF 3851, ITU-T X.509). Это позволяет поддерживать в рамках системы две системы сертификации – сетевую (OpenPGP) и иерархическую (S/MIME). Существуют реализации GPG для всех наиболее распространённых операционных систем. Дистрибутивы системы можно скачать с сайта проекта https://www.gnupg.org.

2.2. Изготовление сертификатов ключей

Рассмотрим обобщенный порядок работы с системой GPG с использованием стандарта OpenPGP. Выполнение криптографических действий над информацией невозможно без наличия ключей шифрования. Для генерации пар открытый/закрытый ключ система предлагает использовать сервер сертификации Cleopatra или консольную команду gpg с набором параметров запуска. Рассмотрим процесс генерации ключей с использованием Cleopatra. На рис. 3 представлен внешний вид программы.

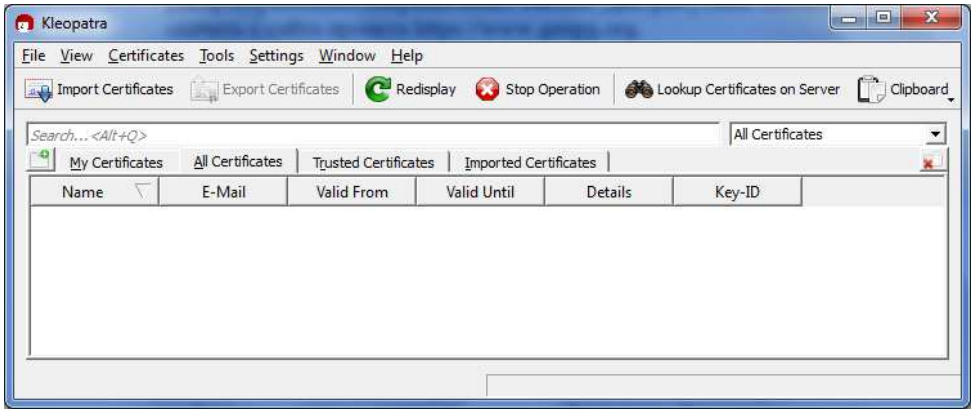


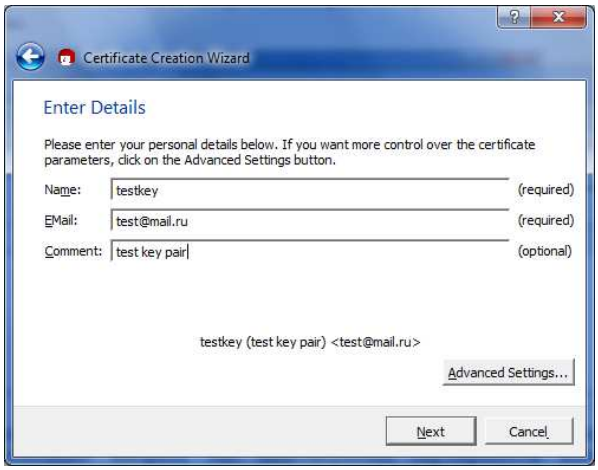
Рисунок 3. Окно администратора ключей Cleopara

Непосредственно после установки в системе не зарегистрировано сертификатов ключей, чтобы они появились в окне необходимо либо создать, либо импортировать сертификаты. Процесс создания новой пары ключей инициируется выбором пункта меню *File->New Certificate.* В открывающемся окне необходимо выбрать тип используемых сертификатов ключей – OpenPGP (PGP/MIME) или X.509 (S/MIME).

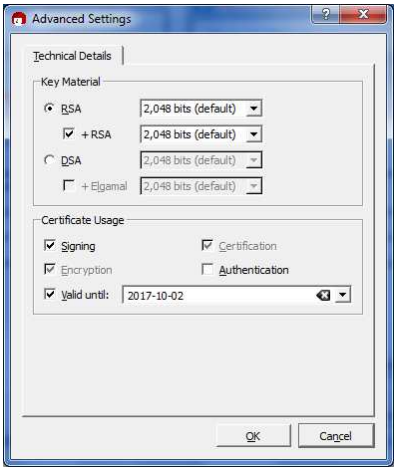
Рисунок 4. Выбор типа генерируемых ключей

Рассмотрим процесс генерации ключей на примере стандарта OpenPGP. После выбора соответствующей опции и нажатия клавиши *Next* начинается работа мастера, который управлеят процессом создания ключей, предлагая пользователю сделать выбор в последовательности диалоговых окон. (рис.5). Первое окно предлагает задать персональные данные владельца пары ключей (имя, адрес электронной почты и комментарий, рис. 5а). Нажав кнопку *Advanced Settings* этого окна можно задать параметры генерируемой ключевой пары (рис. 5б) – размер ключей от 1536 до 4096 бит, алгоритм, который будет использоваться для шифрации и верификации ЭЦП, дату истечения срока действия ключевой пары, для каких целей (шифрование, ЭЦП, сертификация, аутентификация) она будет использоваться. Перед началом непосредственной генерации мастер предлагает еще раз посмотреть выбранные параметры генерируемых колючей (рис. 5в).

После нажатия кнопки *CreateKey* в этом окне начинается непосредственный процесс генерации, помочь в котором пользователь может, вводя случайные символы в окне, изображенное на рис.3г, для определения базы генерации случайных чисел для создаваемых ключей. Наконец, сстема предложит ввести парольную фразу (рис 5д), которая в дальнейшем будет защищать пользователя от несанкционированных манипуляциях стронних лиц с его закрытым ключем. Когда парольная фраза будет введена и корректно подтверждена, генератор создаст пару ключей и выведет окно об успешном завершении опреации (рис. 5е).

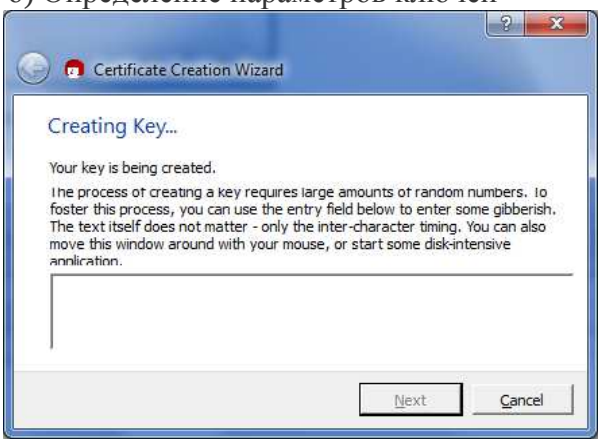


а) Ввод информации о владельце

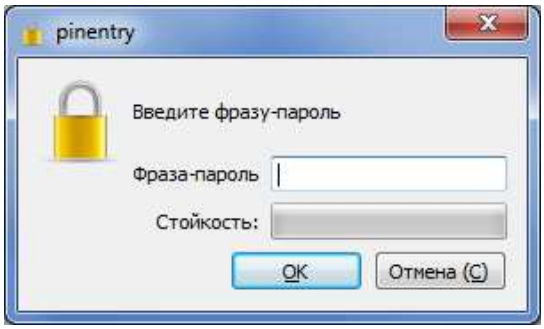


б) Определение параметров ключей

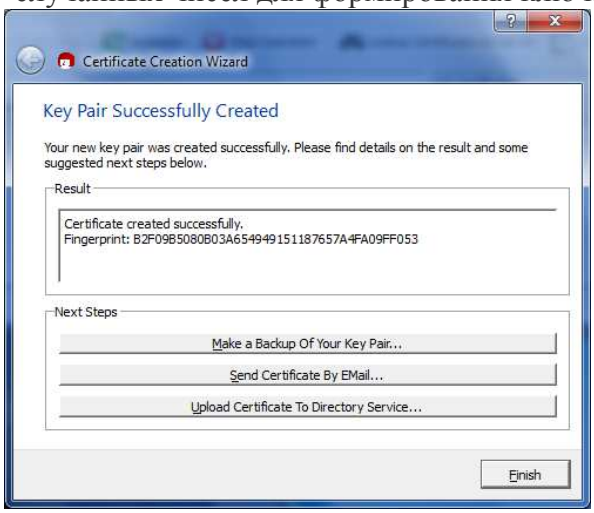
в)Поверка введенных данных



г)Атоматизированная генерация случайных чисел для формирования ключа



д) Ввод парольной фразы для защиты го ключа



е) Отчет об окончании генерации ключе зарытовой пары

Рисунок 5. Окна мастера генерации ключей

В этом окне отображается 40-цифровой *fingerprint* («отпечаток пальца») созданного сертификата ключевой пары, уникальность которого гаарантируется с очень высокой доли вероятности, что позволяет использловать этот отпечаток в качестве уникального в мировом масштабе идентификатора сгенерированной ключевой пары. Обычто в качестве

одентификатора сертификата используются последние 8 цифр отпечатка, что все равно гарантирует низкую вероятность коллизии именования разных ключевых пар.

По окончании процедуры генерации в окне Cleoptra вы можете увидеть строку, соответствующую вновь созданному сертификату (рис.6).

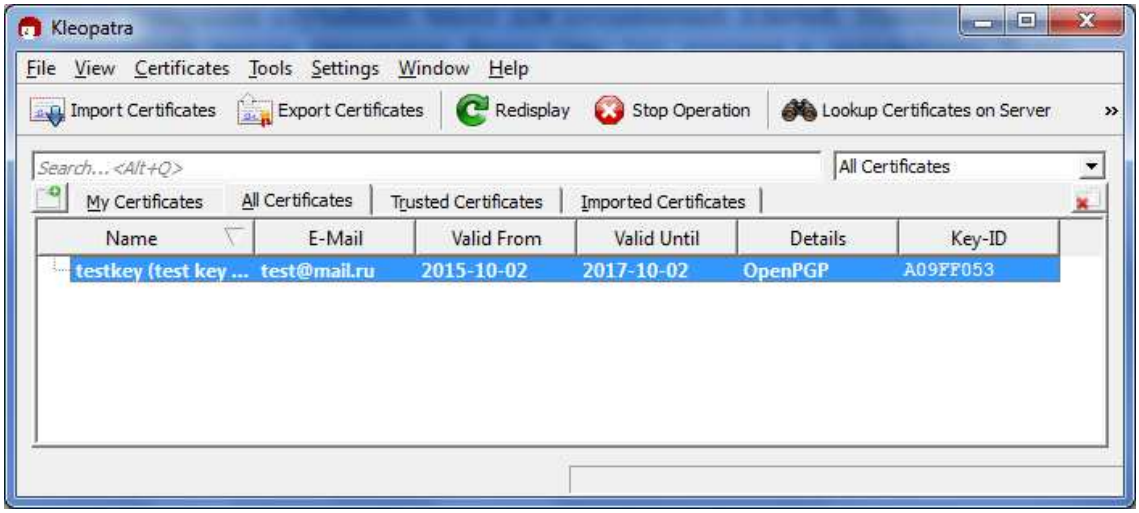


Рисунок 6. Сертификат ключевой пары отображается в окне менеджера ключей.

2.3. Распространение ключей.

Основной принцип асимметричной криптографии – хранить в секрете закрытый ключ для целей формирования ЭЦП и расшифровки пришедших сообщений и распространять публичный ключ своим клиентам, коллегам и собеседникам, чтобы они могли проверить вашу подпись под документом или зашифровать направляемое вам сообщение. Это делает актуальной задачу предоставления доступа к вашему публичному ключу всех желающих. Существуют два пути реализации этой задачи:

1) Через файл экспорта публичного ключа.
2) С использованием сервера сертификатов.

В первом случае необходимо воспользоваться командой *File->Export Certificates*. В ответ на эту команду необходимо выбрать папку и имя файла, в который будет экспортирован сертификат открытого ключа. Если выбрать в качестве расширения файла сертификата *.asc*, то сертификат будет экспортирован в текстовом виде:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFYOiIoBCACuwNxDPNiFu+DiVb6r4bmBCHmycOcce8JkQSHpwGlZcWdIZGZZ
V34pEt4gnKBVBkDLchZDriyODRRYfbXyzaZ5kg5U/Lk5yWE6TVh4/mmhgvzX1A0w
G4CgMD5Nfc+1KIztNgovOqmj7rTs8UpuZfzYNh06R8godvyX9WuVXLiAGxKeEhUS
KZ8T8qIq137brgHpH8HtKtdZuCp8r0lo22O4XzQ9a/24jXt3fu/rt5Emf2BkI7H3
3QfHxtYjVRlsAreq8mJEEKI7v04xMiGvaJUQpcCOB3scyF4YRk6kQR3AIsXoedGV
gPEcXg+ki/0JiQQi2Uq+OF2JrZmK6O6aH8tXABEBAAG0JnRlc3RrZXkgKHRlc3Qg
a2V5IHBhaXIpIDx0ZXN0QG1haWwucnU+iQE/BBMBCAApBQJWDoiKAhsDBQkDw3eG
BwsJCAcDAgEGFQgCCQoLBBYCAwECHgECF4AACgkQh2V6T6Cf8FNx1ggAlquzKSQU
```

OoXIfScM3vyjulj25Jlaga5LcxZ/wlh4eaSUnJIpAA+KTe/gUDYX0WMSDtoxYPXw
z5wYslpGwA4VXhyv/Qhoyv+y/Ue3M9oXqyJf8nX5FQodbC+hHQFLbUibeg90UgWt
RM7I7QuQxmJ3VeRlu5QxTNqhUrxKK8sJ4zuPdSoAlWSoI+Oey4OLj6e/sCEkCqEf
bzKwjXhEScMje5RDCKbcx9AyHK90tsZf0qvNCj4K+7KZvqKAZa7Vyhdg7cIVPbqq
KnfbBJb+1mrIAnKc4gOyDOCiMmDc5yQRVyhojrdaZeYepzFJgDFBf8jCQW1hphie
b2Wa9XP+ipGReLkBDQRWDoiKAQgAs3pcXCNjJ6TWV2x4eTXnKW1FECgzAA1ZTfou
KUVGu840yXKigTs5+2okjLVLkY0095s74Bmsxm2/h1DngjEV718PylVhFx4KISUS
3rQj9lZ2soF0hg3v7wtol8PPCm0unnxzb4HJtJ5PZg0QV5fBPld01eajQzbB8qaI
CaIFcx+/wNE8O4EoWkhCix+EpESLxQfF8OKLEly12VEOrXMWGtJ393AtNCmuU0U8
kNlTj5mxp35vS7LSg4vzrFHNQkmCxUhijRfqwuMbU81t2/MTlz86Ul6nNFh/k/pF
j9yHhPAhv8HQ/JlUE9Liwgj6t3K+wOpBsE6RNUANCz2V/p8VmwARAQABiQElBBgB
CAAPBQJWDoiKAhsMBQkDw3eGAAoJEIdlek+gn/BTcw8H/0IIby3+gZTmZeFdQ+ys
doimMkSfJ3QqWv0JPKcRnZiQ5rJpjv2MYctdBrkxUVv5Xvk2RBrwfU1tn4J5nyi8
ywEbpQMeRlj0ke61s4O03b2Gnf/oAemilqLZW4TwN17mppXqk8wHyuZHUVewpu3q
iyKxMht9D9kfIUuV6l4sK6kz+oc+mfmp3AU19LZf71+WET8B4rg1q/Jc4zjQqvxZ
jT/C0HE1sB8U1WbVX9IqEmHGy7JkVupATN5TrMl/og7wabOnz+VJx3/43o+P9UuC
9yAqsTKxO6ROsDeDzU686CFR+udHkm3JmrBO0sjq0B9gzRvdDQ9ec+I+m53viUEh
0Po=
=OFeM
-----END PGP PUBLIC KEY BLOCK-----

Если же экспортировать сертификат в файл с расширением. gpg или *.pgp*, то экспорт осуществится в двоичном формате.

После этого файл сертификата можно передать любому желающему любым удобным способом (по электронной почте, через разделяемые файловые хранилища или через flash-накопители при личной встрече). Чтобы воспользоваться вашим открытым ключом, обладатель сертификата должен импортировать его в свой менеджер ключей. Для этого используется команда *File->Import Certificates* (или одноименная кнопка на панели инструментов Cleopatra). Выбрав в открывшемся окне файл с сертификатом открытого ключа, пользователь импортирует его в систему (см сертификат с ID= B358DF69) на рис. 7.

Для того, чтобы убедиться в принадлежности ключа лицу, от имени которого фал сертификата был вам отправлен (например, по электронной почте), необходимо сверить его *fingerprint* (*“отпечаток пальца”*). Как уже было отмечено ранее, этот отпечаток (и даже его последние 8 цифр) уникально идентифицирует ключевую пару и ее владельца. Поэтому

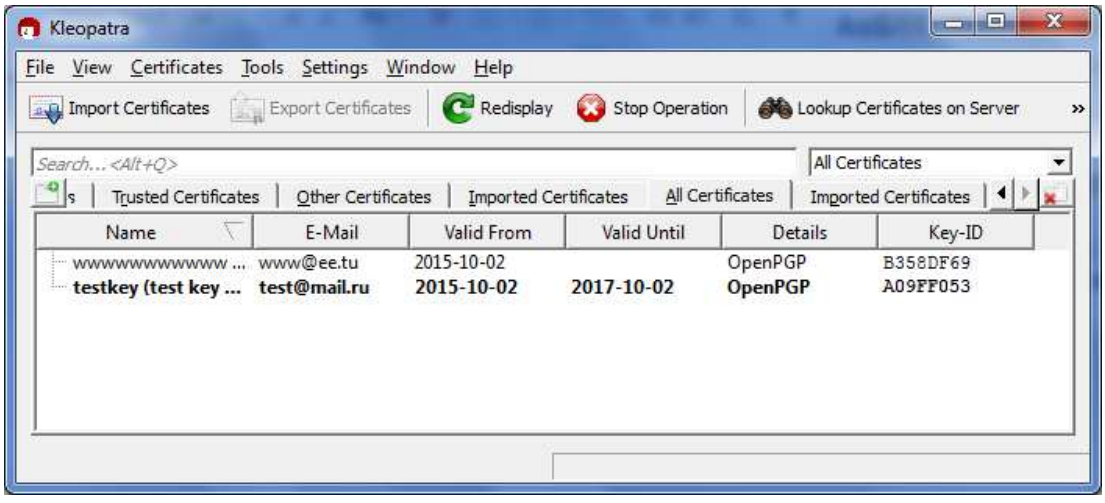


Рисунок 7. Импортированный сертификат открытого ключа в Cleopatr’e

достаточно в окне свойств импортированного ключа (открывается двойным щелчком по сертификату в окне рис.7) убедиться в соответствии отпечатка установленного ключа (рис.8) тому отпечатку, который заявляет истинный владелец ключевой пары и можно считать истинность ключа доказанной. Проблема получения информации о значении отпечатка ключевой пары от ее владельца может решаться различными способами: телефонный звонок, размещение отпечатка владельцем на персональной Интернет-странице или визитке. В любом случае, аутентичность этой информации должен проверять импортирующий.

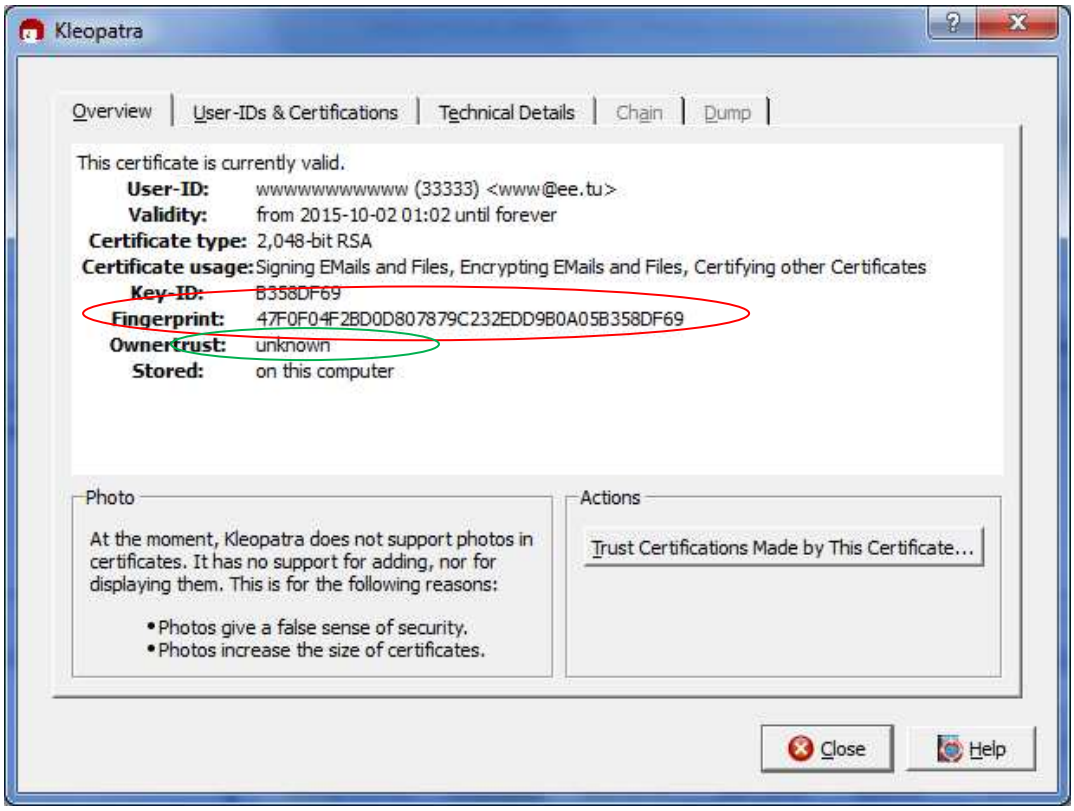
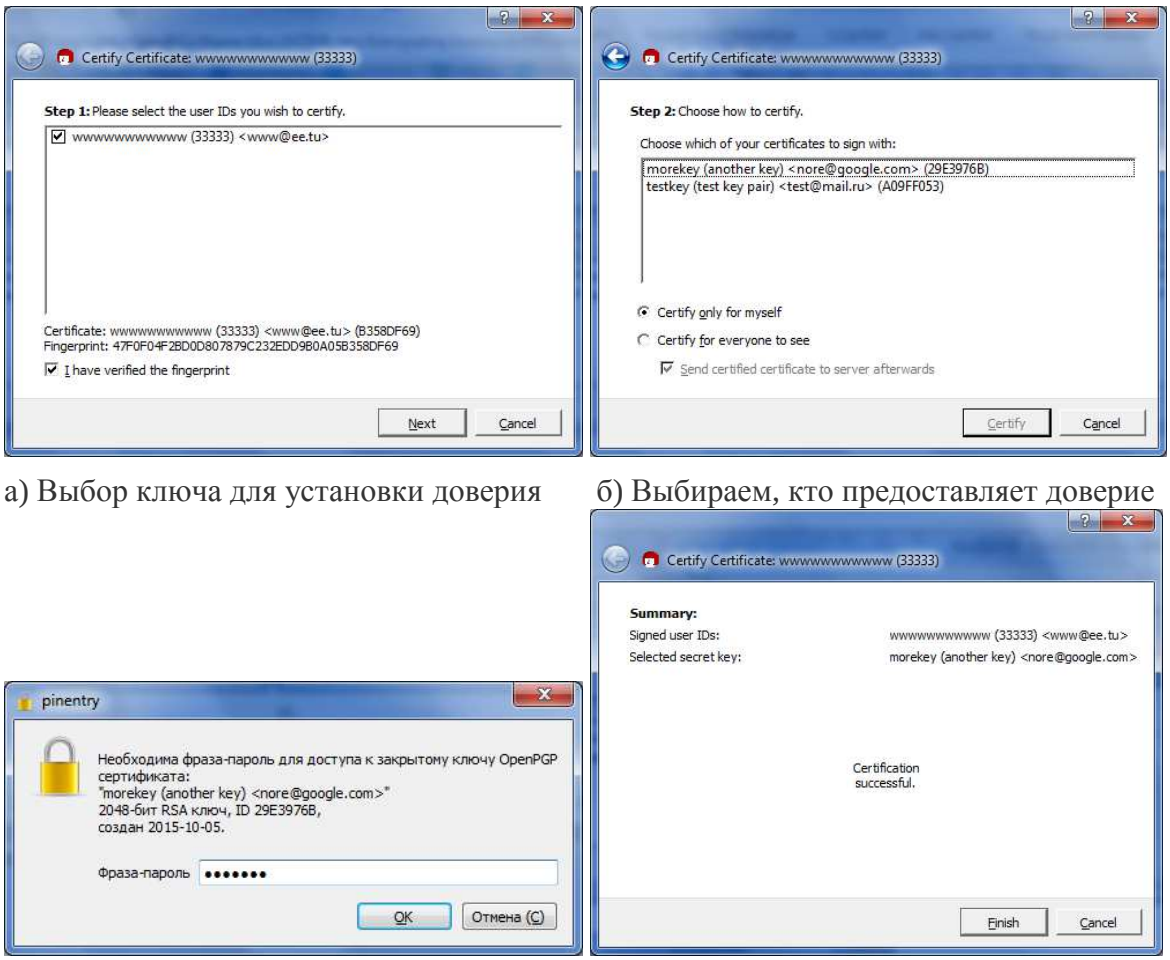


Рисунок 8. Информация об импортированном сертификате

Специально для аутентификацию ключей пользователям GPG предусмотрен механзм установки доверия: пользователь, который убедился в истинности импортированного сертификата может сообщить остальным пользователям сстемы об этом, установив сертификату доверие от своего имени. Для сертификации стороннеого сертификата использлуется команда *Certificates->Certify Certificate*. В появившемся окне (рис.9а) необходимо выбрать, какой из установленных публичных ключей необходимо сертифицировать и указать, что отпечаток ключа был проверен. В следующем окне (9б) необходимо указать, от имени каокго пользователя (каким из зарегистрироанных закрытых ключей) будет сертифицироваться ключ. Далее следует стандартная защита от несанкционированного использования закрытого ключа – ввод парольной фразы (рис. 9в). Успешное завершение процесса сертификации сопровождается выводом сообветствуего окна (рис. 9г). После этого импортированный ключ можно увидеть в списке *Trusted Certificates* окна менеджера ключей.



а) Выбор ключа для установки доверия

б) Выбираем, кто предоставляет доверие

в) Парольня фраза защищает закрытый ключ от несанкционированного использования

г) Подтвержение предоставления доверия

Рисунок 9. Процесс утановления доверия сертификату

Таким образом формируется сеть доверия «Web of Trust», в которой я доверяю некоторому ключу незнакомого мне пользлователя, если ему установил доверия другой пользователь, которому я доверяю. Подобная цепочка доверия может быть и более длинной: «Я доверяю А, он доверяет В, В дверяет С, а, следовательно, я могу доверять С». Чтобы эта цепочка помогала принимать более ответственные решения о доверии, система позволяет определять степень доверия тому или иному ползователю: если в окне свойтсв сертифиакта (рис. 8) нажать кнопку «Trust Certifications Made by This Certificate…» («Доверять сертификациям, сделанным от имени данного сертификата»), то в открывшемся окне (рис.10) можно указать, насколько вы доверяете рекомендациям владельца этой ключевой пары.

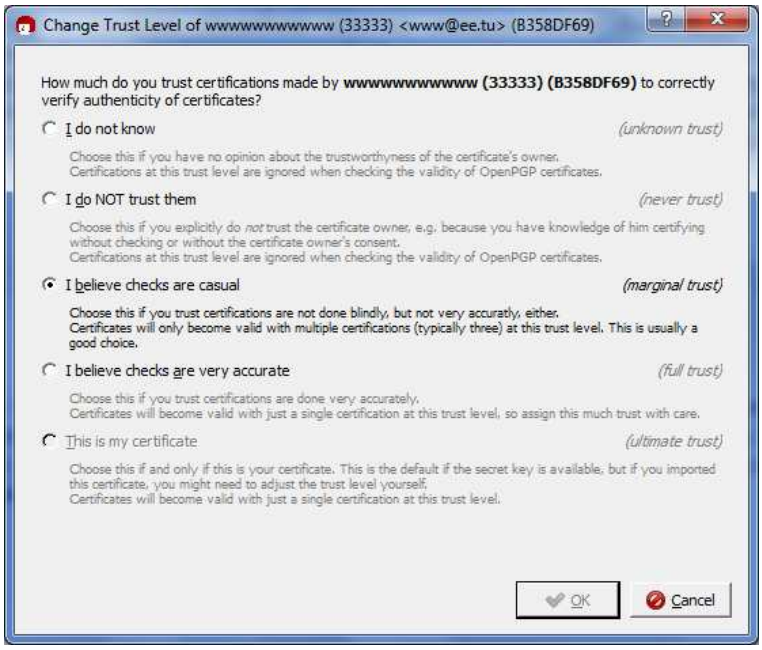


Рисунок 10. Оценка доверия владельцу ключевой пары.

Возможны варианты:

I do not know Я его не знаю	Неизвестный пользователь. Игнорируется при проверке полинности сертификатов OpenPGP
I do NOT trust them Я му не доверяю	Влалельцу ключа отказывается в доверии – он уличен в установке доверия непроверенным сертификатам
I believe checks are casual Частичное доверие владельцу сертификата	Владелец ключа старается сертифицировать сторонне ключи с проверкой, но не всегда. Если более 3-х пользователей установят такой уровень доверия сертификату, он принимается как заслуживающий доверия
I believe checks are very accurate Полное доверие владельцу сертификата	Владелец ключа всегда строго проверяет сертифицируемые ключи. Если у ключа есть хотя бы один подобный уровень доверия, он считается заслужаваюим доверия
This is my certificate Это мой сертификат	Этот уровень устанавливаюся для собственных сертификатов, они становятся доверенными

Установленный уровень доверия владельцу сертификата со стороны текущего пользователя указывается в окне свойств сертификата (рис. 6 под зеленым овалом).

Еще одной замечательной возможностью импорта/экспорта открытых ключей является *сервер ключей*. Выбрав пункт меню *File->Export Certificates to Server…*, можно опубликовать свой открытый ключ на общедоступном сервере ключей (предварительно адрес одного или нескольких серверов ключей должен

быть добавлен в настройках Cleopatra). Пользователь может выбирать из множества проверенных серверов ключей OpenPGP, например:

```
hkp://keys.gnupg.net
hkp://blackhole.pca.dfn.de
hkp://pks.gpg.cz
hkp://pgp.cns.ualberta.ca
hkp://minsky.surfnet.nl
hkp://keyserver.ubuntu.com
hkp://keyserver.pramberger.at
http://keyserver.pramberger.at
http://gpg-keyserver.de
```

Открытый ключ, опубликованный на сервере, можно импоритировать в менеджер ключей, выбрав команду *File->Lookup Certificates on Server*. В прделагаемом окне (рис.11) предлагается найти сертификат по имени или почтовому адресу и импортировать, нажав соответствующую кнопку.

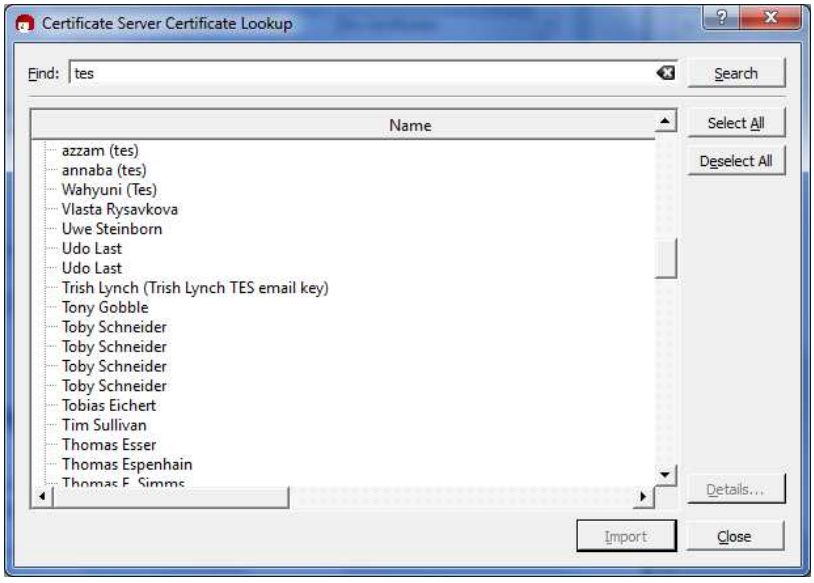


Рисунок 11. Импорт открытого ключа с сервера ключей

2.4. Шифрование и подпись документов.

Теперь, когда создана пара ключей и открытый ключ передан получателю сообщений, настало время приступить непосредственно к защите информации. Для этих целей подойдет как менеджер ключей Cleopatra, так и специальное расширение для ОС Windows под названием GPG Explorer eXtension (GpgEX), которое делает доступными команды шифрации/дешифрации и формирования/верификации ЭЦП доступными из контекстного меню Windows. Для того, чтобы зашифровать файл (сформировать электронную подпись для него) выбираем команду *File->Sign/Encrupt Files*… в Cleopatra или пункт *Подписать и зашифровать*… в контекстном меню шифруемого файла или папки в проводнике Windows (рис.12).

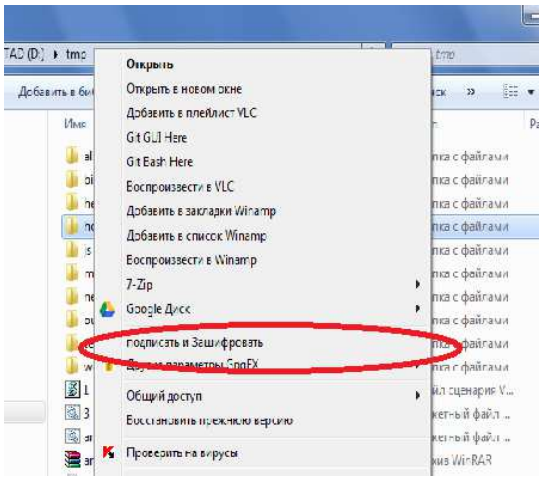


Рисунок 12. Команда шифрации/подписи файла (папки) доступна в контекстном меню проводника

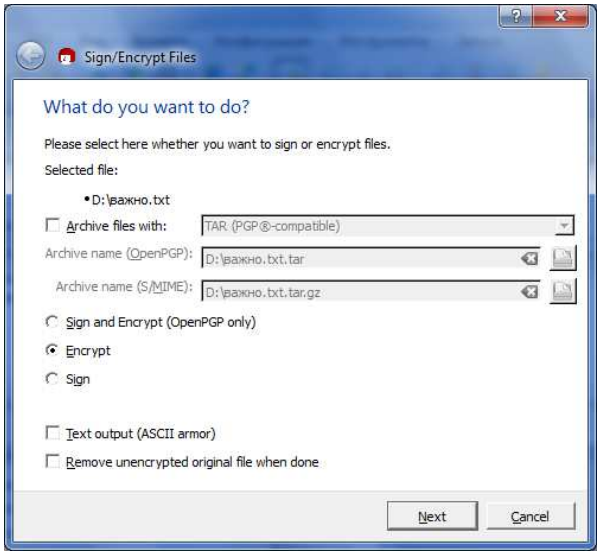
В ответ на команду откроется окно (рис.13а), в котором необходимо выбрать требуемую операцию (только шифровать, только подписать, и то, и другое). Помимо непосредственно защиты система предлагает сжать файл с результатом работы, а также осуществить транспортное кодирование (перевод результата в поток ASCII-байт), а также удалить исходный документ (если выбран режим с шифрованием).

Если был выбран режим шифрования, в следующем окне (рис. 13б) система предложит выбрать сертификат(-ы) открытого (-ых) ключа (-ей), с помощью которого(-ых) будет шифроваться документ. Можно выбрать несколько сертификатов, тогда сообщение смогут расшифровать несколько получателей – владельцев парных закрытых ключей. Рекомендуется (а если файл будет удаляться после шифрования, то настоятельно) включить в список сертификатов для шифрации и одни из своих сертификатов – иначе невозможно будет самому расшифровать документ.

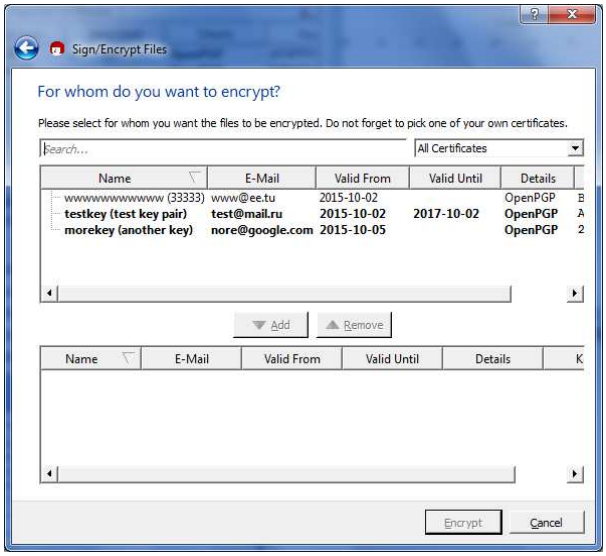
Если выбранный режим работы предполагает подпись документа, то откроется окно выбора сертификата закрытого ключа, которым будет подписан документ. В выпадающем списке *OpenPGP Signing Certificate* этого окна будут отобраны только сертификаты закрытых ключей, зарегистрированных в Cleopatra. Для защиты закрытого ключа от несанкционированного использования система предложит ввести парольную фразу, заданную при генерации ключевой пары (см. пункт 2.2). Удачное завершение процесса шифрации/подписи будет подтверждено выводом позитивного окна (рис. 13.д).

В результате выполнения операций шифрации/формирования подписи на диске в папке с документов появятся документы:

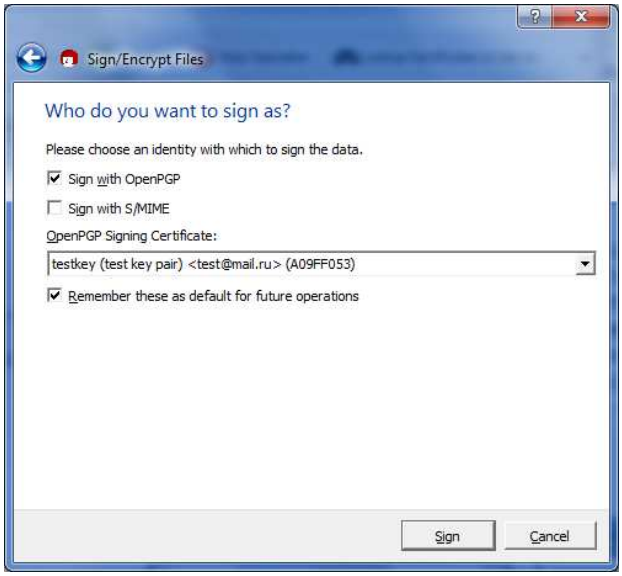
<имя_файла>.gpg	Файл с зашифрованным содержимым исходного файла
<имя_файла>.gpg.asc	Файл с зашифрованным содержимым исходного файла (если в окне 13а был выбран режим *for output as text/ASCII armor*)



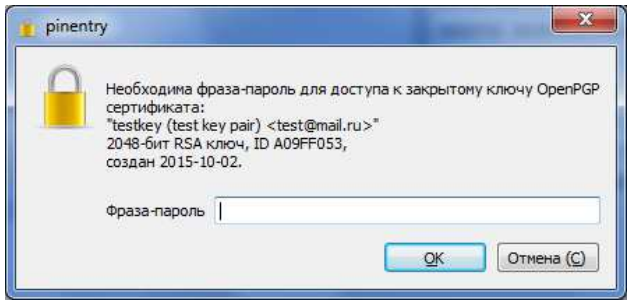
а) Выбор режима



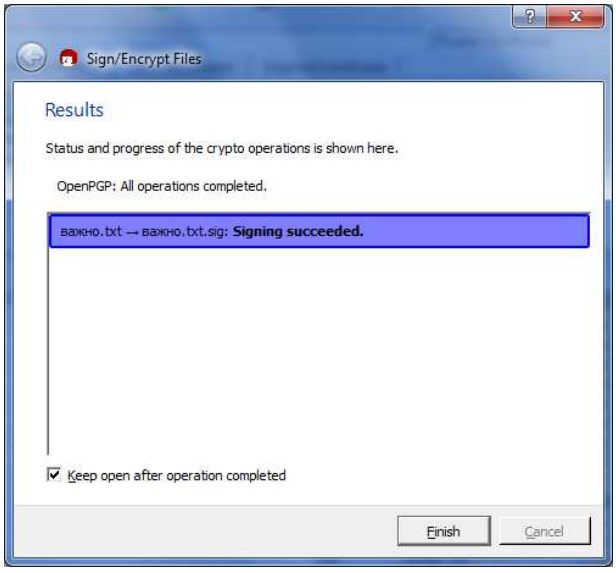
б) Выбор сертификата ключа шифрации



в) Выбор серификата ключа подписи



г) Ввод парольной фразы



д) отчет об успешном выполнении операции

Рисунок 13. Процесс шифрации/формирования подписи

<имя_файла>.sig	Файл с электронной подписью документа
<имя_файла>.asc	Файл с электронной подписью документа (если в окне 13а был выбран режим *for output as text/ASCII armor*)

Эти файлы можно передать получателю, чтобы он смог их расшифровать/проверить подпись под документом. Для этих целей получатель использует либо команду *File->Decrypt/Verify Files…* , либо *команду Расшифровать/ Проверить…* контекстного меню файла с шифром/подписью в проводнике Windows.

В открывшемся окне (рис. 14) можно уточнить параметры процесса дешифрации/проверки подписи: указать, какой файл был подписан, если происходит проверка электронной подписи (поле *Input file is a detached signature*), уточнить каким алгоритмом был дополнительно упакован файл в поле *Input file is an archive, unpack with*, а также задать папку, в которую будут помещены результаты выполнения операции в поле *Output folder*.

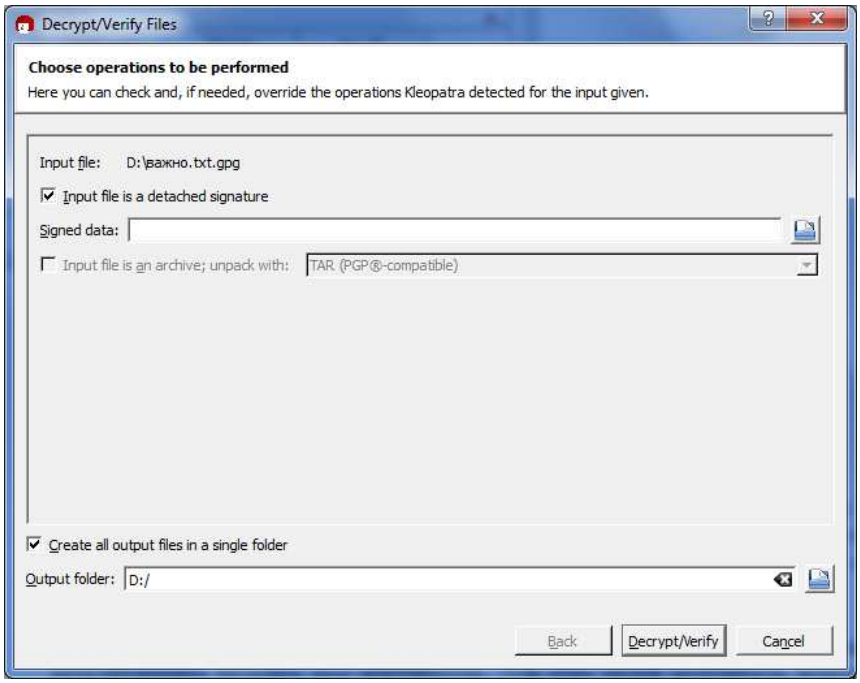
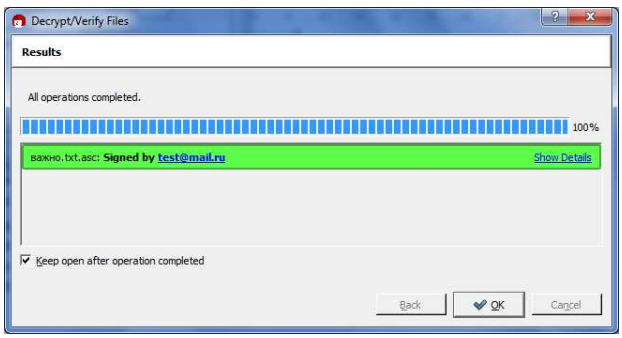


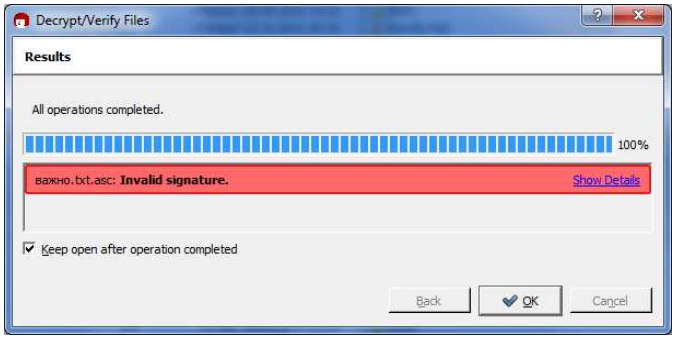
Рисунок 14. Окно настройка процесса дешифрации файла/верификации ЭЦП.

После нажатия кнопки Decrypt/Verify программа выполнит над выбранным файлом указанную операцию (в случае выполнения дешифрации предварительно запросит парольную фразу) и результат работы будет отображен в окне вида, представленного на рис. 15.

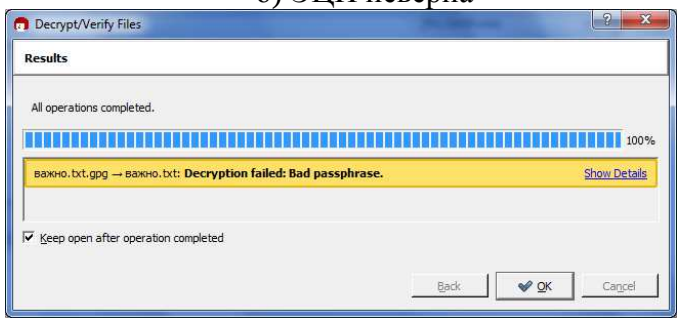
Если при проверке ЭЦП был использован верный открытый ключ и подписанный файл не изменялся с момента формирования подписи, то окно результата выполнения команды будет иметь вид как на рис.15а, если файл был изменен, то результатом операции будет окно вида 15б. Если при дешифрации будет троекратно введена некорректная парольная фраза – пользователь увидит окно 15в. Если же дешифрация будет произведена успешно – на диске в указанной в окне рис.14 папке будет размещен результат дешифрации.



а) ЭЦП верна



б) ЭЦП неверна



в) Введена неверная парольная фраза

Рисунок 15. Виды окон с результатами проверки ЭЦП или дешифрации файла

Помимо шифрации файлов GPG поддерживает шифрование/дешифрацию, а также подпись/проверку ЭЦП для буфера обмена Windows (кнопка *Clipboard* на панели инструментов Cleopatra), а также почтовых сообщений в поддерживаемых почтовых клиентах (подробности см. в документации, поставляющейся вместе с системой).

Еще одной востребованной функцией GPG является формирование/проверка контрольной суммы документов с использованием хеш-функции SHA.

Подав команду *File->Create Checksum Files..*, можно выбрать несколько файлов на диске, для которых будут вычислены криптостойкие контрольные суммы. Их значения будут занесены в файл sha1sum.txt в виде:

```
7b21848ac9af35be0ddb2d6b9fc3851934db8420  F:/gpg/важно.txt
7b21848ac9af35be0ddb2d6b9fc3851934db8420  F:/gpg/важно2.txt
7b21848ac9af35be0ddb2d6b9fc3851934db8420  F:/gpg/важно4.txt
```

где в каждой строке указывается хеш-дайджест файла, а также его имя и путь к нему.

Впоследствии можно проверить целостность файлов, сверив их текущие контрольные суммы с записанными в файле. Выполнить эту операцию можно с помощью команды *File->Verify Checksum Files..* Результаты проверки будут красноречиво отображены в окне отчета (рис. 16).

Как и для прочих уже упоминавшихся команд, для команд формирования/проверки контрольной суммы есть альтернативная форма инициализации через контекстное меню файлов (команды *Создать контрольные суммы/Проверить контрольные суммы*) в подменю *Другие параметры GpgEX*.

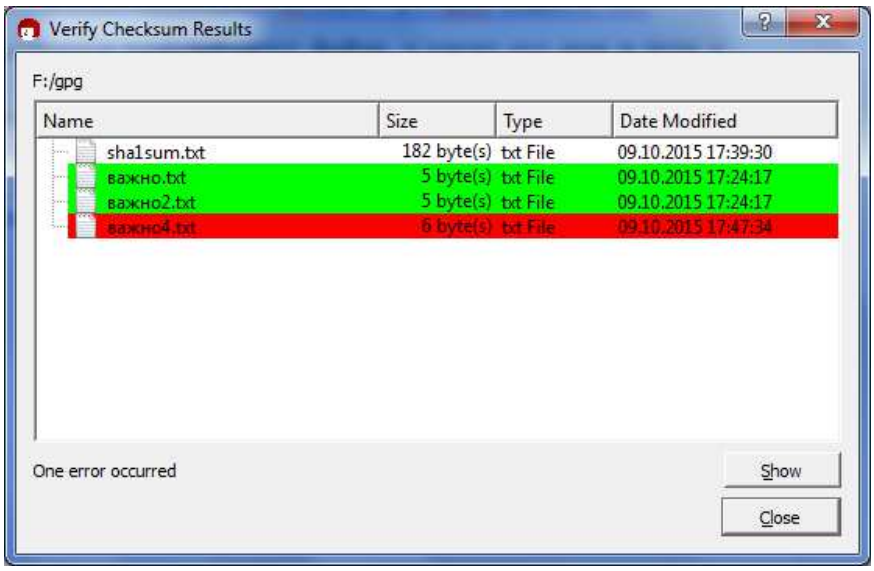


Рисунок 16. Результат проверки контрольных сумм файлов.

2.4. Интерфейс командной строки.

Функции GnuPG можно использовать не только в графическом интерфейсе. Как всякая система, зародившаяся в среде Unix, GnuPG позволяет выполнять все вышеперечисленные действия по консольным командам. Такой интерфейс работы с программой, возможно, не настолько нагляден и прост для обычного пользователя, но он позволяет автоматизировать и масштабировать криптографические процедуры, включая их в состав командных файлов или исполняя по расписанию планировщика задач.

Для использования возможностей системы GnuPG в консольном режиме предназначена внешняя команда gpg. Ее можно вызвать с параметрами как в командной строке, так и в составе командных файлов. Команда gpg поддерживает все вышеперечисленные режимы работы системы: генерацию, экспорт и импорт ключей, шифрование и дешифрацию файлов, формирование и проверку электронной подписи. Для gpg актуально соглашение о статусе завершения – если команда завершилась с кодом 0, то команда выполнилась успешно, код завершения 1 и выше соответствует ошибке, возникшей при выполнении команды.

Конкретное действие, выполняемое командой gpg зависит от параметров запуска. Рассмотрим основные:

-s, –sign

Создать подпись для документа. Команда может быть подана вместе с параметром –encrypt.

-b, –detach-sign

Создать отдельный файл с подписью.

-e, –encrypt

Зашифровать данные. Эта опция может быть подана вместе с –sign.

-c, –symmetric

Зашифровать только с использованием симметричного шифра.

–decrypt <имя_файла>

Дешифровать файл (или стандартный поток ввода, если файл не задан) и вывести результат в стандартный поток вывода (или в файл, заданный в опции –output). Если файл был дополнительно подписан, будет проверена корректность электронной подписи.

–verify <имя_файла>

Проверка подписи под файлом как внедренной в файл (расширение gpg), так и в отдельном файле (расширение sig)

–list-keys [<имена>]

Вывод списка всех зарегистрированных открытых ключей (всех или тех, которым соответствует заданное имя).

–list-secret-keys [<имена>]

Получить список установленных закрытых ключей (всех или тех, которым соответствует заданное имя).

–fingerprint [<имена>]

Вывести список всех ключей вместе с их отпечатками.

–gen-key

Создать новую пару ключей.

–edit-key <ID ключа>

Открывает меню с возможностью выполнения операций с ключом: смена имени пользователя, парольной фразы, установление доверия ключу, запретить/разрешить ключ и др.

–delete-key имя

Удалить публичный ключ

–delete-secret-key имя

Удалить ключевую пару

–export <имя>

Экспортировать все ключи или ключ с заданным именем. Имя файла экспорта можно задать с помощью опции --output.

–import <имя_файла>

Импортировать ключ из указанного файла.

Помимо опций, задающих характер выполняемых действий, команде gpg можно задать ряд конфигурационных опций, рассмотрим назначение некоторых из них (полный список можно посмотреть в [5])

-a, –armor

Выводить информацию в ASCII-формате

-o, –output <файл>

Выводить информацию в указанный файл.

-u, –local-user <имя>

Использовать ключ пользователя с указанным именем для формирования подписи

–default-key <имя>

Использовать указанного пользователя по умолчанию при формировании подписи.

–keyserver <имя>

Использовать указанный сервер ключей, если ключ не найдется среди зарегистрированных.

–cipher-algo <название>

Использовать указанный алгоритм шифрования. Список доступных алгоритмов можно получить, используя опцию –version.

–digest-algo <название>

Использовать указанный алгоритм формирования подписи. Список доступных алгоритмов можно получить, используя опцию –version.

–throw-keyid

Не размещать информацию об используемом ключе в зашифрованное сообщение. Повышает стойкость шифротекста к криптоанализу, но замедляет дешифрацию, т.к. приходится перебирать все доступные ключи

–s2k-cipher-algo <название>

Использовать указанный шифр для защиты закрытого ключа. По умолчанию используется Blowfish

Примеры выполнения криптографических операций с использованием консольной команды gpg:

Зашифровать файл публичным ключом указанного пользователя

```
gpg –u <ID пользователя> –e <имя_исходного_файла>
```

Зашифровать файл симметричным алгоритмом (ключ будет запрошен)

```
gpg –-output <имя_файла_с_шифром> –c <имя_исходного_файла>
```

Расшифровать зашифрованное сообщение в указанный файл

```
gpg –output <имя_файла> --decrypt <зашифрованный_файл>
```

Подписать файл от имени пользователя

```
gpg –u <ID пользователя>  --sign <подписываемый_файл>
```

Проверить электронную подпись под файлом (подпись будет размещена в отдельном файле с расширением asc)

```
gpg –u <ID пользователя>  --detach-sign
                                    <подписываемый_файл>
```

Проверить электронную подпись под файлом

```
gpg –verify <имя_файла_подписи>
```

2.5. Порядок выполнения лабораторной работы.

Лабораторная работа выполняется студентами в паре для полноценного обмена ключами, зашифрованными и подписанными сообщениями. Каждый из студентов в паре работает на компьютере с установленной системой GnuPG для Windows. Компьютеры должны быть объединены в сеть для оперативного обмена файлами. Порядок работы каждого из студентов в паре:

2.5.1. Создайте пару ключей в менеджере ключей Cleopatra.
2.5.2. Скопируйте произвольный текст в буфер обмена. Зашифруйте содержимое буфера обмена с помощью своего открытого ключа. Вставьте содержимое буфера обмена в текстовый редактор, убедитесь, что оно зашифровано. Теперь скопируйте в буфер шифротекст, дешифруйте его своим закрытым ключом, вновь вставьте содержимое буфера обмена в текстовый редактор, убедитесь, что текст был успешно расшифрован.
2.5.2. Экспортируйте сертификат открытого ключа из своей пары ключей в файл и передайте его своему напарнику.
2.5.3. Получив файл с экспортированным ключом от напарника, импортируйте его в менеджер ключей. Установите для импортированного ключа полное доверие.
2.5.4. Зашифруйте с использованием импортированного ключа напарника произвольный текст на диске. Передайте зашифрованный текст напарнику.
2.5.5. Получив зашифрованный файл от напарника, дешифруйте его своим закрытым ключом. Убедитесь, что файл был успешно дешифрован.
2.5.6. Используя свой закрытый ключ, подпишите произвольный файл на диске электронной подписью. Передайте подписанный документ вместе с подписью напарнику.
2.5.7. Получив от напарника документ с подписью, убедитесь, что подпись верна. Измените подписанный документ и убедитесь, что подпись стала неверна. Верните документ к первоначальному состоянию и вновь убедитесь, что подпись верна.
2.5.8. Скопируйте во временную папку несколько документов. Сформируйте для этих документов файл с контрольными суммами. Внесите изменения в один или несколько документов и убедитесь, что система обнаружит расхождения контрольных сумм.
2.5.9. Выполните индивидуальное задание в соответствии с заданием. Задание предполагает написание командного скрипта, выполняющего преобразования данных в соответствии с заданием

2.5.9.1. Написать скрипт, который с ключом /e будет шифровать все текстовые документы в заданной параметром скрипта папке, а с ключом /d дешифровывать их.

2.5.9.2. Написать скрипт, который будет с использованием ЭЦП контролировать целостность файлов в заданной папке.

2.5.9.3. Написать скрипт, который будет с ключом /s сканировать заданную папку и для всех новых файлов в ней создавать электронную подпись, а с

ключом /v проверять подписи под файлами и формировать отчет о корректности подписей в файле-отчете.

2.5.9.4. Написать скрипт, который с будет формировать/проверять общую подпись для всех текстовых файлов (как единого целого) в заданной папке.

2.5.9.5. Написать скрипт, который будет шифровать и подписывать текстовые файлы в заданной папке с ключом /e, а дешифровывать при запуске с ключом /d только те файлы, которые прошли проверку на корректность электронной подписи.

2.5.9.6. Написать скрипт, который будет шифровать и дешифровывать файлы в папке указанной в качестве параметра скрипта, от имени пользователя, идентификатор которого также передается в скрипт как параметр.

2.5.6.7. Написать скрипт, который будет выполнять сое содержимое только после проверки собственной целостности с помощью электронной подписи.

2.5.10. Составьте отчет по лабораторной работе

2.5.11. Ответьте на контрольные вопросы к лабораторной работе. Отчитайте лабораторную работу преподавателю.

2.6. Контрольные вопросы

1. Перечислите достоинства и недостатки асимметричных алгоритмов в сравнении с симметричными.
2. Перечислите основные трудновычислимые задачи, использующиеся в современной асимметричной криптографии.
3. Какую роль в современных криптосистемах играют симметричные и асимметричные алгоритмы шифрования?
4. Как обеспечивается доверие к открытым ключам асимметричных шифров и верификации ЭЦП? Как решается этот вопрос в системе GPG?
5. Какую роль в стандарте OpenPGP играют сервера ключей? Как их можно использовать в системе GPG?
6. Какие стандарты поддерживает система GPG?
7. Какая схема сертификации ключей реализована в системе GPG? Перечислите ее достоинства и недостатки.
8. Опишите схему работы асимметричной криптосистемы.
9. Опишите схему формирования-верификации ЭЦП.
10. Объясните, чем отличается процесс проверки контрольных сумм и ЭЦП в системе GPG. Какой из методов контроля надежнее?

2.7. Содержание отчета

Отчет по лабораторной работе должен содержать следующие сведения:

- Название и цель работы;
- Текст индивидуального задания к лабораторной работе;

- Экранные формы основных этапов работы с системой GPG при выполнении заданий пп 2.5.1-2.5.9;
- Текст скрипта, реализующего индивидуальное задание.

Список литературы:

1. Лясин Д.Н. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ: Асимметричное шифрование Волгоград. гос. техн. ун-т. - Волгоград, 2014, 15 с.
2. Лясин Д.Н. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ: Блочное симметричное шифрование Волгоград. гос. техн. ун-т. - Волгоград, 2014, 20с.
3. В. Ященко. Введение в криптографию.- М. : МЦНМО, 2012. – 352с.
4. Д.Н. Лясин, А.А. Рыбанов, В.А. Носенко. Защита информации. Лабораторный практикум. Волгоград: РПК Политехник, 2012г, 97с.
5. https://www.gnupg.org/documentation/manpage.html. Справка по использованию команды gpg на официальном сайте GnuPG.

Учебное издание

Дмитрий Николаевич **Лясин**
Оксана Федоровна **Абрамова**

Асимметричная криптография и электронная цифровая подпись на примере системы GnuPG

Методические указания

План электронных изданий 2016 г. Поз. №
Подписано на « Выпуск в свет» . .16. Уч-изд. л. .
На магнитоносителе.

Волгоградский государственный технический университет.
400131, г. Волгоград, пр. Ленина, 28, корп. 1.