Лабораторная работа №2 по курсу криптографии

Выполнил студент группы М8О-308Б-18 Коростелев Дмитрий

Условие

1. Создать пару OpenPGP ключей, указав в сертификате свою почту.
2. Установить связь с преподавателем, используя созданный ключ, следующим образом:
   2.1. Прислать собеседнику от своего имени по электронной почте сообщение, во вложении которого поместить свой сертификат открытого ключа и сам открытый ключ (как правило, они умещаются в одном файле).
   2.2. Дождаться письма, в котором собеседник Вам пришлет сертификат своего открытого ключа.
   2.3. Выслать сообщение, зашифрованное на ключе собеседника.
   2.4. Дождаться ответного письма.
   2.5. Расшифровать ответное письмо своим закрытым ключом.
3. Собрать подписи под своим сертификатом открытого ключа.
   3.1. Получить сертификат открытого ключа одногруппника.
   3.2. Убедиться в том, что подписываемый Вами сертификат ключа принадлежит его владельцу -путём сравнения отпечатка ключа или ключа целиком, по доверенным каналам связи.
   3.3. Подписать сертификат открытого ключа одногруппника.
   3.4. Передать подписанный Вами сертификат полученный в п.3.2 его владельцу, т.е. одногруппнику.
   3.5. Собрать 10 подписей одногруппников
   3.6. Прислать преподавателю свой сертификат открытого ключа с 10ю или более подписями одногруппников
4. Подписать сертификат открытого ключа преподавателя и выслать ему.

Метод решения

Для выполнения данной работы была использована unix утилита gpg, в которой есть возможность выполнить создавать, экспортировать, импортировать, подписывать сертификаты и многое другое.

Для создания ключа нужно ввести следующую команду:

```
gpg –full-generate-key
```

После чего будет выведено диалоговое окно, в котором надо выбрать размер ключа, ввести почту и придумать кодовую фразу, которую нужно обязательно запомнить, так как с помощью этой фразы будет выполняться большинство взаимодействий с ключом.

После выполнения операции был сгенерирован новый PGP ключ размером 4096 байт,

Мой отпечаток ключа: **C9ED79ECD41A5A1BB9EA0D1080188575AEB9334A**

Чтобы экспортировать ключ нужно написать команду:

```
gpg -a --export C9ED79ECD41A5A1BB9EA0D1080188575AEB9334A > public.asc
```

После выполнения данной команды, в файле public.asc содержится сертификат ключа и сам публичный ключ.

По заданию требовалось отправить преподавателю публичный ключ и отпечаток ключа по зашифрованному каналу связи.

После чего было получено зашифрованное сообщение.

Чтобы расшифровать сообщение нужно ввести команду:

```
gpg -d encrypted.gpg > decrypted.txt
```

encrypted.gpg – зашифрованный файл

decrypted.txt – расшифрованный итоговый файл

В расшифрованном итоговом файле содержался набор байтов в специфическом формате и для его перевода была написана функция, которая может отобразить байты в виде нормальной строки

```
public class crypt {
    static byte[] fromByteStringToByteArray(String s){
        String[] bytes = s.split("=");
        for(int i = 0;i<bytes.length;++i){
            //bytes[i] = "0x" + bytes[i];
            bytes[i] = bytes[i].toLowerCase();
        }
        byte[] b = new byte[bytes.length];
        for(int i = 0;i<bytes.length; ++i){
            b[i] = (byte)Integer.parseInt(bytes[i],16);
        }
        return b;
    }
    static void exec(){

System.out.print(fromByteStringToString("D0=9F=D0=BE=D0=BB=D1=83=D1=87=D0=B8=
D0=BB") + ' ');

System.out.print(fromByteStringToString("D0=94=D0=BC=D0=B8=D1=82=D1=80=D0=B8=
D0=B9")+ ' ');

System.out.print(fromByteStringToString("D0=9A=D0=BE=D1=80=D0=BE=D1=81=D1=82=
D0=B5=D0=BB=D0=B5=D0=B2")+ ' ');

System.out.print(fromByteStringToString("D0=BF=D0=B8=D1=88=D0=B5=D1=82")+ '
');

System.out.print(fromByteStringToString("D0=94=D0=BE=D0=B1=D1=80=D1=8B=D0=B9"
)+ ' ');
```

```
        System.out.print(fromByteStringToString("D0=B4=D0=B5=D0=BD=D1=8C")+ '
');

System.out.print(fromByteStringToString("D0=BF=D1=80=D0=BE=D1=88=D1=83")+ '
');

System.out.print(fromByteStringToString("D0=BF=D1=80=D0=BE=D1=89=D0=B5=D0=BD=
D0=B8=D1=8F")+ ' ');

System.out.print(fromByteStringToString("D0=BE=D1=82=D0=BF=D1=80=D0=B0=D0=B2=
D0=B8=D0=BB")+ ' ');
        System.out.print(fromByteStringToString("D0=B2=D0=B0=D0=BC")+ ' ');
        System.out.print(fromByteStringToString("D0=BD=D0=B5")+ ' ');
        System.out.print(fromByteStringToString("D1=82=D0=BE=D1=82")+ ' ');

System.out.print(fromByteStringToString("D0=BF=D1=83=D0=B1=D0=BB=D0=B8=D1=87=
D0=BD=D1=8B=D0=B9")+ ' ');
    }
    static String fromByteStringToString(String s){
        return new String(fromByteStringToByteArray(s));
    }
}
```

После полной расшифровки строк, получил раскодированное сообщение в котором была мой переписка с преподавателем и слово – «Получил»

После, преподаватель отправил свой публичный ключ, который нужно было импортировать с помощью следующей команды:

```
gpg --import key.asc
```

key.asc – ключ, который нужно импортировать.

Далее нужно подписать ключ:

```
gpg --sign-key awh@cs.msu.ru
```

Вместо почты, может находится отпечаток ключа, либо его id.

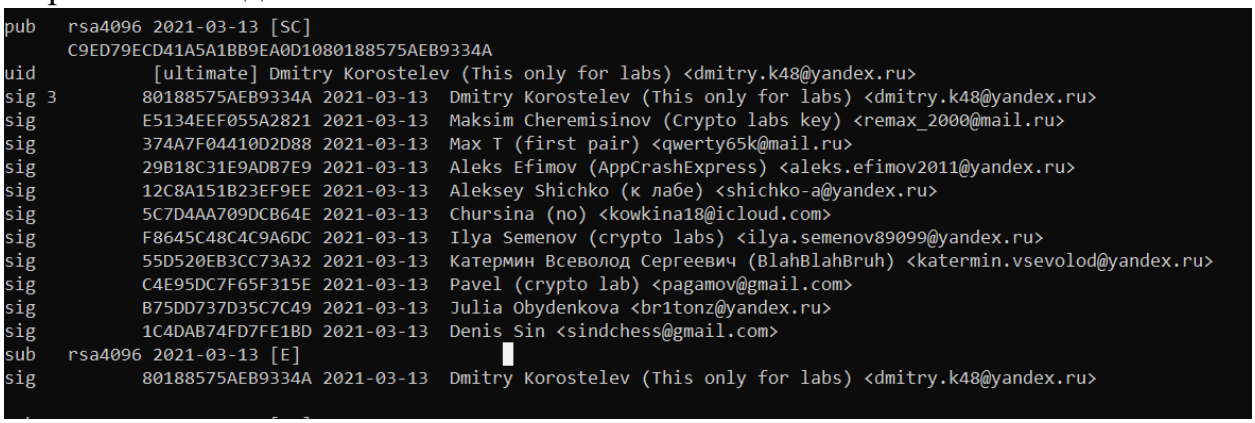
После подписи ключа, подписанный ключ был экспортирован и отправлен преподавателю

По такому алгоритму были собраны подписи и сверены отпечатки ключей одногруппников. Свой ключ с подписями я экспортировал и также отправил преподавателю.

Чтобы посмотреть подписи нужно ввести команду:

```
gpg --list-signatures
```

Скриншот с подписями:



Результат работы программы

Выводы

В ходе выполнения лабораторной работы научился работать с утилитой gpg, сгенерировал свою пару gpg ключей, научился импортировать и экспортировать чужие ключи, а также подписывать своим публичным ключом чужие сертификаты, также смог зашифровать сообщение при помощи публичного ключа собеседника и расшифровать сообщение при помощи своего приватного ключа.