Peter the Great St.Petersburg Polytechnic University

Institute of Computer Science & Technologys

Department of Computer Systems & Software Engineering

**Laboratory №1 Report Discipline:** Information Security

**Theme:** Encryption and Signing with GPG, Gpg4win package

Made by student of group. 13541/1 Smirnov M.I.

(signature)

Lecturer Bogach N.V.

(signature)

“ ” 2017 г.

Saint-Petersburg

2017

**Contents**

[**1.** **Encryption and Signing with GPG, Gpg4win package** 3](#_Toc500720410)

[1.1 Objectives 3](#_Toc500720411)

[1.2 Task 3](#_Toc500720412)

[1.3 Theory 4](#_Toc500720413)

[**2. Work Progress** 5](#_Toc500720414)

[2.1 Study the description and launch graphic tool Kleopatra 5](#_Toc500720415)

[2.2 Create a key pair with OpenPGP (File *→* New Certiﬁcate) 6](#_Toc500720416)

[2.3 Export Certiﬁcate (File *→* Export Certiﬁcate) 8](#_Toc500720417)

[2.4 Sign/Encrypt Files (File *→* Sign/Encrypt Files) 9](#_Toc500720418)

[2.5 Load other users certiﬁcate, import, sign and verify it 12](#_Toc500720419)

[2.6 Using your partner certiﬁcate encrypt, sign and send her a ﬁle 14](#_Toc500720420)

[2.7 Accept, check and decrypt a ﬁle from your partner 14](#_Toc500720421)

[2.8 Following the instructions in GNU Privacy handbook, play with gpg by CLI, i.e. without graphic tool. 15](#_Toc500720422)

[**Conclusion** 18](#_Toc500720423)

# **Encryption and Signing with GPG, Gpg4win package**

## 1.1 Objectives

After completing this module you will be able to:

* + 1. Create digital certiﬁcates;
    2. Encrypt ﬁles;
    3. Sign ﬁles.

## 1.2 Task

* + 1. Study the description and launch graphic tool Kleopatra;
    2. Create a key pair with OpenPGP (File *→* New Certiﬁcate);
    3. Export Certiﬁcate (File *→* Export Certiﬁcate);
    4. Sign/Encrypt Files (File *→* Sign/Encrypt Files);
    5. Load other users certiﬁcates;
    6. Import a certiﬁcate, sign it;
    7. Verify the signature;
    8. Using your partner certiﬁcate encrypt, sign and send her a ﬁle;
    9. Accept, check and decrypt a ﬁle from your partner;
    10. Following the instructions in GNU Privacy handbook (a link is in REFERENCE

section in a bottom of this module) play with gpg by CLI, i.e. without graphic tool.

## 1.3 Theory

**GnuPG**

GnuPG is a complete and free imple- mentation of the OpenPGP standard as de- ﬁned by RFC4880 (also known as PGP). GnuPG allows to encrypt and sign your data and communication, features a ver- satile key management system as well as access modules for all kinds of public key directories. GnuPG, also known as GPG, is a command line tool with features for easy integration with other applications.

Features:

* Full alternative to PGP;
* Does not use proprietary algorithms;
* Distributed under the GNU General Public License;
* Expansion and authentication of e-mail messages created with PGP 5, 6 and 7;
* Support for electronic signature using ElGamal, DSA, RSA and hash functions MD5, SHA- 1, SHA-2, RIPE-MD-160 and TIGER;
* Work with asymmetric encryption ElGamal and RSA (key length from 1024 to 4096 bits);
* Support for block symmetric encryption algorithms AES, CAST5, 3DES, Twoﬁsh;
* Blowﬁsh, Camellia, and IDEA using a plug-in;
* Support for compression algorithms: ZIP, ZLIB, BZIP2.

**Gpg4win**

Gpg4win is a Windows version of GnuPG featuring a context menu tool, a crypto man- ager, and an Outlook plugin to send and re- ceive standard PGP/MIME mails.

It is maintained by the developers of GnuPG and the software included with Gpg4win are Free Software.

# **2. Work Progress**

## 2.1 Study the description and launch graphic tool Kleopatra

Kleopatra is a certiﬁcate manager and GUI for GnuPG. The software stores OpenPGP cer- tiﬁcates and keys. It is available for Windows and Linux.

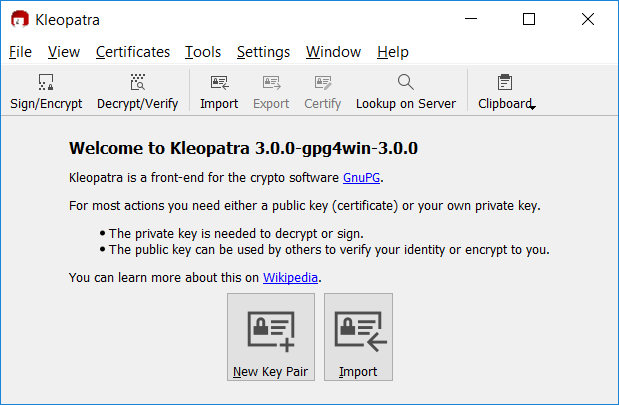


Figure 2.1: Main window

On the main window, kleopatra suggests creating a new key pair or import it.

## 2.2 Create a key pair with OpenPGP (File *→* New Certiﬁcate)

In the following dialog, you can choose format: OpenPGP(PGP/MIME) or X.509 (S/MIME).

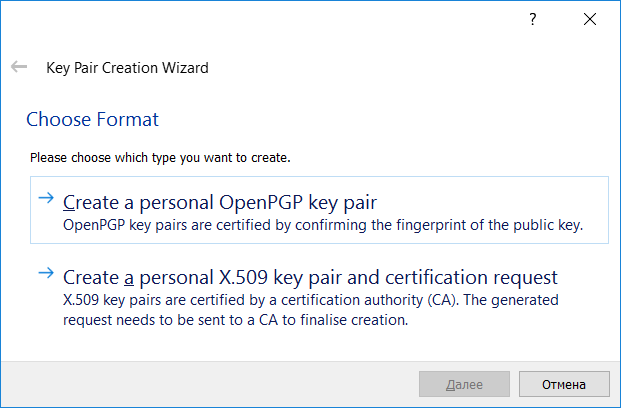
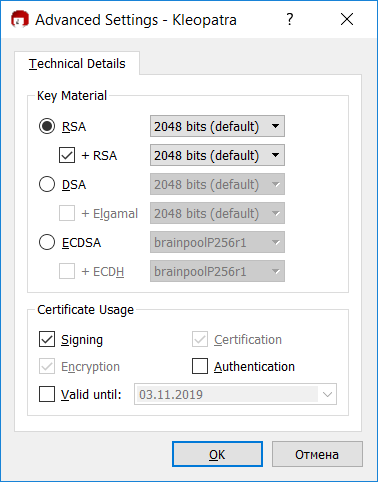
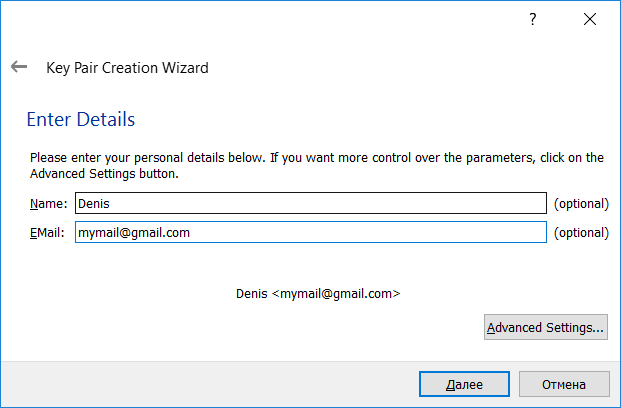


Figure 2.2: Key Pair Creation windows

After that, it is suggested to enter a name and mail, and set up additional settings, if needed.



* + 1. Input of name and email (b) Advanced options

Figure 2.3: Certiﬁcate options

after click an next button, we see all inputed params.

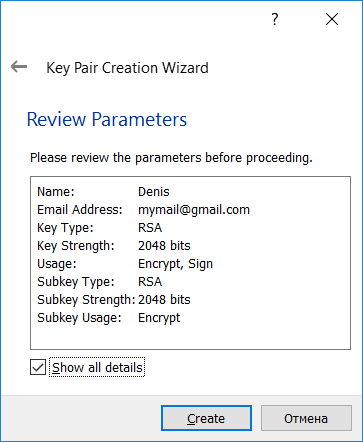


Figure 2.4: Result params

Now need to type password.

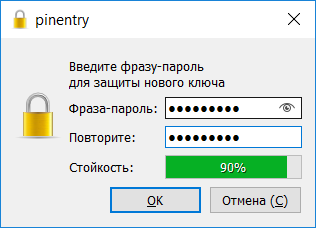


Figure 2.5: Creating passphrase

And after it, we see window with message about successfuly created key pair.

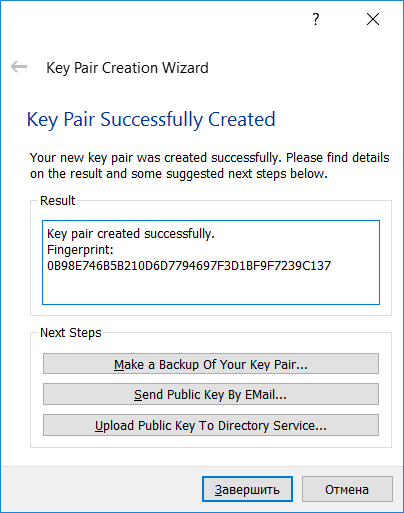


Figure 2.6: Result of creating

In result box we see ﬁngerprint. Fingerprint required to identify the certiﬁcate and its owner.

## 2.3 Export Certiﬁcate (File *→* Export Certiﬁcate)

To export certiﬁcate, right click at certiﬁcate, than choose Export... or just Ctrl+E.

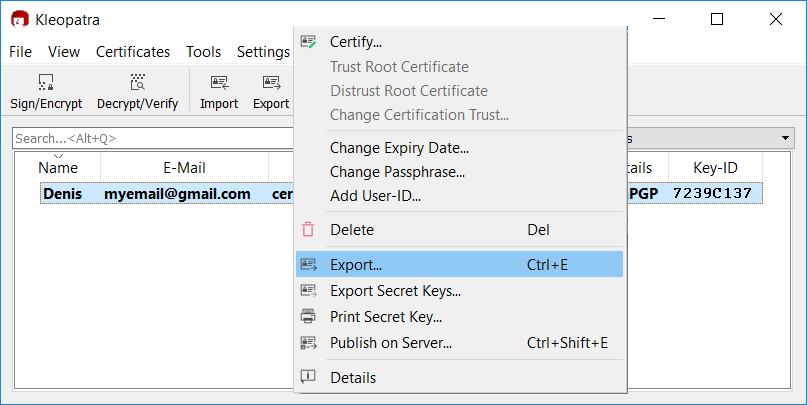


Figure 2.7: Expoting certiﬁcate

File can be exported in theese ﬁle formats:

* asc
* gpg
* pgp

If open it in text editor, we see following:

1 *−−−−−*BEGIN PGP PUBLIC KEY BLOCK*−−−−−*

2

1. mQENBFn8oe0BCAC0RJQ7bGdoaSX863K2NWbpx8lV5yJSYttNcoSW3Ril6SNMsrs+
2. mER1/ dglsfiVNtQvehv89Va6HUlGp7EkM9saOG2ve9heBwZXrD4Xt0XxZaxLhb0C
3. NKxP2Q0PQlJbx /20 PxVFkvMYOIThK2ySbPY0OdbXgC3QldfhQAL+BLFZC1yuWZk5
4. LVQuO8QAK3YF/3 pnhZkX92DACmVOUVncfunObOVVCSQ3pl1jSL4xHkVVUyTSuyZU

7 +P1kX0P2NjsA8hDk+k0RfFjocUQ88wQaVq4oFV88GyWDgu1ngz1F8IR7OiuljRBW

1. wuFaP+TyI285FlpiqYyaMASiZtEQWOVJ / QHDABEBAAG0GURlbmlzIDxteWVtYWls
2. QGdtYWlsLmNvbT6JAU4EEwEIADgWIQQLmOdGtbIQ1td5Rpfz0b+fcjnBNwUCWfyh
3. 7QIbAwULCQgHAgYVCAkKCwIEFgIDAQIeAQIXgAAKCRDz0b+fcjnBN7gGB /43 ea3H
4. Bp57KbHcWKjh932qB0yXbOzrN / s Z f l E e +1/ t S L l 1 +fOKTh0IlNA5yWrs / YZTvts9
5. FxdicUztUi7BjLKDCB1lkQVkUuKWgaCXNEoqxvTcy2aMRYAtEH+Bj9uEaEOVJt1R
6. v3os3IJeG26dJUpNlYpFNhkXhY8TTH0xLb7lgwBT8D7NMSQLhIDH3Mq7Oifr5Pgc
7. 167 tkstjBUAYZzwSY5uqaO5+ukMJ4KvHjDnVjSlraEZkxIrMv0W34canp7 / Hk179
8. U37NXE / f2JH4Kdy / dq / zkF3XyyAPKEoysSuy9krxbKnpuLHtxIuNxdm4 / hG0HlaA
9. Ufw6bRV7Xi1tzV8ruQENBFn8oe0BCADk0qPWiQtul8CibriAL0Jv8tMRqt+oveGg
10. nSi7ke2nCjkrTPcTGl3NBd8zyR7IkWOi /9 FbRiFVJtd5QQSzdV5oWVOFkZZFLX8U
11. klaRvtuBibbZT6brJTkxI9Rw2XTmQVZOw6yJnO57LXMU3rFDzP5PoxpK8mQKEs6Z
12. 3vZG7Gd5FYEGo+Ts+ TciZ3oLvn0p94 / CXwPRt4ri+kRHD0jGO0Vfe / tx r 3 Z J S S h t
13. m+Zw95oreZv7TWeS / nyLTvtDVVad2+gTrOVuIduZNT1sV3Kmbq15IqOFJqQ392KW
14. sO1DECPa / ar8kxkTFf16wCvrjp22pbpTBO8hWW5HkFNVoA6obfH5ABEBAAGJATYE
15. GAEIACAWIQQLmOdGtbIQ1td5Rpfz0b+fcjnBNwUCWfyh7QIbDAAKCRDz0b+ f c j n B
16. N0cUB/429 a t3 b H A fK 2 d w 7 A k C i F z I r j j Z S +bW8zJBSHzrTrtVplb5MKHPSEvkRQFI
17. 6WolkQD3bqhefNY / rzcby2q6EiME1a / / CDOzOEFRkKTvQJFsk66SHd5t4tgTb5s9
18. BbFzh9fAasygpyAP94+MMTQpzEcaqvf+XxUE4b9vQDoEAjMUPqYQxILwKYURr8v9
19. poFcWlT5iyerl7SQQfCKofzpnJ8iR0zuTNG +vhhunRVHfSbLwYLCOb9A7OTte7nX
20. VdGnGbTxI0RcZhFKovvxQkfLefJZT2uEIJpGLI / pAvRu8B52OpP / v L F fq E S P ss 6 l
21. cOOZvHGi2deYH4uLonHgDZe8FStK

29 = RJzf

30 *−−−−−*END PGP PUBLIC KEY BLOCK*−−−−−*

Listing 2.1: 0B98E746B5B210D6D7794697F3D1BF9F7239C137.asc

With this certiﬁcate we can sign and encrypt ﬁles.

## 2.4 Sign/Encrypt Files (File *→* Sign/Encrypt Files)

Let’s create new ﬁle, with following text:

1

Listing 2.2: test.txt

My s e c r e t message

And choose this ﬁle to encrypt

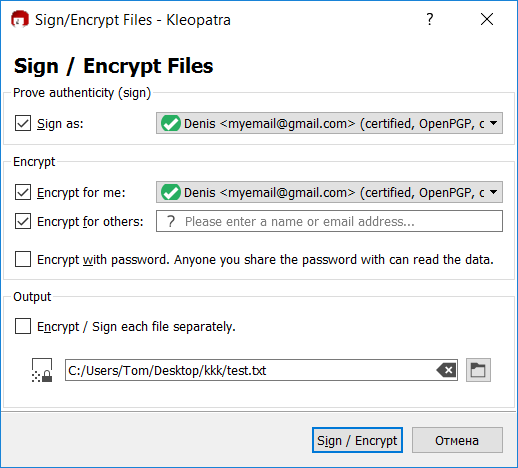


Figure 2.8: Sign/Encrypt Files

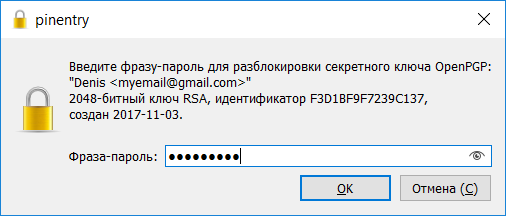


Figure 2.9: Password input

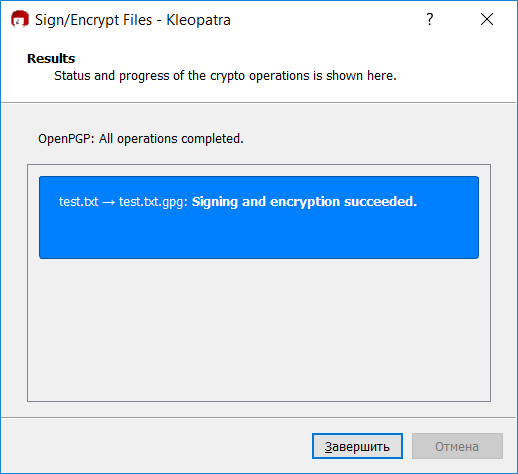


Figure 2.10: Signing and encryption succeeded

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 8501 | 0c03 | 8c02 | 7204 | 1 e 1 f | 91b2 | 0108 | 0087 |
| 2 | 114c | a623 | 1617 | 88db | 2 f4 7 | 0a91 | 288c | 0 c a f |
| 3 | 2b8c | 59 f1 | 3eb1 | f47d | d62e | 0577 | c 4 a f | f 4 f 0 |
| 4 | b846 | 3 f5 5 | 11b9 | 1d68 | a4c7 | 92e8 | 7a22 | d8f0 |
| 5 | ceb9 | 59e2 | d342 | 062a | 3e34 | f 2 f 1 | 9268 | 24e1 |
| 6 | 71e5 | 4c13 | 2c00 | 82c5 | 544b | b7ca | e74c | 78d6 |
| 7 | 1879 | 4913 | 6dcc | 6a87 | f 1 8 f | 014 f | 5088 | d3bf |
| 8 | cd10 | 30ea | 6eda | cd04 | d3f2 | 8e31 | 6364 | b61d |
| 9 | 6aa3 | 95 b f | 4c37 | 2890 | 47c1 | 9702 | 5802 | 42a5 |
| 10 | 6be4 | a291 | 8cd0 | 65c1 | b9d4 | a673 | 5ced | f965 |
| 11 | 817d | a47d | 5168 | 0 f4 1 | 4711 | 75 f8 | 377c | a8d7 |
| 12 | d04a | f1a9 | 7a94 | 80 fb | a2c9 | bb31 | f5a4 | 7743 |
| 13 | e723 | 0e63 | 099c | 6dd3 | d57e | 4618 | e689 | 4728 |
| 14 | 5b9c | 6c9c | e914 | 4a5d | c7b2 | 7c8d | db4b | de10 |
| 15 | c192 | 7d18 | eace | 1c1a | 8daa | f1d1 | a8e6 | 6395 |
| 16 | a9da | a087 | 9421 | 4 b 9 f | 0500 | 0939 | d8ca | f4e 0 |
| 17 | 2781 | a8b4 | aa6c | 458b | 1a7d | f126 | b5fc | f f d 2 |
| 18 | c0d2 | 016c | a6e5 | 3a13 | 3b90 | 7733 | 2eb0 | cea9 |
| 19 | 695 f | f43a | b0d3 | 7979 | 4e57 | d94d | 8eee | 2 f 8 f |
| 20 | 387a | bd77 | d4a6 | d5e2 | 11 f4 | 60bb | f 4 f 4 | 7b7d |
| 21 | ee5b | be68 | 2496 | 1d76 | dc74 | c239 | 4 b 4 f | 0eea |
| 22 | 31c5 | c477 | 704a | 23eb | 1 d 9 f | 4922 | ec93 | f48e |
| 23 | 92c5 | 4242 | 7794 | e fd 1 | bf6b | 898d | 0 a 1 f | 378 f |
| 24 | 0574 | 804 f | 9815 | e9ac | 91cd | d ef0 | 08db | 8 fa 9 |
| 25 | 80bc | 0b8d | 4 f0 5 | abc1 | 4b05 | 200a | 1437 | d179 |

Now let’s see what we got:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 26 | 345d | 1 f9 8 | ea26 | e559 | f1a8 | 7da8 | d630 | 9429 |
| 27 | 645a | 4057 | 90e6 | 1 fe 6 | 8096 | 0 f1 3 | e2a1 | 16ac |
| 28 | bca2 | c61a | 5 f5 6 | 16db | c018 | a c a f | a94a | 4 c f1 |
| 29 | f834 | 05c5 | 7d41 | 0042 | 4ab0 | a653 | e f94 | 7c76 |
| 30 | a25c | 1d45 | 03d2 | d178 | a f2 e | 0ce1 | c2c3 | e190 |
| 31 | 0d52 | 5 a fc | e83d | fc e e | d66a | 54e6 | a f4a | 14 f9 |
| 32 | 6556 | 39dd | ad9b | 8181 | 05 c f | 2 d f9 | 9 e f0 | 6b0d |
| 33 | 13e4 | b170 | 696 f | 82c7 | 25b6 | 7d0c | f8e d | 30e9 |
| 34 | 16ba | 79a5 | e1d2 | 8d0e | 80 f4 | 6e04 | 57d9 | 0604 |
| 35 | 9 f4 7 | c010 | ccb1 | cc61 | dbf4 | b4bc | 0019 | a0e8 |
| 36 | fc e a | d6c2 | 2219 | 8c65 | c294 | 9280 | d755 | 721b |
| 37 | 907d | 0b3e | fdb5 | bf14 | bde6 | 5ac6 | 6c95 | 9a30 |
| 38 | b82b | 9 fd 6 | d8e8 | 4a16 | 89bc | 052e | 9e42 | 9 f f f |
| 39 | 74b5 | b5aa | 6aeb | 0d94 | 00 f6 | ae27 | e937 | 1 e 2 f |
| 40 | 4896 | c303 | 9725 | a4f3 | 1 f 8 f | 5a4c | c0db | 571b |
| 41 | d8b1 | daf9 | f4d6 | 245a | 03a3 | 6073 | 70a3 | ab10 |
| 42 | a11e | 736c | 83 d f | 0acc | 7ba7 | 49ca | 5493 | cd5d |
| 43 | 113 f | 4c8b |  |  |  |  |  |  |

As expected the message was encrypted.

Listing 2.3: test.txt.gpg

## 2.5 Load other users certiﬁcate, import, sign and verify it

Click at import button and choose other user certiﬁcate.

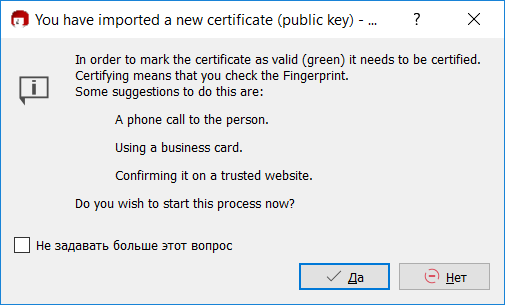
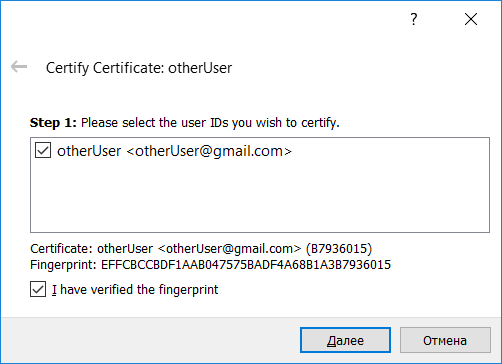


Figure 2.11: Importing new certiﬁcate

We see his name, email and ﬁngerprint.



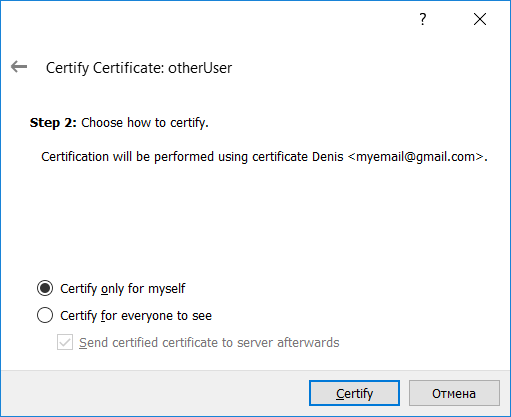
Figure 2.12: Certiﬁcate information Now we choose for whom we certify this.

Figure 2.13: Type of certiﬁcation

As result, we see fully trusted certiﬁcate.

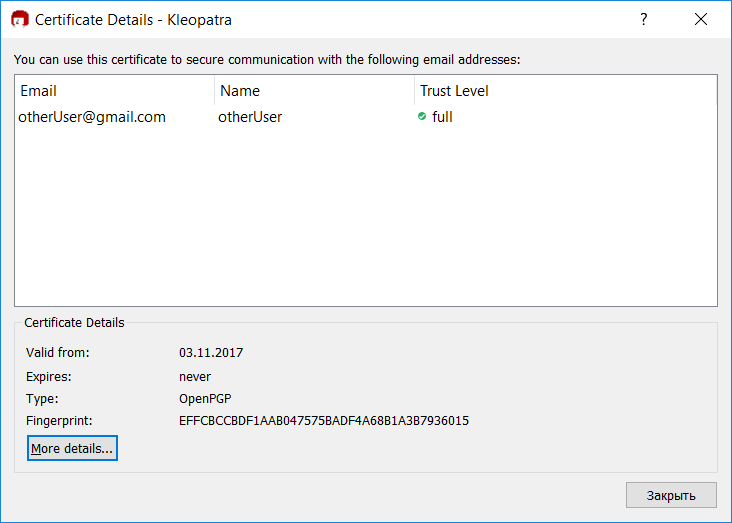


Figure 2.14: Result

## 2.6 Using your partner certiﬁcate encrypt, sign and send her a ﬁle

## 2.7 Accept, check and decrypt a ﬁle from your partner

Let’s decrypt ﬁle(test.txt.gpg) from paragraph 2.4. Click File *→* Decrypt/Verify...

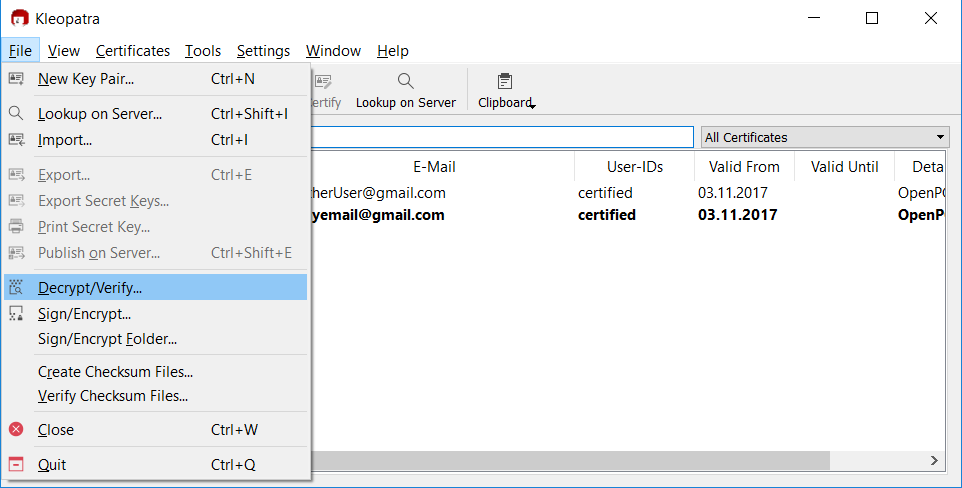


Figure 2.15: Decrypt/Verify

After choosing ﬁle, kleopatra using known certiﬁcate decrypt this, and than we got following message.

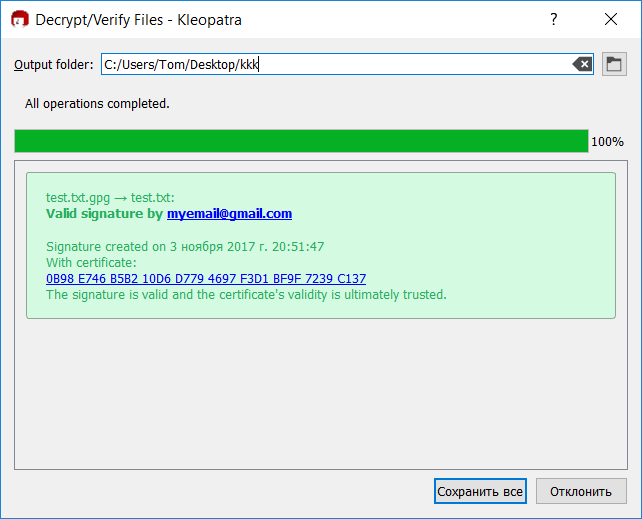


Figure 2.16: Successful decrypt

## 2.8 Following the instructions in GNU Privacy handbook, play with gpg by CLI, i.e. without graphic tool.

In this part of the report, i will use kali linux as virtual machine. First check version of gpg, and then generate key, using following command:

gpg –gen-key

Below is a full action log.

1

gpg ( GnuPG ) 2 . 2 . 0 l i b g c r y p t 1 . 7 . 9

r o o t @ k a l i : ~/ Desktop / t e s t F o l d e r 2 *# gpg −−v e r s i o n*

C o p y r i g h t ( C ) 2017 F r e e S o ftw a r e Foundation , I n c .

L i c e n s e GPLv3 + : GNU GPL v e r s i o n 3 o r l a t e r < h t t p s : / / gnu . org /

T h i s i s f r e e s o ft w a r e : you a r e f r e e There i s NO WARRANTY , t o th e e x t e n t

*‹→* l i c e n s e s / g p l . html >

t o change and r e d i s t r i b u t e i t . p e r m i t t e d by law .

Home : / r o o t / . gnupg Supported a l g o r i t h m s :

Pubkey : C i p h e r :

RSA , ELG , DSA , ECDH , ECDSA ,

EDDSA

IDEA , 3DES , CAST5 , BLOWFISH , AES , AES192 , AES256 , TWOFISH , CAMELLIA128 , CAMELLIA192 , CAMELLIA256

Hash : SHA1 , RIPEMD160 , SHA256 , SHA384 , SHA512 , SHA224

Compression : Uncompressed , ZIP , ZLIB , BZIP2

2

3

4

5

6

7

8

9

10

11

12

13

14

15

1. r o o t @ k a l i : ~/ Desktop / t e s t F o l d e r 2 *# gpg gen key*

*−− −*

1. gpg ( GnuPG ) 2 . 2 . 0 ; C o p y r i g h t ( C ) 2017 F r e e S o ftw a r e Foundation ,

*‹* I n c .

*→*

1. T h i s i s f r e e s o ft w a r e : you a r e f r e e t o change and r e d i s t r i b u t e i t .
2. There i s NO WARRANTY , t o th e e x t e n t p e r m i t t e d by law . 20

21 Note : Use ” gpg *−−* f u l l *−*g e n e r a te *−*key ” f o r a f u l l f e a t u r e d key

*‹→* g e n e r a t i o n d i a l o g .

22

23 GnuPG needs t o c o n s t r u c t a u s e r I D t o i d e n t i f y y o u r key . 24

1. R e a l name : D e n i s
2. E ma i l address : myemail@gmail . com
3. You s e l e c t e d t h i s USER I D :

*−*

1. ” D e n i s <myemail@gmail . com> ” 29
2. Change ( N) ame , ( E ) ma il , o r ( O ) kay / ( Q ) u i t ? O
3. We need t o g e n e r a t e a l o t o f random b y t e s . I t i s a good i d e a t o

*‹* perform

*→*

1. some o t h e r a c t i o n ( ty p e on t h e keyboard , move th e mouse , u t i l i z e

*‹* th e

*→*

1. d i s k s ) d u r i n g th e prime g e n e r a t i o n ; t h i s g i v e s th e random number
2. g e n e r a t o r a b e t t e r chance t o g a i n enough e n t r o p y .

Listing 2.4: terminal log

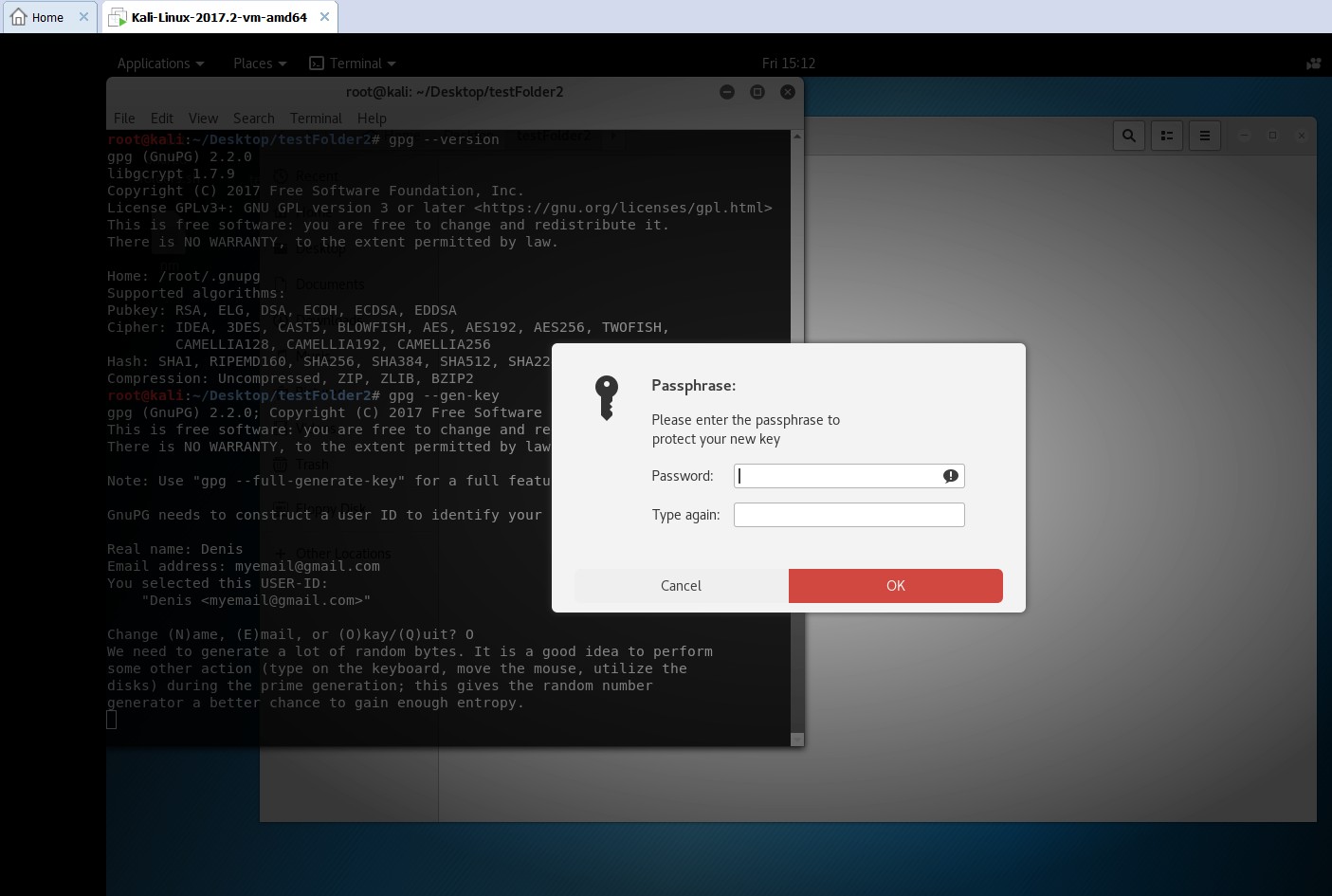


Figure 2.17: Password input

After some action’s with mouse moving, we see following:

1

2

/ r o o t / . gnupg / t r u s t d b . gpg : t r u s t d b c r e a t e d

key 0F3B233F4F373F3A marked as u l t i m a t e l y t r u s t e d

gpg : gpg : gpg : gpg :

3

4

d i r e c t o r y ’ / r o o t / . gnupg / openpgp*−*r e v o c s . d ’ c r e a t e d

5

*‹→* . d / AF3AA2D6F72A7B38B5ED151B0F3B233F4F373F3A . r e v ’

r e v o c a t i o n c e r t i f i c a t e s t o r e d as ’ / r o o t / . gnupg / openpgp*−*r e v o c s

6

p u b l i c and s e c r e t key c r e a t e d and s i g n e d .

7

8

rsa3072 2017*−*11*−*03 [ SC ] [ e x p i r e s : 2019*−*11*−*03]

pub

9

AF3AA2D6F72A7B38B5ED151B0F3B233F4F373F3A

10

u i d D e n i s <myemail@gmail . com> sub rsa3072 2017*−*11*−*03 [ E ] [ e x p i r e s : 2019*−*11*−*03]

11

Listing 2.5: successful key generation

My key was successfully created, now to export it, need to type following command:

1

*‹→ e x p o r t myemail@gmail . com*

r o o t @ k a l i : ~/ Desktop / t e s t F o l d e r 2 *# gpg −−armor −−o u t p u t DENIS . asc −−*

Listing 2.6: export

Key was identiﬁed by email, and now in current directory we have exported certiﬁcate ﬁle.

1

r o o t @ k a l i : ~/ Desktop / t e s t F o l d e r 2 *# l s − l*

2

t o t a l 4

*−*rw*−*r *−−*r *−−* 1 r o o t r o o t 2444 Nov

3

Listing 2.7: directory

3 1 5 : 2 9 DENIS . asc

For import we can use following command:

gpg –import someCert.asc

# **Conclusion**

As result in this report i learned how to use encryption tool’s with GUI(kleopatra) and with console(pgp). Kleopatra is easy to use, because of the intuitive interface. In the console it was more diﬃcult, but using The GNU Privacy Handbook helped me to understand how it work’s. Encryption is extremely important in the modern world, especially when transferring over the Internet important ﬁles or data. So, using of encryption in everything(web, email, text messanger etc) is considered normal.