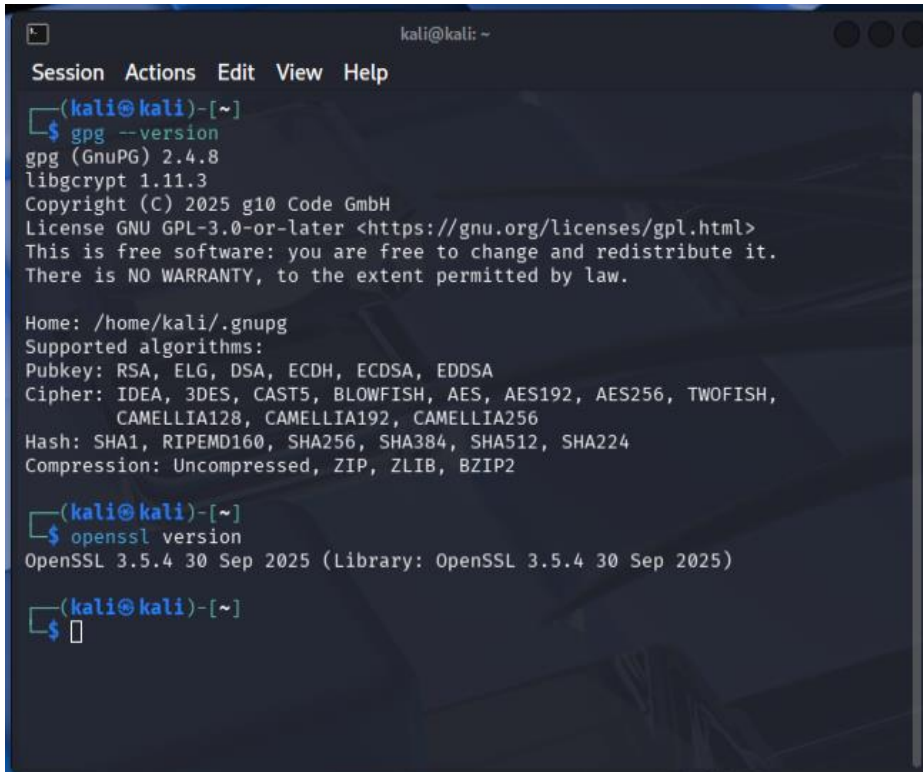


## 1. Sprawdzenie środowiska

`gpg --version`

`openssl version`



```
kali@kali: ~  
Session Actions Edit View Help  
(kali@kali)-[~]  
$ gpg --version  
gpg (GnuPG) 2.4.8  
libgcrypt 1.11.3  
Copyright (C) 2025 g10 Code GmbH  
License GNU GPL-3.0-or-later <https://gnu.org/licenses/gpl.html>  
This is free software: you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
  
Home: /home/kali/.gnupg  
Supported algorithms:  
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA  
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,  
CAMELLIA128, CAMELLIA192, CAMELLIA256  
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224  
Compression: Uncompressed, ZIP, ZLIB, BZIP2  
  
(kali@kali)-[~]  
$ openssl version  
OpenSSL 3.5.4 30 Sep 2025 (Library: OpenSSL 3.5.4 30 Sep 2025)  
  
(kali@kali)-[~]  
$
```

## 2. Generowanie kluczy

`gpg --full-generate-key`

```

(kali@kali)-[~]
$ gpg --full-generate-key
gpg (GnuPG) 2.4.8; Copyright (C) 2025 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: keybox '/home/kali/.gnupg/pubring.kbx' created
Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
  (10) ECC (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 3072
Requested keysize is 3072 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: Marcel
Email address: goska@gmail.com
Comment: comment
You selected this USER-ID:
    "Marcel (comment) <goska@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/kali/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/home/kali/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/kali/.gnupg/openpgp-revocs.d/8ECA72463FCA56E2C84C99826D57053D80290030.rev'
public and secret key created and signed.

pub   rsa3072 2026-01-21 [SC]
      8ECA72463FCA56E2C84C99826D57053D80290030
uid           Marcel (comment) <goska@gmail.com>
sub   rsa3072 2026-01-21 [E]

(kali@kali)-[~]
$

```

### 3. Sprawdzenie kluczy

gpg --list-keys

gpg --list-secret-keys

```
(kali㉿kali)-[~]
$ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
/home/kali/.gnupg/pubring.kbx

pub   rsa3072 2026-01-21 [SC]
      8ECA72463FCA56E2C84C99826D57053D80290030
uid           [ultimate] Marcel (comment) <goska@gmail.com>
sub   rsa3072 2026-01-21 [E]

(kali㉿kali)-[~]
$
```

```
(kali㉿kali)-[~]
$ gpg --list-secret-keys
/home/kali/.gnupg/pubring.kbx

sec   rsa3072 2026-01-21 [SC]
      8ECA72463FCA56E2C84C99826D57053D80290030
uid           [ultimate] Marcel (comment) <goska@gmail.com>
ssb   rsa3072 2026-01-21 [E]

(kali㉿kali)-[~]
$
```

#### 4. Utworzenie pliku testowego

echo "To jest tajny plik projektu kryptograficznego" > tajne.txt

```
(kali㉿kali)-[~/Documents/Projekt]
$ echo "To jest tajny plik projektu kryptograficznego" > tajne.txt
```

#### 5. Szyfrowanie pliku (kluczem publicznym)

gpg --encrypt --recipient twoj@email.pl tajne.txt

```
(kali㉿kali)-[~/Documents/Projekt]
$ gpg --encrypt --recipient goska@gmail.com tajne.txt

(kali㉿kali)-[~/Documents/Projekt]
$ cat tajne.txt.gpg
***;Y5?A
**!yH****8{****WHv+S!7**=jrcK*****,*c~9yf****d**y****)U**7I5*1*F**
3**#****]*z*Dg**j****z**:>
                                4`g**I|p+***+**H***o**w*I9*h\ys5L***r*gA**
{**{{_**A~XN*6 *2S**X*Ee3*\***Wx*****#j*4m**

(kali㉿kali)-[~/Documents/Projekt]
$
```

6. Próba odczytu bez odszyfrowania

cat tajne.txt.gpg

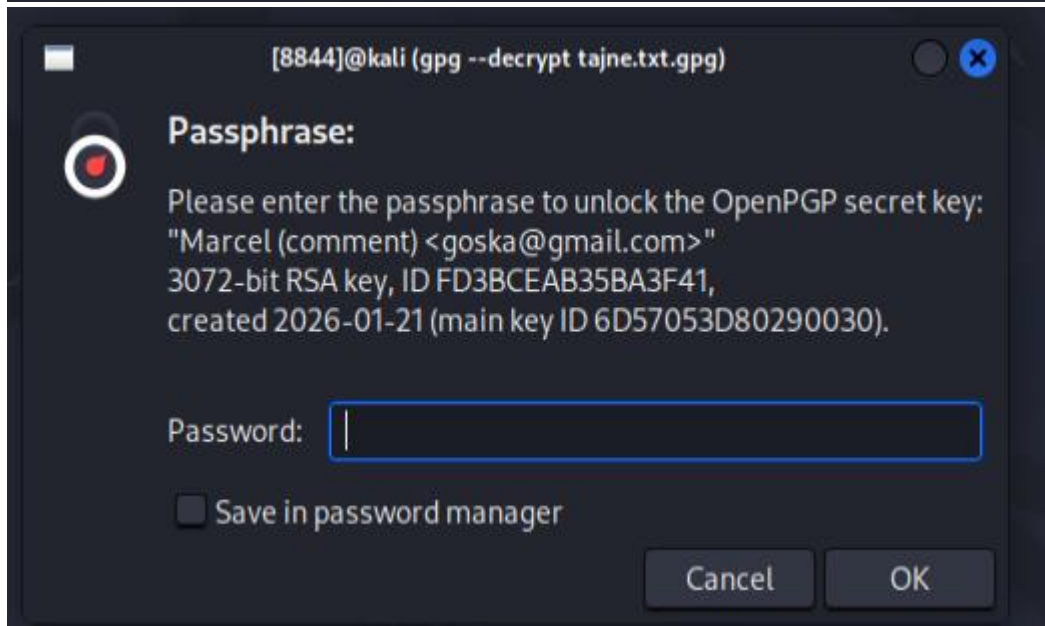
```
(kali㉿kali)-[~/Documents/Projekt]
$ cat tajne.txt.gpg
***;Y5?A
**!yH****8{****WHv+S!7**=jrcK*****,*c~9yf****d**y****)U**7I5*1*F**
3**#****]*z*Dg**j****z**:>
                                4`g**I|p+***+**H***o**w*I9*h\ys5L***r*gA**
{**{{_**A~XN*6 *2S**X*Ee3*\***Wx*****#j*4m**
```

7. Deszyfrowanie pliku

gpg --decrypt tajne.txt.gpg

```
(kali㉿kali)-[~/Documents/Projekt]
$ gpg --decrypt tajne.txt.gpg
gpg: encrypted with rsa3072 key, ID FD3BCEAB35BA3F41, created 2026-01-21
"Marcel (comment) <goska@gmail.com>"

```



```
(kali㉿kali)-[~/Documents/Projekt]
$ gpg --decrypt tajne.txt.gpg
gpg: encrypted with rsa3072 key, ID FD3BCEAB35BA3F41, created 2026-01-21
"Marcel (comment) <goska@gmail.com>"
To jest tajny plik projektu kryptograficznego

(kali㉿kali)-[~/Documents/Projekt]
$ 

```

## 8. Podpisanie Pliku

```
gpg --sign tajne.txt
```

Jawny podpis

```
gpg --clearsign tajne.txt
```

```
(kali㉿kali)-[~/Documents/Projekt]
$ gpg --sign tajne.txt
File 'tajne.txt.gpg' exists. Overwrite? (y/N) y

(kali㉿kali)-[~/Documents/Projekt]
$ gpg --clearsign tajne.txt

```

## 9. Weryfikacja podpisu

```
gpg --verify tajne.txt.gpg
```

```
(kali㉿kali)-[~/Documents/Projekt]
$ gpg --verify tajne.txt.gpg
gpg: Signature made Wed 21 Jan 2026 12:36:53 PM EST
gpg:          using RSA key 8ECA72463FCA56E2C84C99826D57053D80290030
gpg: Good signature from "Marcel (comment) <goska@gmail.com>" [ultimate]
```

## 10. Test integralności

echo "Zmiana" >> tajne.txt

gpg --verify tajne.txt.gpg

Podpis cyfrowy gwarantuje integralność danych – nawet minimalna zmiana pliku powoduje niepowodzenie weryfikacji podpisu.

