


## Raport 2: Podstawy praktycznej kryptografii – Linux, gpg

### 1.1 Generator liczb losowych

Wpisuję komendę, która losuje ciąg znaków w systemie szesnastkowym.

A terminal window with a dark purple background. The prompt is 'sansforensics@siftworkstation: ~'. The command entered is '\$ xxd -l 16 -p /dev/random'. The output is '7ba1fd205d240cd8b8ea66cbf6870a89'. The prompt is repeated, and the command '\$' is entered again.

```
sansforensics@siftworkstation: ~  
$ xxd -l 16 -p /dev/random  
7ba1fd205d240cd8b8ea66cbf6870a89  
sansforensics@siftworkstation: ~  
$
```

**/dev/random** – wirtualne urządzenie w systemach operacyjnych z rodziny Unix, pełniące funkcję generatora losowych liczb z losowością pochodzącą ze sterowników urządzeń i innych źródeł. Przy odczycie z /dev/random wygenerowane zostaną przypadkowe bajty. /dev/random jest odpowiednie w przypadku wymaganej wysokiej przypadkowości danych oraz odporności na przewidywalność kolejnych danych, np. przy tworzeniu kluczy kryptograficznych.

Znaczenie znaków :

l - zatrzymaj się po wpisaniu <len> oktetów.

p | -ps | -postscript - wyjście w postscriptowym ciągłym stylu zrzutu heksowego

16 – tyle wykonuje się powtórzeń

### 1.2 Hashowanie plików

*Dzięki temu ustalisz, jakie programy obliczające hashe masz zainstalowane. Zastanów się, które z nich obliczają znane Ci hashe. Za pomocą tych programów oblicz hashe dowolnego obrazka (lub innego pliku binarnego).*

Wpisuję komendę:

```
find /usr -name '*sum'
```

```
sansforensics@siftworkstation: ~
$ find /usr -name '*sum'
/usr/bin/sha512sum
/usr/bin/b2sum
/usr/bin/cksum
/usr/bin/sum
/usr/bin/sha384sum
/usr/bin/md5sum
/usr/bin/sha256sum
/usr/bin/sha1sum
/usr/bin/sha224sum
/usr/bin/shasum
/usr/share/openssh/sshd_config.md5sum
/usr/share/unattended-upgrades/50unattended-upgrades.md5sum
/usr/share/gdm/greeter.dconf-defaults.md5sum
/usr/share/ufw/user6.rules.md5sum
/usr/share/ufw/user.rules.md5sum
/usr/share/ufw/before.rules.md5sum
/usr/share/ufw/before6.rules.md5sum
/usr/share/ufw/after.rules.md5sum
/usr/share/ufw/after6.rules.md5sum
/usr/share/grub/default/grub.md5sum
sansforensics@siftworkstation: ~
```

Komendy i wywołane hashe:

```
sansforensics@siftworkstation: ~
$ cd Desktop
sansforensics@siftworkstation: ~/Desktop
$ ls
cases
DFIR-Smartphone-Forensics-Poster.pdf
Hex-File-Regex-Cheatsheet.pdf
Hunt-Evil.pdf
iOS-3rd-Party-Apps-Poster.pdf
mount_points
Network-Forensics-Poster.pdf
Poster_Threat-Intelligence-Consumption.pdf
SIFT-Cheatsheet.pdf
SIFT-REMnux-Poster.pdf
SQLite-Pocket-Reference.pdf
Windows-Forensics-Poster.pdf
Windows-to-Unix-Cheatsheet.pdf
Zimmerman-Tools-Poster.pdf
sansforensics@siftworkstation: ~/Desktop
$ shasum SIFT-Cheatsheet.pdf
8d6a75cc0b147ee559599f79ff7d87b2daede825  SIFT-Cheatsheet.pdf
sansforensics@siftworkstation: ~/Desktop
$ sha256sum SIFT-Cheatsheet.pdf
8fda96c8f7bc32844b843dd3d95e793316e23b765b440b240681d3bc9724ad5c  SIFT-Cheatsheet.pdf
sansforensics@siftworkstation: ~/Desktop
$ sha224sum SIFT-Cheatsheet.pdf
423d3df678b2fb440bf3bcbcff9da47d75ba4de9fd502df85a7d83db  SIFT-Cheatsheet.pdf
sansforensics@siftworkstation: ~/Desktop
$ cksum SIFT-Cheatsheet.pdf
3434992663 338715 SIFT-Cheatsheet.pdf
sansforensics@siftworkstation: ~/Desktop
```

```

$ sha1sum SIFT-Cheatsheet.pdf
8d6a75cc0b147ee559599f79ff7d87b2daede825  SIFT-Cheatsheet.pdf
sansforensics@siftworkstation: ~/Desktop
$ sha512sum SIFT-Cheatsheet.pdf
651d231fc49ad2dcea6c0f492c44c53c5d19a00adcafb823aaa6ec3ccf1a830a146686c51956078854189d630b5cd5f7d3f5486202d37882823c690374d04b4a  SIFT-Cheatsheet.pdf
sansforensics@siftworkstation: ~/Desktop
$ sha384sum SIFT-Cheatsheet.pdf

Command 'sha384sum' not found, did you mean:

  command 'sha384sum' from deb coreutils (8.30-3ubuntu2)

Try: sudo apt install <deb name>

sansforensics@siftworkstation: ~/Desktop
$ sha384sum SIFT-Cheatsheet.pdf
15c9b10666d96354c20fe3eddd5622e7671f93df2b5c4c3c91cb3c38b7cf789223fac0dd0fd6a826f1c3e9e334726b29  SIFT-Cheatsheet.pdf
sansforensics@siftworkstation: ~/Desktop
$ sum SIFT-Cheatsheet.pdf
15430 331
sansforensics@siftworkstation: ~/Desktop
$ mdg5sum SIFT-Cheatsheet.pdf

Command 'mdg5sum' not found, did you mean:

  command 'md5sum' from deb coreutils (8.30-3ubuntu2)

Try: sudo apt install <deb name>

sansforensics@siftworkstation: ~/Desktop
$ md5sum SIFT-Cheatsheet.pdf
60cc462dca495e69c5ca833c1edf654f  SIFT-Cheatsheet.pdf

```

## 2.1 Szyfrowanie symetryczne

Zaszyfruj wiadomość tekstową z użyciem hasła, w trybie symetrycznym (`--symmetric`). Prześlij zaszyfrowaną wiadomość koledze/koleżance podając użyte hasło. Następnie odszyfruj i odczytaj otrzymaną wiadomość.

Aby zaszyfrować, wykonuję komendę:

```
gpg2 --symmetric ~/Documents/wiadomość.odt:
```

```

sansforensics@siftworkstation: ~
$ gpg2 --symmetric ~/Documents/wiadomosc.odt
sansforensics@siftworkstation: ~
$ gpg2 --symmetric --armor ~/Documents/wiadomosc.odt
sansforensics@siftworkstation: ~

```

Oryginalny tekst:

```

1 |pokad tupta noca jez

```

Zaszyfrowany tekst bez użycia `--armor`:

```

1 |8C
2 |A0AB838E dx00FC02ZC6CEDE4
3 |ESBE07F2DE_k
4 |839F08C20EB8D078F881Rq!F8YyB4dBA5~va9FCE-k&-lmK6U0C5%:9E 819E92jDC\ES89g829,kY85A

```

Zaszyfrowany tekst po użyciu `--armor`:

```

1 |-----BEGIN PGP MESSAGE-----
2 |
3 |jA0ECQMqCcxwJ5BThGP80l0BVF5B6n0vX+fCluoKNrVh7Kl66NgV5zX+0zzLi/D8
4 |sZ+w2XSNmWJwJBfG+1+rIgrW5rUYuNsam8NYd1pkBgYJFZ6TyVwB8xAC1CzZWT42
5 |J+kLFKmiUfwzkZ4=
6 |=0Ko6
7 |-----END PGP MESSAGE-----

```

Komenda `-armor` zmienia rozszerzenie pliku i zmienia tekst na bardziej przystępny dla użytkownika – o wiele łatwiej jest go przeczytać.

Przesyłam koledze moją zaszyfrowaną wiadomość, a on przesyła mi swoją.

Aby odszyfrować, wykonujemy komendę:

`gpg2 --decrypt ~/Downloads/FILIPEK.asc`

```
sansforensics@siftworkstation: ~
$ gpg2 --decrypt ~/Downloads/FILIPEK.asc
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
ZDAZYC PRZED BOGIEM - FILIPEK (3 zwrotka)

Kiedys bylem dzieckiem, ktory chcial oddac serce
za koncerty dla ludzi, pare wyswietlen wiecej
chociaz nie mialem hajsu, to moglo tu byc gorzej
bef floty sie rzucilem, tu na gleboka wode
chce zeby kiedys wreszcie byli dumni rodzice
zeby brutto czy netto nie robilo rozniczy
wybic sie poza schemat, tyry i kredytu
moze jeszcze na free wywalczyc jakis tytul
jestem normalny, nie tylko przy tych wersach
przedzej odmowie hajsu, niz foty po koncertach
jesli mialem sodowe, to wrocilem na ziemie
i przeprosilem ziomow za swoja pajacere, ta
najgorszy z QueQuality
ten najgorzej ubrany, nieatrakcyjny na featy
ale to Fifi slucha mnie osob grono
dojde do glosu skoro, w zanadrzu sztosow sporo
sansforensics@siftworkstation: ~
```

Udaje mi się odszyfrować plik, który zawiera tekst piosenki.

## 2.2 Szyfrowanie asymetryczne - generowanie i przekazywanie kluczy

Wspomagając się tutorialiem (w odnośnikach), wygeneruj dla siebie parę kluczy PGP. Następnie zaimportuj klucz publiczny kolegi/koleżanki. Aby to zrobić:

1. Wygeneruj swoją parę kluczy.

Aby to zrobić używam komendy:

`Gpg2 --gen-key`

```
sansforensics@siftworkstation: ~
$ gpg2 --gen-key
gpg (GnuPG) 2.2.19; Copyright (C) 2019 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

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

Real name: Patrycja
Email address: patrycjax1717@gmail.com
You selected this USER-ID:
"Patrycja <patrycjax1717@gmail.com>"

Change (N)ame, (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: key CBD73859F753F6DE marked as ultimately trusted
gpg: directory '/home/sansforensics/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/sansforensics/.gnupg/openpgp-revocs.d/3BCEB1C9AC2512261ED21C75CBD73859F753F6DE.rev'
public and secret key created and signed.

pub   rsa3072 2022-03-15 [SC] [expires: 2024-03-14]
       3BCEB1C9AC2512261ED21C75CBD73859F753F6DE
uid           [ultimate] Patrycja <patrycjax1717@gmail.com>
sub   rsa3072 2022-03-15 [E] [expires: 2024-03-14]

sansforensics@siftworkstation: ~
$
```

## 2. Wyświetl klucze publiczne i prywatne.

Używam do tego komendy:

```
gpg2 --list-keys
```

```
sansforensics@siftworkstation: ~
$ gpg2 --list-keys
/home/sansforensics/.gnupg/pubring.kbx
-----
pub   rsa3072 2022-03-15 [SC] [expires: 2024-03-14]
       3BCEB1C9AC2512261ED21C75CBD73859F753F6DE
uid           [ultimate] Patrycja <patrycjax1717@gmail.com>
sub   rsa3072 2022-03-15 [E] [expires: 2024-03-14]

sansforensics@siftworkstation: ~
$ gpg2 --list-secret-keys
/home/sansforensics/.gnupg/pubring.kbx
-----
sec   rsa3072 2022-03-15 [SC] [expires: 2024-03-14]
       3BCEB1C9AC2512261ED21C75CBD73859F753F6DE
uid           [ultimate] Patrycja <patrycjax1717@gmail.com>
ssb   rsa3072 2022-03-15 [E] [expires: 2024-03-14]
```

## 3. Wyeksportuj swój klucz publiczny do pliku:

Eksportuję klucz wykorzystując adres e-mail: patrycjax1717@gmail.com



```

$
sansforensics@siftworkstation: ~
$ gpg --export --armor patrycjax1717@gmail.com patrycjax1717@gmail.com.pub
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQGNBGIXGE4BDACHVGHYwMLLWgZVYIYIsikS+s+EdmIpdypIkIoqc/P/JUDL7hi
EiPMqJD1c4GGxL+MQBeVqQOisMq3Ln4bkKEfQmZaMmgHQS84gxQbfm+Zw1FsEd0
ASRamQ7k1S828Ahx4LF8tHqe2XjLCJ1BdeAfAZRelyIbuN56NR+Day2sLwu4lXg/
KuvGrCwZAWMsTko05zCVzp3Jwz3GBqG9gamkTleotK8unrd5lviFcLz9H+3sxfjt
rcbTl4gmJ2EDrhAScjQONAUT9BxIfosyUUzSCEFBxqcWwnoCk0ZbUk3oMF09p4ev
Zvz2Ig0Xyt2Vm3giuD02XTxrS8Lgzj6IBt75FUfcoGK4Z77QmSLSI9fwqjQSEkQf
S2UBeBoVRuox4pcnnjoJk3cpAJWJMr+JI8vWw1Dy6mRKOMdzYhPqI8rIKSZRH67i
b2Njnd6tu1pECP3NiLD1tb29JRBhWUjwKGkqn4o8KUWH1bxG80C3FiNRhTpcFTE
n2CQjkdRj5jYjPKAEQEAAQIUGF0cnljamEgPHBhdHJ5Y2pheDE3MTdAZ21haWwu
Y29tPokB1AQTAQAPhYhBDvOscmsJRImHtIcdcvXOFn3U/beBQJiMRhOAhSDBQkD
wmCABQsJCACCBHUKCQgLAGQWAgMBAh4BAheAAAJEMvXOFn3U/beJJ0L+wauPgrg
y5cm8/U3v6n8FWS/sLQHFG4UjuckvmIPdkVfngraGsMzkzLEu/yxJmGZu79JtuLE
nSpzqc0ipq0sKvV3sXz3fRSukJeC/e8vIVqBfL8zYsrlvdVE3IdBiSspVXpyZJgx
PuiQw94xz5drLqPWG2eGcXr3GVnyrLBsWDaz0xH00oifHN0HI0911ClRp10ENhTA
TuBJSS+pYhKdOQkkWWH22lrpWKHYeTGIhidBFlozbn3QPCzRhokBwfTc6t7nCe/
P6UFGl/OCpF0jmbamLVC0m1qsHLJQpc/EIaQH89as1tm2qWshn0T4cRxZ2zvH8k
7Jewnd8byeb0HVXiFrXSV1E0kicSLoaX0QyXn0jHZy+kErR4ijZzdgdPj4ICLh9
TZB9wzh0NuUfbtIwCrtCXz9FcJQU7KXBtCZ+0dhwL3dFzF5Arr1Uh73h80pbsFi5
G25fXg7vB95ji6rTLIK0QRLxFNNN4dbLMg2ZBjwk468+j69N95efy/l+RbkBjQRi
MRhOAwA100/KWmpWe/QNliOiG0KyghtL06Us/g4fQiAU8ALChZnHcfXK4MIWgVe
6CvGXHESXiVLV08mp8d4ZNevcBLCL3K9Rm35d5Cwph6CYrcs8o5NBhXNSSnKleBj
rwwSzkscseHEsBl6v6qSJm7X7XMWXTGSAP/UV3z8sBj+LtbfESMWqdvCpOn8cPhH
4VDkA4gX67kQvquL0tGTfVvZ/njCgf6Ex+m7xdzIJuRK5CJwmHN/rmnc5/EGVvbG
Xd7BlrdD9dCeLvkVrqN0YP/4q8z92oTB4zYadA3RVA58cFC00lvAN/9XZWLLlViX
tuqJ3n1aSKtAwLCfYKAF7VkuIKc3yYw/k1LBqS7WEi08gyoMlccZK9QJx2SrDyTP
VjuYBsJxjgk8HGgJP4YgnFUtFNd8PZ0YJ76nTzXyla8pBexGE6RSalieuVPMCeUT
imRUhsJnNHij+pYYNl3L5oVwr2b4r1MC1xwtqwpM1aFvik7j5paNeZ6e2WH/z6e3
OYPyEhIfABEBAAGJABwEGAekACyWIIQ7zrHJrCUSJh7SHHXL1zhZ91P23gUCYjEY
TgIbDAUJA8JnAAAKCRDL1zhZ91P23n50C/9SwoUyZPB09tC67+bnTD4IIFu06TLB
n1nZ7EEVQZkp9e5yeA0t5+nFDwgnf1pTzlrVW2M0/1dZIsBMsZKW/QWC53ozkRXj
y/6UI/KiHm26GRrYKKo9vAxj0/7PNm3dsuk4QFu14aLTWgkIle3lw6dEcX5gJ11j
9Skb1e0d0i8vPhJrIcIriF8x8qLW6hx1YS8dCf+kJZUZC4iWPGYlZ8eVwIx4e3enz
+IuxjpsJjYNFAuC/cfHLznI+FZDxnyTzAxxMQzUavURWL76/Ir+1JE9tEz5fNn+s
hL/BboxKMvgvWL/69CDemGzLxII03Mxrs3mUy0JCa7y5sfH9dMGZuBbVHGwV1lQy
+GjoL1NTUjPKQ7uiYfEXyReeEobx5QwB6WhmkwmZKR1a/vskcN0PxHtfZQlZQbjJ
xxZq6S094Lza8d0yeLvGi59HkiOqRv2pHXaR+kiT6R6HKu40xU+VyIdPTQjLufim
YIRXjP0p07TaWyp6U9L7zLCz00t3Wel0a6s=
=JPKG
-----END PGP PUBLIC KEY BLOCK-----
sansforensics@siftworkstation: ~

```

4. Prześlij utworzony plik koleżance/koledze.

5. Zaimportuj też u siebie otrzymany klucz publiczny:

Importuję u siebie klucz kolegi wykorzystując jego adres e-mail: miloszgronowski@gmail.com

```

sansforensics@siftworkstation: ~
$ cd Downloads
sansforensics@siftworkstation: ~/Downloads
$ gpg --import miloszgronowski@gmail.com.pub
gpg: key DC9EBF2514C2A0DE: public key "Miłosz <miloszgronowski@gmail.com>" imported
gpg: Total number processed: 1
gpg:             imported: 1
sansforensics@siftworkstation: ~/Downloads

```

Klucz został prawidłowo zaimportowany.

## 2.3 Szyfrowanie asymetryczne

Zaszyfruj wiadomość tekstową za pomocą klucza publicznego kolegi/koleżanki. Następnie przekaż wiadomość koledze/koleżance. Po otrzymaniu wiadomości odszyfruj ją używając swojego klucza prywatnego.

Szyfruję utworzony przeze mnie plik przy użyciu klucza publicznego kolegi z wykorzystaniem jego adresu e-mail.

```
sansforensics@siftworkstation: ~
$ gpg2 --recipient miloszgronowski@gmail.com --sign --encrypt --armor ~/Document
s/zaszyfrowana.txt
gpg: 053A5350697142E9: There is no assurance this key belongs to the named user

sub rsa3072/053A5350697142E9 2022-03-16 Miłosz <miloszgronowski@gmail.com>
Primary key fingerprint: 35BA 9054 7179 D2FA 57E9 32E0 DC9E BF25 14C2 A0DE
Subkey fingerprint: ACDE 80B1 18D0 CFA4 333E 83B8 053A 5350 6971 42E9

It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```

Odszyfrowuję otrzymany plik z rozszerzeniem .gpg oraz .asc.

```
Use this key anyway? (y/N) y
sansforensics@siftworkstation: ~
$ gpg2 --decrypt ~/Downloads/cytat.gpg
gpg: encrypted with 3072-bit RSA key, ID 2A7B2EE780D58DE6, created 2022-03-15
"Patrycja <patrycjax1717@gmail.com>"
"Dzis tylko na krotko sle tu zatrzymuje. Ale tutaj serce moje zostalo na zawsze" - Jan Pawel II na rynku w Kalwarii Zebrzydowskiej, 14 sierpnia 1991
sansforensics@siftworkstation: ~
$ gpg2 --decrypt ~/Downloads/cytat.asc
gpg: encrypted with 3072-bit RSA key, ID 2A7B2EE780D58DE6, created 2022-03-15
"Patrycja <patrycjax1717@gmail.com>"
"Dzis tylko na krotko sle tu zatrzymuje. Ale tutaj serce moje zostalo na zawsze" - Jan Pawel II na rynku w Kalwarii Zebrzydowskiej, 14 sierpnia 1991
sansforensics@siftworkstation: ~
$
```

Otrzymuję cytat Jana Pawła II z rynku w Kalwarii Zebrzydowskiej.

## 2.4 Podpisywanie wiadomości

Wykonaj te same czynności co w poprzednim zadaniu, ale tym razem dodatkowo zadbaj o to, żeby wiadomość była podpisana Twoim kluczem. Po otrzymaniu wiadomości odszyfruj ją i sprawdź czy jest podpisana.

Szyfruję wiadomość i używam komendy --sign, która sprawi, że wiadomość będzie podpisana moim kluczem. Następnie wysyłam plik do kolegi.

```
sansforensics@siftworkstation: ~
$ gpg2 --recipient miloszgronowski@gmail.com --sign --encrypt ~/Documents/gra.txt
gpg: 053A5350697142E9: There is no assurance this key belongs to the named user

sub rsa3072/053A5350697142E9 2022-03-16 Miłosz <miloszgronowski@gmail.com>
Primary key fingerprint: 35BA 9054 7179 D2FA 57E9 32E0 DC9E BF25 14C2 A0DE
Subkey fingerprint: ACDE 80B1 18D0 CFA4 333E 83B8 053A 5350 6971 42E9

It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
sansforensics@siftworkstation: ~
```

Pobieram otrzymany od kolegi plik i rozszyfrowuję go.

```
sansforensics@siftworkstation: ~
$ gpg2 --decrypt ~/Downloads/kamien_papier_nozyce.gpg
gpg: encrypted with 3072-bit RSA key, ID 2A7B2EE780D58DE6, created 2022-03-15
"Patrycja <patrycjax1717@gmail.com>"
Kamień
gpg: Signature made Wed 16 Mar 2022 06:51:50 PM UTC
gpg: using RSA key 35BA90547179D2FA57E932E0DC9EBF2514C2A0DE
gpg: Good signature from "Miłosz <miloszgronowski@gmail.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.
Primary key fingerprint: 35BA 9054 7179 D2FA 57E9 32E0 DC9E BF25 14C2 A0DE
sansforensics@siftworkstation: ~
```

Uzyskana wiadomość brzmi „Kamień”.

Używam komendy –verify, która weryfikuje tylko podpis, ale pojawia się błąd, którego przyczyny nie znam.

Następnie wykonuję odłączny podpis przy użyciu komendy –detach-sign.

```
Use this key anyway? (y/N) y
sansforensics@siftworkstation: ~
$ gpg2 --verify ~/Downloads/kamien_papier_nozyce.asc
gpg: verify signatures failed: Unexpected error
sansforensics@siftworkstation: ~
$ gpg2 --detach-sign ~/Downloads/kamien_papier_nozyce.gpg
File '/home/sansforensics/Downloads/kamien_papier_nozyce.gpg.sig' exists. Overwrite? (y/N) y
sansforensics@siftworkstation: ~
$ gpg2 --detach-sign ~/Downloads/kamien_papier_nozyce.asc
sansforensics@siftworkstation: ~
$
```

Otrzymuję plik, który wygląda następująco:

```
1 |-----BEGIN PGP MESSAGE-----
2
3 |hQGMAyp7LueA1Y3mAQv/bI5+/rNVYQsTQdlw09J3FDIEQwb/wFCbRiQTwaM/4FYi
4 |MAE8ybyAMw2wbrPyN0SwZG0zX8LE7G1t+UQRyJYPmy6eIM+qKqu8DQMCdFV8myZR
5 |KocDR5RH0ciX0h1ZLV0YRN6ndPGXpRMAeWKJX6TX8HkUAp/Sh9vNbpB28GH4ofUn
6 |FrdLXTUMQo+BwV+npZuhWCDXRe49dxo8uYVswDCPKS1fm3ziuxQ3BWILY+ROVNSL
7 |Bru75PaRnx2ukwyJ7osoz2s5Ngcg5Cqy8ZNBp8os3NRtH5z+MHG10gofP4Jnw5X0
8 |F4evxug3g8TmDuT23RuuE8Tg9CIR/q0304jiKB+5BVVL32QzQ7n/sUUEfJe3V/F
9 |BBzawE65qdT4z8FWI0Nl9Ja6Q6uuu4s93Cs04GDUiDA0EqyvToKdMRLuJPRV8PYm
10 |7inFoVBAYReNslz5RbzeRxpni+nKPDJMp2JJozxbJSsnRJ/kSwr7tjNPoL5UAsG8
11 |Uarfb0ApAFbgnWT4GaEA0ukB0fVxs5pdk6AajGz5c3YGLshGI1QjSzfAIZX9bjr
12 |26CtXK5tYU/z2ei9rCMFsJK+603s0I8D7PEmgo62nv+9Lo7P10jkbQ6Qq+GCF0tQ
13 |kpJi1mN7r/AshoGoKu/yL09IDSuSC0mZSKkwynEwugipxXjKLS1fqbo45P925U4q
14 |pzEc7Jaz+2WgsXyOUz0Hz9pipCCTRReitNBvRJP4GLZlLFkM0K31Uq/z/z1j4ZJ
15 |X1mm4CTHRGfG1wZ24pZvjhgGdH83gryF96X2QTOrT80hSner8HtQGF3oioMK2ns2
16 |jtuuhjBFBKTK4cy88HCpxiNB9cN9Wtu0SQyNb0y6hyylC/KAlziHwXW1/RQ+aVX
17 |pPG7097KpHWS5qhA0BzxIVFs8I+JsLIA7X4w9QRxmZpL2EuEt0Npr55VuSJ0p73X
18 |DBz3FQb0LTQAW2oiYhbHJpLkMdpLhsXG/wtCr1zKjIMKrrdKHVPurnfZy74dMHY6
19 |b3dzjurFgg+mZzNHEGZNRll6jFwGhVB77YUdfTYj+SP9xXFJWZ+Gqw80aLxx1b
20 |skJyGgy7x8qQkxgDfLzLUiorRuIRUMEB3WyAGOXofIIQ6nQz8eGy3Uz0UmLKLgub
21 |TYf0irUI+VyFmkFRSPQAaj4J+HuZJjckVEzJD17tKuM42c/07RE3EEIxy7Zhtw6f
22 |FB90pz0nHY6AyHLv47GRgeo2LBiM9DGur/JN6o3af/a+
23 |=e7bD
24 |-----END PGP MESSAGE-----
```