Wrocław, 02.11.2020

# Bezpieczeństwo sieci komputerowych

## Laboratorium 2

## Kryptografia

Prowadzący: dr inż. Marcin Markowski

Autorka: Agnieszka Płoszaj 218353

Zadanie zostało wykonane przy wykorzystaniu 2 komputerów:

- Pierwszy komputer z systemem macOS Catalina.
- Drugi komputer z systemem windows 8.1.

Zadanie 1

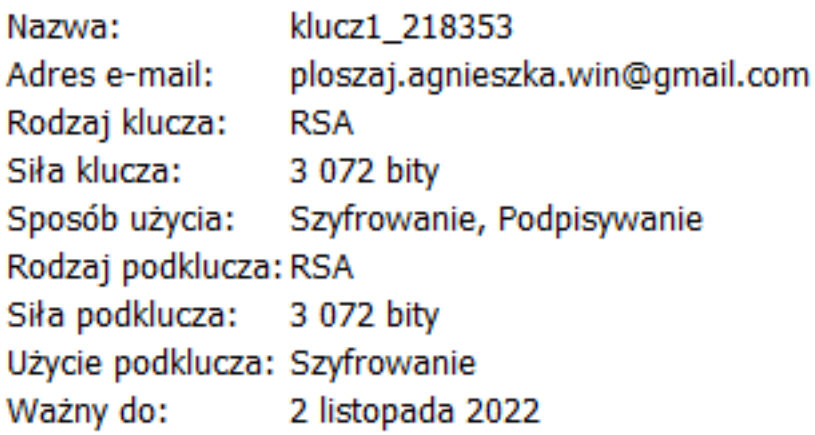
Użyte programy:
- Mozilla Thunderbird
- wtyczka enigmail
- gpg4win (windows)
- GPG Keychain (macOS).

Zadanie 2

Na podstawie ilustracji 1, 2 i 3 została zrobiona tabela 1 ilustrująca utworzone pary kluczy, wraz z parametrami (użyty do ich utworzenia program oraz system operacyjny, algorytm, długość oraz termin ważności). Polecenie jakie zostało użyte przy generowaniu kluczy korzystając z konsoli to:
**gpg --full-generate-key .**

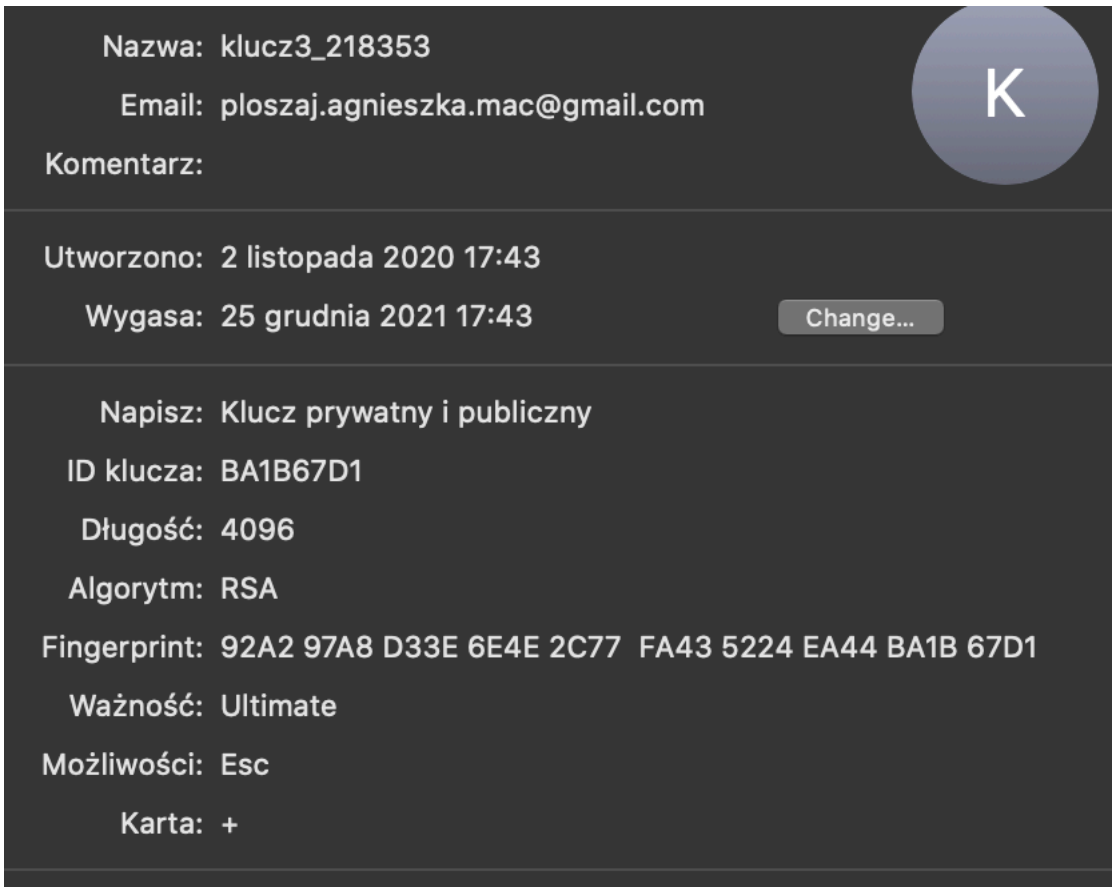
Tabela 1 Wygenerowane komplety kluczy

| Nazwa klucza | klucz1_218353 | klucz2_218353 | klucz3_218353 |
|---|---|---|---|
| Program | Kleopatra | konsola | GPG Keychain |
| System operacyjny | Windows 8.1 | Windows 8.1 | macOS Catalina |
| Algorytm | RSA | DSA i Elgamala | RSA |
| Długość | 3072b | 2048b | 4096b |
| Termin ważności | 02.11.2022 | 22.11.2020 | 25.12.2021 |




Ilustracja 1 Pierwsza para kluczy


Ilustracja 2 Druga para kluczy




Ilustracja 3 Trzecia para kluczy

Zadanie 3

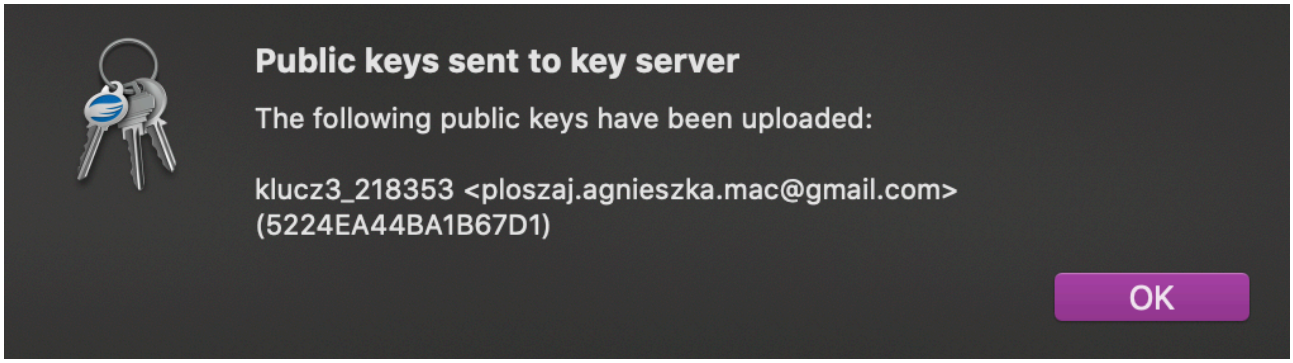
Ilustracja 4 przedstawia fragment wyeksportowanego klucza prywatnego i publicznego do pliku w formacie ASCII (przy pomocy GPA).

Odcisk klucza publicznego:
0C98 D2E7 8E7C 4335 A035  3F6E C3E6 85C5 0337 3118

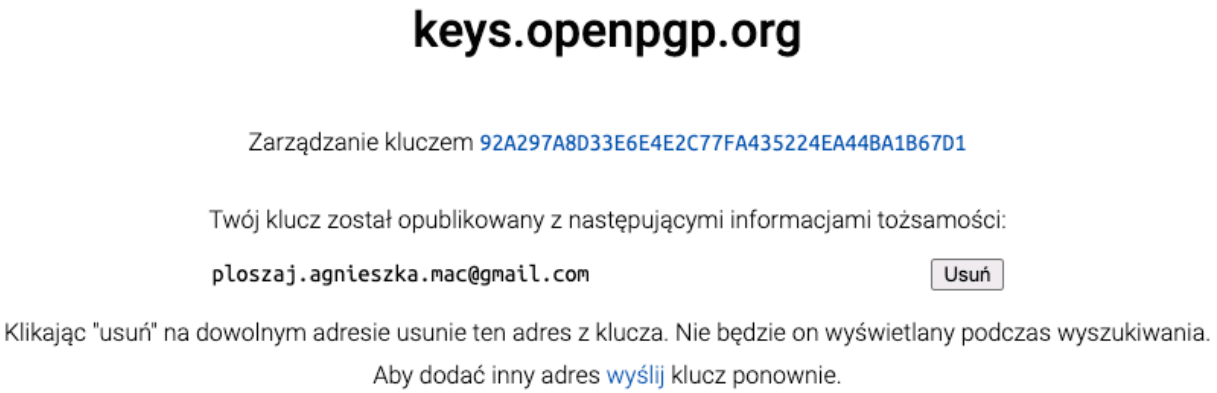
```
yTHuT24LzhgzADdDYGeYfvNCuhzPzQfmsSf/xQbgZ9n3lk7vCcXWcP7qgrCE3UfM
dCmLlktFMaI/41OsnIzK7nEjYBUJIwXnz1eZexkC8IKLLPTwHqZy6UUgzx4BJpwH
qq04lxoGL2VwDz3rNXCC5+9L4z/5wWB2/CMMuce113iWqClyA8/djmKmFS0ow5td
0kP55/mfC82+aHfkCu7ybRCoofSCCMwNosYt3GyIa5T2qtc0HpsjEa5qh2EChUI3
XRtkhMK2T4gku+eeLyGRkdiFGc/4EpboeImLFYos0muwpoUgxMem3ojG1gvCKBwD
eoELrSCIGz/vUH81UHEeoXnkZ42z+KA22s84YCwrmByH1wZ1fJnt8UC54WLgvmDJ
vD7IFguCqkT1Ac7LLAodNzXOYKok//nbSd7c9WdTgMYs2b01AoQZlTBEgmrkPiau
uTW+pPtoT2qAdEbMuDvu3wjSxjqo8H82rUllxuEOuez7xHIcKYfK
=d1kN
-----END PGP PUBLIC KEY BLOCK-----
-----BEGIN PGP PRIVATE KEY BLOCK-----
```

Ilustracja 4 Wyeksportowane klucze do pliku

Klucz 3 (publiczny) został wyeksportowany na serwer kluczy przy pomocy programu GPG Keychain (ilustracja 5). Po przeszukaniu serwera kluczy, znaleziony został klucz 3, natomiast po kliknięciu w zarządzanie kluczem serwis wysyła na skrzynkę pocztową link, po kliknięciu w który ukazuje się strona pokazana na ilustracji 6. Usunięcie klucza nie stanowi więc większego problemu.

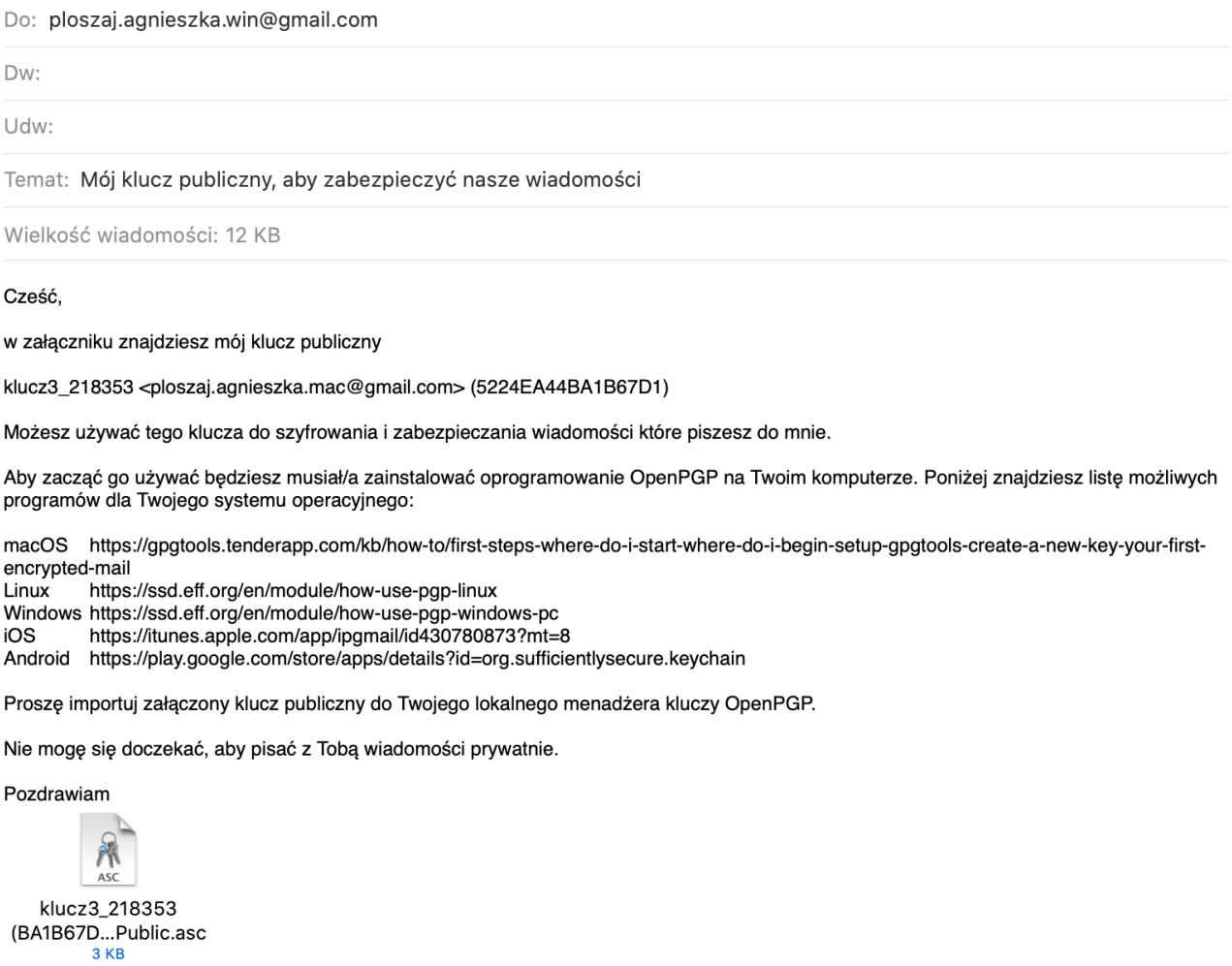



Ilustracja 5 Wyeksportowanie klucza na serwer kluczy




Ilustracja 6 Serwer kluczy

Klucz publiczny (klucz3_218353) został wysłany przy pomocy programu GPG Keynote, który posiada opcję „wysłania klucza przez e-mail”, treść utworzonego w ten sposób e-mail została pokazana na ilustracji 7. Następnie na drugim komputerze został on zaimportowany oraz podpisany (ilustracja 8 i 9). Poprawność podpisu została sprawdzona korzystając z programu Kleopatra (ilustracja 10).



Do: ploszaj.agnieszka.win@gmail.com

Dw:

Udw:

Temat: Mój klucz publiczny, aby zabezpieczyć nasze wiadomości

Wielkość wiadomości: 12 KB

Cześć,

w załączniku znajdziesz mój klucz publiczny

klucz3_218353 <ploszaj.agnieszka.mac@gmail.com> (5224EA44BA1B67D1)

Możesz używać tego klucza do szyfrowania i zabezpieczania wiadomości które piszesz do mnie.

Aby zacząć go używać będziesz musiał/a zainstalować oprogramowanie OpenPGP na Twoim komputerze. Poniżej znajdziesz listę możliwych programów dla Twojego systemu operacyjnego:

macOS https://gpgtools.tenderapp.com/kb/how-to/first-steps-where-do-i-start-where-do-i-begin-setup-gpgtools-create-a-new-key-your-first-encrypted-mail
Linux https://ssd.eff.org/en/module/how-use-pgp-linux
Windows https://ssd.eff.org/en/module/how-use-pgp-windows-pc
iOS https://itunes.apple.com/app/ipgmail/id430780873?mt=8
Android https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain

Proszę importuj załączony klucz publiczny do Twojego lokalnego menadżera kluczy OpenPGP.

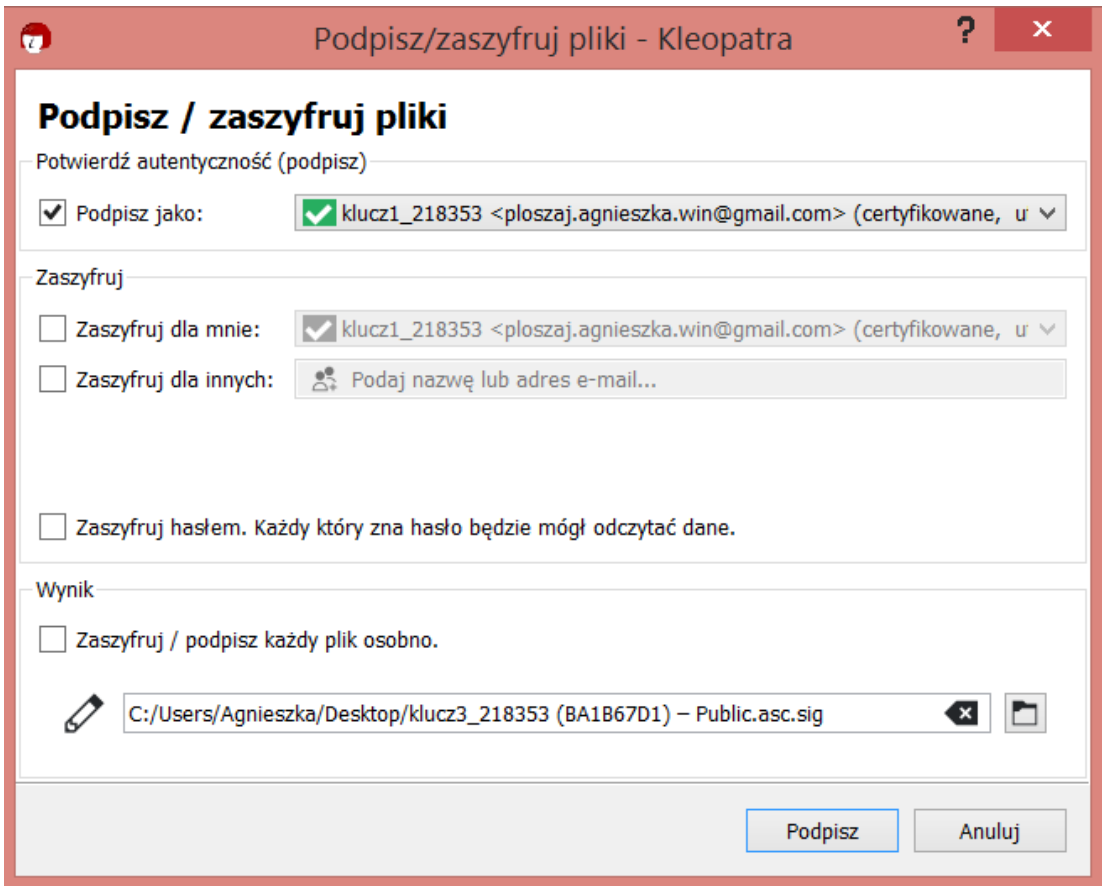
Nie mogę się doczekać, aby pisać z Tobą wiadomości prywatnie.

Pozdrawiam

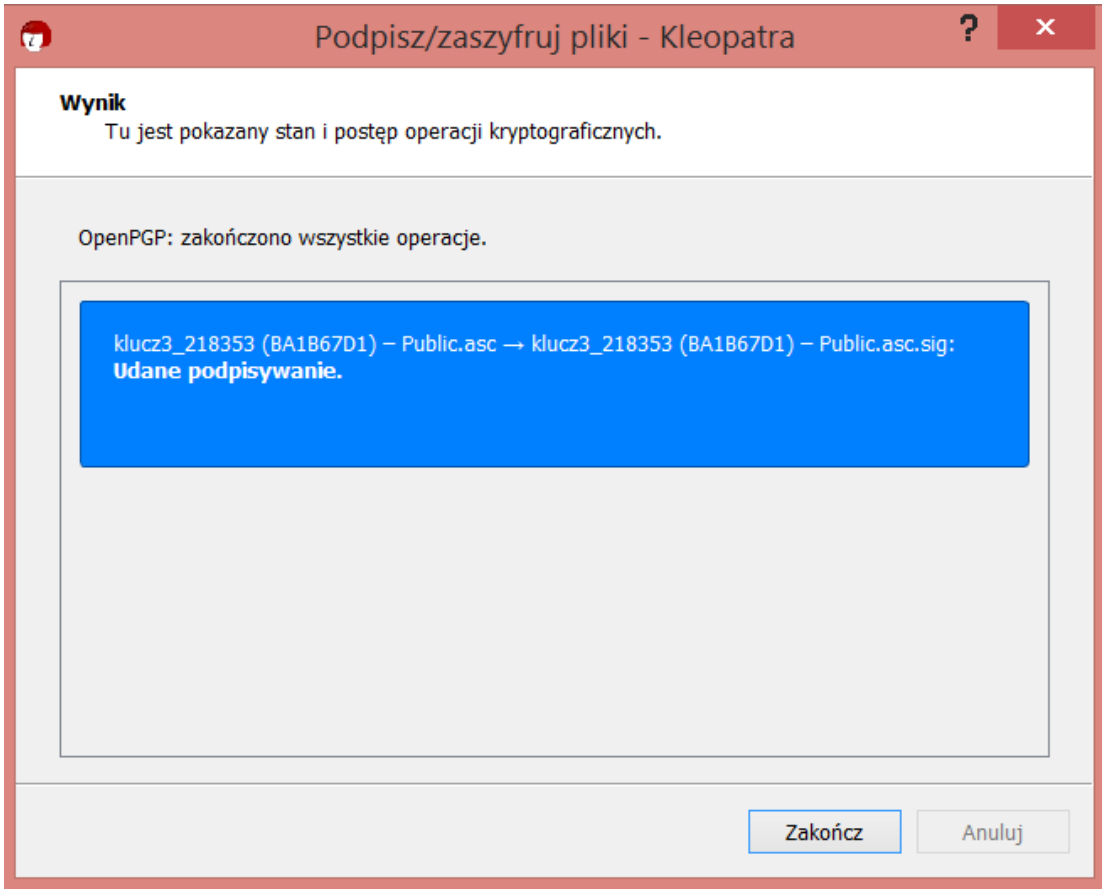
ASC

klucz3_218353
(BA1B67D...Public.asc
3 KB

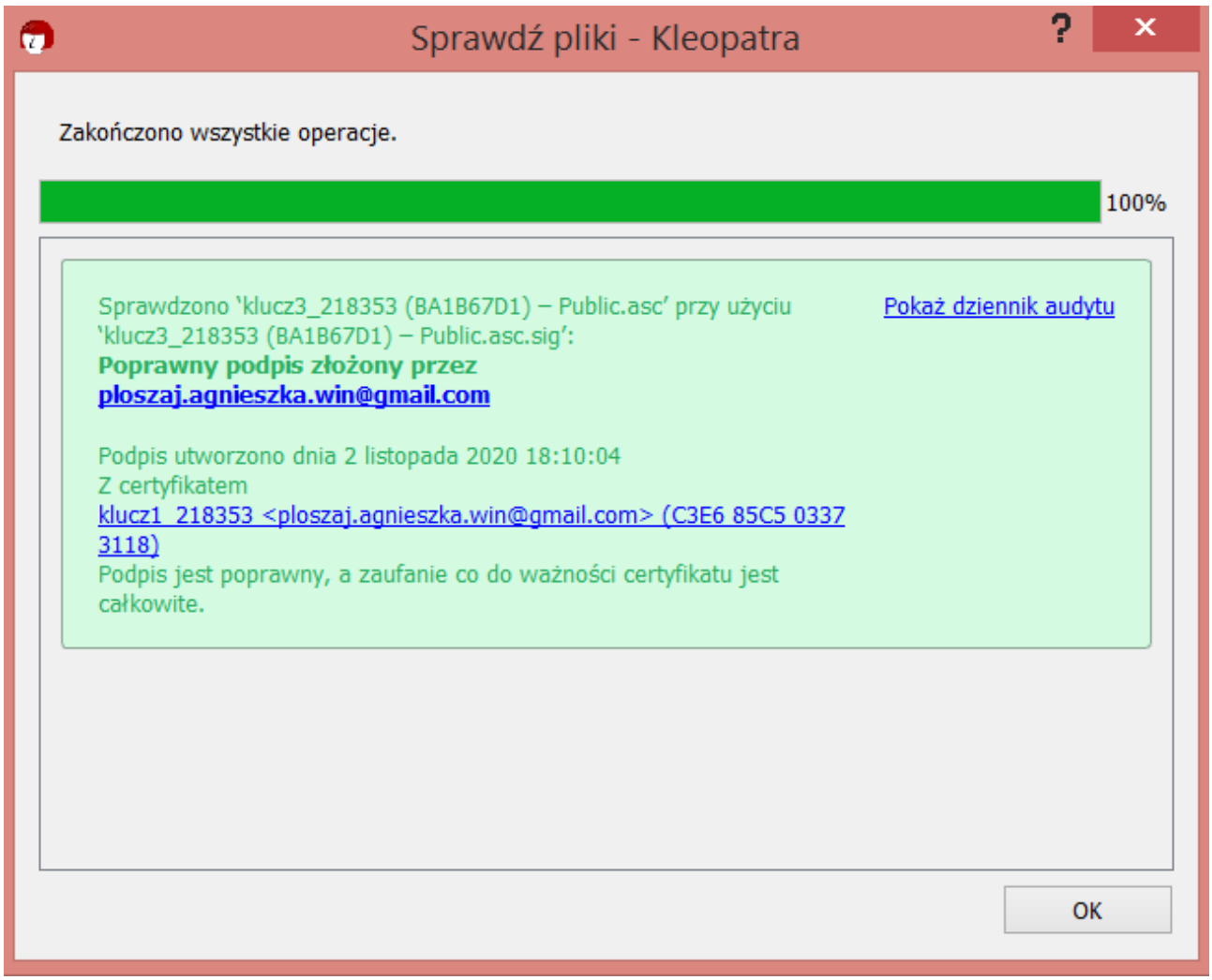
Ilustracja 7 Wysłanie klucza publicznego




Ilustracja 8 Podpisanie klucza




Ilustracja 9 Udane podpisywanie klucza




Ilustracja 10 Sprawdzenie poprawności podpisu

Na początek został utworzony plik tekstowy. Następnie korzystając z konsoli został on podpisany przy pomocy poniższego polecenia:

**gpg** --**celarsign** 218353.txt

Na ilustracji 11 widać zawartość utworzonego w ten sposób pliku, natomiast na ilustracji 12 została przedstawiona weryfikacja podpisu przy wykorzystaniu nakładki graficznej (Kleopatra).

```
C:\Users\Agnieszka\Desktop>gpg --clearsign 218353.txt

C:\Users\Agnieszka\Desktop>type 218353.txt.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Płoszaj
19:01
02.11.2020
-----BEGIN PGP SIGNATURE-----

iQGzBAEBCAAdFiEEfZoU1UbfXzYXAlRy9X0JZKE/Ug4FAl+gSnUACgkQ9X0JZKE/
Ug6piQv+PeCDrNxYu1NsVxG0+Kx/7+TVf+LgRI5Z0CpWGkHoxxHJ1tVzsbCweIf4
wJcUzcTq+yMei3UaHR5fIvkL5u/TaIo7ookzK/XeJBzfV5d4jKEMoriFB7kGUg/i
naRGtJv3rhTHP36phY+IJH5RMdGzyoTHnjvuWKWk3u04X+uhP8xrMDTCYUY5wmsd
f03J09c2f4ftU5cFnMJ0D9UzoJSz6zjtPMy6gt0V3kb1Thcp9z6ftaKuf2znQuCq
sqxNnx4uP9m3VL53jQRMzeV00zNFvWz8TuuZ6xp3c/xfwk2RoKozNRCNZ0a0H1yv
YdQrtB2D0UUe74ngEY/aAwzqjww9/Z0/YIt+BxgnYhVjLB1FQk27djQ8Mz6gguK0
ypCXh+9j5/ZovD/kEaWy8KFzPHdMLMdVzXnZ01En5hdqfLoN9EdkX27348/zo+n2
jPTtFwWt0Ly8rXYEaHdgfy3iEtiRXQxd63WjxY6kqo+WgC9mAnD9QpFmrRdZ6Mve
SPI1dVTb
=d8FM
-----END PGP SIGNATURE-----

C:\Users\Agnieszka\Desktop>
```

Ilustracja 11 Podpis cyfrowy pliku

218353.txt.asc → 218353.txt: Pokaż dziennik audytu

**Poprawny podpis złożony przez**
**ploszaj.agnieszka.win@gmail.com**

Podpis utworzono dnia 2 listopada 2020 19:05:41
Z certyfikatem
Agnieszka Płoszaj win <ploszaj.agnieszka.win@gmail.com> (F573 8964 A13F 520E)
Podpis jest poprawny, a zaufanie co do ważności certyfikatu jest całkowite.

Ilustracja 12 Sprawdzenie poprawności podpisu

Na początek, przy wykorzystaniu konsoli został utworzony podpis do pliku binarnego za pomocą polecenia:

**gpg** --**armor** --**output** wzor_dorobek3.pdf.sig --**detach-sign** wzor_dorobek3.pdf

Następnie została sprawdzona zawartość podpisu, żeby upewnić się, że jest ona w formie czytelnej oraz przy pomocy polecenia:

**gpg** --**verify** wzor_dorobek3.psf.sig

została sprawdzona poprawność złożonego podpisu.

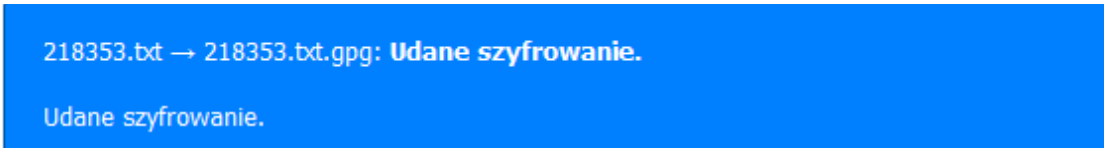
Wszystkie rezultaty zostały pokazane na ilustracji 13.

```
C:\Users\Agnieszka\Desktop>gpg --armor --output wzor_dorobek3.pdf.sig --detach-s
ign wzor_dorobek3.pdf

C:\Users\Agnieszka\Desktop>type wzor_dorobek3.pdf.sig
-----BEGIN PGP SIGNATURE-----

iQGzBAABCAAdFiEEDJjS5458QzWgNT9uw+aFxQM3MRgFAl+hLzYACgkQw+aFxQM3
MRhHlQv9GMKnEFZxq3vExvz/CuFrVwYY4724LNq882AZCsRKfVhzmCA7sGWcQSr0
jvipFSWeLR8hUL2FHIC+Uy6GVyeu0J65190pIMwyFV/vsB3WVnL7+64j4GJkm6ax
6L6PRaRConSlgwlMypFVlqCSWWlvesAxsnDb0Ru7Y58HAVFjhUTESx61JsR0Au4k
X/A0VqqfscGpbOW9xHw4ipwRQf2P/A10fwcCaTFSG2ibCtD/WJbG8Fhsy2ntMgHv
/O48wTihlTJpg/Ew8DXiyIYwUVYM6cuoFQUvpN19k/71L6RBuadM9jIVoXZ5JG94
yhZ69abOVmwGaawaF94n7l9TyYOQZLkAOMFfaLYtjam7YGDF474yNGVhnPZaOy8+
5nzW6XJ3ZxlkfQOfxSNdFPq5Yri5vpGMqKCHw557e9PEeW1wnBCLJzF4qp7vjjeN
hwFVn0F1RmbDTmlzvJ79tEjO74nPeXW+93QtizLnAWBgggIDGSjW0kO5cE9z1VUs
214k/u9J
=1cgw
-----END PGP SIGNATURE-----

C:\Users\Agnieszka\Desktop>gpg --verify wzor_dorobek3.pdf.sig
gpg: przyjęto obecność podpisanych danych w 'wzor_dorobek3.pdf'
gpg: Podpisano w 11/03/20 11:21:42 środkowoeuropejski czas stand.
gpg:                przy użyciu klucza RSA 0C98D2E78E7C4335A0353F6EC3E685C503373
118
gpg: Poprawny podpis złożony przez ,,klucz1_218353 <ploszaj.agnieszka.win@gmail.
com>'' [absolutne]
```

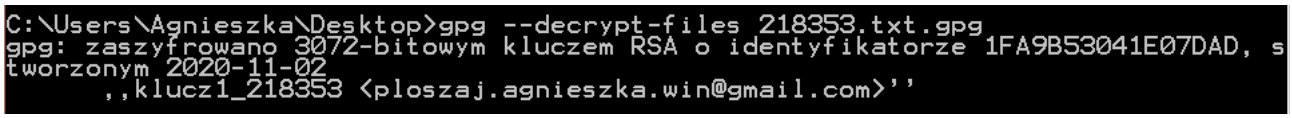
Ilustracja 13 Podpis pliku binarnego

Zadanie 8

Na początku plik *218353.txt* został zaszyfrowany spod nakładki graficznej (ilustracja 14), następnie został on odszyfrowany korzystając z konsoli (ilustracja 15).



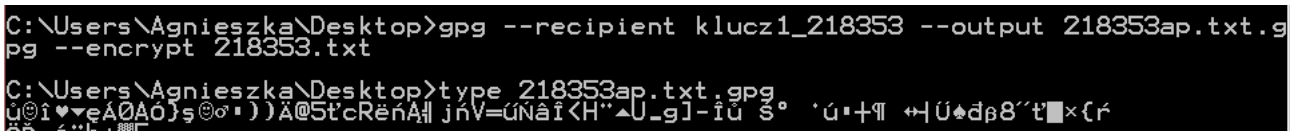

Ilustracja 14 Zaszyfrowanie pliku spod nakładki graficznej



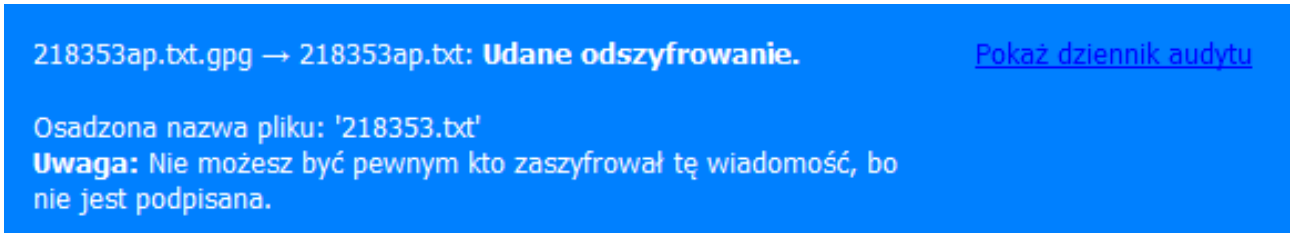

Ilustracja 15 Odszyfrowanie pliku korzystając z konsoli

Zadanie 9

Na początku plik *218353.txt* został zaszyfrowany przy wykorzystaniu konsoli (ilustracja 16), następnie został on odszyfrowany korzystając z nakładki graficznej (ilustracja 17).

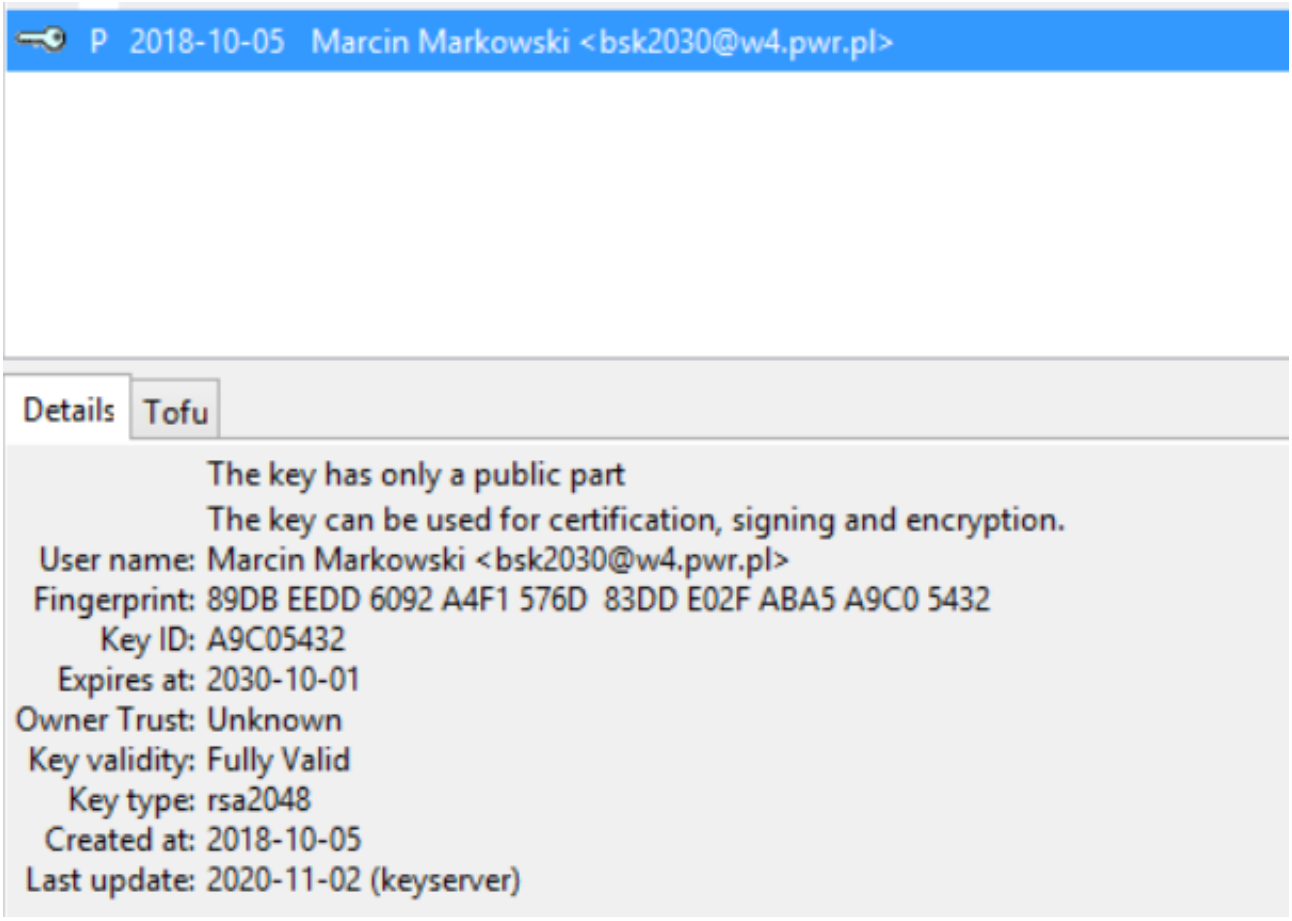



Ilustracja 16 Zaszyfrowanie pliku korzystając z konsoli




Ilustracja 17 Odszyfrowanie pliku korzystając z nakładki graficznej

Na początku korzystając z serwera php.mit.edu zostały pobrane trzy klucze przypisane do podanego adresu e-mail. Następnie porównano odciski wszystkich 3 kluczy, z odciskiem właściwego klucza, aż znaleziono właściwy klucz (ilustracja 18).




Ilustracja 18 Klucz prowadzącego

Korzystając z nakładki graficznej (Kleopatra) oraz klucza z zadania 10, dokonana została weryfikacja podpisu, aż zidentyfikowany został oryginalny plik (ilustracja 19 i 20).


Ilustracja 19 Weryfikacja podpisu – nieprawidłowy podpis


Ilustracja 20 Weryfikacja podpisu – poprawny podpis

Następnie zostało wykonane zadanie na podstawie treści pliku ZAD11_v5.txt (ilustracja 21).

```
*** ZADANIE 11 ***

Zapisać do pliku tekstowego imiona członków grupy.
Plik zaszyfrować za pomocą gpg algorytmem AES192 (tylko symetrycznym) z kluczem 'LABORKA'.
Obliczyć sumę kontrolną SHA-1 pliku (Kleopatra).
Komendy gpg, treść pliku przed i po zaszyfrowaniu oraz sumę kontrolną umieścić w sprawozdaniu.

******************
```

Ilustracja 21 Treść zadania 11

Do pliku *ploszaj.txt* zostało zapisane „Agnieszka”. Plik *ploszaj.txt* został zaszyfrowany korzystając z konsoli, zgodnie z treścią zadania (algorytmem AES192 i kluczem ‘LABORKA’) przy pomocy polecenia:

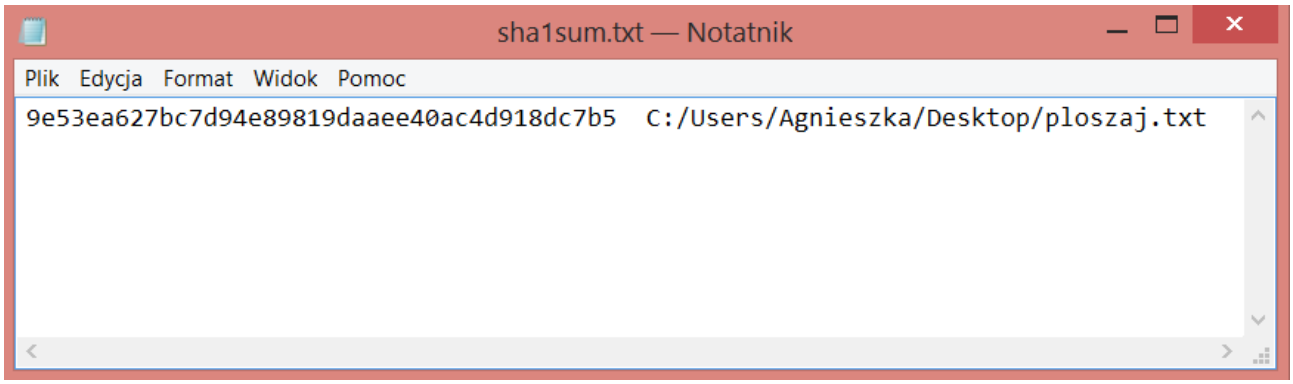
**gpg --symmetric --cipher-algo** AES192 ploszaj.txt

Na ilustracji 22 można zobaczyć, że treść pliku nie uległa zmianie.

```
C:\Users\Agnieszka\Desktop>type ploszaj.txt
Agnieszka
C:\Users\Agnieszka\Desktop>gpg --symmetric --cipher-algo AES192 ploszaj.txt

C:\Users\Agnieszka\Desktop>type ploszaj.txt.gpg
ź~fšIx┐[┴ ┴ĄV|╣↑Aë─3W╩Ä‼ô◄sńÄÓ ShF✲G9Ř[$lôä♣HĚM♠
C:\Users\Agnieszka\Desktop>type ploszaj.txt
Agnieszka
```

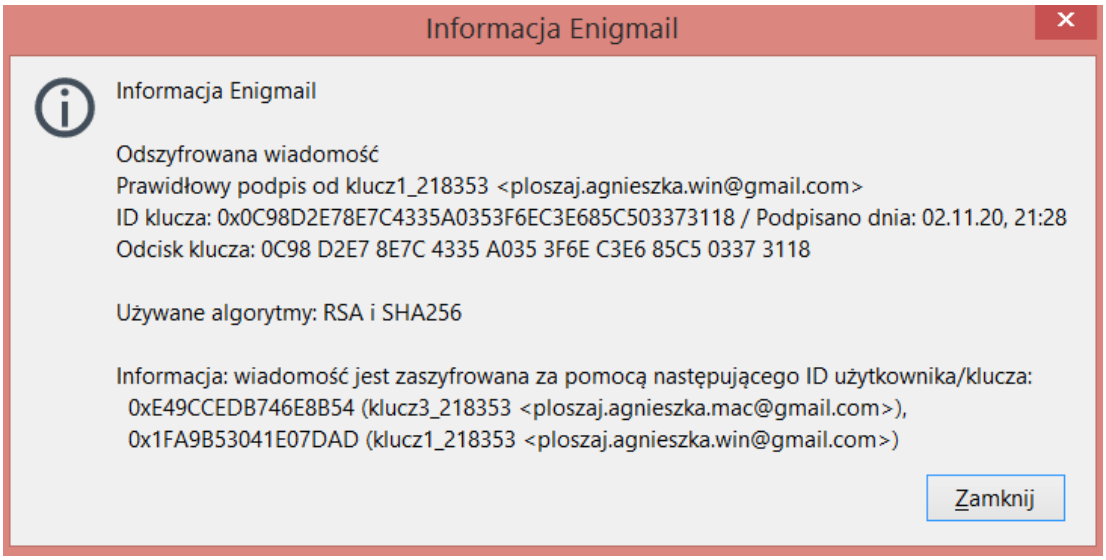
Ilustracja 22 Szyfrowanie symetryczne

W następnej kolejności została obliczona suma kontrolna SHA-1 pliku przy wykorzystaniu nakładki graficznej (ilustracja 23).

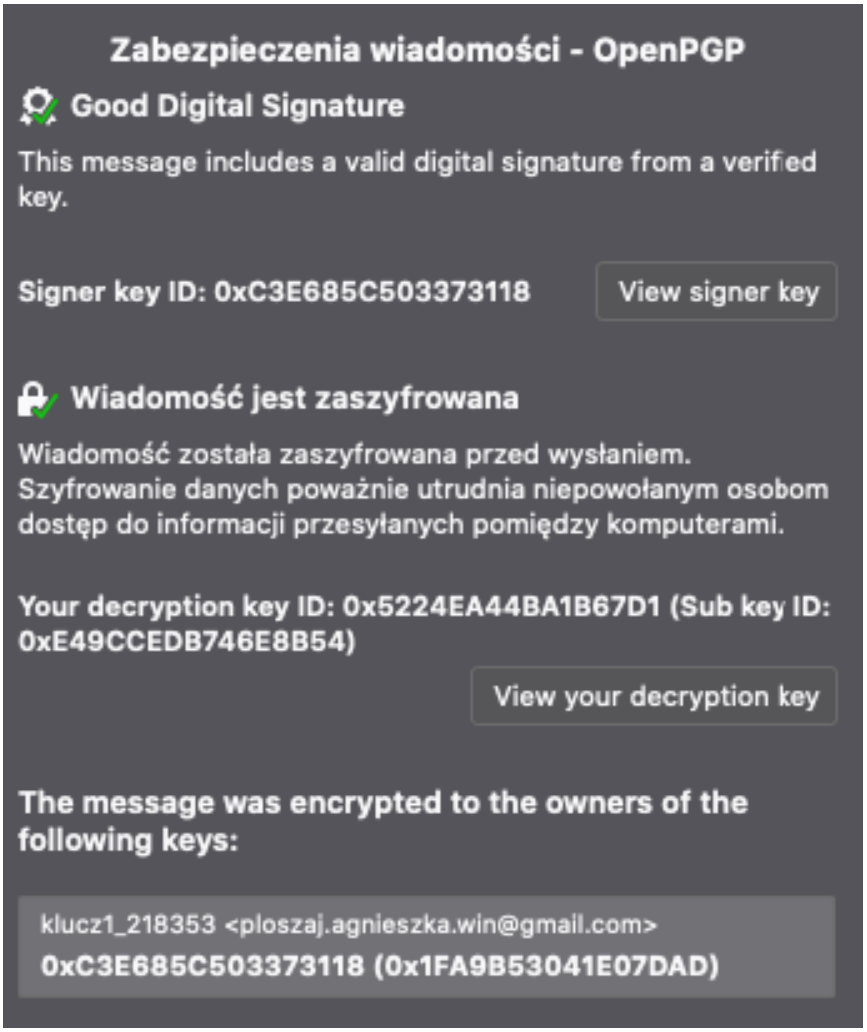



Ilustracja 23 Suma kontrolna SHA-1

E-mail został podpisany cyfrowo i zaszyfrowany, a następnie wysłany z komputera drugiego (ilustracja 24), na komputer pierwszy, gdzie korzystając z Mozilla Thunderbird z wtyczką Enigmail sprawdzono poprawność szyfrowania i podpisu cyfrowego (ilustracja 25).




Ilustracja 24 Informacja Enigmail – komputer drugi




Ilustracja 25 Informacja OpenPGP – komputer pierwszy

Do zadania 13 został użyty plik tekstowy utworzony do zadania 11, w którym zapisane zostało „Agnieszka”. Szyfrowanie we wszystkich poniższych przypadkach odbywało się przy wykorzystaniu konsoli. Zadanie zostało wykonane na komputerze pierwszym (macOS).

Na początek plik został zaszyfrowany domyślnym algorytmem używając polecenia:

**gpg** --**symmetric** ploszaj.txt

W następnej kolejności zostało użyte polecenie:

**od -c** ploszaj.txt.gpg

służące do zrzucania zawartości pliku, a poprzez użycie opcji -c również ustawienie formatu danych wyjściowych na znaki ASCII (gdzie niewidoczne znaki sterujące zostały wyświetlone w postaci sekwencji z użyciem średnika).

Wynik działania powyższych poleceń został pokazany na ilustracji 26. Do sprawdzenia, który algorytm pełni funkcję domyślnego, zostało użyte polecenie:

**gpg -d** ploszaj.txt.gpg

służące do odszyfrowania pliku (jednocześnie podając algorytm za pomocą którego plik został zaszyfrowany (ilustracja 27). Domyślnym algorytmem jest algorytm AES.

```
PA:Desktop coquette$ gpg --symmetric ploszaj.txt
PA:Desktop coquette$ od -c ploszaj.txt
0000000    A   g   n   i   e   s   z   k   a  \n
0000012
PA:Desktop coquette$ od -c ploszaj.txt.gpg
0000000  214  \r 004  \a 003 002 235 345 377 035 234   T 367 344 342 112
0000020    J 001   ]   m   ; 241 235 320 356 365 260 306 031  \b 245 260
0000040  235   m   L 341 340   { 306   x  **   T 254   ; 205 270   Ŷ  **
0000060    .   &   M   ~ 267   M 326   J   E 330 034   ; 002 200   f 275
0000100    +   | 354   ; 245   b   ( 377   u 255   Y   /  ** 345 247   )
0000120  362 235 253 220 336 354   Y 333 354  \a   *
0000133
PA:Desktop coquette$ stat -l ploszaj.txt.gpg
-rw-r--r-- 1 coquette staff 91 Nov  3 23:41:27 2020 ploszaj.txt.gpg
```

Ilustracja 26 Domyślne szyfrowanie symetryczne

```
PA:Desktop coquette$ gpg -d ploszaj.txt.gpg
gpg: dane zaszyfrowano za pomocą AES
gpg: zaszyfrowane jednym hasłem
Agnieszka
```

Ilustracja 27 Algorytm domyślny szyfrowania symetrycznego

Następnie zostały użyte kolejno:

- Algorytm IDEA

```
[PA:Desktop coquette$ gpg --symmetric --cipher-algo IDEA ploszaj.txt
[PA:Desktop coquette$ od -c ploszaj.txt.gpg
0000000  214  \r 004 001 003 002 306 377 251   P 351 222 336 272 342 102
0000020    B 001   R   a  \t 367 365 213 360   O 221 314   \  \f   L  \b
0000040  240 177 370   %   d 253 334   < 223   Q 336 266 365   L   E 150
0000060    h 030   y 247 310   \   #   Y   ~   X 267 341 006     **   k
0000100    N   O 232 372 177 003 223 264   3 035   t   !   &   J 252 147
0000120    g 266   7
0000123
[PA:Desktop coquette$ stat -l ploszaj.txt.gpg
-rw-r--r-- 1 coquette staff 83 Nov  3 23:45:21 2020 ploszaj.txt.gpg
```

Ilustracja 28 Algorytm IDEA

- Algorytm CAMELLIA128

```
[PA:Desktop coquette$ gpg --symmetric --cipher-algo CAMELLIA128 ploszaj.txt
[PA:Desktop coquette$ od -c ploszaj.txt.gpg
0000000  214  \r 004  \v 003 002   y   ? 371   ;   ɜ  **   ŭ  ** 342 112
0000020    J 001 032 375 034 225 231 241 236  \t 202   v   ?   J 244 266
0000040  242 375 262   " 261   )   l 223   c  \t   8 377   T 343 375   }
0000060    B   M   Q   s   0 374 335 372 237 026 347   4   N   _ 244   \
0000100  225 021 203 236 374  \a 227   + 237   `   i 006   ^ 276 270   ,
0000120    K 036   .   h 246   2   7   5 231 234 157
0000133
[PA:Desktop coquette$ stat -l ploszaj.txt.gpg
-rw-r--r-- 1 coquette staff 91 Nov  3 23:46:44 2020 ploszaj.txt.gpg
```

Ilustracja 29 Algorytm CAMELLIA128

- Algorytm TWOFISH

```
[PA:Desktop coquette$ gpg --symmetric --cipher-algo TWOFISH ploszaj.txt
[PA:Desktop coquette$ od -c ploszaj.txt.gpg
0000000  214  \r 004  \n 003 002  \f   D   s 316 333   ij  ** 366 342 112
0000020    J 001 255   e   8   2  ** 261 036 345 256   j   k   Q 246   f
0000040    P 205 375 232 177 232 360 306  \n 364 337   j   } 025 221   -
0000060  274   a 307 031 024 206 233   3   ̩  **   " 245  \b 026 357   =
0000100    B 223   V 233 323   d   d 303   @   U 250 257 006   ؞  **  **
0000120    J   ~  \r 252 246   E 215 241   ,  **   7
0000133
[PA:Desktop coquette$ stat -l ploszaj.txt.gpg
-rw-r--r-- 1 coquette staff 91 Nov  3 23:49:08 2020 ploszaj.txt.gpg
```

Ilustracja 30 Algorytm TWOFISH

Podczas powtarzania tych samych poleceń na różnych plikach tekstowych można było zauważyć, że pewien fragment zrzuconej zawartości pliku jest zawsze taki sam dla danego algorytmu. Fragment ten został zaznaczony na powyższych ilustracjach. Możliwe, że to właśnie sprawdzenie tego znaku, jest niezbędne, żeby można było dowiedzieć się, który algorytm został użyty do zaszyfrowania pliku. Natomiast fragment:
*214 \r 004 znak 003 002*
jest taki sam w przypadku wszystkich sprawdzonych algorytmów symetrycznych i dla różnego typu plików, możliwe więc, że powyższy fragment odpowiada za informację, że plik jest zaszyfrowany przy pomocy algorytmu symetrycznego.

Zbiorcze porównanie algorytmów szyfrowania symetrycznego na podstawie ilustracji 26, 28, 29 oraz 30 prezentuje tabela 2.

Tabela 2 Porównanie algorytmów symetrycznych

| | AES | IDEA | CAMELLIA128 | TWOFISH |
|---|---|---|---|---|
| Wielkość pliku | 91B | 83B | 91B | 91B |
| znak | \a | 001 | \v | \n |