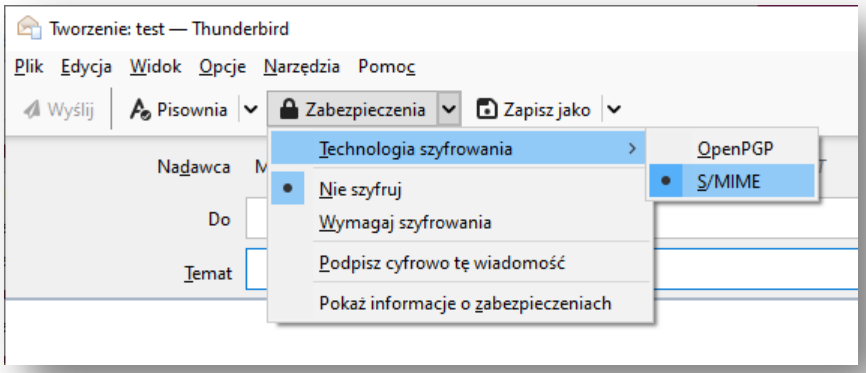
Bezpieczeństwo systemów informatycznych

ĆWICZENIE: Poczta elektroniczna

1. Bezpieczna komunikacja pocztowa

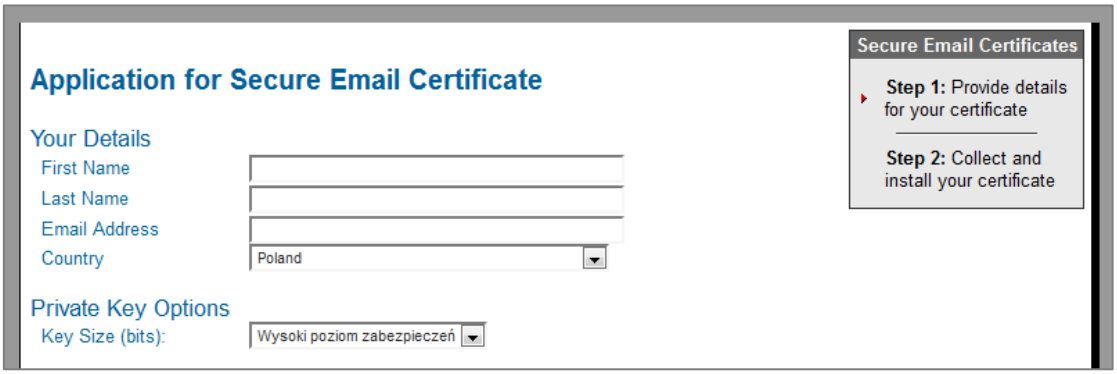
1.1 Wykorzystanie standardu S/MIME

Niektóre programy klientów pocztowych posiadają wbudowaną możliwość wykorzystania mechanizmów kryptograficznych: szyfrowania i / lub podpisywania korespondencji pocztowej zgodnie ze standardem S/MIME (*Secure Multipurpose Internet Mail Extensions*). Taką możliwość posiada m.in. popularny program Thunderbird:



1.2 Certyfikaty adresów pocztowych

W niektórych urzędach certyfikacji można pozyskać darmowy certyfikat standardu X.509 poświadczający wybrany przez siebie adres pocztowy:



Uzyskany w ten sposób certyfikat wraz z kluczami prywatnym i publicznym można wyeksportować do formatu PKCS #12 (`*.pfx`) i następnie zaimportować w kliencie pocztowym obsługującym standard S/MIME. Można wówczas posługiwać się tymi kluczami do szyfrowania i podpisywania korespondencji.

2. System PGP (*Pretty Good Privacy*)

Alternatywnie wobec pocztowego standardu S/MIME, standardy PGP (RFC 1991) oraz OpenPGP (RFC 2440) również umożliwiają podpisywanie i szyfrowanie korespondencji pocztowej (a dodatkowo także plików), oferując certyfikację kluczy publicznych metodą wzajemnego zaufania (*Web of Trust*).

2.1 Zarządzanie kluczami kryptograficznymi

Popularną implementacją standardu PGP jest GnuPG (*Gnu Privacy Guard*). Wszystkie operacje w systemie GnuPG wykonuje uniwersalne narzędzie gpg. Program ten umożliwia użytkownikowi m.in. wygenerowanie pary kluczy asymetrycznych o wybranej długości oraz zarządzanie zbiorami kluczy publicznych innych użytkowników.

2.1.1 Wygenerowanie klucza

```
local> gpg --gen-key
```

Pęki kluczy przechowywane są domyślnie w katalogu ~/.gnupg (lub %AppData%\gnupg w Windows). Bieżącą zawartość pęku posiadanych kluczy publicznych można wyświetlić poleceniem :

```
local> gpg --list-keys
```

lub dla kluczy prywatnych:

```
local> gpg --list-secret-keys
```

Przykładowy wynik działania pierwszego z tych dwóch poleceń może wyglądać następująco:

```
/home/jbond/.gnupg/pubring.gpg
------------------------------
pub  1024D/3BF84E43 2020-06-25 James Bond <agent007@put.poznan.pl>
sub  1024g/8B423A22 2020-06-25
```

Wygenerowany klucz publiczny można udostępnić, np. eksportując do pliku tekstowego w formacie ASCII:

```
local> gpg --export -a -o ~/.gpgkey
```

2.1.2 Pozyskanie czyjegoś klucza publicznego

Pozyskanie klucza publicznego może nastąpić np.:

– z pliku przesłanego lub pobranego (np. poprzez WWW) od właściciela klucza:

```
local> gpg --import plik_z_kluczem
```

– lub z serwera kluczy:

```
local> gpg --keyserver pool.sks-keyservers.net --recv-key 0x3BF84E43
```

2.1.3 Podpisywanie i szyfrowanie wiadomości

Program gpg pozwala podpisać dowolny plik, w szczególności list, swoim kluczem prywatnym:

```
local> gpg --sign plik
```

W powyższym przykładzie wynik pojawi się w postaci skompresowanej w pliku o nazwie plik.pgp. Aby uzyskać podpis w postaci czytelnej (bez kompresji), należy wykonać polecenie:

```
local> gpg --clearsign plik
```

Wówczas powstanie plik tekstowy w formacie ACII o nazwie plik.asc. Nazwę pliku wynikowego możemy zmieniać opcją `-o nazwa_pliku`.

Program gpg pozwala też zaszyfrować wiadomość kluczem publicznym konkretnego odbiorcy:

```
local> gpg --encrypt --recipient odbiorca -at -o list.txt plik
```

a także połączyć szyfrowanie z podpisem elektronicznym:

```
local> gpg -se -r odbiorca -at -o list.txt plik
```

2.1.4 Deszyfrowanie wiadomości i weryfikacja podpisu

Do deszyfrowania i weryfikowania podpisu służą opcje, odpowiednio, `-d` (`--decrypt`) oraz `--verify`:

```
local> gpg -d list
local> gpg --verify list
```

2.1.5 Szyfrowanie plików

Do symetrycznego szyfrowania plików służy opcja `-c` programu gpg:

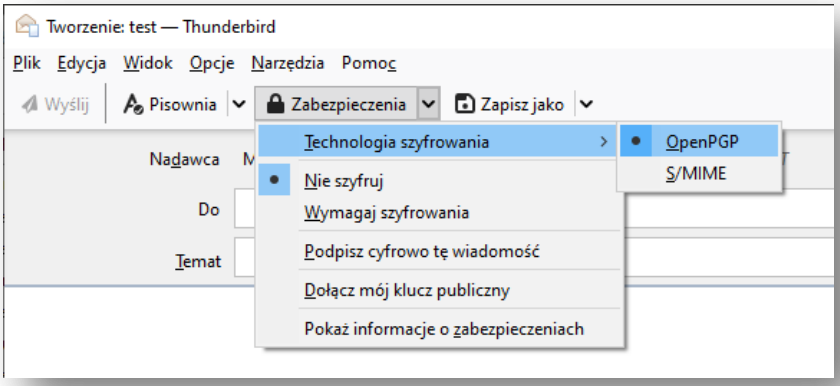
```
local> gpg -c -o szyfrogram plik.txt
```

lub:

```
local> gpg --symmetric --cipher-algo AES256 -o szyfrogram plik.txt
```

2.2 Integracja PGP z klientem pocztowym

Klient pocztowy Thunderbird posiada wbudowane wsparcie dla OpenPGP:



💻 Wykorzystane zasoby:

Mozilla Thunderbird (https://www.thunderbird.net)
GnuPG (https://www.gnupg.org)

🕮 Literatura:

poradnik S/MIME z mozillaZine: https://kb.mozillazine.org/Getting_an_SMIME_certificate
obszerny poradnik PGP: https://www.rossde.com/PGP/

🗒 Problemy do analizy:

- Jaką budowę posiada certyfikat standardu X.509? Do czego pozwala go wykorzystać przeglądarka WWW, a do czego klient pocztowy?

- Zapoznaj się z formatem PKCS #12 plików z certyfikatem.
- Jak potwierdzana jest autentyczność klucza publicznego w systemie PGP?
- Zastanów się od czego mógłby zostać uzależniony poziom zaufania do czyjegoś klucza publicznego w systemie takim jak PGP lub podobnym (w tym: jakie czynniki mogłyby wpływać na wzrost lub spadek poziomu zaufania z biegiem czasu).
- Czy podobnie można zaproponować jakieś kryteria wielopoziomowego zaufania do certyfikatów kluczy publicznych w S/MIME?
- W jaki sposób program gpg wspiera odwołanie (unieważnienie) certyfikatu klucza publicznego użytkownika?