Systemy kryptograficzne
Zarządzanie kluczami

Plan

1. PKI
2. Web Browser i PKI
3. Openssl s_client
4. PGP

PKI

PKI – Public Key Infrastructure:

- CA – *Certification Authority* – urząd certyfikacji – wystawia certyfikaty.
- RA – *Registration Authority* – urząd rejestracji – zbiera wnioski o wydanie certyfikatu, weryfikuje tożsamość subskrybentów.

PKI

PKI wykorzystuje wiele różnych formatów plików w celu przechowywania kluczy i certyfikatów. Poniżej przedstawiam niektóre z najczęściej stosowanych formatów wraz z ich rozszerzeniami plików:

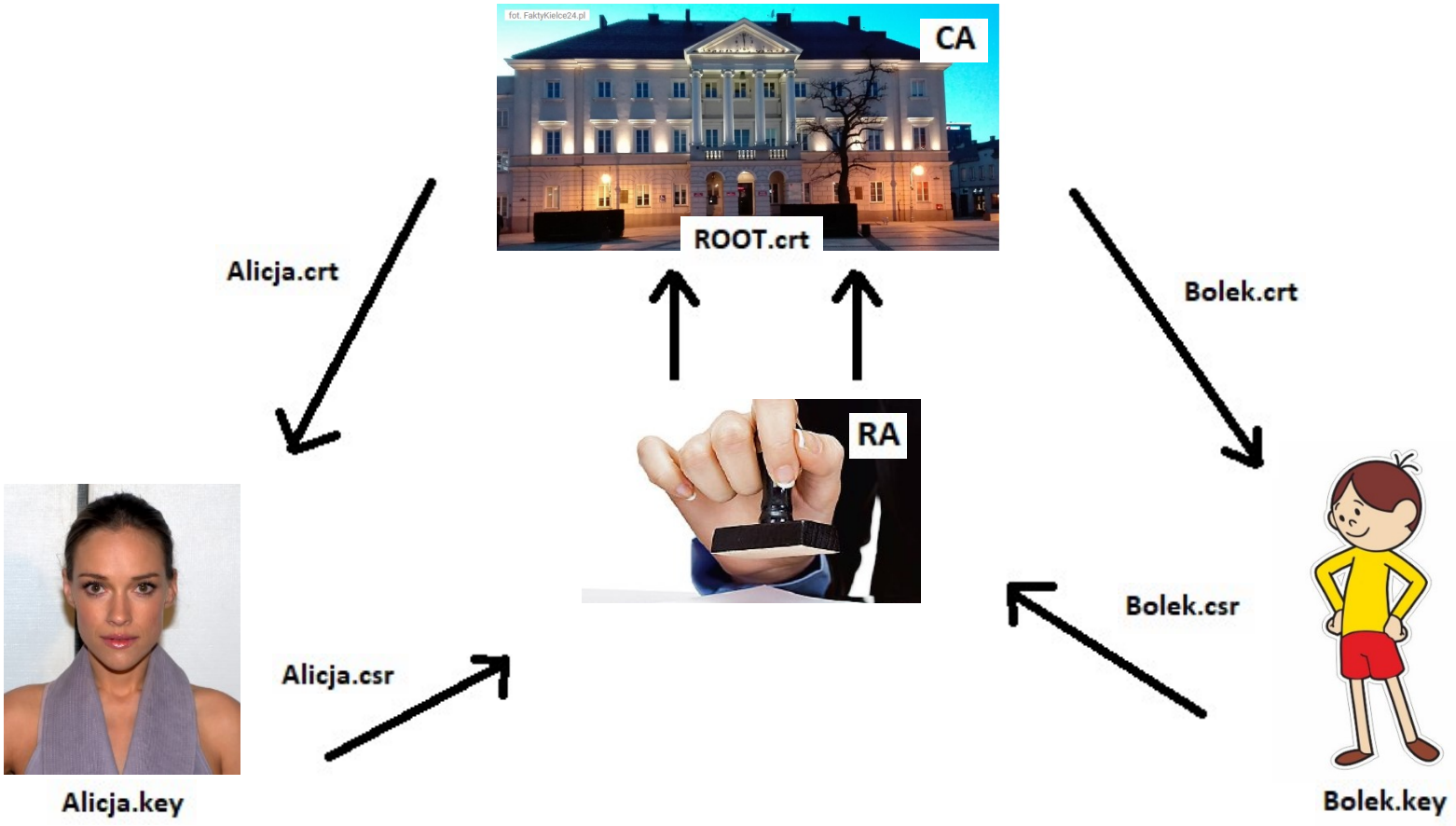
- PEM (Privacy Enhanced Mail) - format tekstowy, często używany do przechowywania kluczy i certyfikatów, rozszerzenie pliku: ".pem", ".key", ".crt", ".cer„, „.csr”.
- DER (Distinguished Encoding Rules) - format binarny, często używany do przechowywania certyfikatów, rozszerzenie pliku: ".der", ".cer".

X.509

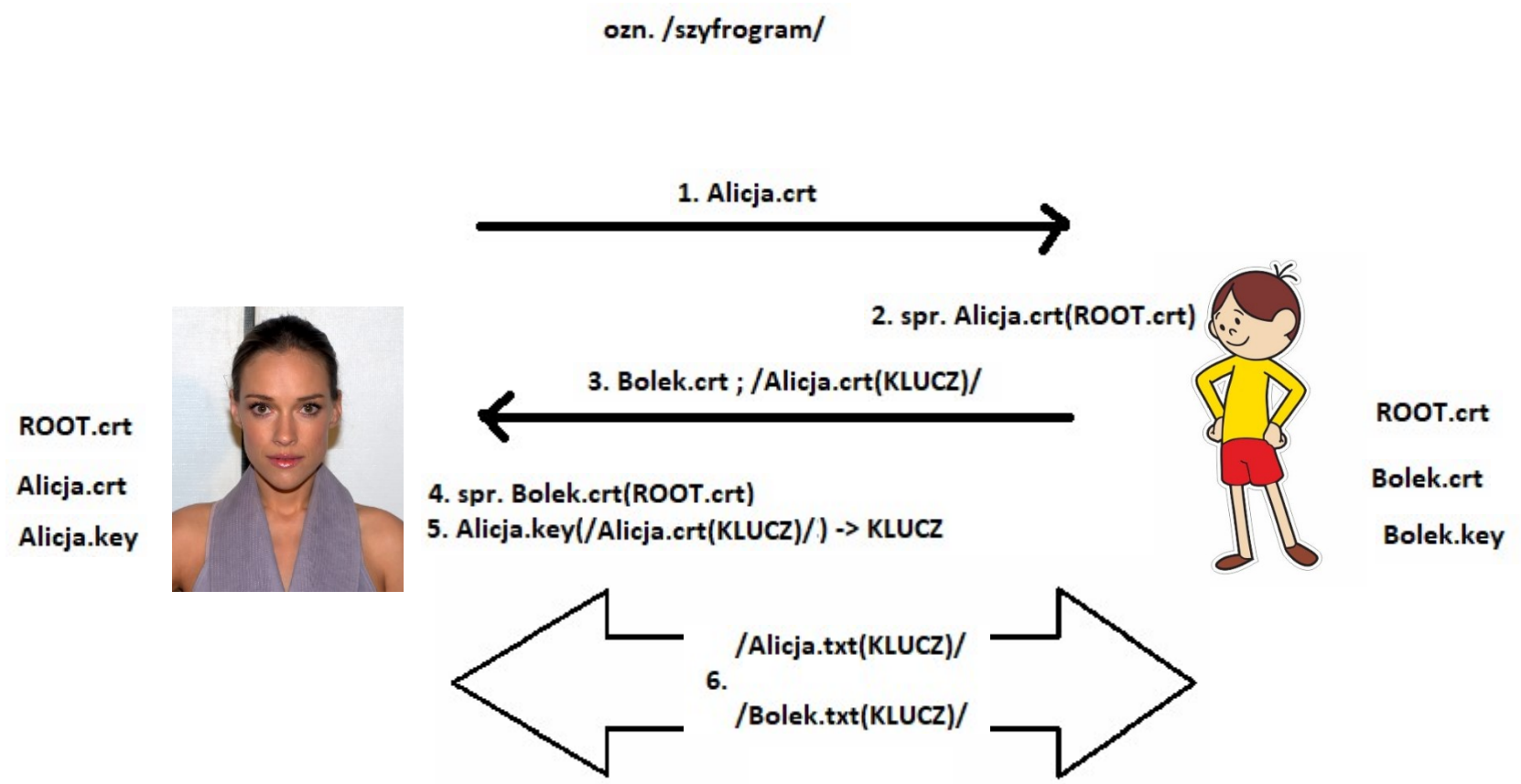
X.509 to standard definiujący format certyfikatów używanych w infrastrukturze klucza publicznego (PKI).

Certyfikaty X.509 są używane do potwierdzania tożsamości użytkowników, serwerów i innych jednostek w sieci. Każdy certyfikat X.509 zawiera publiczny klucz kryptograficzny, informacje o podmiocie (takie jak nazwa i adres e-mail) oraz podpis cyfrowy wydawcy certyfikatu.

PKI

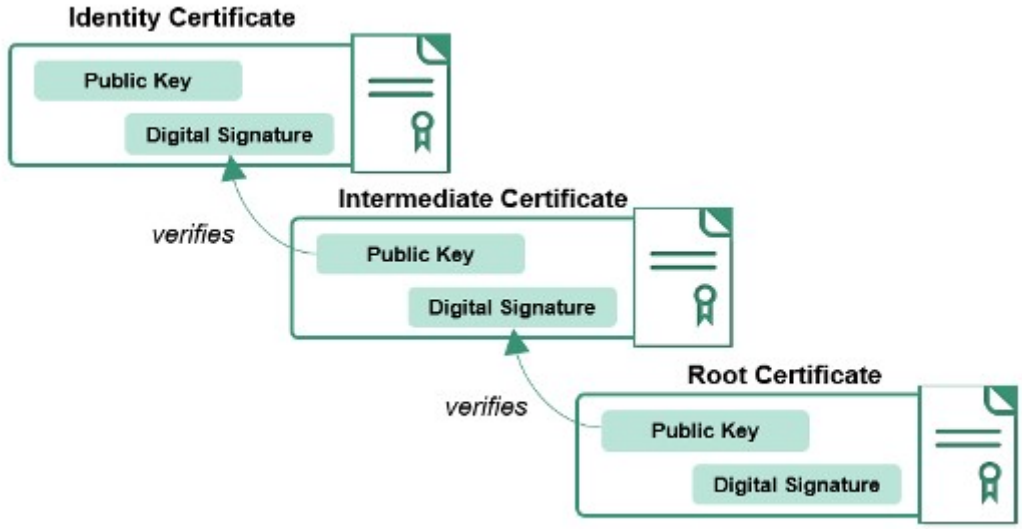

CA
ROOT.crt
Alicja.crt
Bolek.crt
RA
Alicja.csr
Bolek.csr
Alicja.key
Bolek.key

PKI


ozn. /szyfrogram/
1. Alicja.crt
2. spr. Alicja.crt(ROOT.crt)
3. Bolek.crt ; /Alicja.crt(KLUCZ)/
4. spr. Bolek.crt(ROOT.crt)
5. Alicja.key(/Alicja.crt(KLUCZ)/) -> KLUCZ
6.
/Alicja.txt(KLUCZ)/
/Bolek.txt(KLUCZ)/
ROOT.crt
Alicja.crt
Alicja.key
ROOT.crt
Bolek.crt
Bolek.key

PKI - przeglądarki

Przeglądarki internetowe korzystają z certyfikatów,
aby potwierdzić tożsamość witryn internetowych i zapewnić
bezpieczne połączenia z serwerami internetowymi.



https://playbooks.idmanagement.gov/piv/cert-trust/

PKI

W systemach opartych na Unix (takich jak Linux i macOS), magazyn certyfikatów znajduje się w katalogu /etc/ssl/certs.

Windows:

Naciśnij przycisk "Start" i wpisz "mmc.exe" w polu wyszukiwania.

Kliknij prawym przyciskiem myszy na wynik wyszukiwania "mmc.exe" i wybierz "Uruchom jako administrator".

Kliknij menu "Plik" i wybierz "Dodaj/Usuń snap-in".

Wybierz "Menedżer certyfikatów" z listy dostępnych snap-inów i kliknij przycisk "Dodaj".

Wybierz "Konto komputera" i kliknij przycisk "Dalej".

Kliknij "Zakończ" i następnie "OK".

W sekcji "Certificate" wybierz folder "Trusted Root Certification Authorities", aby wyświetlić przechowywane tam certyfikaty roota.

PKI

Przeglądarki internetowe sprawdzają certyfikaty aż do certyfikatu roota, aby potwierdzić tożsamość witryny internetowej i zapewnić bezpieczne połączenie z serwerem internetowym.

Kiedy użytkownik łączy się z witryną internetową przez protokół HTTPS, serwer internetowy przesyła certyfikat SSL (Secure Sockets Layer) lub TLS (Transport Layer Security), który zawiera informacje o tożsamości witryny oraz klucz publiczny, który jest używany do szyfrowania przesyłanych danych. Przeglądarka internetowa weryfikuje ten certyfikat, aby upewnić się, że połączenie jest bezpieczne i że użytkownik łączy się z prawdziwą witryną internetową, a nie z fałszywą stroną stworzoną przez cyberprzestępców.

W procesie weryfikacji certyfikatu przeglądarka sprawdza, czy certyfikat jest ważny i został wydany przez zaufaną instytucję certyfikującą. Przeglądarka następnie łączy się z serwerami certyfikatów, aby zweryfikować podpis cyfrowy certyfikatu, aż do momentu, gdy zostanie odnaleziony certyfikat roota, który jest już uwierzytelniony i zaufany przez przeglądarkę. Jeśli certyfikat roota jest ważny i znajduje się na liście zaufanych certyfikatów przeglądarki, przeglądarka zaakceptuje połączenie i wyświetli witrynę internetową jako bezpieczną.

PKI

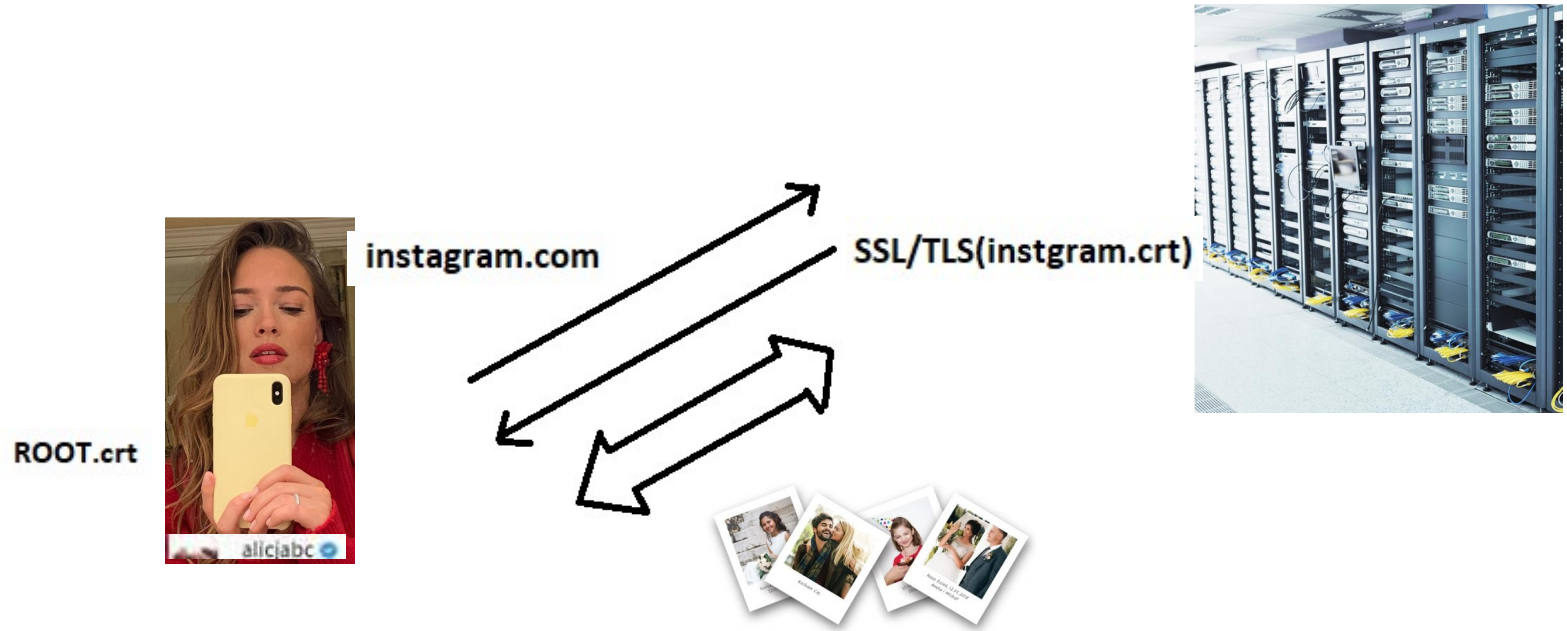
1. Użytkownik wpisuje adres URL witryny internetowej z protokołem HTTPS (np. https://www.przykladowawitryna.com) w pasku adresu przeglądarki i naciska Enter.
2. Przeglądarka nawiązuje połączenie z serwerem internetowym, który hostuje witrynę internetową, poprzez wysłanie żądania HTTP GET.
3. Serwer internetowy odpowiada, przesyłając certyfikat SSL lub TLS, który zawiera informacje o tożsamości witryny oraz klucz publiczny, który jest używany do szyfrowania przesyłanych danych.
4. Przeglądarka weryfikuje certyfikat SSL lub TLS, aby upewnić się, że połączenie jest bezpieczne i że użytkownik łączy się z prawdziwą witryną internetową, a nie z fałszywą stroną stworzoną przez cyberprzestępców.
5. Przeglądarka sprawdza, czy certyfikat jest ważny i został wydany przez zaufaną instytucję certyfikującą.
6. Przeglądarka łączy się z serwerami certyfikatów, aby zweryfikować podpis cyfrowy certyfikatu, aż do momentu, gdy zostanie odnaleziony certyfikat roota, który jest już uwierzytelniony i zaufany przez przeglądarkę.
7. Jeśli certyfikat roota jest ważny i znajduje się na liście zaufanych certyfikatów przeglądarki, przeglądarka zaakceptuje połączenie i wyświetli witrynę internetową jako bezpieczną.

8. Przeglądarka generuje klucz symetryczny, który jest używany do szyfrowania i deszyfrowania przesyłanych danych.
9. Przeglądarka szyfruje klucz symetryczny przy użyciu klucza publicznego, który został przesłany wraz z certyfikatem SSL lub TLS przez serwer internetowy.
10. Przeglądarka przesyła zaszyfrowany klucz symetryczny do serwera internetowego.

Serwer internetowy deszyfruje klucz symetryvczny przy użyciu swojego klucza prywatnego, który jest związany z certyfikatem SSL lub TLS.

11. Po nawiązaniu bezpiecznego połączenia między przeglądarką a serwerem internetowym, wszystkie dane przesyłane między nimi są szyfrowane przy użyciu klucza symetrycznego (AES, 3DES itp), który jest unikalny dla każdego połączenia i jest używany do ochrony danych przed przechwyceniem i odczytem przez osoby trzecie.
12. Przeglądarka wyświetla witrynę internetową z zielonym zamkniętym kłódką na pasku adresu, co oznacza, że połączenie jest bezpieczne i dane są szyfrowane.

PKI



Odczyt certyfikatu

The s_client command implements a generic SSL/TLS client which connects to a remote host using SSL/TLS. It is a very useful diagnostic tool for SSL servers.

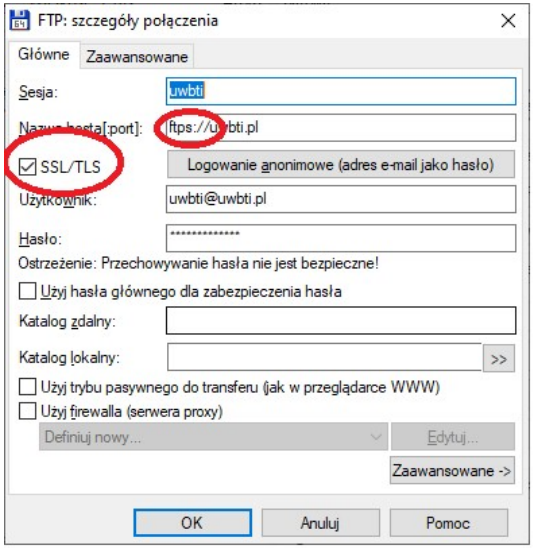
```
openssl s_client -showcerts -connect example.com:443 </dev/null

openssl s_client -connect example.com:443 </dev/null >temp.pem

openssl x509 -in temp.pem -noout -text
```

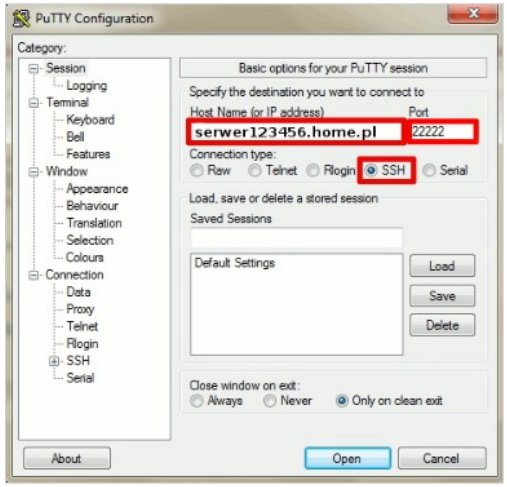
Weryfikacja certyfikatu

```
openssl s_client -connect example.com:443 -showcerts </dev/null > exmpl_chain.pem

cat exmpl_chain.pem

openssl verify -CAfile exmpl_chain.pem /etc/ssl/certs/DigiCert_Global_Root_G2.pem
```

Wireshark

ftp do uwbti.pl:
ftp -p
open uwbti.pl
...





POCZTA

Czy wiecie, że administrator serwera pocztowego MOŻE wiedzieć co piszecie w mailach?

(Fot. 123RF) wyborcza.pl

PGP

PGP (Pretty Good Privacy)
wykorzystuje szyfrowanie symetryczne do szyfrowania treści wiadomości oraz szyfrowanie asymetryczne do szyfrowania klucza symetrycznego, który jest używany do szyfrowania treści wiadomości.

Szyfrowanie treści e-maila rozpoczyna się od wygenerowania losowego klucza symetrycznego, który jest używany do szyfrowania treści wiadomości. Następnie klucz symetryczny jest szyfrowany przy użyciu klucza publicznego odbiorcy, który jest pobierany z serwera kluczy lub z pierścienia kluczy na komputerze nadawcy. Zaszyfrowany klucz symetryczny jest dołączany do wiadomości e-mail jako tzw. "klucz sesji".
Odbiorca wiadomości pobiera zaszyfrowaną treść e-maila wraz z kluczem sesji. Następnie odbiorca używa swojego klucza prywatnego do odszyfrowania klucza sesji, a następnie używa odszyfrowanego klucza do odszyfrowania treści wiadomości.

PGP

man gpg
gpg –h
gpg --delete-secret-keys --all

gpg --gen-key

gpg --export -o Bolek.asc Bolek

gpg --import Bolek.asc

nano message.txt

gpg --encrypt --recipient bolek@bolek.pl message.txt

gpg --decrypt message.txt.gpg

PGP - examples

- gpg -se -r Bob file
- sign and encrypt for user Bob

- gpg --clear-sign file
- make a cleartext signature

- gpg -sb file
- make a detached signature

- gpg -u 0x12345678 -sb file
- make a detached signature with the key 0x12345678

- gpg --list-keys user_ID
- show keys

- gpg --fingerprint user_ID
- show fingerprint

- gpg --verify pgpfile
- gpg --verify sigfile [datafile]
- Verify the signature of the file but do not output the data unless requested. The second form is
- used for detached signatures, where sigfile is the detached signature (either ASCII armored or bi -
- nary) and datafile are the signed data; if this is not given, the name of the file holding the
- signed data is constructed by cutting off the extension (".asc" or ".sig") of sigfile or by asking
- the user for the filename. If the option --output is also used the signed data is written to the
- file specified by that option; use - to write the signed data to stdout.

Systemy kryptograficzne
Zarządzanie kluczami

THE END