

# Infrastruktura klucza publicznego

*Piotr Kucharski*

## Infrastruktura klucza publicznego

- Wstęp do kryptografii
- Podstawy podpisu elektronicznego
- Infrastruktura klucza publicznego

## Wprowadzenie do kryptografii

- Podstawowe terminy
- Podstawowe algorytmy kryptograficzne
- Kryptograficzne funkcje skrótu
- Inne ważne rzeczy
- Kryptoanaliza i ataki na systemy kryptograficzne

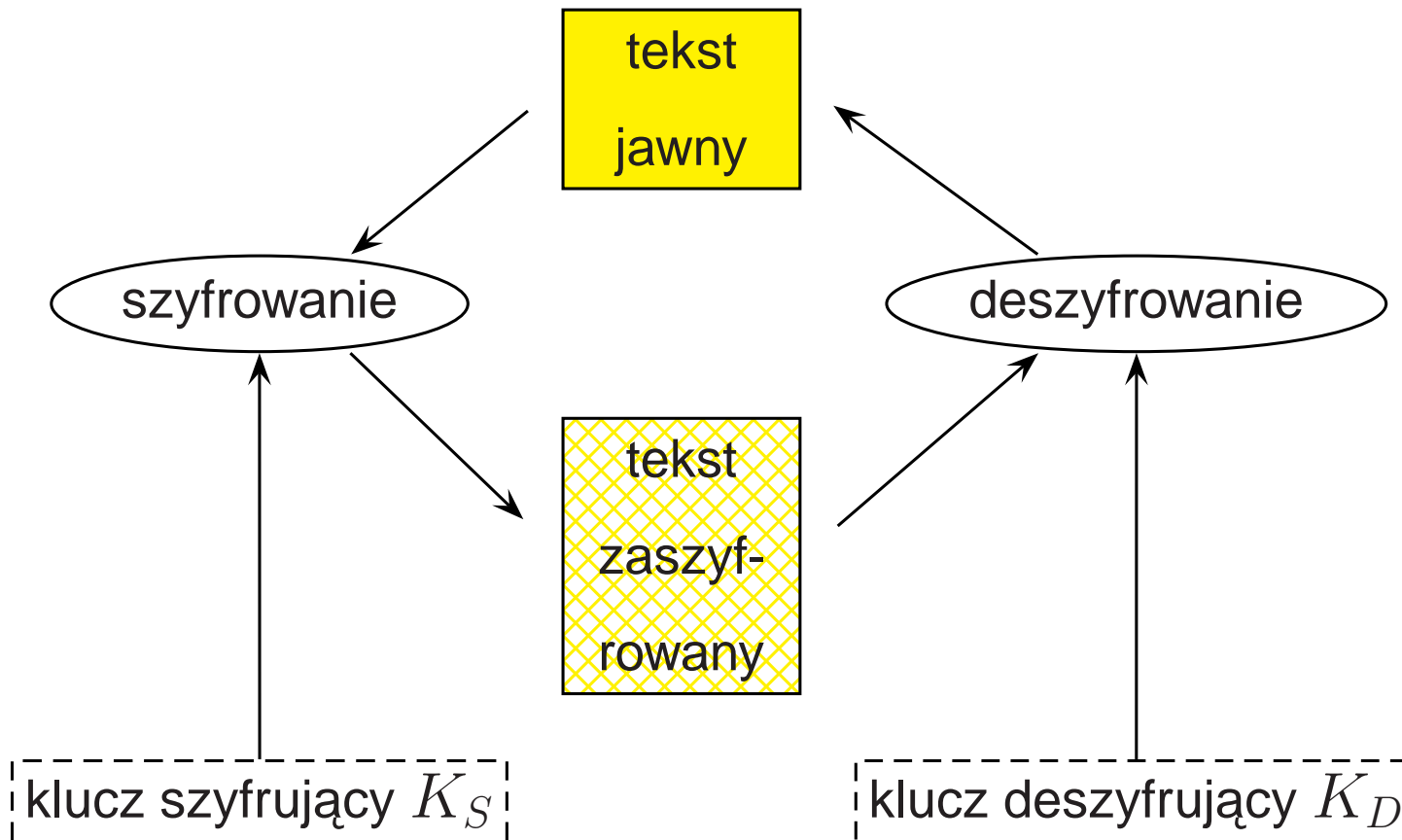
## Kryptografia: podstawowe terminy

**kryptografia** sztuka lub nauka o matematycznych technikach związanych z takimi aspektami bezpieczeństwa, jak poufność, integralność danych, uwierzytelnianie, niezaprzeczalność

**kryptoanaliza** badanie matematycznych metod, które są używane w próbach złamania technik kryptograficznych

**kryptologia** kryptografia i kryptoanaliza

# Kryptografia: podstawowe terminy



## Kryptografia: szyfry

Metoda szyfrowania i deszyfrowania nazywana jest **szyfrem**. Niektóre metody kryptograficzne polegają na tajności algorytmu szyfrującego. Naiwnie.

Holenderski kryptograf Auguste Kerckhoff von Nieuwenhof w 1883 sformułował:

**Zasada Kerckhoffa** *Bezpieczeństwo systemu kryptograficznego nie może być oparte na utrzymywaniu tajności algorytmu, tylko na utrzymywaniu tajności klucza.*

## Kryptografia: algorytmy

Dwie klasy szyfrów:

**symetryczne** (z tajnym kluczem), ten sam klucz jest używany zarówno do szyfrowania, jak i do deszyfrowania, z tego powodu klucz **musi** być utrzymany w sekrecie; te algorytmy możemy podzielić na dwie grupy: szyfry strumieniowe (bit po bicie) i blokowe (64 bity lub więcej na blok).

**asymetryczne** (z kluczem publicznym), jeden klucz jest używany do szyfrowania, a inny do deszyfrowania i *vice-versa*; dzięki temu jeden klucz może być publicznie dostępny, a dla zapewnienia bezpieczeństwa drugi klucz (prywatny) utrzymywany w sekrecie

## Kryptografia: algorytmy symetryczne

**The One-Time Pad** (jednorazowy) udowodniony jako nie do złamania; Vernam w 1917 wymyślił szyfr: tyle samo bitów tekstu jawnego i tyle samo losowego „klucza”. Nie można go użyć drugi raz. W 1949 Shannon udowodnił skuteczność.

**DES** wymyślony w latach 70, NIST zrobił z niego standard, bloki 64-bitowe, klucze 56-bitowe (bezpieczne, ale podatne na przeszukanie przestrzeni wszystkich kluczy), wolny; jego odmiana 3DES poprawia problem krótkiego klucza, ale jest 3 razy wolniejsza...

**AES** National Institute of Standards and Technology zorganizował konkurs na nowy **A**dvanced **E**ncryption **S**tandard. Wszyscy finaliści używali bloków 128-bitowych i akceptowali klucze o długości 128, 192 i 256 bitów.

**Rijndael** Zwycięzca; autorzy: Joan Daemen i Vincent Rijmen; łatwo go zaimplementować sprzętowo, ma krótki czas zestawienia klucza, zużywa mało pamięci (najlepszy atak w 7 rundach na 10... nie tak źle, ciągle wymaga  $2^{120}$  kroków i  $2^{100}$  bajtów pamięci)

**Serpent** autorzy: Anderson, Biham, i Knudsen; nowatorski projekt, może zostać zaimplementowany w logice bramek wektorowych, najbezpieczniejszy, ale wolny (najlepszy atak w 10 z 32 rund)

**Twofish** autorzy: Schneier et al., Counterpane Security; tak pomiędzy Rijndaelem i Serpentem, wygląda całkiem nieźle (najlepszy atak w 8 z 16 rund)

**RC6** autorzy: Rivest, Robshaw i Yin, RSA Laboratories; ewolucja RC5 (najlepszy atak w 17 z 20 rund)

**MARS** autorzy: Zunic et al., IBM; zależy od zestawu instrukcji 32-bitowych procesorów, problemy na innych architekturach, wykonuje wiele operacji, stąd bardzo kosztowny

**inne** Blowfish, CAST-128 (RFC2144), IDEA (uważany jest za bardzo bezpieczny, jest dobrze poznany, ale opatentowany i płatny w celach komercyjnych), Rabbit (zupełnie nowy), RC4 (podobno źródło wypłynęło, nieznane bezpieczeństwo, szybki, wariant OTP)

**przed 1970** Fish (niemieckie dowódzwo armii 2WŚ, doprowadził do powstania pierwszego komputera: Colossus), Enigma (rotory, polscy kryptoanalitycy, Bletchley Park (dziś nie jest bezpieczny)), szyfr Vernama, szyfry podstawieniowe (w tym przesuwające: Cezar, rot13), inne starożytne

**ROT13** stosowany m.in. w Usenecie („fgfbjnal j Hfrarpvr”)

## **Kryptografia: algorytmy asymetryczne**

W późnych latach 70 zaobserwowano, że w oparciu o bardzo, bardzo trudny problem, którego rozwiązanie zajęłoby tysiące lat, mógłby powstać system kryptograficzny z dwoma kluczami, publicznym (do szyfrowania) i prywatnym (do deszyfrowania).

Wymiana kluczy. Zanotowano, że kryptograficzny system z kluczem publicznym mógłby zostać użyty do utworzenia tajnego klucza używanego w szyfrowaniu symetrycznym dużej ilości danych. Whitfield Diffie i Martin Hellman skonstruowali protokół wymiany klucza, który na zawsze zmienił oblicze świata kryptografii.

Niedługo później Ron Rivest, Adi Shamir i Leonard Adleman utworzyli pierwszy prawdziwy system kryptograficzny z kluczem publicznym.

W ogólności system kryptograficzny z kluczem publicznym jest zbudowany z trudnego problemu:

1. weź trudny problem (NP-zupełny), dla którego możesz znaleźć szczególną postać rozwiązywalną w krótkim czasie (trapdoor function with high computational complexity)
2. przekształć jawny tekst do takiej łatwej postaci trudnego problemu
3. użyj klucza publicznego do przekształcenia łatwej postaci z powrotem na trudny problem
4. prześlij po niezabezpieczonym kanale
5. użyj klucza prywatnego do przekształcenia trudnego problemu w łatwą postać
6. rozwiąż łatwą postać w celu uzyskania jawnego tekstu

**RSA** autorzy: **R**ivest, **S**hamir i **A**dleman; najczęściej używany algorytm klucza publicznego; rozkładanie na czynniki iloczynu dwóch dużych (gigantycznych!) liczb pierwszych;

Interaktywny tutorial o RSA <http://www.youdzone.com/rsa.html>

**ElGamal** podobny do DH, wolniejszy, ale bez obciążeń patentowych

**Diffie-Hellman** protokół wymiany klucza z 1976(!); pozwala dwóm stronom bez uprzedniego wspólnego tajnego klucza stworzyć taki; mogą potem użyć go jako klucza w kryptografii symetrycznej (szybkie szyfry blokowe) lub jako podstawę do wymiany klucza; problem logarytmów dyskretnych.

## Schemat protokołu Diffie-Hellman

Mamy liczbę pierwszą  $p = 23$  i generator  $g = 3$  (matematycznie starannie dobrane)

**Alicja** wybiera losowo liczbę  $a = 6$

oblicza  $A = g^a \bmod p = 3^6 \bmod 23 = 16$

i wysyła  $A = 16$  do Bartka

**Bartek** wybiera losowo liczbę  $b = 15$

oblicza  $B = g^b \bmod p = 3^{15} \bmod 23 = 12$

i wysyła  $B = 12$  do Alicji

**wspólny sekret**  $s = g^{a \cdot b} \bmod p$  możemy obliczyć tak:

- Alicja zna  $a$  i  $B$  i oblicza  $s = B^a \bmod p = 12^6 \bmod 23 = 9$
- Bartek zna  $b$  i  $A$  i oblicza  $s = A^b \bmod p = 16^{15} \bmod 23 = 9$

## Kryptograficzne funkcje skrótu

„Kompresują” wiadomość do stałego rozmiaru **wartości skrótu** (hash).  
Tak zaprojektowane, żeby różne wiadomości dawały różne wartości skrótu.  
(Nie do końca to możliwe.) Ale przynajmniej niewielkie zmiany wiadomości dają zupełnie inny wartości skrótu i jest trudno znaleźć dwie takie same wiadomości, które by dawały taką samą wartość skrótu (*kolizja*).

**MD5** rozwijane przez RSA Labs, opublikowane w RFC1321, skraca do wartości 128 bitowych, np.:

MD5 (irc2.11.1p1.tgz) = c5a2b3097a5fbeb91b39412730b02ab5

**RIPEMD-160** miał zastąpić MD5, daje skróty 160 bitowe

**SHA-1** opublikowany przez rząd USA (FIPS PUB 180-1), 160-bit, ma dłuższe (256, 384, 512) odmiany

## Inne ważne rzeczy

**MAC** (message authentication codes) kody autoryzujące wiadomość; oparte na skrótach (HMAC-SHA-1) lub blokowych szyfrach symetrycznych (DES-CBC-MAC), używane do weryfikacji integralności wiadomości

**dzielony tajny klucz** rozdaje się klucz  $N$  ludziom (lub więcej) i gwarantuje, że mniej niż  $N$  osób nie będzie w stanie dostać całego klucza

**generatory liczb losowych** bardzo często używane w kryptografii (jak klucze); potrzebujemy przynajmniej 128 bitów entropii

**prawdziwe** fizyczne, sprzętowe (przepuszczanie półprzewodników) szum, najmniej znaczący bit wejścia audio, czas między przerwami a klawiszami naciskanymi przez użytkownika

**pseudo** najlepiej wziąć trochę fizycznego (nawet jeśli małego) i zamieszać;  
ANSI X9.17; FIPS-186 (DSS)

***GLL musi być bardzo dobry... albo system krypto będzie popsuty.***

# Kryptoanaliza i ataki na systemy kryptograficzne

**brute-force** aka przeszukiwanie całej przestrzeni kluczy

**man in the middle** wejdź między Alicję i Bartka, rób za pośrednika

**kryptoanaliza** sztuka odszyfrowywania zaszyfrowanego

**atak na wiadomość zaszyfrowaną** nic nie wiemy

**atak ze znanym tekstem jawnym** odszyfruj resztę lub zdobądź klucz

**atak z wybranym tekstem jawnym** w celu zdobycia klucza

**przetwarzania kwantowe** rozwiąż *WSZYSTKIE* równania w jednej chwili

**czynnik ludzki** oszukiwanie

## Podstawy podpisu elektronicznego: wstęp

- Podpis tradycyjny
- Podpis elektroniczny
  - koncept
  - tworzenie i weryfikacja
  - problemy
  - przykład

## Podpis tradycyjny

Podpis nie jest treścią transakcji, raczej jej reprezentacją.

Podpis tradycyjny służy następującym celom:

**dowodowy** odręczny podpis jest cechą podpisującego, więc łączy dokument z podpisującym

**ceremonialny** kładzie nacisk na istotę prawną, co pozwala zapobiegać „nierozważnym zobowiązaniom”

**aprobujący** niektóre akty prawne wymagają formy pisemnej pod rygorem nieważności

**skuteczność i logistyka** daje poczucie jasności i ostateczności (negocjacje) i jest łatwy do zrobienia!

*Zapewnia strony o ważności i wymagalności transakcji.*

## Podpis elektroniczny

Zalety dokumentów elektronicznych (łatwość duplikacji, szybki transport, przetwarzania przez komputer) są równocześnie ich wadami. Potrzeby:

**uwierzytelnienie podpisującego** podpis musi wskazywać, kto podpisał  
które dane i powinien być trudny do podrobienia

**uwierzytelnienie dokumentu** podpis musi wskazywać, co dokładnie zostało  
podpisane, bez możliwości podrobienia lub zmiany dokumentu lub  
podpisu

**wydajność** podpis, jego tworzenie i weryfikacja powinny zapewniać możliwie  
największe zaufanie dla podpisującego i autentyczności dokumentu  
przy najmniejszym możliwym koszcie

*To jest zadanie usługi niewypieralności w PKI.*

Podpis elektroniczny korzysta z systemów kryptografii z kluczem publicznym, które używają dwóch matematycznie związanych ze sobą kluczy:

**klucz prywatny** znany tylko podpisującemu,  
używany do tworzenia podpisu

**klucz publiczny** znany powszechnie,  
używany do weryfikowania podpisu

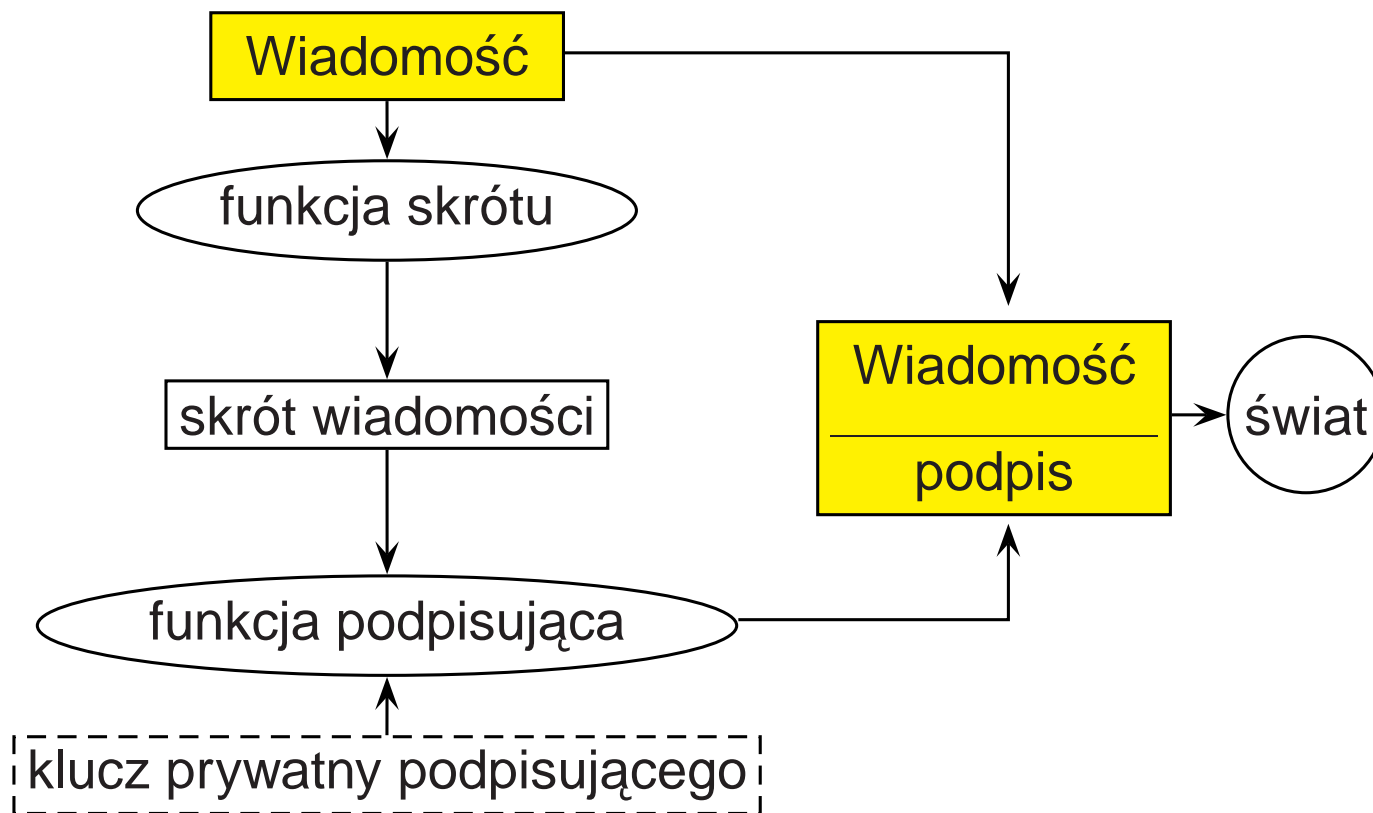
Klucze są związane, ale jest praktycznie niemożliwe obliczeniowo wymyślenie klucza prywatnego na podstawie publicznego.

Ze względu na wydajność nie jest podpisywana cała wiadomość (choć krótkie powinny być), a tylko jej skrót

**skrót wiadomości, hash** wynik funkcji skrótu (hashującej, mieszającej) zaaplikowanej na całej wiadomości

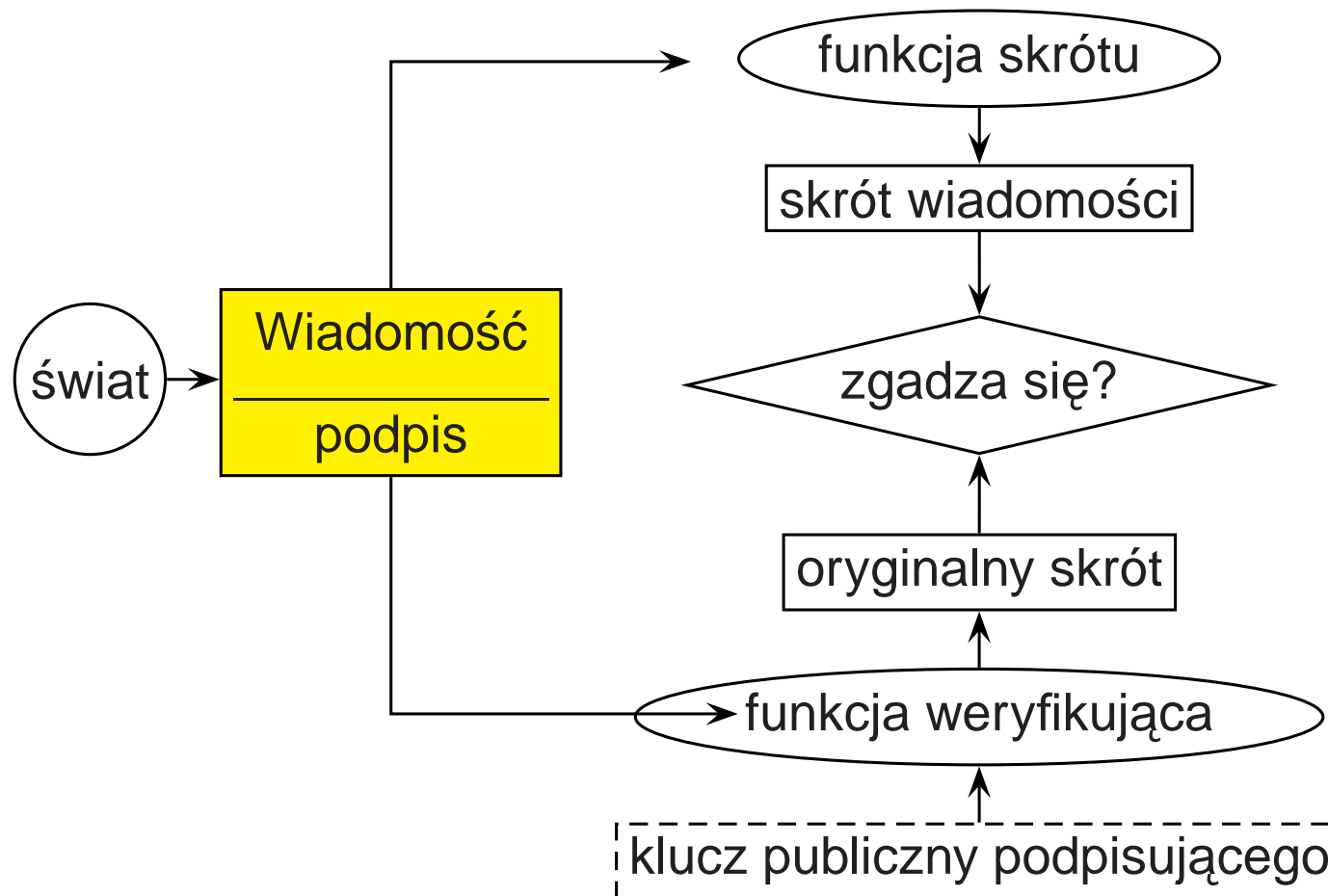
**funkcja skrótu** jednokierunkowy mechanizm produkujący unikatowy, stałej wielkości, „odcisk palca” danych wejściowych

## Tworzenie podpisu elektronicznego



Uwaga: funkcja podpisująca to de facto szyfrowanie skrótu wiadomości

# Weryfikacja podpisu elektronicznego



Uwaga: funkcja weryfikująca to de facto deszyfrowanie podpisu

## Problemy z podpisem elektronicznym

**funkcja skrótu** polegamy na niej, a z definicji istnieje możliwość kolizji. . . może być *więcej niż jedna* wiadomość, która daje tę samą wartość skrótu

⇒ podrobienie

kolizje MD5 <http://www.cits.rub.de/MD5Collisions/>

**wiadomość** podpisujący musi mieć pełną świadomość, co podpisuje

**klucze** skąd wiadomo, czyj jest dany klucz? ⇒ Infrastruktura Klucza Publicznego

**PKI** czy możemy jej wierzyć?

## Przykład podpisu elektronicznego w PGP

```
% cat test.txt
```

```
Cześć, to przykładowy plik na zajęcia z kryptografii  
Mój PIN do komórki: 981cff74a11a7ffacab4ce26656b3f55
```

```
% pgp -sba -u chopin@sgh.waw.pl test.txt
```

```
Pretty Good Privacy(tm) 2.6.3ia - Public-key encryption for the masses.  
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-03-04  
International version - not for use in the USA. Does not use RSAREF.
```

```
A secret key is required to make a signature.  
You need a pass phrase to unlock your RSA secret key.  
Key for user ID: Piotr Kucharski <chopin@sgh.waw.pl>  
1024-bit key, key ID D818A489, created 1995/10/31
```

```
Enter pass phrase: s3kr3t... Pass phrase is good. Just a moment...  
Transport armor file: test.txt.asc
```

```
% cat test.txt.asc
```

```
-----BEGIN PGP MESSAGE-----
```

```
Version: 2.6.3ia
```

```
iQCVAwUAQr899kABjqnYGKSJAQFabgP9GsOfNMs25xNTrMPcdNEodZfmltpzTSJg5  
3vEyR200sVD9txVkl1qchhIRznssgYNIeTOYQwVrt1XRedgEnkAUyO8iRjPn4DQm  
ETP17NReV3bcwHZI5SFdNsZiLNe7YIujycIbywuP4C/a/MrteAdp5tLgHTBX01l/  
uFVYM6fnoLw=
```

```
=IRV1
```

```
-----END PGP MESSAGE-----
```

## Sprawdźmy

```
% pgp +verbose=0 test.txt.asc -o /dev/null
Good signature from user "Piotr Kucharski <chopin@sgh.waw.pl>".
Signature made 2005/06/20 23:20 GMT using 1024-bit key, key ID D818A489
```

Teraz usuwam linijkę z PIN-em i sprawdzam jeszcze raz.

```
% pgp +verbose=0 test.txt.asc -o /dev/null
WARNING: Bad signature, doesn't match file contents!
```

```
Bad signature from user "Piotr Kucharski <chopin@sgh.waw.pl>".
Signature made 2005/06/20 23:20 GMT using 1024-bit key, key ID D818A489
```

Przywracam linijkę z PIN-em, ale zmieniam podpis i sprawdzam jeszcze raz.

```
% pgp +verbose=0 test.txt.asc -o /dev/null
ERROR: Bad ASCII armor checksum
```

```
Error: Transport armor stripping failed for file test.txt.asc
```

```
For a usage summary, type: pgp -h
For more detailed help, consult the PGP User's Guide.
```

## Infrastruktura Klucza Publicznego (PKI)

- model
- oferowane usługi
- centra certyfikujące
- model zaufania
- problemy

## **PKI: model infrastruktury**

**infrastruktura** w ogólności: jak prąd czy sieć lan

**potrzeby** związane z aplikacjami, bezpieczne (i raz) logowanie, przezroczystość

**zyski** oszczędność kosztów, swobodna wymiana informacji (wewnętrzna i zewnętrzna), jednorodne rozwiązania (łatwość administracji), *droga do możliwego osiągnięcia bezpieczeństwa*

**model infrastruktury** na następnym slajdzie

## Infrastruktura

---

- centra certyfikacyjne
- archiwizacja kluczy
- zarządzania historią kluczy
- repozytoria certyfikatów
- odzyskiwanie kluczy
- certyfikacja krzyżowa
- odwoływanie certyfikatów
- automatyczna aktualizacja kluczy
- oprogramowanie użytkownika

## Usługi

---

- uwierzytelnianie
- bezpieczny znacznik czasu
- bezpieczna archiwizacja danych
- integralność
- notariat
- tworzenie uprawnień/polityki
- poufność
- niezaprzeczalność
- weryfikacja uprawnień/polityki

## Przegląd infrastruktury PKI

**Centrum certyfikujące (CA)** zaufana trzecia strona, wiąże dany klucz publiczny z jakimś podmiotem przez wydawania certyfikatów podpisanych przez CA

**Repozytorium certyfikatów** mechanizm dostarczający certyfikaty; wydajność, skalowalność, dostępność on-line

**Odwoływanie certyfikatów** zmiana nazwy lub kradzież klucza prywatnego – musi być sposób, żeby oznaczyć dany certyfikat jako nieważny; publikowane listy certyfikatów odwołanych (CRL), używanie Online Certificate Status Protocol (OCSP)

**Archiwizacja i odzyskiwanie kluczy** klucze (do podpisu i szyfrowania) będą gubione (ułomna ludzka pamięć lub uszkodzenie sprzętu) – dla organizacji może być nie do zaakceptowania, żeby *nie* mogli odszyfrować zaszyfrowanych danych

**Automatyczna aktualizacja kluczy** certyfikaty nie są wieczne – jeśli aktualizacja będzie obowiązkiem użytkownika, to na pewno zostanie zapomniana

**Zarządzanie historią kluczy** nowe klucze nie będą w stanie odszyfrować starych danych zaszyfrowanych starymi kluczami (a reszyfrowanie jest praktycznie niewykonalne)

**Krzyżowa certyfikacja** uniwersalna (światowa) PKI to mit; różne współpracujące systemy PKI potrzebują sposobu, żeby ich użytkownicy sobie ufali

**Oprogramowanie klientów** cała infrastruktura (serwery) jest niczym bez klientów; to klienci ściągają certyfikaty, przetwarzają CRL-e, weryfikują procedury...

## Przegląd usług PKI

**Uwierzytelnianie** zapewnienie jednego podmiotu, że drugi podmiot jest tym, za kogo się podaje

**Integralność** zapewnienie, że dane nie zostały zmienione

**Poufność** zapewnienie, że nikt nie może odczytać danych oprócz explicite wybranych odbiorców

**Bezpieczny znacznik czasu** sekwencje zdarzeń nie mogą pozostawiać wątpliwości

**Notariat** certyfikacja poprawności danych (acz „poprawność” to bardzo subiektywny termin)

**Niezaprzeczalność** jak podpiszesz, nie możesz twierdzić, że nie podpisałeś; nie można być na 100% pewnym, ale to dużo pomaga

**Bezpieczne archiwum danych** wszystkie stare certyfikaty, listy CRL, dokumenty ze znacznikami czasu itd. muszą być zachowywane w celach dowodowych

**Tworzenie uprawnień/polityki** mechanizmy do tworzenia szczegółowej polityki kontroli dostępu dla osób lub grup (a w zasadzie ich certyfikatów)

**Weryfikacja uprawnień/polityki** mechanizmy do sprawdzania tego, co wyżej

## Podstawowe usługi PKI: uwierzytelnianie

**Uwierzytelnianie** to zapewnienie jednego podmiotu, że drugi podmiot jest tym, za co się podaje. W pewien sposób.

**Uwierzytelnienie podmiotu** bez zważania na to, co robi; to pierwszy krok przed sprawdzaniem czegokolwiek innego

- lokalne uwierzytelnienie: poziom podstawowy, zwykle wymaga osobistej interwencji i objawia się w sposób bezpośredni i oczywisty (przyciśnięcie kciuka, wpisanie hasła itp.)
- zdalne uwierzytelnienie: gdy użytkownik korzysta z serwisów zdalnych, które potrzebują wiedzieć, kim ten użytkownik jest; może bezpośrednio angażować użytkownika, ale z powodu problemów z zabezpieczaniem danych w transporcie oraz (co ważniejsze) dyskomfortem użytkownika (może nie być SSO), przetwarzany jest tylko *wynik lokalnego uwierzytelnienia*

**Uwierzytelnienie podmiotu** można uzyskać na wiele sposobów

**coś, co masz** smartcard czy token

**coś, co wiesz** hasło lub PIN

**coś, czym jesteś** unikatowy odcisk palca, wzór siatkówki

**coś, co robisz** wzór odręcznego pisma

Pojedynczy czynnik jest mniej bezpieczny niż kombinacja. Świat biznesu zwykle używa dwóch czynników, np. smartcard (PIN i chip elektroniczny), securID (kod tokena i PIN).

**Identyfikacja źródła danych** identyfikuje podmiot jako źródło jakichś danych

– statycznie i nieodwołalnie (umożliwia działanie usługi niezaprzeczalności)

## Podstawowe usługi PKI: integralność

- zapewnienie niezmienności (celowej lub nie), zarówno w trakcie transportu (między *tam* a *tu*), jak i przechowywania (między *wtedy* a *teraz*)
- bity parzystości, kody CRC mogą wykryć przypadkowe błędy w pojedynczych bitach, ale nic nie poradzą na świadomych włamywaczy
- rozwiązaniem jest podpisywanie swoim kluczem prywatnym (jeden do wielu (w zasadzie to pochodna uwierzytelniania) lub kluczem publicznym odbiorcy (jeden do jednego) — nikt poza właścicielami klucza nie będzie mógł manipulować danymi *oraz* zapewnić integralności podpisu

Kody Uwierzytelniające Wiadomość (Message Authenticating Code, MAC) są zbudowane z blokowych szyfrów symetrycznych np. DES-CBC-MAC (FIPS113) lub kryptograficznych funkcji skrótu np. HMAC-SHA-1 (RFC2104) – używają kryptografii symetrycznej, ale klucze są zwykle rozpowszechniane przy pomocy PKI.

Co ma zrobić Alicja, by uwierzytelnić wiadomość dla Bartka?

| jeśli Bartek ma klucz publiczny PKI                     | jeśli Bartek nie ma takiego klucza   |
|---|--|
|   | niech Bartek wygeneruje sobie klucz publiczny DH i użyje go z jej kluczem prywatnym DH, by |
| wygenerować nowy klucz symetryczny                      |  |
| użyć tego klucza do podpisania danych kodem MAC         |  |
| zaszyfrować klucz symetryczny kluczem publicznym Bartka |  |
| dołączyć zaszyfrowany klucz symetryczny do danych       | dołączyć jej publiczny klucz DH do danych  |
| wysłać dane do Bartka                                   |  |

## Podstawowe usługi PKI: Poufność

Poufność to zapewnienie podmiotu, że nikt nie może odczytać danej wiadomości oprócz *explicite* wybranych odbiorców. Pierwsza i oczywista myśl: zaszyfrować wszystkie dane kluczem publicznym odbiorcy.

To. Jest. P.O.W.O.L.N.E... Lepiej użyć szybkiej kryptografii symetrycznej (żeby uprościć, to niezależnie od wielkości danych):

- utwórz klucz symetryczny
  - użyj go (przy użyciu szyfru blokowego) do zaszyfrowania danych
  - wyślij zaszyfrowane dane do Bartka
    - z własnym publicznym kluczem DH lub
    - z kopią klucza symetrycznego zaszyfrowanego kluczem publicznym
- Bartka

## Usługi dodane PKI: bezpieczna komunikacja

Transmisja danych zabezpieczona przez podstawowe usługi PKI (uwierzytelnianie, integralność, poufność) i nałożone na istniejącą infrastrukturę sieciową daje *bezpieczną komunikację*:

**e-mail** S/MIMEv2 (RFC2311, RFC2312) lub PGP

**web** SSL/TLS (RFC2246)

**VPN** IPsec/IKE (RFC2401, RFC2411)

## Usługi dodane PKI: bezpieczny znacznik czas

- kolejność zdarzeń musi być poza wątpliwością
- zwłaszcza gdy korzystamy z certyfikatów, ich czasem życia i odwołaniami
- zwłaszcza gdy używamy podpisu elektronicznego

## Usługi dodane PKI: wsparcie niezaprzeczalności

Produkt innych usług PKI: uwierzytelniania i bezpiecznego znacznika czasu.

- gdy już raz bezpiecznie podpiszesz dokument (czy jako autor, czy jako odbiorca), **nie możesz** wyprzeć się faktu podpisania
- chyba że jesteś w stanie dowieść, że podpis został wcześniej skradziony... może w sądzie uwierzą

## Usługi dodane PKI: notariat

- notariusz (trzecia strona) zaświadcza, że jakieś dane są „poprawne”
- co jest „poprawne”, to bardzo subiektywna sprawa

## Usługi dodane PKI: zarządzanie uprawnieniami/polityką

- mechanizmy do tworzenia szczegółowej kontroli dostępu dla użytkowników i grup (czy raczej ich certyfikatów)
- kto (certyfikat klienta) może zrobić co i kiedy
- czasem zawarte wprost w certyfikacie, ale to jest trudne w zarządzaniu

## Certyfikaty

**X.509v3** standard ISO, z kluczem publicznym, ogólne zaufanie do jednego centrum, próbuje powiązać osoby (ich nazwiska) z pojedynczymi certyfikatami na całym świecie; opisane przez IETF jako PKIX w RFC 3280

**SET** Secure Electronic Transaction: płatności kartami kredytowymi, „prywatne” rozszerzenie X.509v3

**Atrybuty** oparte na rolach, RFC 3281, bazowane na X.509, bez klucza publicznego, podstawa dla Infrastruktury Zarządzania Uprawnieniami

**SPKI/SDSI** (Simple Public Key Infrastructure/Simple Distributed Security Infrastructure), odłącza nazwiska osób od certyfikatów; miało być proste, ale nikt tego nie używa

**certyfikaty PGP** (Pretty Good Privacy), przenosi „władzę” z globalnej na każdego użytkownika oddzielnie (więcej o tym później, na slajdzie Modele Zaufania)

## struktura X.509 (RFC 3280)

- Certyfikat
  - Version** identyfikuje wersję certyfikatu
  - Serial Number** niepowtarzalny w obrębie danego CA identyfikator liczbowy certyfikatu
  - Algorithm ID** algorytm użyty do podpisania certyfikatu
  - Issuer** jednoznaczna nazwa wydawcy certyfikatu
  - Validity** ważne nie przed i nie po
  - Subject** jednoznaczna nazwa właściciela
  - Subject Public Key Info** klucz publiczny (i jego algorytm) właściciela
  - Issuer Unique ID** opcjonalny niepowtarzalny identyfikator wydawcy
  - Subject Unique ID** opcjonalny niepowtarzalny identyfikator właściciela
  - Extensions** opcjonalne rozszerzenia
- Algorytm podpisu elektronicznego certyfikatu
- Podpis elektroniczny certyfikatu

## Rozszerzenia X.509

**Authority Key Identifier** niepowtarzalny identyfikator klucza służącego do weryfikacji podpisu elektronicznego certyfikatu

**Subject Key Identifier** niepowtarzalny identyfikator klucza publicznego w tym certyfikacie (dla rozróżnienia różnych kluczy tego samego właściciela)

**Key Usage** zastrzega użycie klucza do jednego lub więcej pól eksploatacji: podpis elektroniczny, niezaprzeczalność, szyfrowanie kluczy, szyfrowanie danych, uzgadnianie kluczy, podpisywanie certyfikatów, podpisywanie list CRL, tylko szyfrowanie, tylko deszyfrowanie

**Extended Key Usage** lista identyfikatorów określających użycie klucza publicznego: uwierzytelnianie TLS serwera, uwierzytelnianie TLS klienta, podpisywanie kodu, zabezpieczenie poczty, znacznik czasu, podpisywanie OCSP

**CRL Distribution Point** miejsce (zwykle URI), skąd można ściągnąć najnowsze listy CRL (pełne lub różnicowe)

**Private Key Usage Period** kiedy klucz prywatny może być używany (może

być inne niż czas życia klucza publicznego); pomaga zarządzać kluczami

**Certificate Policies** identyfikatory polityki i opcjonalnych kwalifikatorów: praktyki centrum certyfikującego (Certification Practice Statement, CPS) i informacje dla użytkownika

**Policy Mappings** identyfikatory polityk ekwiwalentnych między dwoma CA

**Subject Alternative Name** alternatywne nazwy właściciela certyfikatu, np. adres e-mail, IP, URL itp. (musi być wypełnione, jeśli pole **Subject** jest puste)

**Issuer Alternative Name** alternatywne nazwy wydawcy certyfikatu

**Basic Constraints** podstawowe zastrzeżenia: podmiot końcowy (jeśli fałsz)  
lub certyfikat CA (jeśli prawda)

**Path Length Constraint** maksymalna liczba CA, które mogą być w łańcuchu  
certyfikatów za tym certyfikatem

**Name Constraints** dozwolone/wykluczone poddrzewa atrybutów w postaci  
DN, URI, e-mail, czegokolwiek hierarchicznego; pozwala elastycznie  
zarządzać (delegując władzę/uprawnienia)

**inne** Subject Directory Attributes, Policy Constraints, Inhibit Any Policy,  
Freshest CRL Pointer, prywatne rozszerzenia np. Authority Information  
Access czy Subject Information Access

```
% openssl s_client -connect www.sgh.waw.pl:443 -showcerts | \
openssl x509 -noout -text
```

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 208317 (0x32dbd)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=PL, O=Unizeto Sp. z o.o., CN=Certum Level III

Validity

Not Before: Apr 18 16:29:54 2006 GMT

Not After : Apr 18 16:29:54 2007 GMT

Subject: C=PL, O=Szkola Glowna Handlowa, OU=Centrum Informatyczne,  
CN=\*.sgh.waw.pl/emailAddress=unix@sgh.waw.pl

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

```
00:db:4d:35:90:3e:18:1c:c2:d8:f1:a1:e3:f4:da:
54:4c:f0:5f:8e:48:36:09:5a:63:79:83:66:8e:4f:
ff:9c:82:4e:12:3f:e6:3a:df:cc:62:bd:0d:7f:da:
9d:ff:92:2a:72:0b:47:c2:a8:4a:ad:ad:7d:e2:e4:
a4:c5:7f:2b:d3:27:82:a1:6e:a4:dc:52:1e:91:78:
83:b1:be:6c:b7:fb:da:9f:ce:17:8e:24:a0:09:0e:
13:79:0f:97:07:56:77:2f:7b:4f:11:8c:1e:2f:1c:
55:0e:15:ae:0c:0d:09:cb:fa:39:cd:0c:40:91:7c:
f7:a2:65:ef:b3:1e:de:65:3b
```

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Extended Key Usage:

TLS Web Server Authentication

X509v3 Basic Constraints:

CA:FALSE

X509v3 CRL Distribution Points:

URI:http://crl.certum.pl/class3.crl

Authority Information Access:

OCSP - URI:http://ocsp.certum.pl

X509v3 Certificate Policies:

Policy: 1.2.616.1.113527.2.2.3

CPS: http://www.certum.pl/CPS

User Notice:

Organization: Unizeto Sp. z o.o.

Number: 1

Explicit Text: Usage of this certificate is strictly subjected to the Certum Certification Practice Statement (CPS) incorporated by reference herein and in the Certum Repository at <https://www.certum.pl/repository>. This CPS is also available by mail at Unizeto Sp. z o.o. 70-486 Szczecin, Krolowej Korony Polskiej 21, Poland. Copyright (c) 1998-2004 Unizeto Sp. z o.o. All Rights Reserved.

Signature Algorithm: sha1WithRSAEncryption

8c:db:01:95:75:33:ff:2e:4a:8e:24:16:97:3b:7f:65:73:f0:  
e3:a7:23:61:d3:2a:f4:40:97:58:20:b7:69:39:70:b8:63:5d:  
2c:3e:c8:8b:ce:a6:86:02:64:df:4c:7f:a2:55:5b:d7:2f:e4:  
cd:8e:0c:16:42:10:ec:e7:8a:0c:96:f0:ff:cc:2d:06:bf:11:  
2f:7b:ff:94:39:5a:08:54:08:31:d9:31:ea:e3:d4:bd:e0:cc:  
4d:d2:23:64:f3:db:f9:8c:35:f8:61:d7:1e:2d:e2:0f:be:7f:  
bf:8c:f1:ae:10:52:f0:f5:f5:f4:fd:25:94:7e:0b:46:aa:17:  
ce:fb

## Centra Certyfikacyjne

Po angielsku Certificate Authorities, CA.

CA dają poczucie pewności, że klucz publiczny zawarty w certyfikacie faktycznie należy do podmiotu wymienionego w certyfikacie.

Podpis elektroniczny „złożony” przez CA na certyfikacie zapewnia *kryptograficzne powiązanie* między kluczem publicznym podmiotu, jego nazwą i innymi informacjami w certyfikacie, jak np. okresem ważności.

## Strony PKI

**centra certyfikacyjne** (certification authorities) wydają certyfikaty

**abonenci** (subscribers) podmioty (indywidualni lub organizacje) korzystające z klucza prywatnego odpowiadającego publicznemu w certyfikacie

**klienci** (relying parties) podmioty polegające na certyfikacie

**centra rejestracyjne** podmioty wspierające CA w procesie uwierzytelniania tożsamości lub innych przymiotów aplikującego o certyfikat, inicjujące odwołanie certyfikatu na prośbę abonenta, akceptujące lub odrzucające żądania zmiany kluczy w certyfikacie

**repozytoria** podmioty świadczące usługi publikacji, przechowywania i dostępu do certyfikatów, list CRL i innych dokumentów PKI

## Podstawowe obowiązki CA

**rejestracja abonentów** identyfikacja i uwierzytelnianie (często delegowane do centr rejestracyjnych (Registration Authorities, RA))

**bezpieczne zarządzanie certyfikatem** tworzenie i wydawanie, dystrybucja, odnawianie certyfikatów i podpisywanie nowych kluczy (rekey), odwoływanie, zawieszanie

**zapewnienie informacji o statusie certyfikatu** wydawanie list odwołanych certyfikatów (CRL), utrzymywanie mechanizmów sprawdzania statusu on-line (OCSP)

**dostęp do certyfikatów i list CRL** przez Repozytorium Certyfikatów (katalog online, może być zarządzany inną firmę) dostępnego dla klientów

## Rejestracja abonenta

- abonent tworzy swoją parę kluczy publiczny + prywatny
- abonent dostarcza jakiś dowód tożsamości – zależnie od polityki certyfikatu i wymaganego poziomu ufności
- abonent pokazuje, że jest w posiadaniu klucza prywatnego (podpisanie jakichś danych, CA zweryfikuje kluczem publicznym)
- CA wydaje certyfikat
- CA podpisuje certyfikat własnym kluczem prywatnym
- abonent publikuje certyfikat (żeby inni mogli z niego korzystać)
- CA publikuje certyfikat w repozytorium

## Centra certyfikacyjne: podstawowa biurokracja

**Polityka Certyfikacyjna** (Certificate Policy, CP) określony zestaw zasad opisujących dozwolone przypadki użycia certyfikatu w konkretnym zastosowaniu czy klasie aplikacji o wspólnych wymaganiach co do bezpieczeństwa

**Komunikat o Praktykach Certyfikacyjnych** (Certification Policy Statement, CPS) opis praktyk i procedur, bardziej szczegółowy niż w CP – gdy CP mówi o wymaganiach, CPS opisuje, jak CA je spełnia

**Porozumienie Abonenckie** wbrew nazwie: między CA a RA; koncentruje się na odpowiedzialności abonenta i dozwolonych warunkach użycia certyfikatu

**Porozumienie Klienckie** określa warunki, po spełnieniu których klient może polegać na certyfikacie (jak np. sprawdzenie statusu certyfikatów w łańcuchu)

**Komunikat Opisujący PKI** prosto i zwięźle najważniejsze punkty z CP i CPS

**inne porozumienia** dotyczące kompatybilności, porozumienia producentów, porozumienia związane z dystrybucją certyfikatów, wewnętrzne porozumienia związane z poziomem zaufania PKI

**inne biznesowe praktyki CA** plan zarządzania kluczami, instrukcje i polityka bezpieczeństwa, instrukcje treningów, używania, instalacji i dla użytkownika, przewodniki dla działu kadr i książki dla pracowników, polityka używania e-maili...

## **Centra Certyfikujące: więcej biurokracji**

Opisy praktyk biznesowych CA dotyczące integralności usługi:

**Kontrola środowiska CA** opisuje środowisko pracy

**Zarządzanie kluczami** opisuje procedury związane z kluczami samego CA

**Zarządzanie cyklem życia certyfikatów** opisuje procedury związane z certyfikatami abonentów

## Kontrola środowiska CA

- CPS i zarządzanie polityką dot. certyfikatów
- zarządzanie bezpieczeństwem
- klasyfikacja i zarządzanie zasobami
- bezpieczeństwo osobiste
- bezpieczeństwo fizyczne i środowiskowe
- zarządzanie operacyjne
- zarządzanie dostępem do systemów
- zarządzanie i rozwój systemów
- zarządzanie planami ciągłości działania
- monitorowanie i zgodność
- zapisywanie zdarzeń

## Zarządzanie kluczami samego CA

- tworzenie kluczy CA
- przechowywanie, archiwizacja i odtwarzanie kluczy CA
- dystrybucja kluczy publicznych CA
- (opcjonalne) odzyskiwanie klucza CA
- używanie klucza CA
- niszczenie klucza CA
- archiwizacja
- zarządzanie cyklem życia sprzętu kryptograficznego CA
- (opcjonalne) usługi zarządzania kluczami abonenta

## Zarządzanie cyklem życia certyfikatów

- rejestracja abonentów
- (opcjonalne) odnawianie certyfikatów
- ponowne podpisywanie kluczy w certyfikacie (rekey)
- wydawanie certyfikatów
- dystrybucja certyfikatów
- odwoływanie certyfikatów
- (opcjonalne) zawieszanie certyfikatów
- przetwarzanie informacji o statusie certyfikatów

## Odwoływanie Certyfikatów

- certyfikat wiąże podmiot z kluczem publicznym
- jeśli podmiot zmienia nazwę (nazwisko, pracę) lub klucz prywatny przestał być tajny... trzeba odwołać certyfikat!
- **opóźnienie odwołania** to czas pomiędzy uzyskaniem informacji, że certyfikat powinien zostać odwołany, a wysłaniem samej informacji o odwołaniu – bardzo ważny czynnik

- metody

**publikacje okresowe** jak listy certyfikatów odwołanych (CRL)

**systemy online** jak Online Certificate Status Protocol

## Publikacje okresowe (listy CRL)

**Pełna lista CRL** problemy ze skalowalnością (duża!) i poprawnością danych

**Lista Odwołanych Centrów** (Authority Revocation Lists) do odwoływania certyfikatów CA, korzysta z rozszerzenia CRL nazwie "issuing distribution point"

**Punkty Dystrybucji CRL** (częściowe listy CRL) kilka list CRL z jednego CA, łatwiejsze w zarządzaniu, mogą wskazywać na URI; jednak muszą być statyczne

**Rozszerzone Punkty Dystrybucji CRL i Przekierowane CRL** wskaźniki do częściowych list CRL, ograniczone jakimś filtrem

**Listy różnicowe** różnice od ostatniej publikacji pełnej listy; małe i często

**Pośrednie listy CRL** sumy listy CRL z różnych CA (łatwiejsza weryfikacja)

**Drzewa Odwołanych Certyfikatów** (Certificate Revocation Trees) wymyślone przez Valicert, oparte na drzewach skrótu Merkle'a, wydajnie prezentują duże listy odwołanych certyfikatów

## Systemy online

Online certificate status protocol (OCSP) RFC2560

**żądanie** wersja protokołu, żądanie usługi, identyfikatory certyfikatów

**odpowieź (ogólnie)** wersja, nazwa odpowiadającego, odpowiedzi na każdy z certyfikatów z pytania, dodatkowe rozszerzenia, algorytm podpisu, podpis

**odpowieź na certyfikat** identyfikator docelowego certyfikatu, status (dobry, odwołany, nieznany), ważność odpowiedzi, dodatkowe rozszerzenia

## CRL

**Listy Odwołanych Certyfikatów** (Certificate Revocation Lists) elektronicznie podpisane struktury danych

- zawierają listę odwołanych certyfikatów
- integralność i uwierzytelnienie dzięki podpisowi elektronicznemu
- podpisujący to zwykle wydawca odwołanego certyfikatu
- mogą być trzymane w pamięci podręcznej

**CRL v1** specyfikacja X.509 z 1998; problemy:

- skalowalność (taka lista może być niezwykle duża)
- problemy z rozszerzaniem
- atak przez podłożenie innej listy

**CRL v2** dodaje ważną poprawkę: *rozszerzenia*

## struktura CRL

**version** 2 lub brak (wtedy 1)

**signature** algorytm podpisu (np. md5WithRSAEncryption)

**issuer** jednoznaczna nazwa wydawcy CRL

**thisUpdate** data wydania

**nextUpdate** data następnego wydania (aka czas ważności tego wydania)

**revokedCertificates** lista numerów seryjnych odwołanych certyfikatów, czas odwołania (plus, tylko w wersji 2, ewentualne rozszerzenia per wpis)

**extensions** tylko w wersji 2

- rozszerzenia dla każdego wpisu (numeru seryjnego certyfikatu)
  - reason code** nieznan, utracony klucz, utracone CA, zmiana afiliacji, nadpisany, zaprzestanie działania, wstrzymanie, usunięcie z CRL, wycofanie uprawnień...
  - invalidity date** czas nieważności kluczy prywatnych i publicznych
  - certificate issuer** wystawca certyfikatu dla przekierowanej listy CRL
  - hold instruction code** co zrobić przy zawieszeniu certyfikatu (nic, zadzwonić do wystawcy, odrzucić)
- rozszerzenia dla samej listy CRL
  - authority key identifier**
  - issuer alternative name**
  - CRL number**
  - issuing distribution point**
  - Delta CRL indicator**

```
% wget http://crl.thawte.com/ThawteServerCA.crl
% openssl crl -inform der -text -noout -in ThawteServerCA.crl
Certificate Revocation List (CRL):
    Version 1 (0x0)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: /C=ZA/ST=Western Cape/L=Cape Town/O=Thawte Consulting
           cc/OU=Certification Services Division/CN=Thawte Server
           CA/emailAddress=server-certs@thawte.com
    Last Update: Dec  1 11:00:00 2006 GMT
    Next Update: Dec 29 11:00:00 2006 GMT
Revoked Certificates:
    Serial Number: 4B3D02FD9DCA2807A8E4FD4CC029385F
    Revocation Date: Oct 27 05:30:19 2006 GMT
    Serial Number: 4B463AB9B2FF900A6377EF14A330458B
    Revocation Date: Oct 23 20:47:28 2006 GMT
    Serial Number: 4B48D9D9606511B0B1AE3B61F00C9D8B
    Revocation Date: Mar 22 13:18:47 2006 GMT
[... ]
Signature Algorithm: sha1WithRSAEncryption
cb:f8:f2:51:cf:96:a2:97:9a:8d:eb:7b:c3:71:5d:8e:e0:b9:
83:1d:7e:3a:d1:ff:1b:f5:e5:27:3c:e4:12:20:0a:29:55:11:
60:da:df:2d:ec:08:e7:95:e2:f5:0a:1e:24:f3:d5:27:fe:6a:
e6:c1:a6:90:57:c2:e0:f8:03:25:17:8c:50:00:98:c7:de:a9:
cc:68:2f:6f:c1:d6:c3:93:bd:2e:c5:be:70:76:20:9f:3b:44:
2a:2e:d7:20:04:fc:b9:14:2a:cb:b8:2b:41:09:58:b4:b3:75:
52:52:ad:df:c0:36:b9:2d:d5:85:47:7b:45:4b:70:a7:a1:3a:
79:aa
```

## Gargantuizm PKI

**infrastruktura** centra certyfikujące, repozytoria certyfikatów, odwoływanie certyfikatów, archiwizacja i odzyskiwanie kluczy, automatyczna aktualizacja kluczy, zarządzanie historią kluczy, certyfikacja krzyżowa, oprogramowanie klienckie

**usługi** uwierzytelnianie, integralność, poufność, bezpieczny znacznik czasu, notariat, niezaprzeczalność, bezpieczne archiwum danych, tworzenie uprawnień/polityki, weryfikacja uprawnień/polityki

Czy to *wszystko* jest naprawdę potrzebne?

www i poczta (ssl)

|                           |              |          |
|---------------------------|--------------|----------|
| Centrum<br>Certyfikacyjne |              |          |
|                           |              |          |
|                           |              |          |
| Uwierzytelnianie          | Integralność | Poufność |
|                           |              |          |
|                           |              |          |

www i poczta (ssl) **które są ważne**

|                           |  |  |
|---------------------------|--|--|
| Centrum<br>Certyfikacyjne |  | <b>Odwoływanie<br/>Certyfikatów</b>      |
|                           |  |  |
|                           |  | <b>Oprogramowanie<br/>Klienckie</b>      |
| Uwierzytelnianie          | Integralność                           | Poufność                                 |
|                           |  |  |
|                           | <b>Tworzenie<br/>Uprawnień/Polityk</b> | <b>Weryfikacja<br/>Uprawnień/Polityk</b> |

www i poczta (ssl) **które są ważne dla kilku instytucji**

|                                |                                |                                  |
|--------------------------------|--------------------------------|----------------------------------|
| Centrum<br>Certyfikacyjne      | Repozytorium<br>Certyfikatów   | Odwoływanie<br>Certyfikatów      |
| Archiwizacja Kluczy            |                                |                                  |
| Zarządzanie Historią<br>Kluczy | Krzyżowa<br>Certyfikacja       | Oprogramowanie<br>Klienckie      |
| Uwierzytelnianie               | Integralność                   | Poufność                         |
| Bezpieczny Znacznik<br>Czasu   |                                | Niezaprzeczalność                |
|                                | Tworzenie<br>Uprawnień/Polityk | Weryfikacja<br>Uprawnień/Polityk |

www i poczta (ssl) **które są ważne dla kilku instytucji rządowych**

|                                |                                |                                     |
|--------------------------------|--------------------------------|-------------------------------------|
| Centrum<br>Certyfikacyjne      | Repozytorium<br>Certyfikatów   | Odwoływanie<br>Certyfikatów         |
| Archiwizacja Kluczy            | Odzyskiwanie Kluczy            | Automatyczna<br>Aktualizacja Kluczy |
| Zarządzanie Historią<br>Kluczy | Krzyżowa<br>Certyfikacja       | Oprogramowanie<br>Klienckie         |
| Uwierzytelnianie               | Integralność                   | Poufność                            |
| Bezpieczny Znacznik<br>Czasu   | Notariat                       | Niezaprzeczalność                   |
| Bezpieczne<br>Archiwum Danych  | Tworzenie<br>Uprawnień/Polityk | Weryfikacja<br>Uprawnień/Polityk    |

## Model Zaufania PKI

Komu możesz ufać?

**zaufanie** zdefiniowane przez ITU-T:

*Podmiot "A" ufa podmiotowi "B", gdy "A" zakłada, że "B" zachowa się dokładnie tak, jak "A" oczekuje*

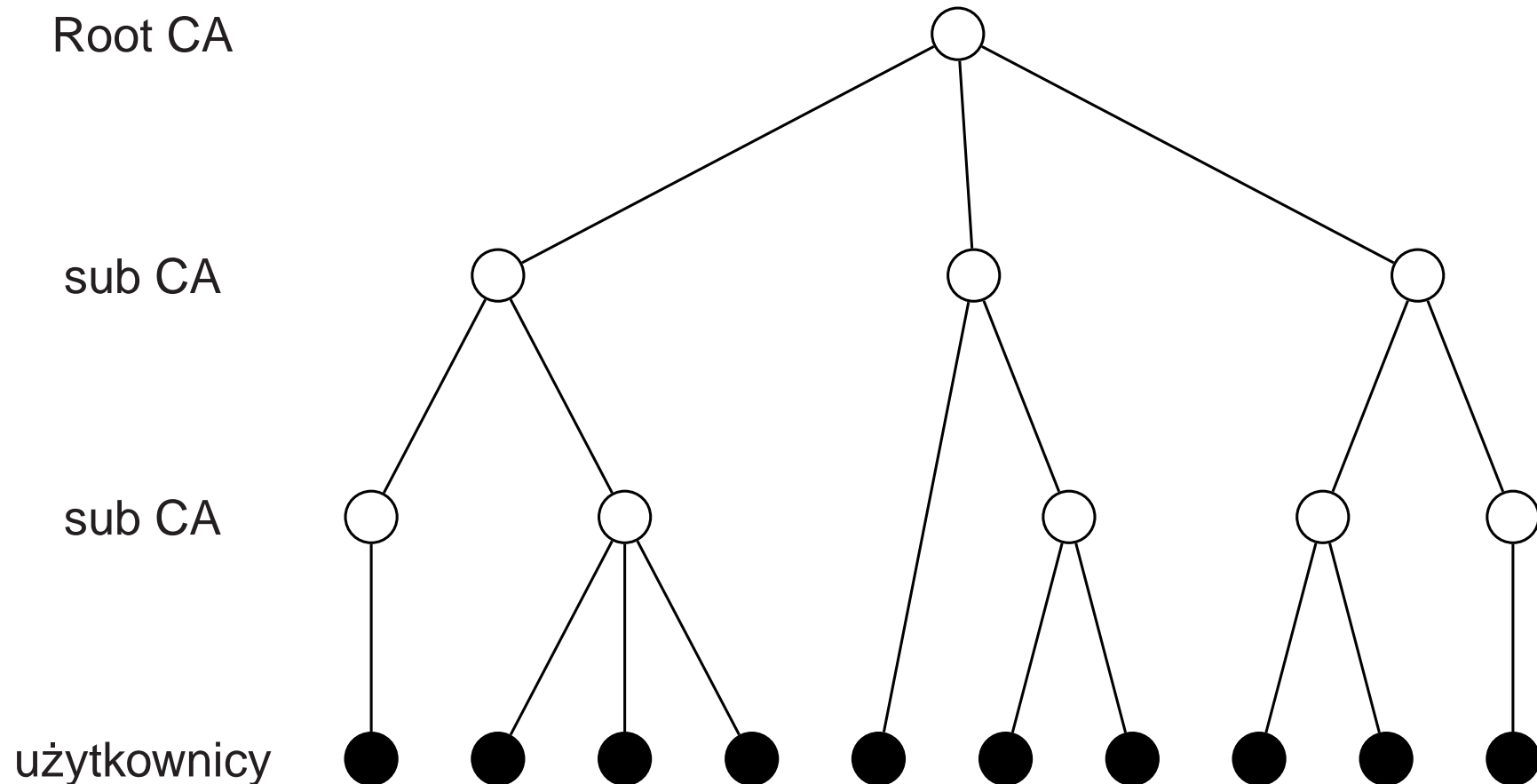
**zaufanie PKI** wywodzi się z powyższego:

*Podmiot ufa CA, gdy podmiot zakłada, że CA utworzy i będzie utrzymywać dokładne i poprawne powiązanie atrybutów z kluczem publicznym*

**zaufanie do klucza publicznego** jeśli jesteś przekonany, że klucz publiczny i prywatny należą do tego samego podmiotu

Założenia, oczekiwania, ludzkie zachowania...

# Ścisła hierarchia Centr Certyfikujących



Root CA jest *kotwicą zaufania*

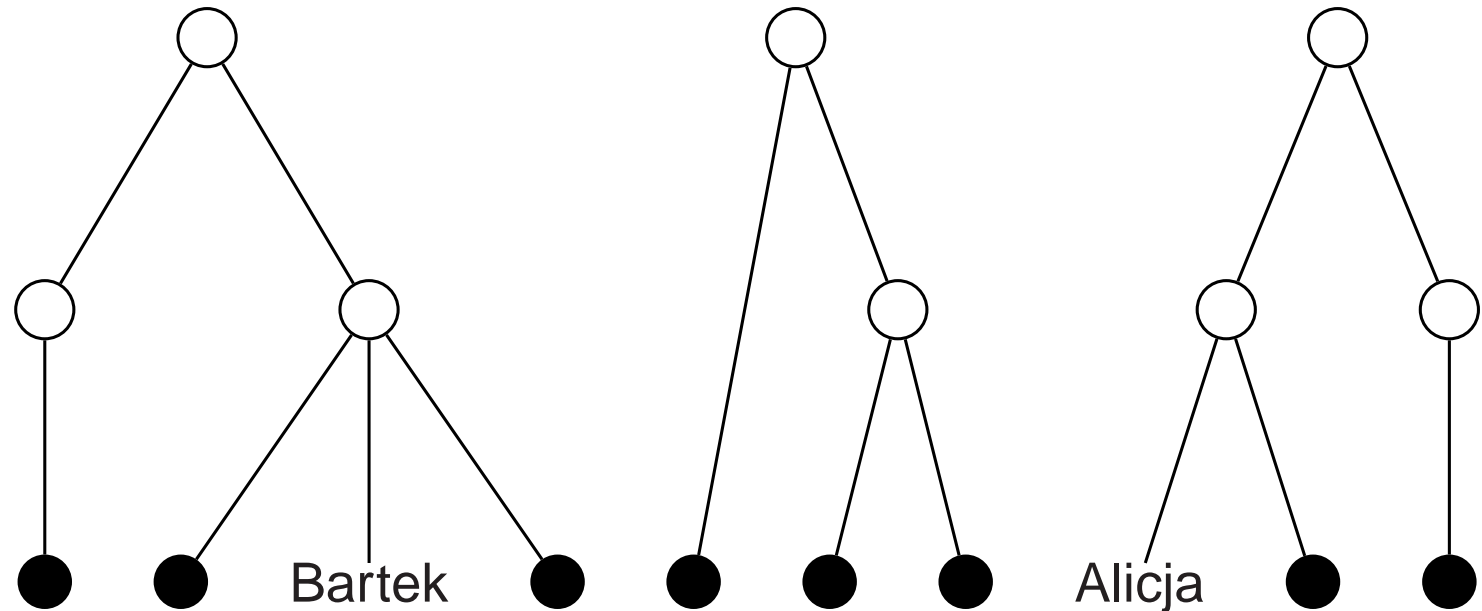
Możesz utworzyć własny, podpisany przez siebie Root CA i się pobawić!

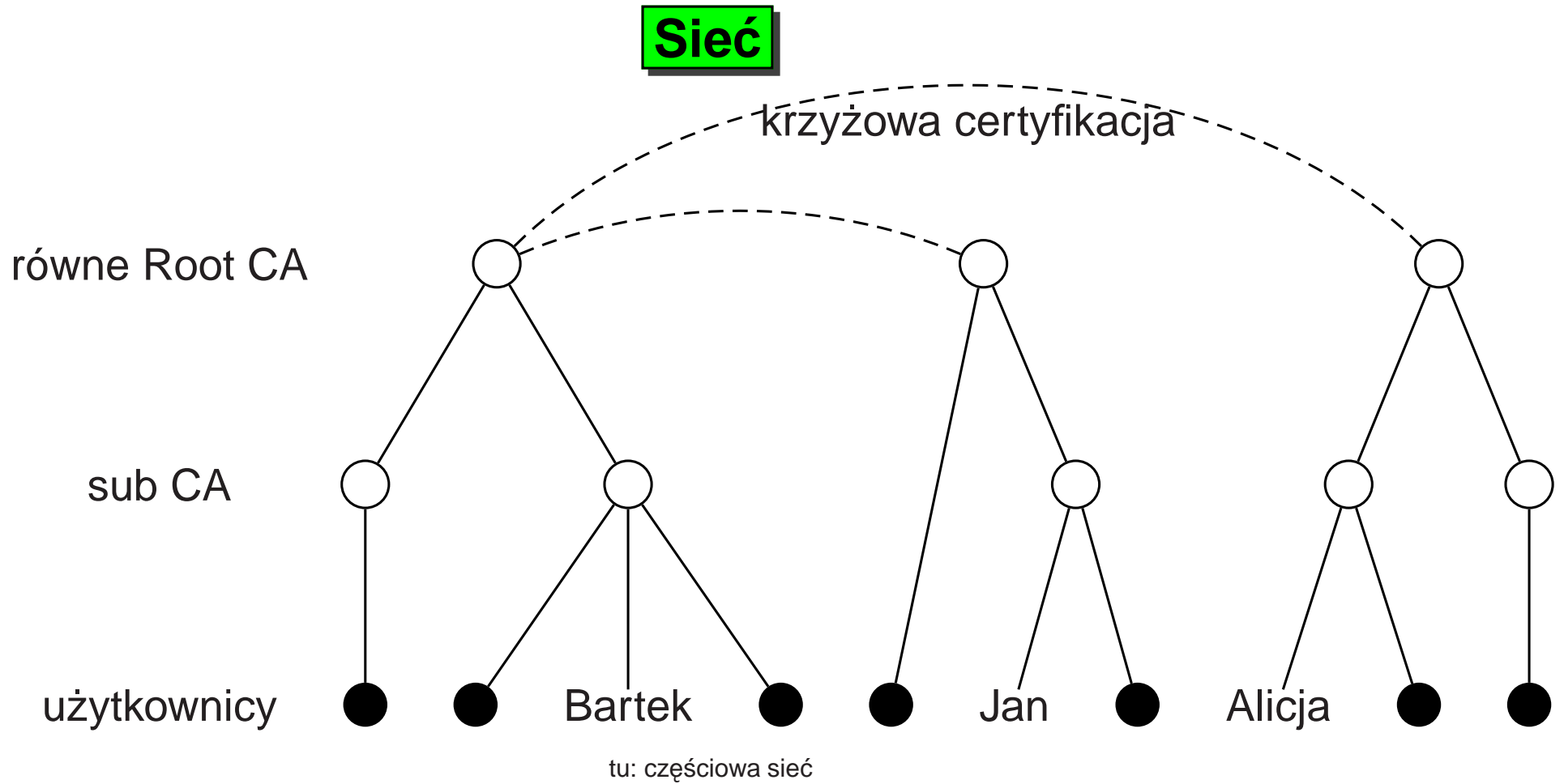
# Architektura z Rozdzielonym Zaufaniem

równe Root CA

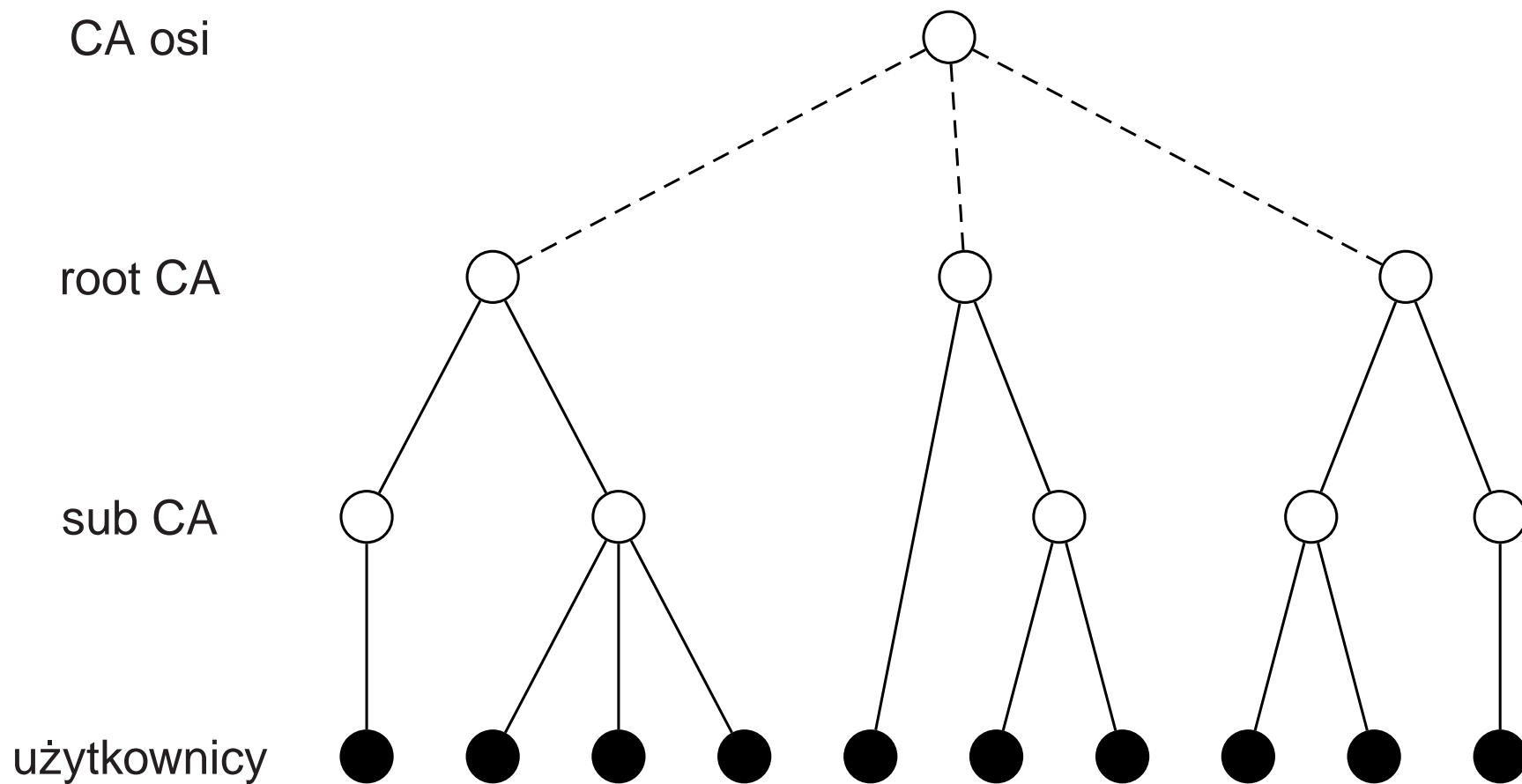
sub CA

użytkownicy

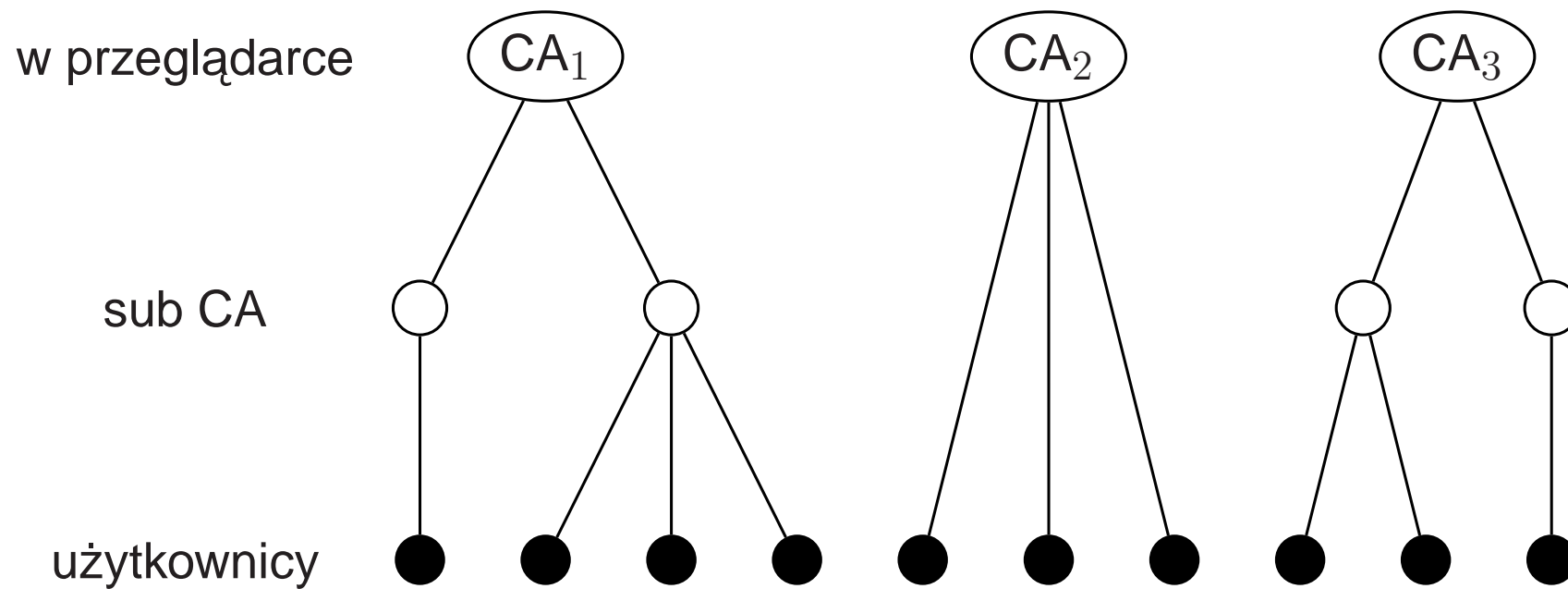




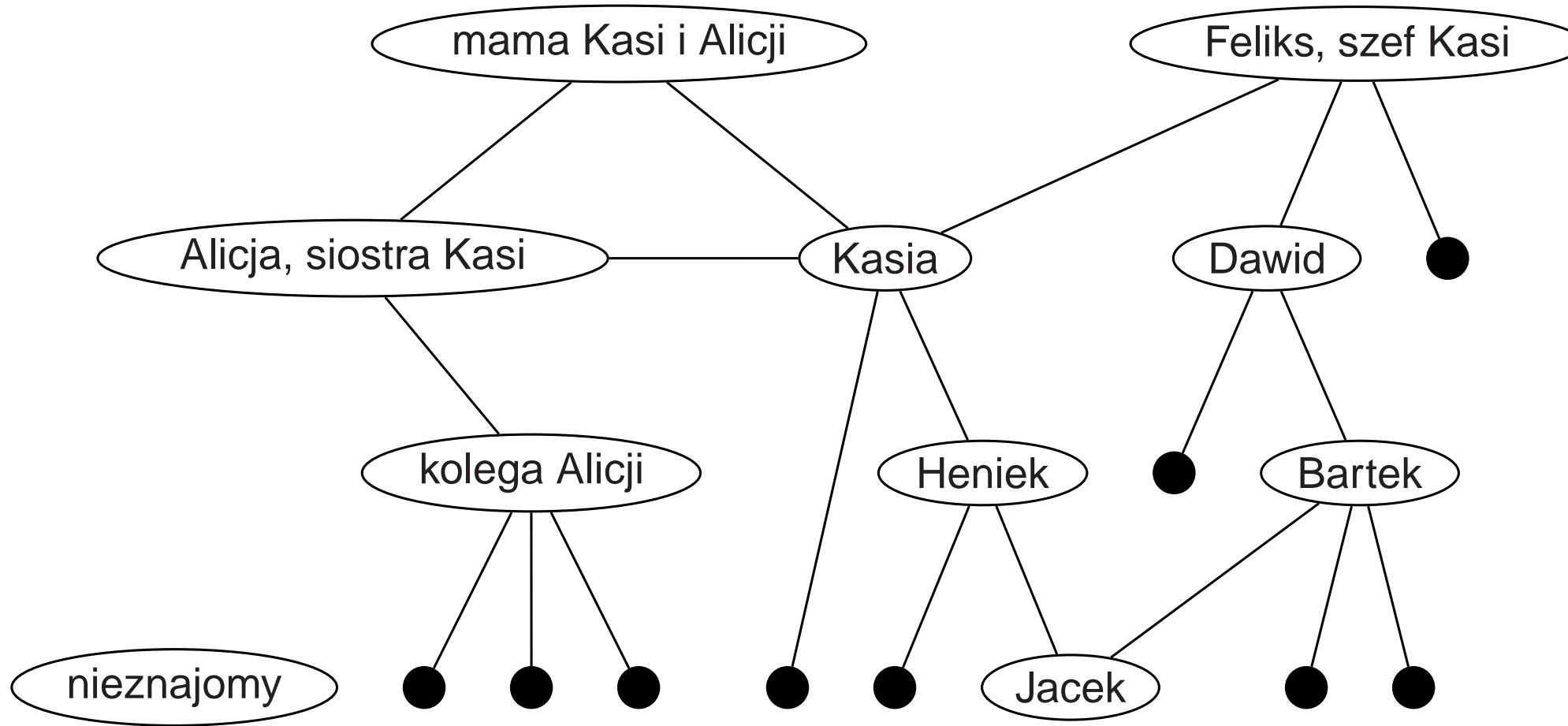
# Oś i szprychy (Hub and Spoke)



# Model WWW



# Zaufanie użytkownika (user-centric)



model PGP

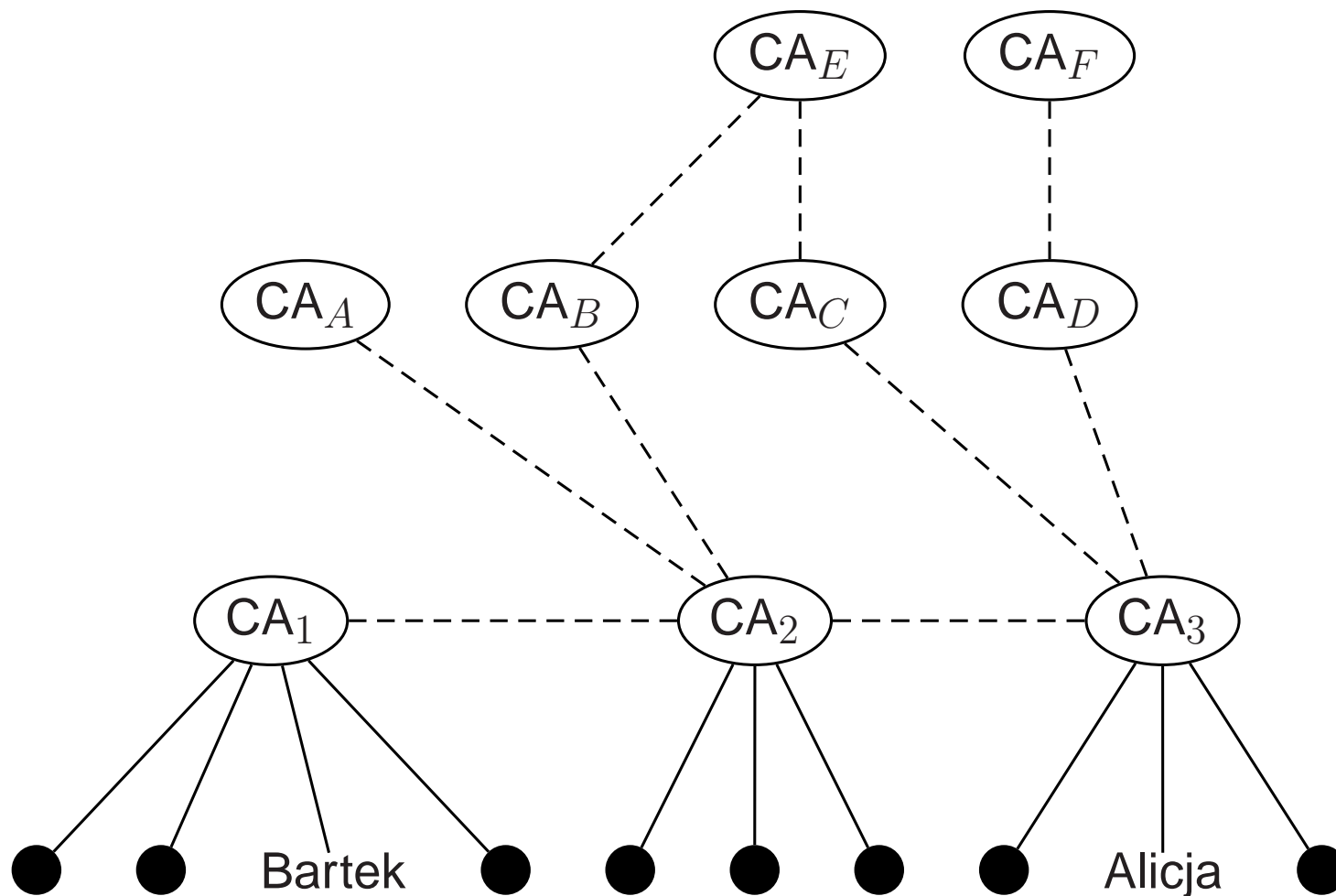
„Sześć uścisków dłoni”

```
% pgp +verbose=0 -kvv chopin@sgh
Type Bits/KeyID      Date           User ID
pub   1024/D818A489  1995/10/31    Piotr Kucharski <chopin@sgh>
sig       D818A489                Piotr Kucharski <chopin@sgh>
sig       61094A7D               Szymon Sokol <szymon@uci.agh>
sig       56A1520D               Tomasz R. Surmacz <ts@wroc.apk>
% pgp +verbose=0 -kvv ts@wroc
Type Bits/KeyID      Date           User ID
pub   1024/56A1520D  1995/04/23    Tomasz R. Surmacz <ts@wroc.apk>
sig       D818A489                Piotr Kucharski <chopin@sgh>
sig       C38B2AAD                Jan Rychter <jwr@itc.pw>
sig       61094A7D               Szymon Sokol <szymon@uci.agh>
sig       56A1520D               Tomasz R. Surmacz <ts@wroc.apk>
% pgp +verbose=0 -kvv szymon@uci
Type Bits/KeyID      Date           User ID
pub   512/61094A7D  1993/08/26    Szymon Sokol <szymon@uci.agh>
sig       8DF27F3D                <Wojtek.Sylwestrzak@icm>
sig       DA9F6825                Zbigniew Zych <zych@onet>
sig       56A1520D               Tomasz R. Surmacz <ts@wroc.apk>
```

MIT PGP Public Key Server <http://pgp.mit.edu/>

```
% pgp +verbose=0 -kvc chopin@sgh
Type Bits/KeyID      Date           User ID
pub   1024/D818A489  1995/10/31    Piotr Kucharski <chopin@sgh>
Key fingerprint =
      CE B8 CA 5D AC 3B 97 85   94 D5 8A 25 F3 4E 55 C6
```

# Tworzenie i weryfikacja ścieżki certyfikatu



## Problemy z tradycyjną PKI: technologiczne

<http://www.anu.edu.au/people/Roger.Clarke/II/PKIMisFit.html>

**problemy z podpisem elektronicznym** trzecia strona musi sprawdzić, czy klucz prywatny jest we właściwych rękach, trzecia strona musi być zaufana, klucz prywatny musi mieć zapewnione wysokie bezpieczeństwo (by nikt nie uzyskał do niego dostępu), klucz publiczny musi być odpowiedni, trzeba mieć wsparcie infrastruktury, informacje o wykradzionych kluczach i odwołaniach certyfikatów musi być szybko przekazana wszystkim (ale bez nadużywania sieci), podpis elektroniczny opiera się na funkcjach skrótu, które z natury są kolizyjne

**klucze publiczne** mogą być podrabiane (te wysyłane z wiadomością, zapisane na ftp/www lub w repozytoriach)... z tego powodu potrzebujemy certyfikatów

## Problemy z tradycyjną PKI: technologiczne

**proces uwierzytelniania przy wydawaniu certyfikatu** (powiązanie klucza publicznego z jakimś podmiotem) musi być bezpieczny na wszystkich etapach (przyjście do RA, okazanie dokumentu ze zdjęciem, podanie klucza publicznego, dowód na posiadanie klucza prywatnego (podpisywanie na oczach RA), *pokazanie dowodów na bezpieczeństwo klucza prywatnego*, podanie punktu/osoby kontaktowej, wybieranie punktu otrzymania certyfikatu

**CA** musi być zaufane i nie ma łatwego sposobu na odkrycie, czy CA nie zostało skompromitowane i czy certyfikat został wydany poprawnie

## Problemy z tradycyjną PKI: technologiczne

**certyfiakat X.509v3** jedna para kluczy na osobę, „wyróżniona nazwa” jednoznaczna w całej przeogromnej przestrzeni nazw, nie pozwala na pseudonimy, mały wpływ na sposób generowania pary kluczy, a w wielu przypadkach taka generacja jest poza kontrolą danej osoby (co może oznaczać, że klucz prywatny nie jest prywatny na starcie), mały wybór nośników przechowujących klucz prywatny (token, dyskietka), certyfikaty należą do wydawcy (użytkownik tylko nabywa praw do używania); problemy z odwoływaniem

## Problemy z tradycyjną PKI: model zaufania

**hierarchiczny model zaufania** zaufanie do CA nie jest automatyczne, każde CA musi być podpisane przez jakieś wyższego rzędu, póki nie osiągnie się mitycznego CA, do którego wszyscy mają bezwarunkowe zaufanie

**dyktatura** wszystkie strony muszą odkryć swoją tożsamość, nawet jeśli potrzebują ujawnić tylko niektóre swoje atrybuty (wiek, kwalifikacje)

**jednoznaczna wyróżniona nazwa** nie może być dwóch „Janów Kowalskich”, muszą być jakoś rozróżnieni; to także zabrania posiadania więcej niż jednego certyfikatu na osobę

## Problemy z tradycyjną PKI: (nie)bezpieczeństwo klucza prywatnego

**serwery korporacyjne** mają włamania, klucze prywatne użytkowników mogą zostać skradzione

**przechowywanie** użytkownicy zwykle zapisują swoje klucze prywatne na swoich stacjach roboczych, które są bezpośrednio podłączone do Internetu! to daje wiele możliwości odkrycia, skopiowania lub użycia kluczy prywatnych przez złośliwe oprogramowanie

## Problemy z tradycyjną PKI: ograniczone ubezpieczenie

**ubezpieczenia** praktycznie brak; CA starają się minimalizować odpowiedzialność finansową w „Komunikatach Polityki Certyfikacyjnej” oraz, co gorsza, wydają certyfikaty za różną cenę z różnym stopniem dokładności sprawdzania tożsamości przy wydawaniu certyfikatu

**BRAK ubezpieczenia** (i pewności) odnośnie:

- czy klucz prywatny był (w trakcie wydawania certyfikatu) dostępny tylko „właścicielowi”
- czy klucz prywatny jest *teraz* dostępny tylko „właścicielowi”
- czy klucza prywatnego użył „właściciel”
- czy klucza prywatnego użył „właściciel” bez przymusu

## Problemy z tradycyjną PKI: prywatność

**identyfikacja** rejestracja wymaga narzucania się z żądaniem dokumentów  
lub biometryk

**rejestry** certyfikatów są pełne danych osobowych

**listy CRL** powinny być sprawdzane przy każdej transakcji – stwarza to  
zagrożenie śledzenia e-aktywności danej osoby i daje władzom możliwość  
wymazania tożsamości danej osoby

**anonimowość** zwiększone użycie (nazwanych) certyfikatów może prowadzić  
do załamania się tradycji anonimowości i pseudonimowości

## Problemy z tradycyjną PKI: alternatywy

Modele różne od (naiwnego) wojskowego konceptu absolutnego zaufania:

**sieć zaufania PGP** nie ma CA, każdy może wydać certyfikat, odporność na awarie przez mnogość (ścieżek do) certyfikatów, zaufanie jest względne nie absolutne – jak w prawdziwym świecie

**SPKI/SDSI** porzucona globalna przestrzeń nazw, odrzucone założenie, że certyfikat pewnie jest związany z jakimś podmiotem, niech klienci decydują, atrybuty są związane z kluczami publicznymi, nie z tożsamościami z prawdziwego świata; wspiera pseudonimy

**inne** Stefan Brand's Alternative Certificates, Reputation and Brand, Trust-Management systems