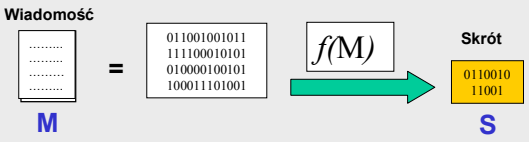
Ochrona integralności i autentyczności danych

Podpisy cyfrowe

Funkcje skrótu

Sformułowanie zagadnienia



= taki zbiór danych, że:

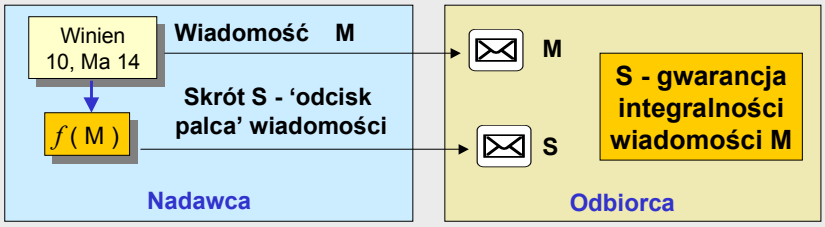
- Mając dane S nie sposób odtworzyć M — Jednokierunkowość
- $\forall_{M \neq Y} : f(M) \neq f(Y)$ — Unikatowość
- Dla dowolnego M, skrót S_M jest podobnie nieuporządkowany jak skrót losowego ciągu U - S_U

Maksymalizacja entropii skrótu

Funkcje skrótu

- Stosowane algorytmy wyznaczania skrótu ...
  - MD5 - 128 bitów (16 bajtów)
  - SHA-1 - 160 bitów (20 bajtów)

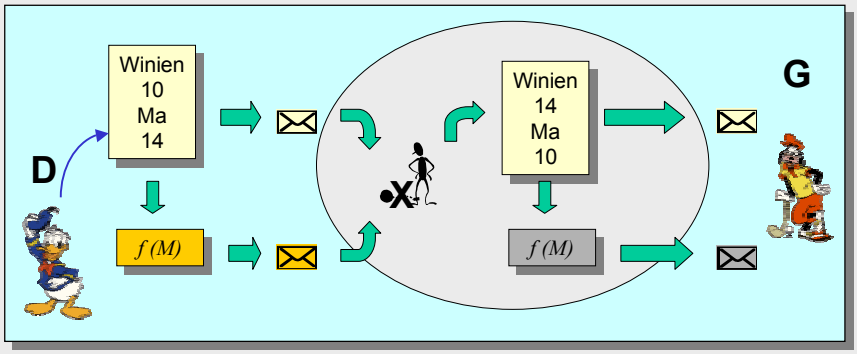
...są złożone i nie ma dowodu, że spełniają postawione założenia.



Zastosowanie funkcji skrótu pozwala na ochronę integralności wiadomości

Ochrona autentyczności wiadomości

Zastosowanie funkcji skrótu nie pozwala na ochronę autentyczności wiadomości

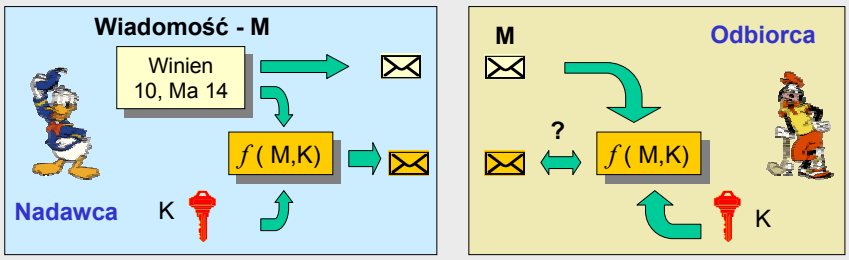


Każdy może zmienić wiadomość i wyznaczyć dla niej skrót

MAC (message authentication code)

Istota metody

- Nadawca: Wyznaczenie skrótu na podstawie wiadomości powiększonej o tajną informację, identyfikującą autora (klucz)
- Odbiorca: Sprawdzenie skrótu na podstawie wiadomości i współdzielonego klucza autora



MAC zapewnienia autentyczność wiadomości

Odbiorca musi posiadać ten sam klucz w celu uwiarygodnienia treści (klucz symetryczny)

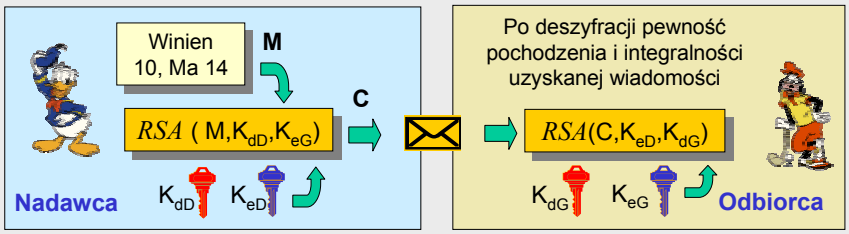
Podpisy cyfrowe

MAC - pozwala sporządzić podpis cyfrowy wiadomości, ale ma wady:

- Konieczność posiadania tego samego klucza przez obydwu uczestników korespondencji
- Możliwość wyparcia się autorstwa tekstu - tekst może być podpisany przez dowolną osobę posiadającą klucz

RSA - ?

Pożądana metoda wykorzystująca ideę kluczy publicznych



Podpisy cyfrowe

Podstawową wadą metody RSA jako metody zapewniającej ochronę autentyczności i integralności wiadomości jest duża złożoność obliczeniowa algorytmu

↓ Zmniejszenie czasochłonności operacji

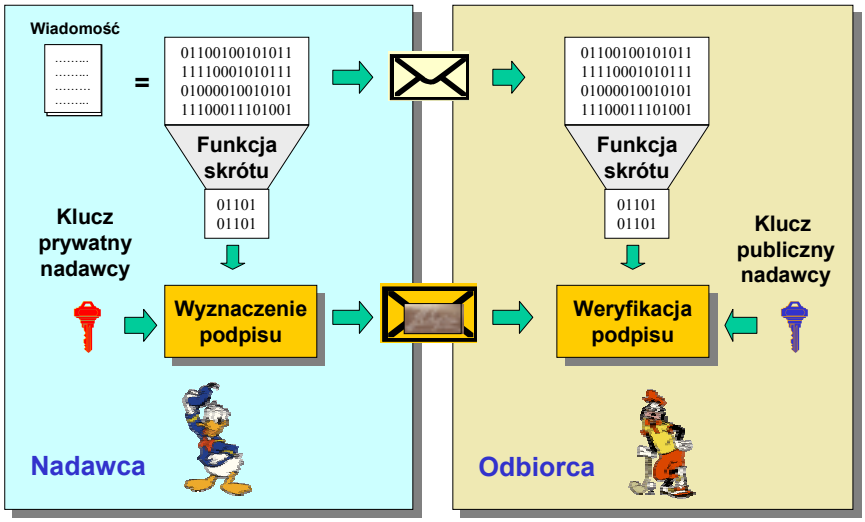
Zamiast przetwarzania całej wiadomości przy użyciu RSA wyznaczyć i podpisać wyłącznie skrót wiadomości

↓

Efekt to przetwarzanie niewielkiego bloku danych o stałym rozmiarze (16 lub 20 bajtów)

Realizacja idei - algorytm DSA (digital signature algorithm)

Algorytm DSA



Algorytm DSA

Podstawowe charakterystyki

- Długość kluczy - jak w RSA
- Struktura klucza - składniki p,q,g (wspólne dla kluczy tajnego i jawnego) oraz x (tajny) i y (jawny)
- Stosowana funkcja skrótu - SHA-1
- Wielkość podpisu cyfrowego - 20 bajtów

Funkcja

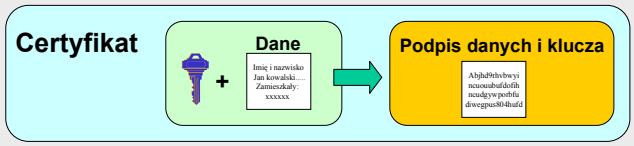
- Zapewnia autentyczność wiadomości - może ją sporządzić tylko posiadacz klucza tajnego
- Zapewnia integralność podpisywanej wiadomości - podpis zgadza się tylko dla wiadomości w oryginalnej postaci

Rozpowszechnianie kluczy - certyfikaty

Powszechnie przyjętym sposobem dystrybucji kluczy publicznych stały się certyfikaty

Składniki certyfikatu

- Klucz publiczny
- Informacje o posiadaczu klucza (imię, nazwa instytucji itp.)
- Podpis (podpisy) cyfrowe całej zawartości

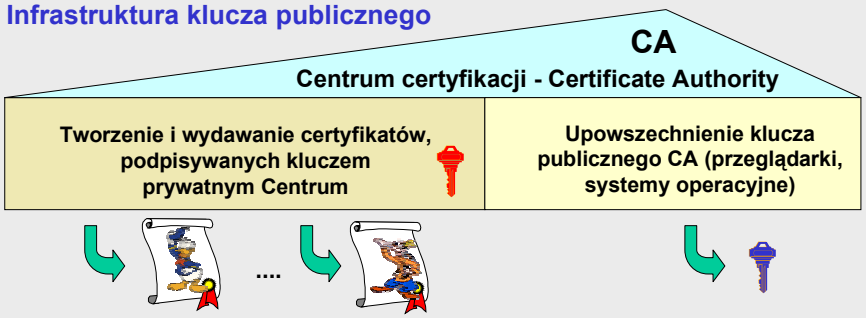


Dystrybucja certyfikatów

Metody dystrybucji certyfikatów

- Bezpośrednia (osobista - e-mail, dyskietka z plikiem)
- Serwery certyfikatów (bazy danych kluczy)
- Infrastruktura klucza publicznego (PKI)
  - bazy danych kluczy + struktura i procedury

Infrastruktura klucza publicznego



Korzystanie z certyfikatów

- Aplikacje i certyfikaty

Aplikacje wykorzystujące kryptografię klucza publicznego korzystają z certyfikatów jako standardu upowszechniania kluczy

Zaufanie do oprogramowania = zaufanie do klucza

Konieczne upewnienie się czy klucz nie został spreparowany

Konieczne sprawdzenie, czy certyfikat nie został cofnięty

Certyfikaty

Sprawdzanie certyfikatów

Dla celów sprawdzania autentyczności certyfikatu, w certyfikacie znajduje się ‘odcisk palca’ (skrót) certyfikatu, który można porównać z opublikowanym (na przykład wydrukowanym w prasie) oryginałem

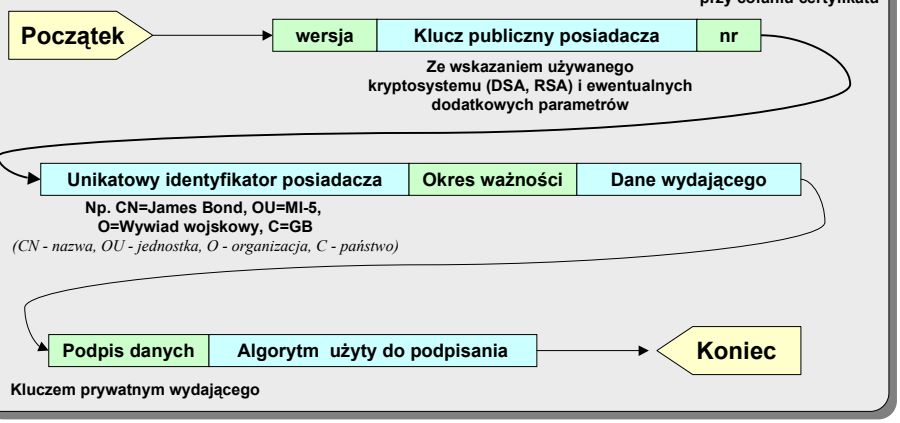
Łańcuchy certyfikatów

Jeżeli certyfikat nie jest wydawany bezpośrednio przez zaufane i znane CA, ale przez pośrednika, pośrednik może poświadczyć własną wiarygodność dołączając podpis swojego klucza publicznego, wykonany kluczem podstawowego CA. Dla większej liczby szczebli pośrednich, procedura poświadczania jest zagnieżdżana tworząc łańcuch certyfikatów

Stosowane standardy certyfikatów

X.509 Przeglądarki

Struktura certyfikatu X.509



Certyfikaty X.509

Postać wydruku certyfikatu

-----BEGIN CERTIFICATE-----
CERTIFICATE DATA
CERTIFICATE FINGERPRINT
-----END CERTIFICATE-----

Owner: CN=ll, OU=ll, O=ll, L=ll, S=ll, C=ll
Issuer: CN=ll, OU=ll, O=ll, L=ll, S=ll, C=ll
Serial Number: 59092b34
Valid from: Thu Sep 25 18:01:13 PDT 1997
until: Wed Dec 24 17:01:13 PST 1997
Certificate Fingerprints:
MD5: 11:81:AD:92:C8:E5:0E:A2:01:2E:D4:7A:D7:5F:07:6F
SHA1: 20:B6:17:FA:EF:E5:55:8A:D0:71:1F:E8:D6:9D:C0:37:13:0E:5E:FE

Wersje certyfikatów X.509

- v.1 - standard - 1988
- v.2 - dodane elementy o spornej przydatności - nie rozpowszechniona
- v.3 - 1996 - tzw. rozszerzenia np. przeznaczenie ('key usage')
  (n.p. tylko do podpisu, a nie do wykorzystania do nawiązania połączenia SSL)
  dodatkowo, poświadczane nowe elementy identyfikujące (e-mail...cokolwiek)

Certyfikaty

Ubieganie się o certyfikat

- Generacja własnej pary kluczy
- Utworzenie certyfikatu uwierzytelnionego przez własny klucz (self-signed certificate),
- Wysłanie do CA CSR (certificate signing request) zawierającego utworzony certyfikat
- Odpowiedź CA - zwrócony certyfikat z początkowym podpisem zastąpionym podpisem CA

Bezpieczne korzystanie z komputerów

Uwierzytelnianie użytkownika i sesje w sieci lokalnej

Hasła i logowanie

- **Ataki słownikowe**
  - Próbowane słowa ze słownika (kilkadziesiąt - kilkaset tysięcy, z uwzględnieniem przypadków, rodzajów - rząd miliona)
  - Używane małe i duże litery
  - Dodawane na początku / końcu cyfry i znaki interpunkcji

  Złożoność do trylionów kombinacji (10^{12}) nie stanowi przeszkody przy współczesnym poziomie mocy obliczeniowych

- **Budowanie dobrych haseł**

  Oczywiście, pamiętam hasło logowania na swoje konto - to imię mojego psa ...

  - mój pies wabi się 7&uo(#DF-<Qa
  - zmieniam mu imię co trzy tygodnie

Hasła

- **Budowanie dobrych haseł**
  - >6,8 znaków, wstawianie znaków innych niż litery

    50 przycisków klawiatury, SHIFT - 2*50 = 100 symboli ⇒ Jeżeli hasło przypadkowe $100^6 - 100^8$

    Atak metodą prób i błędów - 10^{16} kombinacji dla hasła 8-znakowego

    Uwzględnienie ALT i CTRL - daje ok. 200 różnych symboli (ASCII)
  - Skróty (lub elementy) fraz — Pmps (Polak mądry po szkodzie)
  - Mieszanie dwóch (lub więcej) wyrazów

    kloss bond ⇒ kbloosnsd
- **Przechowywanie haseł w systemie**
  - Kradzież pliku z hasłami

    Hasła przechowywane w formie zaszyfrowanej lub jako skróty - kradzież nie daje bezpośrednio możliwości logowania (ale dla skrótu, hasło można odgadnąć w drodze ataku słownikowego)

Bezpieczeństwo systemów informatycznych

Hasła

- **Inne problemy związane z hasłami**
  - Ataki ‘psychologiczne’

    Ktoś podaje się za technika i mówi że wystąpił problem w sieci, prosząc jednocześnie o podanie hasła

    ⇓ wykorzystanie naturalnej skłonności niesienia pomocy

    Działa zaskakująco skutecznie - w 2000 w USA zanotowano ok. 60 tysięcy takich przypadków
  - Problemy z zapamiętywaniem haseł

    Redakcja NYT donosi o średnio 1000 przypadkach zapominania haseł na tydzień
- **Ochrona przed atakami słownikowymi**

  Zawieszanie możliwości logowania na określony czas po kilku nieudanych próbach (timeout-y)

Krzysztof Ślot © 2002

Zdalne logowanie

- Problem zdalnego logowania - hasło może być przechwycone

  Powszechnie stosowane aplikacje przesyłające hasła tekstem jawnym:

  FTP, telnet, POP3, SMTP, HTTP... ☹

- Wysyłanie hasła w formie zaszyfrowanej lub w postaci skrótu

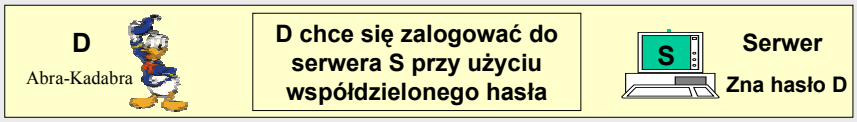
  W niczym nie rozwiąże problemu, bo można przechwycić i wykorzystać bez odgadywania hasła ☹

- Wysyłanie hasła w formie zaszyfrowanej (lub skrótu hasła) powiększonego o losowo zmienianą informację

  Technika challenge-response- CR (hasło i odzew) ☺

Logowanie do serwera

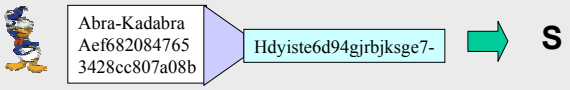
Technika CR



1. S generuje liczbę losową i wysyła ją do D

2. D wyznacza skrót hasła, powiększonego o liczbę losową i odsyła do S

3. S wyznacza skrót znanego hasła i wygenerowanej liczby losowej i porównuje go z otrzymaną wiadomością

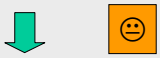
Logowanie do serwera

- Implementacja idei CR - protokół CHAP

Zastosowanie - autoryzacja użytkownika w PPP

Nie jest odporne na atak słownikowy

Atakujący zna skrót i zna przesłaną liczbę losową - może więc samemu dla różnych haseł próbować wygenerować skrót (algorytm jest znany) taki sam, jak przesłany serwerowi



Uwierzytelnienie nie jest bezpieczne gdy hasło jest słabe

Logowanie do serwera

- Modyfikacja uodparniająca na atak słownikowy

S i D mają współdzielony klucz szyfrowania

1. S szyfruje kluczem liczbę losową i wysyła do D
2. D rozszyfrowuje kryptogram i odtwarza liczbę losową
3. D zmienia w umówiony sposób liczbę losową (dodaje 1 itp.) i odsyła do S
4. S sprawdza kryptogram

Atakujący nie ma dostępu do liczby losowej - nie może przeprowadzić ataku słownikowego

Jeżeli do liczby losowej dodana zostanie dodatkowa informacja, np. ID użytkownika, wtedy metoda pozwala na uwierzytelnienie obydwu stron

Inne metody uwierzytelniania

- Uniemożliwienie ataku słownikowego - mocne hasło

  Żetony (bilety) i kalkulatory haseł

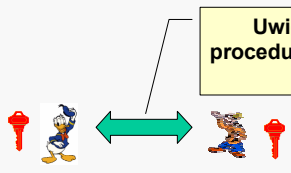
  Stosowanie metody CR, tyle że samo hasło jest długie, przydzielone danemu użytkownikowi, pamiętane w formie pliku, autonomicznego ‘kalkulatora’, karty elektronicznej

- Biometria
  - Odcisk palca
  - Obraz tęczówki
  - Głos
  - Kształt dłoni
  - Obraz siatkówki
  - Podpis, dynamika pisania

Sesje w sieci lokalnej

- Zadanie - zapewnić bezpieczne logowanie i bezpieczny przebieg sesji między komputerami pracującymi w sieci lokalnej

1. Wykorzystanie współdzielonego klucza tajnego



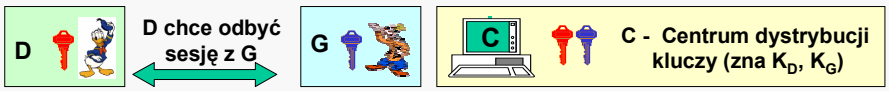
Tajny klucz jest inny dla każdej pary uczestników komunikacji

Wada #1 metody - problemy z gospodarką kluczami

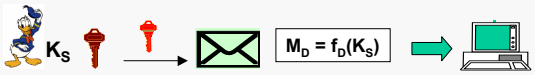
Wada #2 metody - problemy z generacją dobrego klucza

Sesje w sieci lokalnej

2 Ustanowienie centrum dystrybucji kluczy sesyjnych



➡ D generuje losowy klucz sesji K_S, szyfruje go kluczem K_D i wysyła do C



➡ C deszyfruje klucz sesyjny, szyfruje go kluczem K_G i wysyła do G

➡ G rozkodowuje klucz sesyjny 

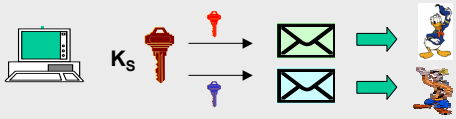
Wada - D, G mogą nie potrafić wygenerować silnego klucza, więc...

Sesje w sieci lokalnej

3 Zmiana roli centrum dystrybucji kluczy sesyjnych - centrum generuje klucze sesji

➡ D wysyła prośbę o klucz sesji do C

➡ C generuje losowy klucz sesji K_S, szyfruje go kluczem K_D i wysyła do D oraz szyfruje go kluczem K_G i wysyła do G



➡ D i G deszyfrują klucz sesyjny