Bezpieczeństwo systemów komputerowych

Zastosowania kryptografii

Marcin Peczarski

Instytut Informatyki Uniwersytetu Warszawskiego

14 listopada 2011

Jednokierunkowa funkcja skrótu

- Argumentem jednokierunkowej funkcji skrótu H jest wiadomość M o dowolnej długości.
- Wartość funkcji ma ustaloną długość.
- Mając daną wiadomość M, łatwo jest obliczyć jej skrót $h = H(M)$.
- Mając dane h, trudno jest znaleźć taką wiadomość M, że $h = H(M)$.
- Mając daną wiadomość M, trudno jest znaleźć inną wiadomość M', taką że $H(M) = H(M')$.
- Trudno jest znaleźć dwie losowe wiadomości M i M', takie że $H(M) = H(M')$.

Atak urodzinowy

- Korzysta z paradoksu dnia urodzin.
- Dla skrótu m-bitowego wystarczy wylosować liczbę wiadomości rzędu $2^{m/2}$, aby z dużym prawdopodobieństwem znaleźć takie dwie, że $H(M) = H(M')$.
- Brutalny atak polega na znalezieniu dla danego h wiadomości M, takiej że $h = H(M)$.
- Jeżeli m-bitowy skrót (na danym etapie rozwoju technologii) nie poddaje się brutalnemu atakowi, to aby obronić się przed atakiem urodzinowym, należy zastosować skrót o podwojonej długości $2m$.

Atak urodzinowy, cd.

- Można wykorzystać do wygenerowania dwóch różnych dokumentów o tym samym skrócie m-bitowym.
- W każdym dokumencie znajdujemy $m/2$ miejsc, w których można dokonać subtelnych zmian:
  - dodanie spacji na końcu wiersza,
  - wstawienie sekwencji spacja-cofnij-spacja,
  - itp.
- To umożliwia wygenerowanie $2^{m/2}$ wersji każdego dokumentu.
- Jest spore prawdopodobieństwo, że istnieje para posiadająca ten sam skrót.

Ciąg uwierzytelniania wiadomości

- MAC – Message Authentication Code
- Funkcja skrótu zawierająca klucz
- Zapewnia autentyczność i integralność bez wprowadzania tajności.
- Zaszyfrowanie skrótu algorytmem symetrycznym

$$E_K(H(M)).$$

- Zaszyfrowanie skrótu kluczem prywatnym autora wiadomości

$$E_{K_s}(H(M)).$$

- Szyfrowanie wiadomości algorytmem blokowym w trybie CBC lub CFB – skrótem jest ostatni blok szyfrogramu.
- Użycie jednokierunkowej funkcji skrótu:

$$H(K \parallel |M| \parallel M),$$

$$H(K_1 \parallel M \parallel K_2),$$

$$H(K_1 \parallel H(K_2 \parallel M)).$$

HMAC

- Hash Message Authentication Code
- Keyed-Hashing for Message Authentication, RFC 2104
- Dla danej funkcji skrótu H definiujemy

$$HMAC(K, M) = H((K \oplus 0\text{x}5\text{c}\ldots5\text{c}) \parallel H((K \oplus 0\text{x}36\ldots36) \parallel M)).$$

Funkcje skrótu (1)

- MD (Message Digest) [nie opublikowana oficjalnie], MD2 [RFC 1319]:
  - powstały w latach 80 ubiegłego wieku;
  - autor Ron Rivest;
  - pierwsze powszechnie stosowane funkcje skrótu;
  - były wykorzystywane w systemach pocztowych.
- SNEFRU:
  - powstała w 1990;
  - autor Ralph Merkle (Xerox);
  - algorytm HMAC;
  - kilkakrotnie szybsza od MD2;
  - oryginalna wersja złamana w 1992;
  - zwiększenie liczby iteracji poprawia bezpieczeństwo, ale czyni wolniejszą od MD5 i SHA.
- MD4 [RFC 1320]:
  - odpowiedź Rivesta na SNERFU;
  - zbyt słaba.

Funkcje skrótu (2)

- MD5 [RFC 1321]:
  - autor Ron Rivest;
  - udoskonalona wersja MD4, ale wolniejsza od niej;
  - wykonuje 64 iteracje (4 rundy po 16 kroków);
  - generuje skrót o długości 128 bitów;
  - obecnie uważa się, że ma zbyt słabą odporność na kolizje;
  - zdecydowanie niezalecana.
- SHA (Secure Hash Algorithm):
  - opracowana przez NSA;
  - przyjęta przez NIST jako standard federalny w 1993;
  - wersja oryginalna SHA-0 jest zbliżona do MD4;
  - stosunkowo szybko wykryto słabości SHA-0 (ich natury nie opublikowano) i opracowano SHA-1 (ratyfikowany przez NIST);
  - wykonuje 80 iteracji (4 rundy po 20 kroków);
  - generuje skrót o długości 160 bitów;
  - obecnie uważa się ją za zbyt słabą;
  - niezalecana.

Funkcje skrótu (3)

- RIPEMD (RACE Integrity Primitives Evaluation Message Digest):
  - opracowana w ramach projektu finasowanego przez UE;
  - odmiana MD4 uodporniona na znane ataki;
  - generuje skrót o długości 128 bitów;
  - dwie wersje algorytmu, różniące się wartościami stałych, są realizowane równolegle;
  - ciągi wyjściowe obydwu wersji są po każdym bloku dodawane do zmiennych łańcuchowych;
  - znaleziono kolizję.
- RIPEMD-160:
  - ulepszona wersja RIPEMD.
  - autorzy: Hans Dobbertin, Antoon Bosselaers, Bart Preneel;
  - generuje skrót o długości 160 bitów;
  - są też wersje RIPEMD-128, RIPEMD-256 i RIPEMD-320.

Funkcje skrótu (4)

- HAVAL:
  - może generować skrót o długości 128, 160, 192, 224 lub 256 bitów;
  - można ją uznać za wariant MD4;
  - stosuje wyrafinowane funkcje nieliniowe siedmiu zmiennych o własności lawinowości;
  - prawdopodobnie posiada dużą odporność na kryptoanalizę.

Funkcje skrótu (5)

- ElGamal:
  - oparta na trudności rozwiązania problemu logarytmu dyskretnego;
  - p – liczba pierwsza;
  - g, x – losowe liczby, całkowite dodatnie, mniejsze niż p;
  - obliczamy $y = g^x \bmod p$;
  - (g, p, y) – klucz publiczny;
  - k – wybrana losowo liczba względnie pierwsza z $p - 1$;
  - obliczamy $a = g^k \bmod p$;
  - korzystamy z rozszerzonego algorytmu Euklidesa do obliczenia liczby b z równania $M = (xa + kb) \bmod (p - 1)$;
  - podpisem wiadomości M jest para (a, b);
  - weryfikacja podpisu następuje pomyślnie, jeśli $y^a a^b \equiv g^M \pmod p$;
  - poprawność wynika z równości $y^a a^b = g^{xa+kb}$;
  - k musi być utrzymywane w tajemnicy i nie może się powtórzyć.

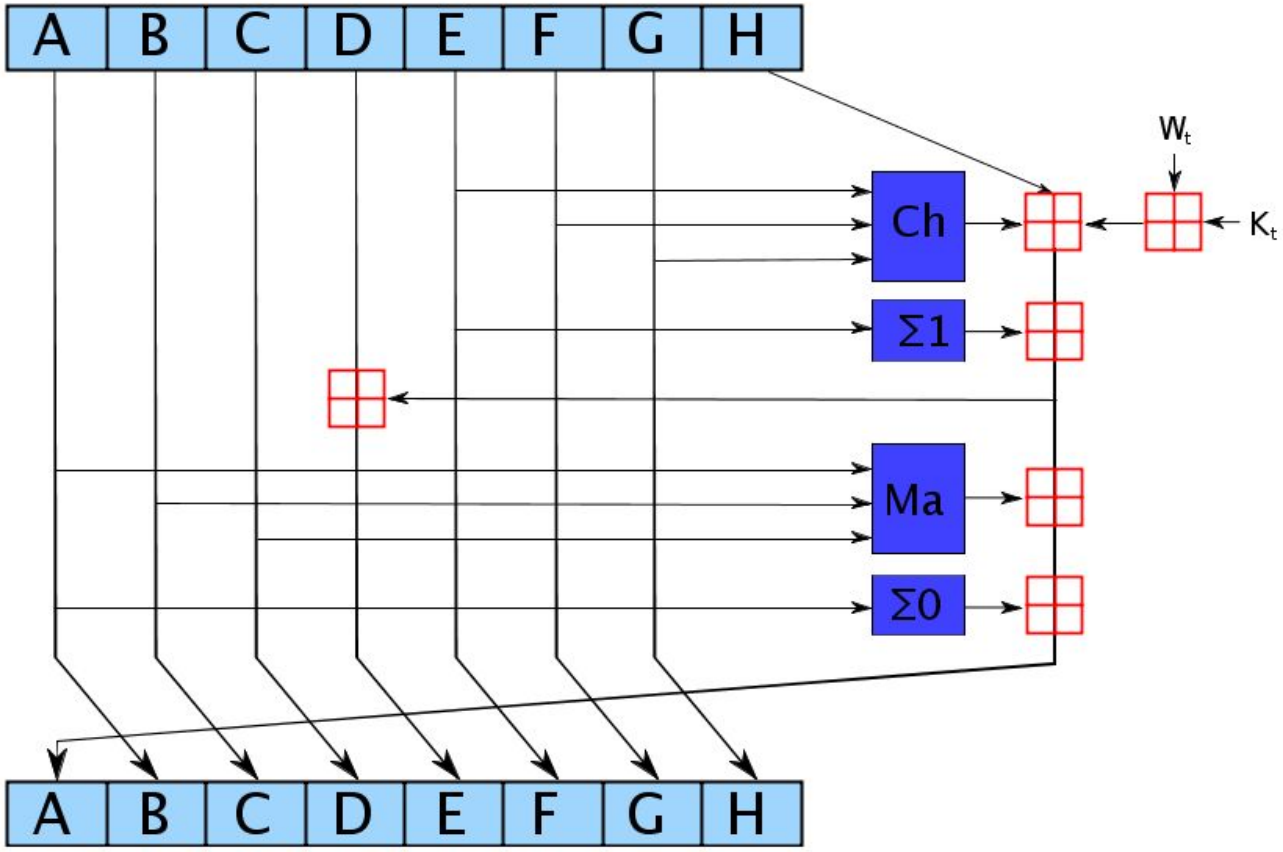
Funkcje skrótu (6)

- SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512):
  - przystosowane do współpracy z kluczami AES;
  - generują skróty o długości odpowiednio 224, 256, 384 i 512 bitów;
  - większa złożoność obliczeniowa od poprzedników;
  - algorytmy SHA-224 SHA-384 mają identyczny koszt obliczeniowy co odpowiednio SHA-256 i SHA-512;
  - powszechnie spotyka się jedynie SHA-256 i SHA-512;
  - dość powszechnie są uznawane za bezpieczne.

	l. rund	dł. słowa	dł. bloku	maks. dł. wiad.
SHA-224/256	64	32 bity	512 bity	$2^{64} - 1$
SHA-384/512	80	64 bity	1024 bity	$2^{128} - 1$

SHA-256

- Jedna runda



- Ch := (E and F) xor ((not E) and G)
- Σ1 := (E ror 6) xor (E ror 11) xor (E ror 25)
- Ma := (A and B) xor (A and C) xor (B and C)
- Σ0 := (A ror 2) xor (A ror 13) xor (A ror 22)

SHA-256, cd.

- Wszystkie operacje są wykonywane w trybie big-endian.
- Wiadomość jest rozszerzana o jedynkę, dopełnienie zawierające zera i 64-bitową oryginalną długość wiadomości, tak aby długość całości w bitach była podzielna przez 512.
- H0, ..., H7 to pierwsze 32 bity części ułamkowej pierwiastka kwadratowego z kolejnych liczb pierwszych od 2 do 19.
- A, ..., H są inicjowane wartościami H0, ..., H7.
- K_t to pierwsze 32 bity części ułamkowej pierwiastka sześciennego z kolejnych liczb pierwszych od 2 do 311.
- W_t to 64 słowa powstałe z jednego 512-bitowego bloku wiadomości z użyciem operacji rotacji, xor i dodawania modulo.
- $H0 := H0 + A \bmod 2^{32}$
- $H1 := H1 + B \bmod 2^{32}$
- ...
- $H7 := H7 + H \bmod 2^{32}$
- Skrót jest konkatenacją końcowych wartości H0, ..., H7.

Pożądane cechy podpisu

- Autentyczność, niepodrabialność, jednorazowość – podpisujący opatrzył nim dokument świadomie, weryfikujący jest przekonany, że został złożony przez podpisującego, nie można go przenieść na inny dokument.
- Dokument po podpisaniu nie może być zmieniony.
- Nie można wyprzeć się podpisu.

Podpis cyfrowy

- Zaproponowano wiele schematów podpisów cyfrowych.
- Najpowszechniej stosuje się podpis używający jednokierunkowej funkcji skrótu i kryptografii asymetrycznej:
  - nadawca oblicza jednokierunkowy skrót wiadomości;
  - nadawca wytwarza podpis przez zaszyfrowanie skrótu za pomocą swojego klucza prywatnego;
  - nadawca wysyła do odbiorcy wiadomość wraz z podpisem;
  - odbiorca oblicza jednokierunkowy skrót wiadomości;
  - odbiorca deszyfruje odebrany podpis za pomocą klucza publicznego nadawcy;
  - jeśli skróty obliczony i odszyfrowany są identyczne, to podpis jest uznawany za autentyczny.
- Często do dokumentu przed podpisaniem dodawany jest znacznik czasu.

Poufności, nienaruszalność i autentyczność

- Nadawca podpisuje wiadomość.
- Nadawca szyfruje podpisaną wiadomość.
- Nadawca wysyła zaszyfrowaną wiadomość do odbiorcy.
- Odbiorca deszyfruje odebraną wiadomość.
- Odbiorca weryfikuje autentyczność podpisu.

Standardy podpisu cyfrowego

- DSS (Digital Signature Standard):
  - NIST FIPS PUB 186 z 1994 r. z kategorii FIST (Federal Information processing STandard);
  - obejmuje użycie skrótu SHA-1 oraz algorytmu z kluczem publicznym DSA (Digital Signature Algorithm) opracowanego przez NSA;
  - jest też wersja wykorzystująca krzywe eliptyczne.
- SHS (Secure Hash Standard):
  - NIST FIBS PUB 180-3 z 2008 r.;
  - opisuje użycie SHA-1, SHA-224, SHA-256, SHA-384, SHA-512.
- ISO 9796
  - ISO/IEC 9796-2:2002 bazuje na problemie trudności faktoryzacji liczb całkowitych (RSA);
  - ISO/IEC 9796-3:2006 bazuje na problemie trudności logarytmu dyskretnego.

Zarządzanie kluczami

- Problem przekazania klucza – jak partnerowi komunikacji przekazać w sposób bezpieczny klucz niezbędny do szyfrowania i deszyfrowania?
- Problem zmiany klucza – jak regularnie zmieniać klucz?
- Problem doboru systemu szyfrowania:
  - przy zastosowaniu szyfru symetrycznego musimy generować nowy klucz dla każdej sesji komunikacji, a nawet zmieniać go w trakcie sesji;
  - szyfry asymetryczne są mniej wydajne od symetryczych i nie nadają się do szyfrowania długich wiadomości;
- Problem potwierdzania autentyczności klucza (publicznego).
- Powszechnie stosuje się:
  - kryptografię asymetryczną do potwierdzania tożsamości;
  - szyfr symetryczny do szyfrowania właściwej komunikacji;
  - metodę zaproponowaną przez Whitfielda Diffiego i Martina Hellmana do generowania jednorazowego klucza sesji.

Certyfikacja

- Certyfikację stosuje się, aby zapobiegać podstawieniu fałszywego klucza publicznego.
- Certyfikat jest poświadczeniem autentyczności, podpisanym przez godną zaufania instytucję, nazywaną urzędem poświadczającym, CA (ang. *Certification Authority*).
- Certyfikat ma formę dokumentu elektronicznego.
- Certyfikat zawiera podstawowe dane identyfikujące właściciela.
- Urząd poświadczający CA potwierdza, że informacja opisująca właściciela klucza jest prawdziwa, a klucz publiczny faktycznie do niego należy.

Certyfikacja, cd.

- Certyfikat posiada okres ważności wyznaczający czas, przez który certyfikowane dane można uważać za poprawne.
- Niezależnie od okresu ważności certyfikowane klucze mogą zostać uznane za niepoprawne, np. gdy zaistnieje podejrzenie, że ktoś nieuprawniony wszedł w posiadanie tajnego klucza prywatnego, odpowiadającego certyfikowanemu kluczowi publicznemu.
- Urząd poświadczający CA musi przechowywać listę niepoprawnych i nieaktualnych certyfikatów.
- Unieważnienie klucza jest także rodzajem certyfikatu.

Budowa certyfikatu ITU-T X.509

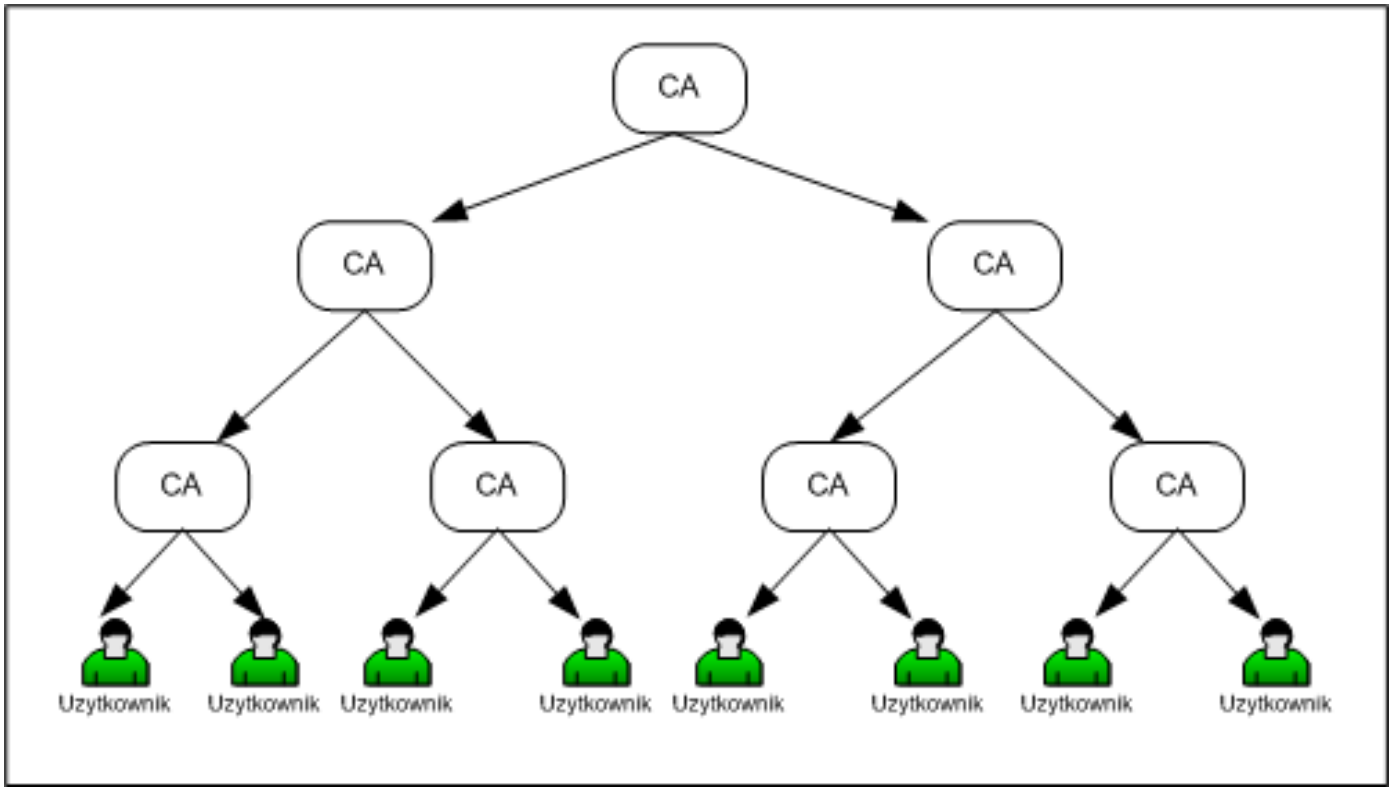
- Certyfikat
  - Wersja: 3
  - Numer seryjny: 12
  - Algorytm sygnatury certyfikatu: SHA-1 z szyfrowaniem RSA
  - Wystawca: E = info@net.icm.edu.pl, CN = DSICM Server Certificate Authority, OU = ICM, Dzial Sieciowy, O = Uniwersytet Warszawski, ST = Mazowieckie, C = PL
  - Ważność
    - Nieważny przed: 2006-10-09 14:53:45
    - Nieważny po: 2008-10-08 14:53:45
  - Podmiot: E = root@mimuw.edu.pl, CN = usosweb.mimuw.edu.pl, OU = Wydział Matematyki, Informatyki i Mechaniki, O = Uniwersytet Warszawski L = Warszawa, ST = Mazowieckie, C = PL
  - Informacje o kluczu publicznym
    - Algorytm klucza publicznego: szyfrowanie RSA
    - Klucz publiczny: 81 02 81 ...

Budowa certyfikatu ITU-T X.509, cd.

- Certyfikat, cd.
  - Rozszerzenia
    - Identyfikator klucza podmiotu certyfikatu: 23 45 67 ...
    - Identyfikator klucza ośrodka certyfikacji: 03 16 78 ...
    - Warunki użycia klucza certyfikatu: podpisywanie, niezaprzeczalność, szyfrowanie klucza
- Algorytm sygnatury certyfikatu: SHA-1 z szyfrowaniem RSA
- Wartość sygnatury certyfikatu: 4c 04 4d ...

Zaufanie do urzędów certyfikujących

- Urzędy certyfikujące tworzą wielopoziomową hierarchię.
- Urzędy danego poziomu wystawiają certyfikaty urzędom lub użytkownikom znajdującym się na niższym poziomie.
- Na szczycie znajduje się organizacja o powszechnie uznanym autorytecie, ostatecznie poświadczająca poprawność całej procedury.



Infrastruktura kluczy publicznych

- PKI (Public Key Infrastructure)
- Obejmuje sprzęt, oprogramowanie, ludzi, polityki bezpieczeństwa i procedury konieczne do utworzenia, zarządzania, przechowywania, dystrybucji i unieważniania certyfikatów kluczy publicznych w skali najczęściej ogólnonarodowej lub światowej.
- Oferuje często dodatkowe certyfikaty, np. Certyfikaty Znacznika Czasu (dla wiarygodnego potwierdzania czasu).
- W Internecie wykorzystywana jest specyfikacja X.509 v3, PKIX, RFC 4158, RFC 5280.

Infrastruktura kluczy publicznych, cd.

- Hierarchiczny system urzędów certyfikujących oferujących publicznie swoje usługi:
  - weryfikacja tożsamości użytkowników ubiegających się o certyfikaty,
  - zarządzanie kluczami kryptograficznymi:
    - generowanie par kluczy dla użytkowników,
    - bezpieczne przechowywanie kluczy,
  - zarządzanie certyfikatami:
    - wystawianie certyfikatów kluczy publicznych,
    - generowanie list certyfikatów unieważnionych, CRL, (ang. *Certificate Revocation List*).

Komponenty infrastruktury kluczy publicznych

- Urzędy certyfikujące
- Punkty rejestrujące, poręczające zgodność kluczy z identyfikatorami (lub innymi atrybutami) posiadaczy certyfikatów
- Użytkownicy certyfikatów podpisujący cyfrowo dokumenty
- Klienci weryfikujący podpisy cyfrowe i ścieżki certyfikacji do zaufanego urzędu certyfikującego

Komponenty infrastruktury kluczy publicznych, cd.

- Repozytoria przechowujące i udostępniające certyfikaty i listy unieważnień:
  - serwery LDAP (Lightweight Directory Access Protocol), RFC 4510,
  - respondery OCSP (Online Certificate Status Protocol), RFC 2560,
  - serwery WWW,
  - serwery FTP,
  - serwery DNS, DNSsec, RFC 3833, RFC 4033, RFC 4034, RFC 4035, RFC 4398, RFC 4509, RFC 4641, RFC 5155,
  - X.509 CRL (Certificate Revocation List), RFC 5280,
  - agenci systemu katalogowego X.500, DSA (Directory System Agents),
  - korporacyjne bazy danych.

Problemy infrastruktury kluczy publicznych

- Jednoznaczna identyfikacja podmiotu:
  - kompromis między pełnymi danymi identyfikującymi a prywatnością podmiotu,
  - różne klasy certyfikacji o różnym poziomie zaufania.
- Pierwszy certyfikat:
  - bezpieczne certyfikowanie urzędów certyfikujących,
  - wzajemna certyfikacja różnych urzędów,
  - wbudowanie certyfikatów wybranych urzędów najwyższego poziomu na stałe w aplikacji, przeglądarce www,
  - weryfikacja podpisu poświadczającego integralność kodu aplikacji.

Przesyłanie certyfikatów

- Powszechnie uznanym za standardowy mechanizmem pobierania certyfikatów jest przekazywanie ich jako typ MIME application/x-x509-ca-cert, RFC 2045 do 2049.
- Protokoły wymiany informacji niezbędnych do właściwego zarządzania infrastrukturą kluczy publicznych:
  - CMP (Certificate Management Protocol), RFC 4210,
  - CRMF (Certificate Request Message Format), RFC 4211,
  - CMC (Certificate Management over CMS), RFC 5273,
  - SCVP (Server-based Certificate Validation Protocol), RFC 5055.

Kategorie certyfikatów

- Polskie ustawodawstwo rozróżnia dwa rodzaje certyfikatów.
- Certyfikaty zwykłe (tzw. powszechne)
  - obejmują takie zastosowania jak szyfrowanie danych, poczta elektroniczna, www, urządzenia sieciowe, oprogramowanie.
- Certyfikaty kwalifikowane:
  - wywołują skutki prawne równoważne podpisowi własnoręcznemu (Ustawa z dn. 18.09.2001 o podpisie elektronicznym);
  - są przeznaczone dla osób fizycznych, wydawane na podstawie umowy i po (osobistej) weryfikacji tożsamości w Punkcie Rejestracji CA;
  - znajdują zastosowanie w każdym przypadku składania oświadczenia woli (również e-faktury);
  - nie służą do szyfrowania dokumentów.

SSH – Secure Shell

- Poufność przesyłanych danych
- Integralność przesyłanych danych
- Kompresja przesyłanych danych
- Uwierzytelnianie maszyny, z którą się łączymy
- Uwierzytelnianie użytkownika
- Zdalny terminal
- Tunelowanie portów TCP
- Przesyłanie plików
- Model klient-serwer, demon domyślnie nasłuchuje na porcie 22
- SSH-1, 1995, podatny na atak „man in the middle”
- SSH-2, 1996
- RFC 4251 – 4255

Podstawy bezpieczeństwa SSH

- Wszystkie algorytmy są znane i jawne.
- Klucze są na tyle długie, aby zabezpieczyć przed brutalnym atakiem na dziesięciolecia.
- Algorytmy są negocjowane – można zaprzestać używania skompromitowanego algorytmu.

Architektura SSH-2

- Warstwa transportowa (ang. *transport layer*):
  - zapewnienie poufności i integralności danych,
  - kompresja danych,
  - negocjacja algorytmów wymiany kluczy, szyfrowania, skrótu kryptograficznego i kompresji,
  - ustalenie kluczy sesji dla szyfru symetrycznego,
  - uwierzytelnienie serwera wobec klienta,
  - udostępnienie wyższym warstwom interfejsu do przesyłania pakietów,
  - zmiana klucza sesji po przesłaniu 1 GB danych lub po upływie godziny.
- Warstwa uwierzytelniania użytkownika (ang. *user authentication*):
  - korzysta z szyfrowanego połączenia udostępnianego przez warstwę transportową.
- Warstwa połączenia (ang. *connection layer*):
  - definiuje pojęcie kanałów i multipleksuje kanały,
  - przykładowe typy kanałów: powłoka tekstowa, port forwarding.

Uwierzytelnianie serwera wobec klienta

- Serwer posiada klucz prywatny i publiczny – może mieć więcej niż jedną parę.
- Klient pamięta nazwy serwerów i ich klucze publiczne.
- Można zastosować hierarchię certyfikatów.
- Problem pierwszego połączenia, gdy brak infrastruktury klucza publicznego:
  - akceptacja klucza serwera przez użytkownika przy pierwszym połączeniu,
  - weryfikacja klucza przy kolejnych połączeniach.

Uwierzytelnianie użytkownika

- Jest inicjowane przez klienta.
- Najczęstszą metodą jest identyfikator i hasło podawane przez użytkownika.
- Identyfikacja za pomocą klucza publicznego i prywatnego użytkownika.
- Identyfikacja maszyny, na której jest uruchomiony klient.
- Może być wyłączone.

Algorytmy wymiany klucza

- Wymagane:
  - diffie-hellman-group1-sha1 [RFC 2409]
  - diffie-hellman-group14-sha1 [RFC 3526]

Algorytmy szyfrowania z kluczem publicznym używane w certyfikatach

- Wymagany:
  - ssh-dss
- Zalecany:
  - ssh-rsa
- Opcjonalne:
  - pgp-sign-rsa
  - pgp-sign-dss

Algorytmy szyfrowania symetrycznego

- Wymagany:
  - 3des-cbc
- Zalecany:
  - aes128-cbc
- Opcjonalne:
  - blowfish-cbc
  - twofish[128,192,256]-cbc
  - aes[192,256]-cbc
  - serpent[128,192,256]-cbc
  - arcfour
  - idea-cbc
  - casr128-cbc
- Niezalecany:
  - none

Algorytmy skrótu

- Wymagany:
  - hmac-sha1
- Zalecany:
  - hmac-sha1-96
- Opcjonalne:
  - hmac-md5
  - hmac-md5-96
- Niezalecany:
  - none

Algorytmy kompresji

- Wymagany:
  - none
- Opcjonalny:
  - zlib

SCP – secure copy – wysyłanie pliku

- Użytkownik wydaje polecenie:
  `scp file1 host:file2`
- Klient SCP buduje połączenie SSH do maszyny `host` – uruchamia (ang. *fork*) klienta SSH.
- W zbudowanym połączeniu przesyła polecenie:
  `scp -t file2`
- Demon SSH uruchamia przesłane polecenie na maszynie `host` – uruchamia (ang. *fork*) serwer SCP.
- Klient SCP przesyła plik `file1` do serwera SCP.
- Serwer SCP zapisuje plik `file2`.

SCP – secure copy – ściąganie pliku

- Użytkownik wydaje polecenie:
  `scp host:file1 file2`
- Klient SCP buduje połączenie SSH do maszyny `host` – uruchamia (ang. *fork*) klienta SSH.
- W zbudowanym połączeniu przesyła polecenie:
  `scp -f file1`
- Demon SSH uruchamia przesłane polecenie na maszynie `host` – uruchamia (ang. *fork*) serwer SCP.
- Serwer SCP przesyła plik `file1` do klienta SCP.
- Klient SCP zapisuje plik `file2`.

SCP

- Nie ma żadnego oficjalnego standardu.
- Potrafi skopiować plik, zachowując standardowe uniksowe uprawnienia.
- Pozwala na używanie metaznaków w nazwach plików.
- Potrafi kopiować rekurencyjnie podkatalogi.
- Protokół jest mieszanką komunikatów tekstowych i binarnych, jest nierozszerzalny.