Bezpieczna komunikacja w sieci

Protokoły bezpiecznej komunikacji

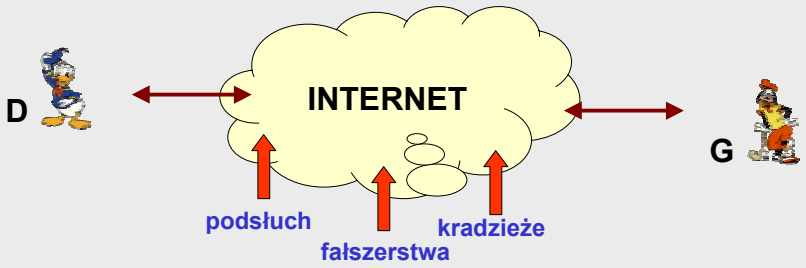
Wprowadzenie

- **Protokoły bezpiecznej komunikacji a model OSI**
  - Warstwa sieciowa - IPSec
  - Warstwa aplikacji - SSH, SSL, S/MIME, PGP
- **Wykorzystanie protokołów bezpiecznej komunikacji**
  - Handel elektroniczny (przeglądarki internetowe) - SSL
  - Zdalny dostęp do poufnych danych (przeglądarki - SSL)
  - Zdalny dostęp do zasobów komputera - SSH
  - Bezpieczna poczta - S/MIME, PGP/MIME
  - Wirtualne sieci prywatne (VPN) - IPSec (L2TP)

Protokoły

- Sformułowanie problemu

Zapewnić bezpieczeństwo sesji pomiędzy dwoma aplikacjami działającymi na dowolnych dwóch komputerach sieci
(komunikujących się za pomocą protokołu TCP/IP)



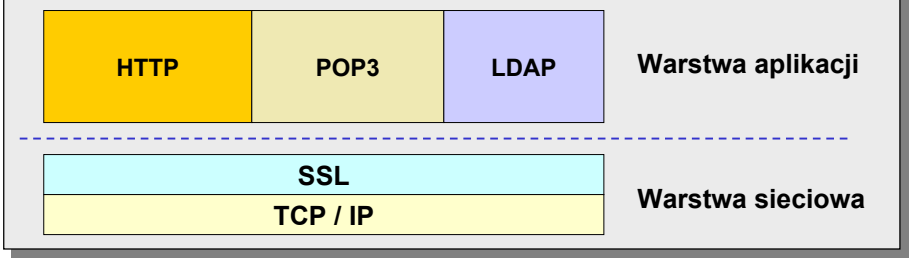
D i G nie są sobie uwierzytelnione przez jednostkę nadrzędną (jak np. centrum autoryzacji w systemie Kerberos)

Protokół SSL

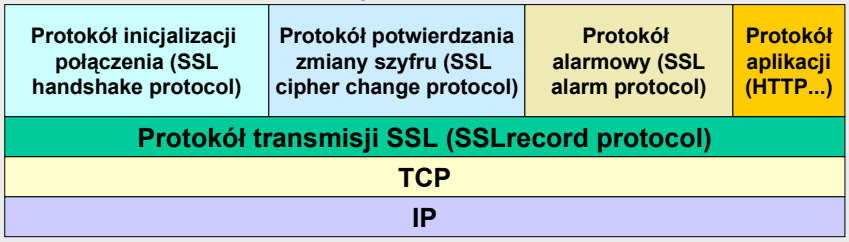
- Autor - Netscape
- Warstwy ochrony
  - Uwierzytelnianie serwera (i opcjonalnie - klienta)
  - Ochrona poufności danych - szyfrowanie
  - Ochrona integralności danych - MAC
- Wersje SSL
  - Wersja 1 - złamana podczas pierwszej prezentacji
  - Wersja 2 - powszechnie używana
- Typowe zastosowanie - przeglądarki (HTTPS)
  - Szeroki wachlarz obsługiwanych technik szyfrowania (symetryczne - DES, 3DES, AES, IDEA...; asymetryczne - RSA, DSA, DH)
  - Ograniczenia eksportowe na silną kryptografię (512b RSA itp.)

Protokół SSL

SSL a warstwowy model OSI



Stos protokołów



SSL - informacje wstępne

- **Połączenie**

  Logiczne połączenie między klientem i serwerem

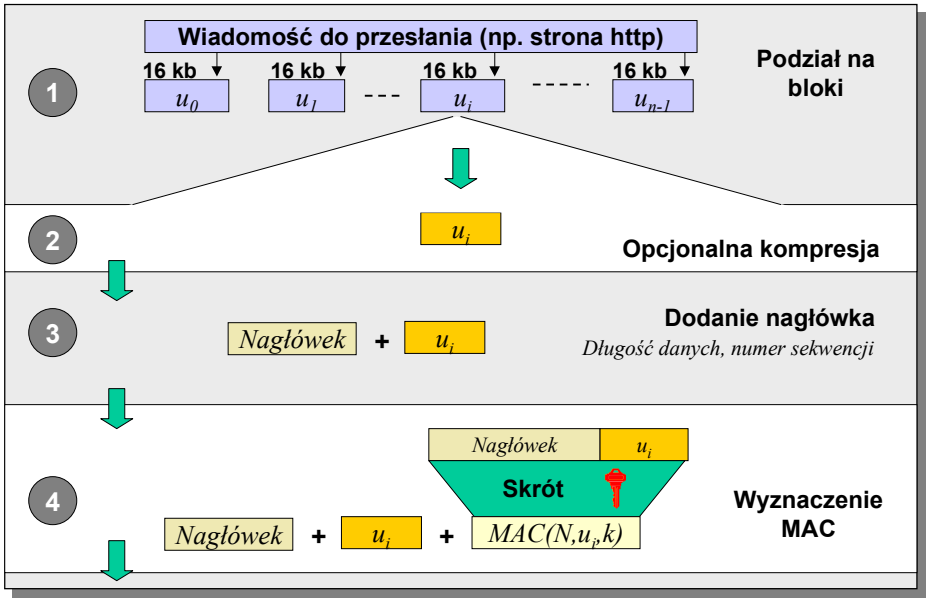
- **Sesja**

  Skojarzenie pary klient-serwer, któremu przyporządkowany jest zestaw atrybutów jednoznacznie charakteryzujących komunikację.

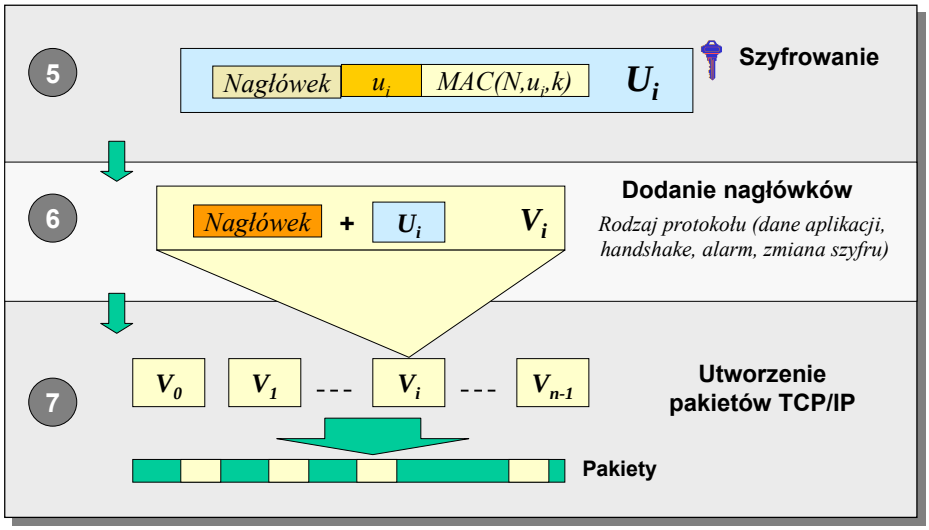
  Pojedyncza sesja = szereg połączeń

- **Atrybuty sesji**
  - Identyfikator (generowany przez serwer dla identyfikacji klienta)
  - Certyfikat uwierzytelniający stronę
  - Uzgodniona metoda kompresji danych
  - Uzgodniona metoda szyfrowania i stosowany algorytm skrótu
  - 48-bajtowy sekret wymieniony w fazie inicjalizacji
  - Inne informacje pomocnicze

Protokół transmisji danych w SSL



Protokół transmisji danych w SSL



SSL - Handshake (ustanowienie sesji)

0

HTTP

Wybranie linku do strony chronionej przez serwer WWW protokołem HTTPS

HTTPS

1

Nawiązanie połączenia

Klient	Serwer
Wiadomość CLIENT_HELLO Wersja SSL, liczba losowa (24B), czas, lista obsługiwanych algorytmów szyfrowania symetrycznego i metod wymiany klucza sesji, ID (gdy nowa sesja-0)	
	Analiza i odesłanie SERVER_HELLO Wersja SSL, liczba losowa (24B) i czas, wybrana metoda wymiany klucza sesji, wybrany algorytm szyfrowania symetrycznego i wyznaczania skrótu, ID przyznany sesji

SSL - ustanowienie sesji

2

Uwierzytelnienie

Klient	Serwer
	Wysłanie certyfikatu Krok jest opcjonalny i jest pomijany w przypadku uzgodnienia do wymiany klucza prostego wariantu metody Diffie-Hellmana. Jeżeli certyfikat jest wysyłany to ma on format określony przez X.509
Weryfikacja certyfikatu Weryfikacji podlegają: data ważności, CA, dane certyfikatu. Jeżeli CA nie znajduje się na liście zaufanych ośrodków - kontynuacja sesji zależy od użytkownika. Niepomyślna weryfikacja to zerwanie sesji. Pomyślna - serwer jest uwierzytelniony	
	Opcjonalne żądanie certyfikatu klienta Zwykle pomijane (serwer nie ma wtedy klucza publicznego klienta)
Wysłanie certyfikatu	
	Weryfikacja certyfikatu klienta

SSL - ustanowienie sesji

3 **Ustanowienie klucza sesji**

Klient	Serwer
Generacja i wysłanie sekretu wstępnego Tzw. sekret wstępny (pre-master secret) to liczba losowa, o długości zwykle 48B. Sekret jest szyfrowany kluczem publicznym - klucz jest w posiadaniu klienta (w wyniku wymiany certyfikatu) lub jest dosyłany na osobną prośbę klienta (ograniczenie eksportowe długości klucza do 512b)	**Deszyfracja sekretu wstępnego** Odszyfrowanie przy użyciu klucza prywatnego

Utworzenie kluczy szyfrowania i uwierzytelniania (MAC)

Klucze są tworzone na bazie wymienionego sekretu wstępnego i wymienionych w pierwszej fazie połączenia liczb losowych (serwera - LS i klienta - LK) w następujący sposób: tworzony jest tzw. sekret główny (master secret) MS = HASH(sekret wstępny, LS, LK). MS ma 48 bajtów i jest używany do generacji kluczy w drodze uzgodnionej procedury (algorytmy wyznaczania liczby losowej inicjalizowanej MS)

Bezpieczeństwo systemów informatycznych

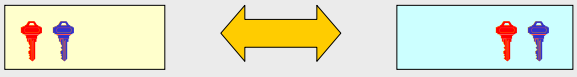
SSL - ustanowienie sesji

4 **Zakończenie procedury ustanawiania sesji**

Klient	Serwer

Wymiana komunikatu zakończenia fazy ustanawiania połączenia

Komunikat jest pierwszą wiadomością zaszyfrowaną według protokołu transmisji, wykorzystującego uzgodnione parametry kryptograficzne - szyfr symetryczny i przyjętą metodę uwierzytelniania



Parametry sesji są zapamiętane na serwerze i używane przy każdym kolejnym połączeniu - procedura ustanawiania sesji nie jest powtarzana

Krzysztof Ślot © 2002

Pozostałe protokoły SSL

- **Protokół wymiany komunikatów o błędach**
  - Dwa rodzaje błędów - FATAL, WARNING
  - Wymiana danych zabezpieczona szyfrowaniem i ochroną autentyczności
- **Protokół potwierdzenia zmiany szyfru**
  - Pojedyncza wiadomość o zakończeniu negocjacji technik kryptograficznych do ochrony sesji
  - Wymiana danych zabezpieczona szyfrowaniem i ochroną autentyczności

Następca SSL - protokół TLS (transport layer security)

- Zasadniczo stanowi rozwinięcie przedstawionej procedury
- **Informacje szczegółowe:** http://www.faqs.org/rfcs/rfc2246.html

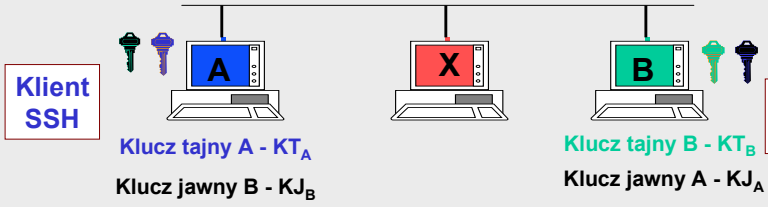
Protokół SSH

- Wprowadzony dla zapewnienia bezpiecznej realizacji operacji zdalnych systemu UNIX - rsh, rcp, rlogin.
- Połączenie odbywa się przez gniazdo TCP/IP (przydzielony port 22)
- Wbudowana kompresja przesyłanych danych (LZ78 - ZIP)
- Dojrzałość - wersja 2 protokołu

Protokół SSH

Czynności konfiguracyjne

Generacja par kluczy RSA na komputerach chronionych protokołem



Połączenie

- Połączenie jest inicjowane zawsze przez klienta
- Rozpoczyna się wymianą tzw. łańcucha identyfikującego (uzgodnienie wersji protokołu, i określenie systemu operacyjnego)
- Serwer przesyła klientowi swój klucz publiczny oraz publiczny klucz tymczasowy, który podlega regeneracji co godzinę (obydwa - RSA)

Protokół SSH

Uwierzytelnienie serwera (B) przez klienta (A)

- Generacja liczby losowej (256b)
- Szyfrowanie kluczem publicznym odbiorcy KJ_B (ściślej, obydwoma kluczami serwera)

Wysłanie wiadomości do B

- Deszyfracja swoim kluczem prywatnym (KT_B)
- Zaszyfrowanie kluczem publicznym nadawcy (KJ_A)

Odesłanie wiadomości do A

- Deszyfracja i porównanie z oryginałem - jeżeli takie same, to autoryzacja połączenia

Protokół SSH

Uzgodnienie metody szyfrowania sesji

- Nie ma ograniczeń co do techniki szyfrowania (IDEA, AES...)
- Zależy od konfigurowanych preferencji i możliwości stron, dokonywane w fazie uwierzytelniania

Tworzenie klucza sesyjnego

- Generacja liczby losowej K_S (klucza sesyjnego)
- Zaszyfrowanie kluczem publicznym KJ_A

$C_S = f_{KJA}(K_s)$  Wysłanie wiadomości do A

- Deszyfracja klucza sesyjnego $K_S = f_{KTA}(C_s)$

Klucz sesyjny jest okresowo regenerowany

Protokół SSH

Sesja

- Po autoryzacji i utworzeniu klucza następuje rozpoczęcie sesji konsolowej, sesji X-windows lub wykonanie innego polecenia
- Transmisja to wymiana pakietów o specjalnym formacie (długość, dane, suma kontrolna)
- Szyfrowanie jest uaktywniane w momencie przesłania klucza

Tunelowanie połączeń

- SSH pozwala na przekierowanie komunikacji odbywającej się między dowolnymi aplikacjami dwóch komputerów do utworzonego, bezpiecznego kanału ('tunelu'). Pakiety adresowane do aplikacji na dowolny port są kierowane na port 22 i odpowiednio oznaczane, a następnie, w komputerze odbiorcy są dostarczane do odpowiedniej aplikacji przez mechanizmy SSH.

Protokół SSH

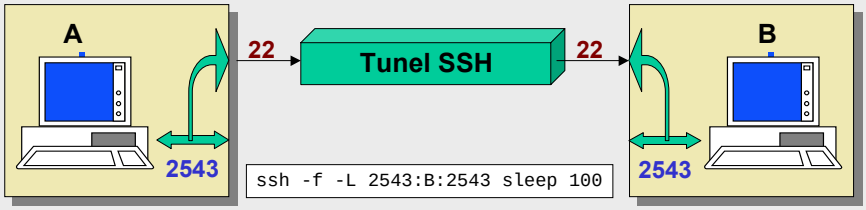
Przekierowanie lokalnego portu TCP

Założenia:

- Serwis X, nieszyfrowany oferowany przez B na porcie p (2543)
- Istnieje tunel SSH między komputerami (B jest serwerem SSH)

Cel

Wykorzystać kanał szyfrowany dla serwisu X




```
ssh -f [user@remote host] -L [local port]:[remote host]:[remote port] [some command]
```

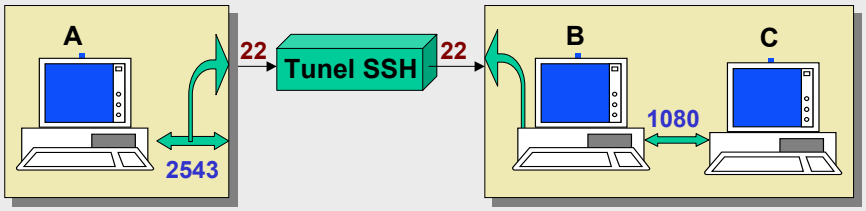

Protokół SSH

Przekierowanie portu zdalnego

Założenia:

- Serwis X, nieszyfrowany oferowany przez C na porcie p (1080)
- C jest połączony z B
- Istnieje tunel SSH między komputerami A i B

Cel Wykorzystać kanał szyfrowany dla komunikacji A - C




```
ssh -f -R 2543:B:1080 C sleep 100
```

Protokół SSH

- Potencjalne zagrożenie bezpieczeństwa - przekierowywanie portów eliminuje możliwość ochrony metodą filtracji portów

- Dostępność oprogramowania
  - Putty - darmowy klient SSH dla MS
  - OpenSSH - darmowy serwer SSH dla MS Windows

- Szczegółowe informacje
  - http://www.free.lp.se/fish/rfc.txt
  - ...

PGP (1991)

- Uwierzytelnianie
  - Algorytm DSA
  - Uwierzytelnianie może dotyczyć każdej wiadomości
- Poufność
  - Wiadomości są szyfrowane, przy czym klucz szyfrowania jest dla każdej wiadomości inny

    Klucz dla kolejnej wiadomości jest dołączany do wiadomości bieżącej. Kluczem jest 128 bitowa liczba losowa. Liczba ta jest generowana przy wykorzystaniu informacji o ruchu myszki i uderzeniach klawiszy mających miejsce w czasie trwania sesji (losowość). Klucz jest szyfrowany kluczem publicznym odbiorcy wiadomości.
- Kompresja
  - Zwiększenie odporności na ataki kryptograficzne

PGP

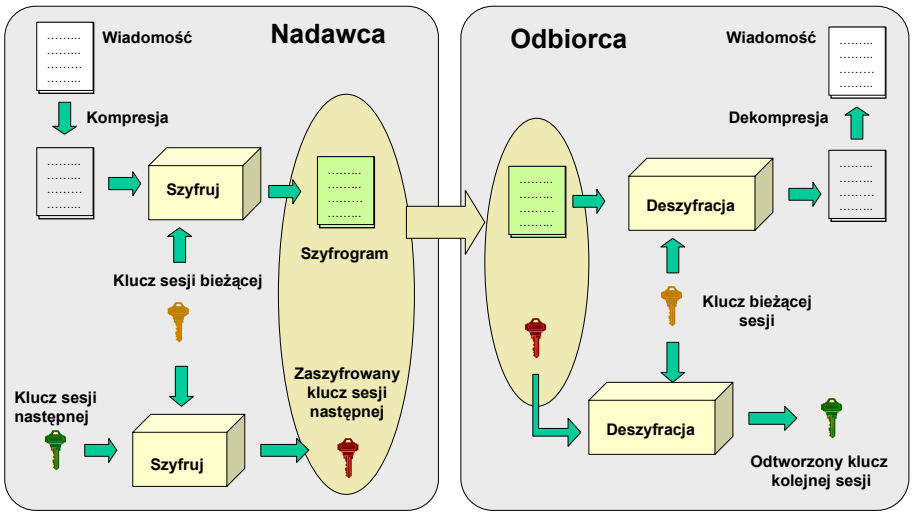
Konwersja radix-64

Zaszyfrowane dane to dowolne wartości 8-bitowe. Ponieważ wiele programów poczty radzi sobie tylko ze znakami ASCII, PGP może dokonać konwersji na znaki ASCII, wg zasady 3B kodu - 4B znaków ASCII. W ten sposób 24 bitowe ciągi znaków są dzielone na cztery 6-bitowe sekcje, a każdej z nich jest przypisywany symbol ASCII, zgodnie z podaną tablicą.

Tablica kodowania radix-64

liczba 6b	znak kodu	liczba 6b	znak kodu	liczba 6b	znak kodu	liczba 6b	znak kodu
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
1	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Komunikacja w PGP



Klucz prywatny PGP szyfrowany skrótem 'passphrase' - frazy

MIME i S/MIME

Multipurpose Internet Mail Extensions

➡ **Informacje dotyczące bezpieczeństwa przesyłane jako uzupełnienie wiadomości**

Format ogólny

```
Content-type: multipart / type, boundary = „Granica”
Content transfer encoding: base64

-- Granica
informacje o szyfrowaniu
-- Granica
wiadomość
-- Granica
podpis
-- Granica --
```

S/MIME wersja 3 - wydaje się że zapewnia wystarczającą ochronę

PGP a poczta elektroniczna

➡ **PGP może być użyte do szyfrowania poczty ...**

```
From: Hans Kloss <hkloss@abwehra.org>
   To: Czapajew Gieroj<cg@pik.su>
   Mime-Version: 1.0
   Content-Type: multipart/encrypted; boundary=nic;
      protocol="application/pgp-encrypted"

   -- nic
   Content-Type: application/pgp-encrypted
   Version: 1
   -- nic
   Content-Type: application/octet-stream
   -----BEGIN PGP MESSAGE-----
   Version: 2.6.2

   HUYGTf(*jlwonemfysiru*uenrbxhBYlE37(B&Bc0&DBHD89nfszfwtm.fosneu-fnebV
   oencjisurgt9OJBFYVywyc56Vhjlflgojhmoom957s7&(T^59n.gJOInf8^%d0eamqVW3
   c-d=)dmgj:firngig-*Bld.KGNUXVydo*hfirl87sgbuebvyVtcdkurbvauduygsv=
   -----END PGP MESSAGE-----
   -- nic --
```

➡ **... i podpisywania**

```
Content-Type: multipart/signed; boundary=granica; micalg=pgp-md5;
    protocol="application/pgp-signature"
```