UNIWERSYTET
OPOLSKI

INSTYTUT MATEMATYKI I INFORMATYKI

ul. Oleska 48, 45–052 Opole
tel. +48 77 452 72 05
fax +48 77 452 72 07
im@math.uni.opole.pl
www.math.uni.opole.pl

Sieci Komputerowe
Bezpieczeństwo sieci komputerowych

dr Zbigniew Lipiński
Instytut Matematyki i Informatyki
ul. Oleska 48
50-204 Opole
zlipinski@math.uni.opole.pl

Zagadnienia

Zasady bezpieczeństwa w sieciach i systemach komputerowych

Zarządzanie bezpieczeństwem

Pojęcia związane bezpieczeństwem systemów i sieci komputerowych

Przestępstwa w sieciach i systemach komputerowych

Przykłady ataków na sieci i systemy komputerowe

Systemy wykrywania i obrony przed intruzami

Firewall , serwery proxy

Bezpieczna komunikacja w sieci

Secure Socket Layer, Transport Layer Security

Protokół HTTPS

Data Encryption Standard, Advanced Encryption Standard

Algorytm RSA, Algorytm Diffie-Hellmana

Bezpieczeństwo poczty elektronicznej , certyfikat

System Kerberos

IPSec

Internet Key Exchange

Internet Security Association and Key Management Protocol

Tryby pracy Ipsec

Bezpieczeństwo sieci w dokumentach RFC.

RFC 1281 'Guidelines for the Secure Operation of the Internet', R.Pethia, S.Crocker, B.Fraser,1991.

RFC 2179 'Network Security For Trade Shows', A. Gwinn, 1997.

RFC 2196 'Site Security Handbook', B. Fraser, 1997.

RFC 2316 'Report of the IAB Security Architecture Workshop', S. Bellovin, 1998.

RFC 2407 'The Internet IP Security Domain of Interpretation for ISAKMP', D. Piper

RFC 2504 'Users' Security Handbook', E. Guttman, L. Leong, G. Malkin [1999]

RFC 2623 'NFS Version 2 and Version 3 Security Issues and the NFS Protocol's Use of RPCSEC_GSS and Kerberos V5', M. Eisler, 1999.

RFC 2709 'Security Model with Tunnel-mode IPsec for NAT Domains', P. Srisuresh.

RFC 2725 'Routing Policy System Security', C.Villamizar, C.Alaettinoglu, D.Meyer, S.Murphy, 1999.

RFC 2828 'Internet Security Glossary', R. Shirey, 2000.

Zagadnienia bezpieczeństwa sieci i systemów komputerowych związane jest z:

- ochroną hostów przed włamaniami, kradzieżą danych lub oprogramowania,
- ochroną przed oszustwami, fałszerstwami w systemach i sieciach komputerowych,
- ochroną przed szpiegostwem komputerowym (podsłuch, kradzież danych),
- ochroną przed niszczeniem danych i programów,
- paraliżowaniem pracy systemów komputerowych (obniżenie dostępności systemu),
- ochroną poufności, integralności danych transmitowanych danych,
- uwierzytelnieniem i autoryzacją użytkowników, hostów i usług,
- ochroną systemów teleinformatycznych przed złośliwym oprogramowaniem.

Zarządzanie bezpieczeństwem

Zarządzanie bezpieczeństwem w sieciach i systemach komputerowych polega na:

- tworzeniu polityki bezpieczeństwa dla sieci, systemów komputerowych,
- tworzenie zasad dostępu do zasobów,
- tworzeniu norm i zaleceń bezpieczeństwa, dobrych praktyk dotyczących bezpieczeństwa,
- tworzenie procedur analizy ryzyka związanego z bezpieczeństwem,
- tworzenie procedur reagowania i dokumentowanie incydentów naruszenia bezpieczeństwa,
- określaniu klas bezpieczeństwa systemów komputerowych,
- monitorowaniu stanu zabezpieczeń, monitorowanie transmitowanych danych, aktywności hostów,
- specyfikowaniu narzędzi służących do analizy zabezpieczeń,
- aktualizacji systemów operacyjnych i aplikacji,
- definiowaniu mechanizmów uwierzytelniania i autoryzacaji hostów, usług i użytkowników.

Pojęcia związane bezpieczeństwem systemów i sieci komputerowych

poufność (ang. confidentiality) ochrona informacji przed nieautoryzowanym jej ujawnieniem.

integralność danych (data integrity) ochrona informacji przed nieautoryzowanym jej zmodyfikowaniem.

identyfikacja (ang. identification) możliwość rozróżnienia użytkowników, np. użytkownicy w systemie operacyjnym są identyfikowani za pomocą UID (user identifier).

uwierzytelnienie (authentication) proces weryfikacji tożsamości użytkownika; najczęściej opiera się na tym co użytkownik wie (proof by knowledge), np. zna hasło, co użytkownik ma (proof by possession), np. elektroniczną kartę identyfikacyjną.

autentyczność (ang. authenticity) pewność co do pochodzenia, autorstwa i treści danych.

niezaprzeczalność (ang. nonrepudiation) ochrona przed fałszywym zaprzeczeniem przez nadawcę – faktu wysłania danych, przez odbiorcę - faktu otrzymania danych.

autoryzacja (ang. authorization) proces przydzielania użytkownikowi praw dostępu do zasobów.

kontrola dostępu (ang. access control) procedura nadzorowania przestrzegania praw dostępu do zasobów.

Przestępstwa w sieciach i systemach komputerowych

hacking - włamywanie do systemów lub sieci komputerowych, naruszenie art. 267 §1 Kodeksu karnego.

sniffing – podsłuchiwanie pakietów w sieci. Skanowanie sieci jest naruszeniem art. 267 § 2 Kodeksu karnego.

session hijacking - polega na uzyskiwaniu nieuprawnionego dostępu do systemów poprzez przechwycenie sesji legalnego użytkownika

browser hijacker - oprogramowanie, które modyfikuje ustawienia przeglądarki internetowej użytkownika w celu przechwycenia sesji HTTP.

spoofing - podszywanie się pod innego użytkownika w sieci.

IP spoofing - rodzaj ataku w którym adres IP napastnika rozpoznawany jest jako adres zaufany.

TCP spoofing - podszywanie bazujące na oszukaniu mechanizmu generowania numerów ISN.

UDP spoofing – modyfikacja pakietów w celu atakowania usług i protokołów wykorzystujących protokół UDP.

phishing (password fishing) - metoda uzyskiwania haseł dostępu do zasobów internetowych (kont bankowych użytkownika) za pośrednictwem e-maili. Atakujący podszywa się pod przedstawiciela instytucji finansowej próbując nakłonić odbiorców e-maili do przesłania im haseł oraz innych danych osobowych (np. poprzez fałszowanie stron WWW).

cracking – łamanie zabezpieczeń programów komputerowych, dostępu do systemów, usług sieciowych.

piractwo komputerowe - to kopiowanie, reprodukowanie, używanie i wytwarzanie bez zezwolenia produktu chronionego przez prawo autorskie.

Przykłady ataków na sieci i systemy komputerowe

Atak Denial of Service (DoS). Celem ataku DoS jest unieruchomienie całego systemu ofiary lub jego komponentów.

Smurf attack. Jest to atak DDOS (Direct Denial Of Service), polega na wygenerowaniu dużej liczby pakietów ‘ICMP echo request’ z adresem IP ofiary ataku jako źródłowym. Ofiara zostanie zalana odpowiedziami ping.

Atak SYN flood. W przypadku ataku SYN flood atakujący wysyła na adres ofiary dużą liczbę segmentów SYN protokołu TCP adresowanych z dowolnych (nieistniejących) adresów IP. Ofiara odpowiada segmentami SYN/ACK i rozpoczyna bezowocne oczekiwanie na segmenty ACK. W trakcie oczekiwania wyczerpują się zasoby hosta ofiary.

W 1997 r. atak SYN flood na WebCom wyłączył z użycia ponad 3000 stron WWW.

Ping of Death. Atak ten przeprowadza się poprzez wygenerowanie pofragmentowanych pakietów ICMP przekraczających w sumie 64 kB, scalanie może powodować błędy prowadzące do zawieszenia się systemu hosta.

Land attack. Atakujący wysyła segment SYN na adres ofiary podając jej własny adres jako źródłowy i nadając ten sam numer portu źródłowego i docelowego. Stacja TCP ofiary nigdy nie zestawi połączenia zapętlając się w nieskończoność.

Ochrona hostów:

- uniemożliwienie startowania systemu z nośników wymiennych,
- ograniczenie stosowania nośników wymiennych (stacji dyskietek, nagrywarek),
- ograniczenie wykorzystania przestrzeni lokalnych dysków twardych,
- rejestracja prób dostępu do systemu i ich limitowanie (kontrola, kto i kiedy korzystał z systemu),
- bezpieczne kasowanie poufnych danych,
- uniemożliwienie usunięcia / wyłączenia zabezpieczeń, np. antywirusowych,
- konsekwentna polityka zarządzania hasłami użytkowników.

Ochrona sieci

- dobór medium transmisyjnego i topologii sieci,
- fizyczna ochrona pomieszczeń z urządzeniami sieciowymi i serwerami usług ,
- zdefiniowanie listy stanowisk, z których dany użytkownik może uzyskać dostęp do systemu (adresy MAC lub IP),
- usuwanie nieużywanych kont użytkowników.

Ochrona usług sieciowych:

- usunięcie z systemu, dezaktywacja wszystkich usług zbędnych,
- zastąpienie usług niezbędnych odpowiednikami o podwyższonym bezpieczeństwie (jeśli to możliwe i takie odpowiedniki są dostępne),
- kontrola dostępu do pozostałych usług (np. poprzez zapory sieciowe firewall) .

Exploit - programy służące do wyszukiwania, wykorzystywania luk w oprogramowaniu.

Przykład. Narzędzie Metasploit służy do wyszukiwania zagrożeń, luk w oprogramowaniu, audytu i testowania bezpieczeństwa sieci i programów komputerowych.

Służy do uruchamiania exploitów w celu testowania zagrożeń i luk w oprogramowaniu.

Baza exploitów umieszczona jest na stronie http://metasploit.com/

Przykłady exploitów opisane w bazie:

- *Microsoft RPC DCOM Interface Overflow*
- *Cisco VPN Concentrator 3000 FTP Unauthorized Administrative Access*
- *Broadcom Wireless Driver Probe Response SSID Overflow*
- *This module exploits a stack buffer overflow in the Broadcom Wireless driver that allows remote code execution in kernel mode by sending a 802.11 probe response that contains a long SSID.*
- *Firefox 3.5 escape() Return Value Memory Corruption*
- *Wireshark LWRES Dissector getaddrsbyname_request Buffer Overflow*
- *PuTTy.exe <= v0.53 Buffer Overflow.*

Systemy wykrywania intruzów

Systemy IDS (Intrusion Detection Systems) do wykrywania zagrożeń stosują:

- analizę sygnatur zagrożeń,
- statystyczną analizę anomalnych zachowań użytkowników.

Rodzaje systemów IDS:

- system wykrywania intruzów na hostach, Host-based IDS.
- system wykrywania intruzów w sieciach, Network IDS.

Przykłady systemów IDS:

- Next-Generation Intrusion-Detection Expert System (NIDES), http://www.sdl.sri.com/projects/nides/
- USTAT, projekt STAT, http://www.cs.ucsb.edu/~seclab/projects/stat/projects.html
- Bro Intrusion Detection System, http://bro-ids.org/
- Advanced Intrusion Detection Environment, http://aide.sourceforge.net/

Systemy ochrony przed intruzami

Systemy IPS (Intrusion Prevention Systems (IPS) służą do:

- monitorwania,
- wykrywania,
- podejmowania działań w celu ochrony przed zagrożeniami sieci i systemów komputerowych.

Systemy IPS do wykrywania zagrożeń stosują:

- analizę sygnatury zagrożeń,
- analizę heurystyczną transmitowanych pakietów w sieciach,
- monitorowanie zachowań użytkowników.

Przykłady systemów IPS:

- System Cisco IPS,
- Snort, Intrusion Prevention and Detection System (IDS/IPS), http://www.snort.org/

Ściany ogniowe – firewall'e

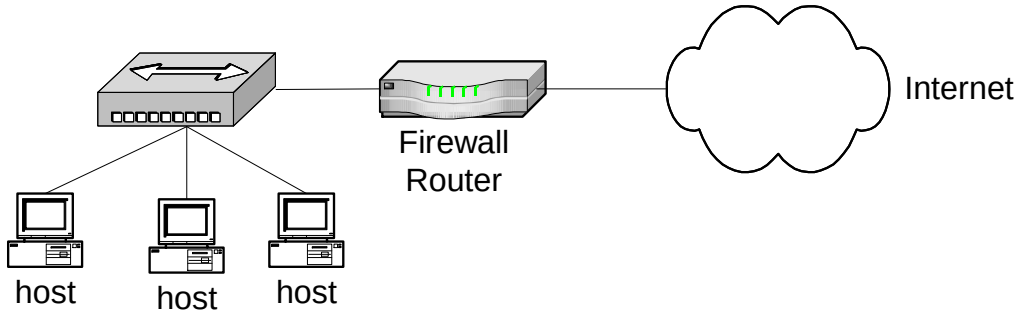
Ściana ogniowa, firewall

jest urządzeniem (lub układem urządzeń), oddzielającym sieć zewnętrzną (np. Internet) od chronionej sieci wewnętrznej.

Firewall pełni rolę gatewaya przez który przechodzi cała komunikacja między siecią zewnętrzną a chroniona podsiecią.

Firewall pozwala na:

- udostępnianie konkretnych usług na konkretnych serwerach,
- udostępnianie usług poprzez określone protokoły i porty,
- monitorowanie transmisji między hostami znajdującymi się w sieci zewnętrznej z hostami w sieci wewnętrznej.



Firewall chroni lokalne sieci

Serwery proxy

pośredniczą w komunikacji między hostami znajdującymi się w sieci lokalnej a hostami znajdującymi w sieciach zewnętrznych.

'Proxy server is a server that performs an action for another computer which cannot perform the action for itself.'

Serwery proxy pośredniczą w komunikacji między klientem a serwerem:

- pełniąc rolę klienta, np. serwer proxy w imieniu klienta generuje zapytanie do serwera WWW,
- pełniąc rolę serwera, np. serwer proxy odpowiada w imieniu serwera WWW klientowi generującemu zapytanie o stronę.

Serwery proxy

Gateway (brama)

jest serwerem pośredniczącym w komunikacji między klientem a serwerem **pełniąc rolę serwera.**

Typy serwerów proxy:

- bramy pośredniczące w transmisji (circuit-level gateways),
- bramy aplikacji (application-level gateways).

Serwery **proxy typu 'circuit-level'** budują wirtualne połączenia z hostami sieci lokalnej, działają w imieniu hosta w sieciach zewnętrznych.

Np. w żądaniach hosta o strony WWW wysyłanych do sieci zewnętrznych serwer proxy zamienia adres IP hosta na swój.

Serwery **proxy typu 'application-level'** pozwalają na konfigurowanie pośredniczenia w komunikacji dla poszczególnych aplikacji i protokołów (ftp,telnet, http, smtp).

Na serwerach proxy może być analizowana treść przesyłanych pakietów, można realizować zasady bezpieczeństwa w sieci wewnętrznej.

Serwer proxy może pełnić role bramy przez którą są udostępnianie usługi sieciowe.

Pozwala to na:

- ukrycie wewnętrznej struktury sieci w której znajdują się serwery usług,
- monitorowanie połączeń klientów z serwerami usług.

Bezpieczna komunikacja w sieci

Protokoły bezpiecznej komunikacji:

- Secure Socket Layer (SSL), Transport-Layer Security (TLS),
- https, ssh,
- Kerberos, Radius.
- Ipsec.

Mechanizmy bezpieczeństwa sieci:

- Standard X.509v3. Infrastruktura klucza publicznego (PKI).
- Security/Multipart (RFC 1847). Mechanizm bezpiecznego przesyłania załączników w poczcie elektronicznej.
- DNSsec (RFC 2065). Mechanizm zabezpieczenia serwerów DNS.

Secure Socket Layer

Protokół SSL, (ang.) **Secure Socket Layer**.

Protokół opracowany przez firmę Netscape Communications.

Specyfikacja: A. O. Freier, P. Karlton, P. C. Kocher. The SSL Protocol Version 3.0. Netscape Communications Corporation, 1996..

Protokół SSL został stworzony dla potrzeb bezpiecznej komunikacji w Internecie.

Służy do zabezpieczenia (szyfrowania) przesyłanych danych w protokole HTTP, FTP, protokołach poczty elektronicznej.

W warstwie transportowej wykorzystuje protokół **TCP**.

Protokół SSL był podstawą opracowania protokołu TLS, (Transport Security Layer).

Secure Socket Layer

Protokół SSL służy do uwierzytelnienia i szyfrowania przesyłanych danych w protokołach warstwy aplikacji modelu referencyjnego dla OSI:

- HTTP (protokół HTTPS, port 443),
- FTP (sFTP, port 22),
- protokołach poczty elektronicznej,
  - Secure SMTP (SSMTP), port 465,
  - IMAP4 over SSL, port 993, port 585,
  - Secure POP3 (SSL-POP), port 995,
- telnet, protokół SSH, port 22.

RFC 4217, P. Ford-Hutchinson, Securing FTP with TLS, 2005.

RFC 2595, C. Newman, Using TLS with IMAP, POP3 and ACAP, 1999.

RFC 2244, C. Newman, J. G. Myers, ACAP -- Application Configuration Access Protocol.

RFC 3207, P. Hoffman, SMTP Service Extension for Secure SMTP over Transport Layer Security, 2002.

RFC 4250, S. Lehtinen, C. Lonvick, The Secure Shell (SSH) Protocol Assigned Numbers, 2006.

Secure Socket Layer

Funkcje protokołu SSL:

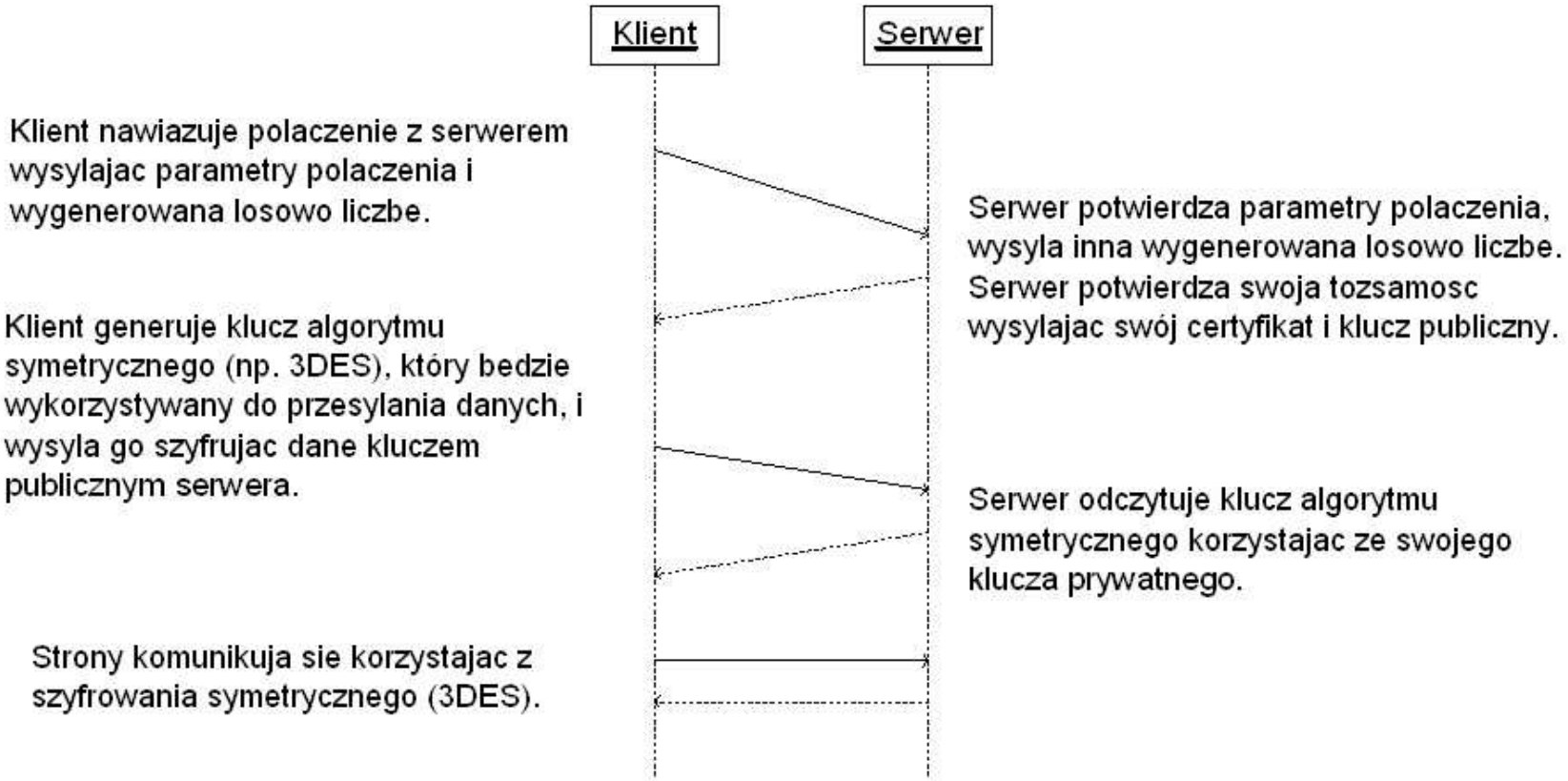
- uwierzytelnienie stron biorących udział w transmisji,
- zapewnienie poufności transmisji poprzez szyfrowanie przesyłanych danych,
- zapewnienie integralności przesyłanych danych (zabezpieczenie danych przed ich modyfikowaniem w czasie transmisji).

Do uwierzytelnienia komunikujących się stron protokół SSL wykorzystuje infrastrukturę klucza publicznego (PKI, Public Key Infrastructure), standard **X509**.

Dla zapewnienia poufności transmisji wykorzystywane są:

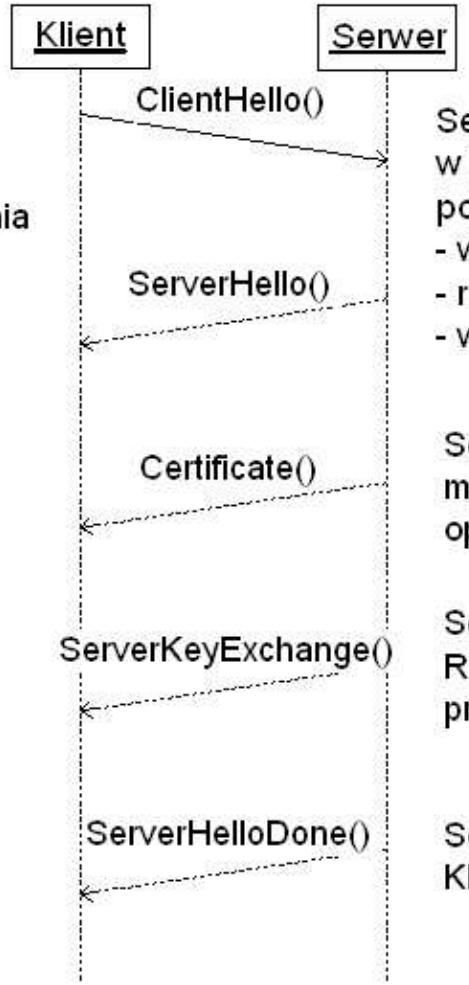
- algorytmy asymetryczne RSA, Diffie-Hellmana, Fortezza's Key Exchange Algorithm, inne,
- algorytmy symetryczne DES, 3DES, RC4, inne,
- algorytmy haszujące MD5, SHA.

Etapy bezpiecznego przesyłania danych
Wymiana klucza szyfrowania symetrycznego.



SSL- schemat bezpiecznej wymiany danych

Klient wysyla do serwera zgloszenie zawierajace m.in. obslugiwana wersje protokolu SSL, mozliwe sposoby szyfrowania i kompresji danych oraz identyfikator sesji. Komunikat zawiera losowa liczbe uzywana do generowania kluczy.



Serwer odpowiada komunikatem 'ServerHello' w którym zwraca klientowi wybrane parametry polaczenia:
- wersje protokolu SSL,
- rodzaj szyfrowania i kompresji,
- wygenerowana przez serwer losowa liczbe.

Serwer wysyla swój certyfikat aby klient mógl sprawdzic tozsamosc serwera (etap opcjonalny).

Serwer wysyla swój klucz publiczny. Rodzaj i dlugosc klucza jest okreslony przez uzyty typ algorytmu.

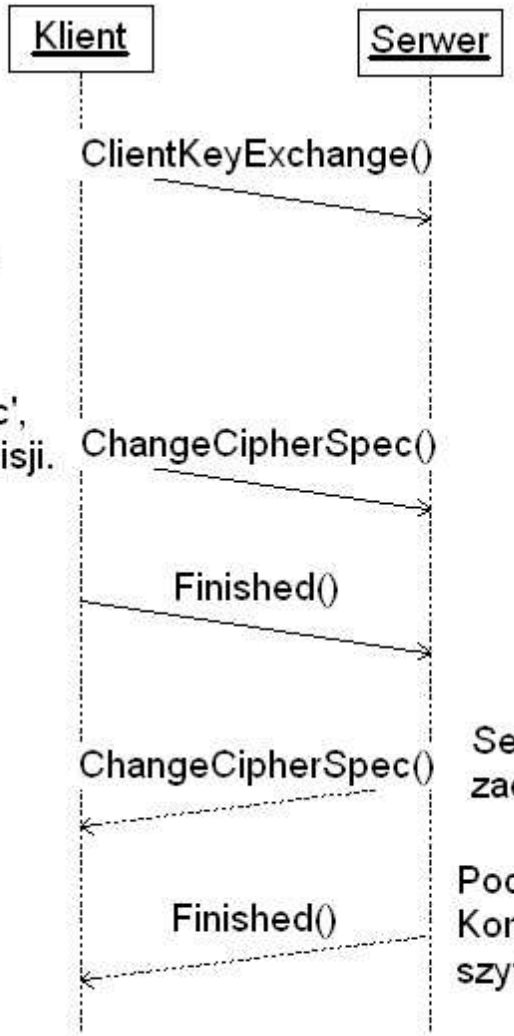
Serwer wysyla komunikat 'ServerHelloDone'. Klient moze rozpoczac budowe polaczenia.

SSL- schemat bezpiecznej wymiany danych

Klient wykorzystuje wymienione, losowe liczby do wygenerowania klucza symetrycznego (sluzacego do szyfrowania transmisji).
Klient wysyla klucz symetryczny do serwera szyfrujac go kluczem publicznym serwera.

Klient wysyla komunikat 'ChangeCipherSpec', serwer moze rozpoczac szyfrowanie transmisji.

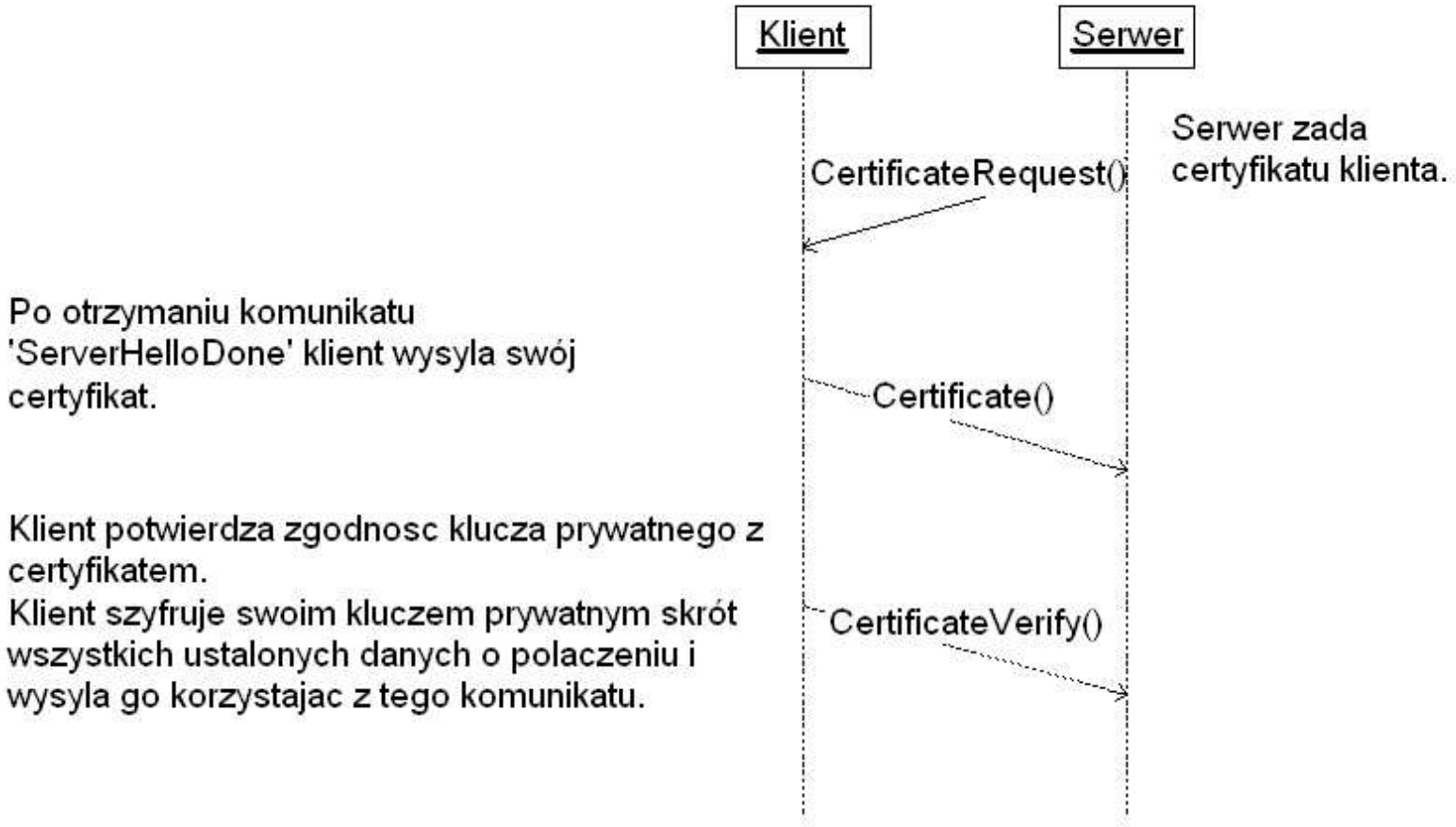
Klient wysyla komunikat 'Finished', klient informuje Serwer, ze moze odbierac szyfrowane dane.



Serwer informuje klienta, ze zaczyna szyfrowac transmisje.

Poczatek transmisji szyfrowanej. Komunikat sluzy równiez testowi szyfrowanej transmisji.

SSL - identyfikacja stron transmisji



Transport Layer Security

Specyfikacje protokołu Transport Layer Security (TLS):

- RFC 2246, T. Dierks, C. Allen, The TLS Protocol Version 1.0, 1999.
- RFC 4346, T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.1, 2006.
- RFC 5246, T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, 2008.

Protokół opracowany w celu zapewnienia poufności i integralności transmitowanych danych w sieciach komputerowych.

Protokół TLS stosuje do uwierzytelnienia wiadomości kryptograficzne funkcje haszujące.

TLS składa się z dwóch protokołów:

- TLS Record protocol (TLSR), służy do szyfrowania transmisji (DES, RC4),
- TLS Handshake protocol (TLSH), służy do bezpiecznej wymiany klucza kryptograficznego.

Protokół TLSR zapewnia bezpieczeństwo transmitowanych danych poprzez

- stosowanie szyfrowania danych (DES, RC4),
- wymianę klucz szyfrowania symetrycznego za pomocą bezpiecznej transmisji (protokół TLSH),
- zabezpieczanie integralności wysyłanych wiadomości poprzez stosowanie kodów MAC (Message Authentication Code).
- stosowanie funkcji haszujących (SHA, MD5) do wyliczania kodów MAC.

Mechanizm HMAC wykorzystuje do uwierzytelnienia wiadomości funkcje haszujące (MD-5, SHA-1) i wspólny dla obu stron wymiany danych tajny klucz (shared key).

RFC 2104, H. Krawczyk, M. Bellare, R. Canetti, HMAC: Keyed-Hashing for Message Authentication, 1997.

Bezpieczeństwo protokołu TLSH opiera się na

- stosowaniu uwierzytelnienia stron wymiany danych za pomocą asymetrycznych algorytmów kryptograficznych, kluczy publicznych (RSA, Digital Signature Standard).
- bezpiecznej wymiany kluczy kryptograficznych algorytmów symetrycznych.

Nowe elementy TLS w wersji 1.1

- Niejawny wektor inicjujący (Initialization Vector IV) został zastąpiony przez jawny wektor (explicit IV).
- Dodano ochronę przeciwko atakom CBC (Cipher block chaining attacks).

  Wprowadzono generowanie klucza inicjującego IV dla każdej wiadomości, zmianę obsługi błędów uzupełnienia.

  Opis ataku CBC:

  B. Moeller, Security of CBC Ciphersuites in SSL/TLS: Problems and Countermeasures, http://www.openssl.org/~bodo/tls-cbc.txt

  H. Krawczyk, The order of encryption and authentication for protecting communications (Or: how secure is SSL?), http://eprint.iacr.org/2001/045.
- Zmiana obsługi błędów uzupełnienia (padding errors).

  W wersji TLS 1.1 zastapiono komunikat *decryption_failed* o błędzie w uzupełnieniu na komunikat *bad_record_mac.*
- Dodano obsługę rejestrów protokołów IANA (lista rejestrów znajduje się na stronie http://www.iana.org/protocols).

HTTPS

Protokół SSL służy do zabezpieczenia transmisji w protokole HTTP.

Serwery WWW wykorzystują do bezpiecznej komunikacji **port 443** (zamiast portu 80).

Do identyfikacji transmisji szyfrowanej stosuje się URL'a z nazwa protokołu **https**.

Przykład: https://www.math.uni.opole.pl/index.htm

Komunikacja klienta WWW (przeglądarki internetowej) z serwerem WWW składa się z wielu krótkotrwałych połączeń.

Aby uniknąć procesu budowania bezpiecznego połączenia dla każdego połączenia serwer WWW buduje trwale połączenie - **sesje**.

Dla **sesji** serwer WWW generuje identyfikator sesji **SessionId**.

Przy tworzeniu połączenia klient może w komunikacie 'ClientHello' podać **SessionId** wygenerowany dla innej sesji.

Serwer traktuje takie połączenie jako kontynuacje sesji z podanym **SessionId**.

Data Encryption Standard

Algorytm DES, (ang.) **Data Encryption Standard**.

IBM 1975.

DES jest przykładem symetrycznego algorytmu szyfrującego, tzn. ten sam klucz jest używany do szyfrowania i deszyfrowania.

Za pomocą algorytmu DES szyfrowane są 64 bitowe bloki danych.

Długość klucza DES - 8 bajtów:

- 56 bitów definiuje klucz
- 8 bitów to bity parzystości każdego bajtu.

Szyfrowanie i deszyfrowanie algorytmem DES składa się z **16 cykli**.

W trakcie każdego cyklu dokonywane są obliczenia, w oparciu o wyniki obliczeń z poprzedniego cyklu i w oparciu o podklucz generowany z 64-bitowego klucza dla danego cyklu.

Zasady generowania podkluczy jest jawny.

Do szyfrowania algorytmem DES stosowane jest **16 tablic** liczbowych (maciarzy 8x8), zwanymi **S-box'ami**.

Odpowiedni dobór liczb w S-box'ach na wpływ na **bezpieczeństwo algorytmu**.

Deszyfrowanie algorytmem DES polega na zastosowaniu kluczy w odwrotnej kolejności.

3xDES

Dla zwiększenia bezpieczeństwa szyfrowanych danych stosowany jest algorytm 3DES.

3DES - trzykrotne szyfrowanie algorytmem DES.
Do szyfrowanie 3DES stosowany są dwa klucze **K_1, K_2** DES.

Dane szyfrowane są w następujący sposób:

$$tekstZaszy\ \ frowany\ \ = K_1 K_2^{-1}(K_1(tekst\ ))$$

Deszyfrowanie:

$$tekst = K_1^{-1}(K_2 K_1^{-1}(tekstZaszyfrowany))$$

$K_i(tekst)$ **szyforwanie kluczem K_i**

$K_i^{-1}(tekstZaszyfrowany))$ **deszyforwanie kluczem K_i**

Tryby pracy algorytmu DES

- **Electronic Codebook (ECB)**. W trybie ECB szyfrowany jest kolejno każdy 64-bitowy blok danych za pomocą tego samego 56-bitowego klucza.

- **Cipher Block Chaining (CBC)** . W trybie CBC, na 64-bitowy bloku danych jest wykonywana bitowa operacja XOR z poprzednim, zaszyfrowanym już blokiem danych, tak otrzymany blok jest szyfrowany za pomocą klucza DES.

  RFC 2451, R. Pereira, R. Adams, The ESP CBC-Mode Cipher Algorithms,1998.

  ESP – oznacza Encapsulating Security Payload.

- **Cipher Feedback (CFB)**, szyfrowanie strumieniowe. Klucz DES jest używany do wygenerowania pseudolosowego ciągu danych, który następnie pełni rolę strumienia szyfrującego mieszanego z danymi za pomocą funkcji XOR.

- **Output Feedback (OFB),** szyfrowanie strumieniowe. Tryb podobny do CFB, w którym pseudolosowy ciągu danych wykorzystywany jest do szyfrowania i deszyfrowania.

Tryby pracy algorytmu DES

Przykład: Cisco Encryption Technology (CET).

CET jest stosowany w systemach operacyjnych firmy Cisco (Cisco IOS) używa algorytmu DES w trybie strumieniowym, aby wielkości zaszyfrowanych pakietów nie zmieniała się, tzn. nie zmieniała się wielkość MTU dla danych podsieci.

W systemach Cisco IOS, protokół IPsec wykorzystuje algorytm DES w trybie CBC (Cipher Block Chaining).

Stosowane motody EDE (encrypt-decrypt-encrypt):

- K1 = K3 oznacza, że klucz ma 112-bitów (2-key 3DES)
- K1 ≠ K3 oznacza, że klucz ma 168-bitów (3-key 3DES)

Moc algorytmu DES dla 2-kluczowego 3DES, i 3-2-kluczowego 3DES:

- DES: 2^56 kombinacji kluczy = 7.2 x 10^16 kombinacji kluczy,
- 2- kluczowy 3DES: 2^112 kombinacji kluczy = 5.2 x 10^33 kombinacji kluczy,
- 3- kluczowy 3DES: 2^168 kombinacji kluczy = 3.7 x 10^50 kombinacji kluczy.

Advanced Encryption Standard

Algorytm Advanced Encryption Standard (AES) zastąpił algorytm DES.

http://www.nist.gov/aes

Specyfikacja NIST standardu AES: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Jako algorytm AES został wybrany algorytm Rijndaela (Rain Doll) stworzony przez Joan Daemen ‘a i Vincent’a Rijmen’a (Uniwersytet Leuven, Belgia).

Cechy algorytmu AES:

- zmienna długość bloku i klucza szyfrującego,
- klucze mogą mieć 128, 192, lub 256-bitów,
- można szyfrować 128, 192, lub 256 bitowe bloki danych i ich kombinacje.

Tryby pracy algorytmu AES (RFC 3602):

- CBC (Cipher Block Chaining), RFC 3602.
- ECB (Electronic CodeBook mode),
- CFB (Cipher FeedBack mode),
- OFB (Output FeedBack mode)
- CTR (Counter mode), RFC 3686, RFC 5288, RFC 5930.

Tryby uwierzytelnienia wiadomości (Message Authentication Code):

- CBC-MAC (Cipher Block Chaining Message Authentication Code), RFC 3565,
- HMAC (Hashed Message Authentication Code), RFC 3537.

Algorytm MAC jest funkcją haszująca stosowaną w celu zapewnienia uwierzytelnienia i integralności wiadomości.

Odbiorca wiadomości, posiadając funkcję haszującą z tym samym kluczem co nadawca może sprawdzić kto nadał wiadomość i czy wiadomość była modyfikowana.

Algorytm RSA

Algorytm RSA, Rivest, Shamir, Adleman.

RSA jest przykładem asymetrycznego algorytmu szyfrującego, tzn. do szyfrowania i deszyfrowania wykorzystywane są różne klucze.

Jeden z kluczy jest jawny, zwany jest kluczem publicznym, drugi klucz jest niejawny zwany kluczem prywatnym.

Dane zaszyfrowane jednym kluczem mogą być rozszyfrowane tylko za pomocą drugiego klucza.
Algorytm RSA może korzystać z kluczy dowolnej długości.

Standardowo wykorzystywane są klucze o długości **1024**, **2048 bitów**.

Przykład:

Bezpieczeństwo **56 bitowego** klucza DES jest równoważne zabezpieczanie **512 bitowym** kluczem RSA.

Schemat generowania kluczy RSA. Przykład

Dwie losowo wybrane liczby pierwsze p, q:

- należy pomnożyć n = p*q
- należy obliczyć g(n) = (p - 1) * (q - 1)
- wybrać losowo liczbę **e** między 1 i g(n), względnie pierwszą z g(n)
- znaleźć liczbę **d** odwrotną do **e** mod g(n), tzn. d * e mod g(n) = 1.

Para kluczy RSA to: (n, e), (n, d).

Przykład:

p = 37 q = 23

n= p*q= 37*23 = 851

g(n) = (37 - 1)*(23 - 1) = 36 * 23 = 792

e i 792 muszą być względnie pierwsze.

Klucz publiczny: e, może być równy np 5.

Klucz prywatny: $d = 5^{-1} \bmod 792 = 317$ (5*317 mod 792=1).

Para kluczy RSA to: (n, e) = (851, 5), (n, d) = (851, 317)

Szyfrowanie RSA. Przykład

Szyfrowanie i deszyfrowanie wiadomości znaku 'A'.

Znakowi 'A' w kodzie ASCI odpowiada liczba 65. Asci(A)=65.

Szyfrowanie:

$$szyfr(Asci(znak)) = Asci(znak)^{e} \bmod n = szyfr(65) = 65^{5} \bmod 851 = 632$$

Deszyfrowanie:

$$dszyfr(szyfr) = szyfr^{d} \bmod n = dszyfr(632) = 632^{317} \bmod 851 = 65$$

Algorytm Diffie-Hellmana

Algorytm Diffie-Hellmana służy do uzgadniania kluczy przeznaczonych do bezpiecznej komunikacji bez potrzeby istnienia wcześniej uzgodnionych kluczy i haseł.

W protokole jawne są:

- liczba p, duża liczba pierwsza (1024 bitów),
- liczba a, generator grupy Zp (Zp - grupa reszt modulo p).

Zasada działania algorytmu Diffie-Hellmana:

1. Użytkownik A wybiera liczbę La , Użytkownik B wybiera liczbę Lb.
   La, Lb < p-1. Liczby La i Lb są utrzymywane w tajemnicy.
2. Użytkownik A przesyła do Użytkownika B liczbę $Lac = a^{\boldsymbol{La}} \ mod\ p$.
3. Użytkownik B przesyła w odpowiedzi Użytkownikowi A liczbę
   $Lbc = a^{\boldsymbol{Lb}} \ mod\ p$.
4. Po wymianie liczb Lac i Lbc następuje obliczenie klucza.
   Jako uzgodniony klucz przyjmowana jest liczba $k = a^{\boldsymbol{Lb}} * La \ mod\ p$.

Użytkownik A oblicza klucz ze wzoru $k = (Lbc)^{\boldsymbol{La}} \ mod\ p$.
Użytkownik B oblicza klucz ze wzoru $k = (Lac)^{\boldsymbol{Lb}} \ mod\ p$.

Certyfikaty

Częścią protokołu SSL jest **mechanizm obsługi certyfikatów**.

Certyfikat

jest zbiorem danych służących do identyfikacji użytkownika sieci.

Do wydawania certyfikatów służą organizacje **Certificate Authorities** (CA).

Serwery CA służą również do przechowywania kluczy publicznych użytkowników sieci.

Największymi wydawcami certyfikatów są:

- Verisign, http://www.verisign.com/
- Thawte, http://www.thawte.com/, .

Certyfikaty mogą być przesyłane w formacie:

- binarnym DER, (ang.) Distinguished Encoding Rules
- ASCII przy użyciu kodowania base64-PEM (pliki *.pem).

Skrót: pem - Privacy Enhanced Mail

Struktura certyfikatu

Nazwa pola	Opis
Subject	Dane właściciela certyfikatu.
Public Key	Klucz publiczny właściciela certyfikatu.
Issuer	Dane wystawcy certyfikatu.
Signature	Podpis elektroniczny certyfikatu wykonany przez wystawcę certyfikatu.
Validity	Termin ważności certyfikatu.
Administrative Information	Dane administracyjne: wersja certyfikatu, numer seryjny, algorytmy użyte przy tworzeniu certyfikatu.
Extended Information	Informacje dodatkowe.

Dane właściciela certyfikatu

Struktura pola 'Subject':

Nazwa pola	Oznaczenie
Common Name	CN
Organization	O
Organizational Unit	OU
Locality	L
State or Providence	ST
Country	C
EmailAddress	Email

Struktura certyfikatu. Zakodowany ASN.1

-----BEGIN CERTIFICATE-----
MIICWDCCAgICAQAwDQYJKoZIhvcNAQEEBQAwgbYxCzAJBgNVBAYTAlpBMRUwEwYD
VQQIEwxXZXN0ZXJuIENhcGUxEjAQBgNVBAcTCUNhcGUgVG93bjEdMBsGA1UEChMU
VGhhd3RlIENvbnN1bHRpbmcgY2MxHzAdBgNVBAsTFkNlcnRpZmljYXRpb24gU2Vy
dmljZXMxFzAVBgNVBAMTDnd3dy50aGF3dGUuY29tMSMwIQYJKoZIhvcNAQkBFhR3
ZWJtYXN0ZXJAdGhhd3RlLmNvbTAeFw05NjExMTQxNzE1MjVaFw05NjEyMTQxNzE1
MjVaMIG2MQswCQYDVQQGEwJaQTEVMBMGA1UECBMMV2VzdGVybiBDYXBlMRIwEAYD
VQQHEwlDYXBlIFRvd24xHTAbBgNVBAoTFFRoYXd0ZSBDb25zdWx0aW5nIGNjMR8w
HQYDVQQLExZDZXJ0aWZpY2F0aW9uIFNlcnZpY2VzMRcwFQYDVQQDEw53d3cudGhh
d3RlLmNvbTEjMCEGCSqGSIb3DQEJARYUd2VibWFzdGVyQHRoYXd0ZS5jb20wXDAN
BgkqhkiG9w0BAQEFAANLADBIAkEAmpIl7aR3aSPUUwUrHzpVMrsm3gpI2PzIwMh3
9l1h/RszI0/0qC2WRMlfwm5FapohoyjTJ6ZyGUUenICllKyKZwIDAQABMA0GCSqG
SIb3DQEBBAUAA0EAfI57WLkOKEyQqyCDYZ6reCukVDmAe7nZSbOyKv6KUvTCiQ5c
e5L4y3c/ViKdlou5BcQYAbxA7rwO/vz4m51w4w==
-----END CERTIFICATE-----

Struktura certyfikatu

Certificate:
Data:
  Version: 0 (0x0)
  Serial Number: 0 (0x0)
  Signature Algorithm: md5withRSAEncryption
  Issuer: C=ZA, SP=Western Cape, L=Cape Town,
      O=Thawte Consulting cc,
  OU=Certification Services, CN=www.thawte.com,
  Email=webmaster@thawte.com
Validity
   Not Before: Nov 14 17:15:25 1996 GMT
   Not After : Dec 14 17:15:25 1996 GMT
   Subject: C=ZA, SP=Western Cape, L=Cape Town,
      O=Thawte Consulting cc,
   OU=Certification Services, CN=www.thawte.com,
  Email=webmaster@thawte.com

Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
    Modulus:
      00:9a:92:25:ed:a4:77:69:23:d4:53:05:2b:1f:3a:
      55:32:bb:26:de:0a:48:d8:fc:c8:c0:c8:77:f6:5d:
       61:fd:1b:33:23:4f:f4:a8:2d:96:44:c9:5f:c2:6e:
       45:6a:9a:21:a3:28:d3:27:a6:72:19:45:1e:9c:80:
       a5:94:ac:8a:67
       Exponent: 65537 (0x10001)
 Signature Algorithm: md5withRSAEncryption
   7c:8e:7b:58:b9:0e:28:4c:90:ab:20:83:61:9e:ab:78:2b:a4:
   54:39:80:7b:b9:d9:49:b3:b2:2a:fe:8a:52:f4:c2:89:0e:5c:
   7b:92:f8:cb:77:3f:56:22:9d:96:8b:b9:05:c4:18:01:bc:40:
   ee:bc:0e:fe:fc:f8:9b:9d:70:e3

Przykład klucza prywatnego

```
Private-Key: (512 bit)
modulus:
    00:9a:92:25:ed:a4:77:69:23:d4:53:05:2b:1f:3a:
    55:32:bb:26:de:0a:48:d8:fc:c8:c0:c8:77:f6:5d:
    61:fd:1b:33:23:4f:f4:a8:2d:96:44:c9:5f:c2:6e:
    45:6a:9a:21:a3:28:d3:27:a6:72:19:45:1e:9c:80:
    a5:94:ac:8a:67
publicExponent: 65537 (0x10001)
privateExponent:
    00:a9:0f:30:6c:bb:75:df:89:50:b1:7c:f5:ad:32:
    1f:fd:5c:b5:26:26:19:87:3a:f4:57:e6:eb:4e:8a:
    d4:a1:ff:6a:99:7b:e3:a0:d0:af:ed:83:4c:db:c3:
    75:2c:d9:8a:13:f6:cb:48:f2:7e:b0:f1:9e:ed:3b:
    09:73:fe:01
prime1:
    00:cc:eb:61:18:66:69:27:ee:38:58:d1:82:4e:bf:
    bd:da:5f:7c:9f:de:49:0e:87:b2:58:dc:8c:c5:ea:
    a1:2c:47
prime2:
    00:c1:19:da:4c:9c:8d:49:16:62:30:a8:a2:88:18:
    4b:2a:fc:cf:f9:75:7c:ad:0a:c5:af:ec:e3:73:27:
    46:60:e1
exponent1:
    00:a1:d0:50:a8:ba:dd:d8:a9:35:17:75:c1:47:3c:
    03:c8:37:d4:aa:4d:16:35:82:13:e4:35:ac:77:f0:
    d1:fa:ab
exponent2:
    36:11:35:0c:6a:71:2c:db:b5:96:86:41:2b:f6:11:
    65:f1:ef:91:9b:91:d3:29:c6:fc:61:49:b6:3e:72:
    f8:41
coefficient:
    05:56:f9:d7:54:93:1f:a4:10:a0:fb:51:c8:82:7d:
    f3:a5:e6:d6:58:8f:d8:31:3d:b2:50:02:f9:07:2e:
    55:af
```

System Kerberos

System (usługa, protokół) Kerberos wersja 5, (ang.) **Kerberos Network Authentication Service (V5)**.

System Kerberos został stworzony w MIT.
Lista dyskusyjna: kerberos@MIT.EDU

RFC 4120, The Kerberos Network Authentication Service (V5) C. Neuman, T.Yu, S. Hartman, K. Raeburn.

Kerberos służy do uwierzytelnienia klientów i serwerów usług w sieciach otwartych, tzn. sieciach niezabezpieczonych.

Do uwierzytelnienia klientów i serwerów służą serwery identyfikacyjne, serwery AS,
(ang.) **Authentication Servers**.

Uwierzytelnienie, autoryzacja

Serwer AS przesyła na żądanie klienta **uwierzytelnienie**, (ang.) **credentials**.

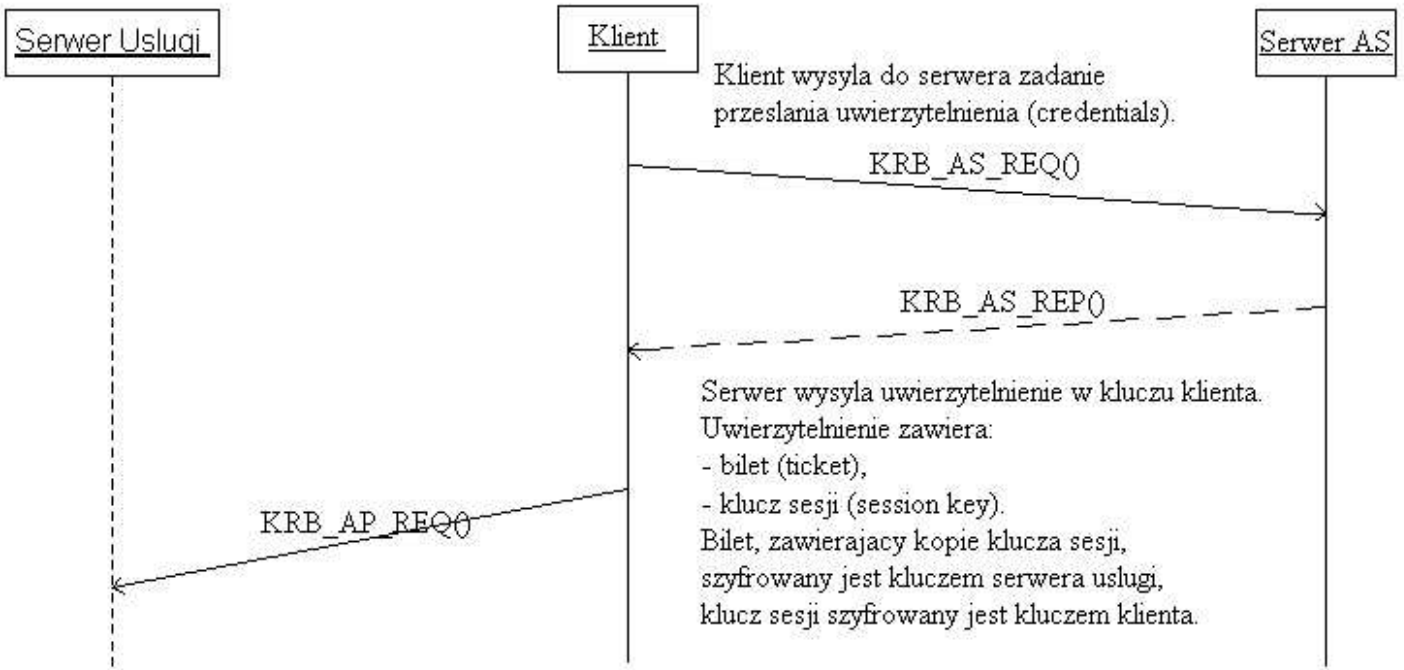
Uwierzytelnienie składa się z:

- **biletu** na dany serwer usługi, (ang.) server ticket
- **klucza sesji**, (ang.) session key.

Uwierzytelnienie, (ang.) **authentication**, oznacza identyfikacje tożsamości użytkownika sieci.

Autoryzacja, (ang.) **authorization,** oznacza proces służący określeniu czy użytkownik (już zidentyfikowany) ma prawo do korzystania z danej usługi lub w jakim zakresie może z niej korzystać.

Schemat uwierzytelnienia klienta (RFC4120)



Klient przesyla bilet na serwer uslugi.
Bilet zawiera dane identyfikujace klienta i kopie klucza sesji.
Bilet zaszyfrowany jest przez serwer AS kluczem serwera uslugi.
Wymieniony klucz sesji sluza do identyfikacji klienta i serwera.
Klucz moze równiez sluzyc do szyfrowania transmisji miedzy klientem a serwerem.

Schemat uwierzytelnienia klienta (RFC4120)

(1) **Klient** chcąc zbudować bezpieczne połączenie z serwerem usługi wysyła do **serwera AS** żądanie przesłania uwierzytelnienia (credentials).

(2) Serwer AS odpowiada wysyłając uwierzytelnienie w kluczu klienta.

Uwierzytelnienie zawiera:

- bilet, (ticket)
- klucz sesji (session key).

Bilet (zawierający kopie klucza sesji) szyfrowany jest kluczem serwera usługi klucz sesji szyfrowany jest kluczem klienta.

(3) Klient przesyła bilet na serwer usługi.

Bilet zawiera dane identyfikujące klienta i kopie klucza sesji.

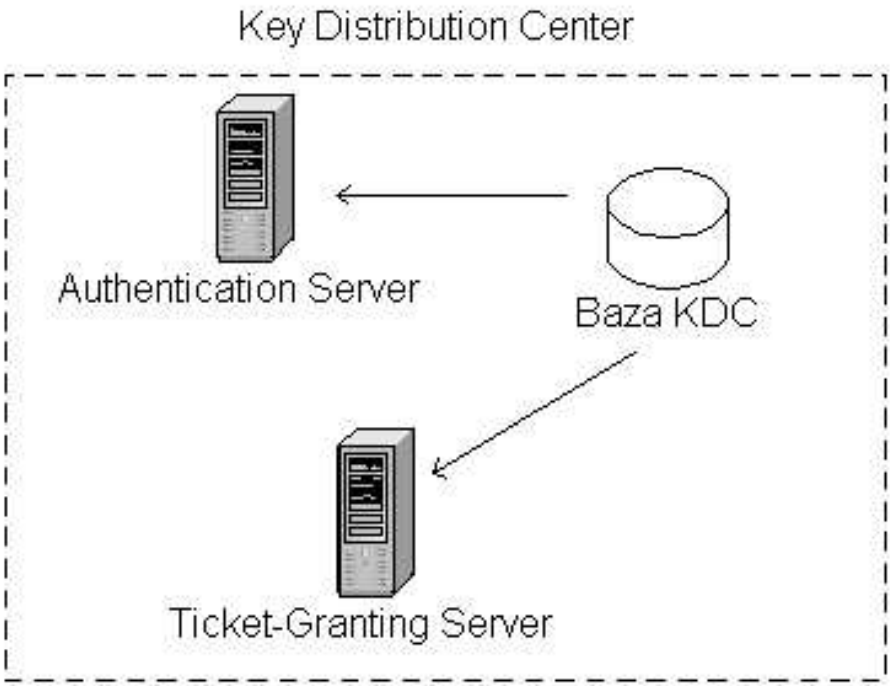
Bilet zaszyfrowany jest przez serwer AS **kluczem serwera usługi**.

Wymieniony klucz sesji służy do identyfikacji klienta i serwera.

Klucz może również służyć do szyfrowania transmisji miedzy klientem a serwerem.

Key Distribution Center

Serwer AS (authentication server) jest częścią **usługi KDC**, (ang.) **Key Distribution Center**.



Usługa udzielania biletów usługi (TGS, Ticket-Granting Service)

Klient wysyłając na **serwer AS** żądanie uwierzytelnienia w odpowiedzi może otrzymać:

- bilet na serwer i klucz sesji,
- **bilet TGT** (T**icket-Granting Ticket)** i **klucz sesji**.

Bilet TGT służy klientowi do uwierzytelnienia na **serwerze TGS** (Ticket-Granting Server)

rządania przez klienta od **serwer TGS** uwierzytelniena na serwer usługi.

TGT służy do komunikacji klienta z serwerem TGS.

Klient żądając uwierzytelnienia na serwer usługi wysyła do **serwera TGS bilet TGT**.

W odpowiedzi **serwer TGS** wysyła **bilet na serwer** i **klucz sesji**.

Bilet TGT jest zaszyfrowany kluczem **serwera AS,** a klucz sesji kluczem **klienta**.

Bilet TGS szyfrowany jest kluczem serwera usługi, klucz sesji szyfrowany jest kluczem **sesji TGT**.

Usługa udzielania biletów usługi (TGS, Ticket-Granting Service)

Klient komunikuje się z serwerem TGS celu uzyskania:

- uwierzytelnienia na serwer usługi (klient musi już mieć przydzielony przez AS bilet TGT),
- przedłużenia lub weryfikacji posiadanego biletu,
- biletu proxy.

Przy wymianie wiadomości między klientem a serwerem TGS **nie jest używany klucz klienta**.

Używany jest klucz biletu TGT, odnawialny bilet lub podklucz sesji.

Przykłady wiadomości Kerberosa

Wiadomości podprotokołu AS:

Klient do serwera Kerberosa: KRB_AS_REQ.

Serwer Kerberosa do klienta: KRB_AS_REP, KRB_ERROR.

Wiadomości podprotokołu TGS:

Klient do serweraKerberosa: KRB_TGS_REQ.

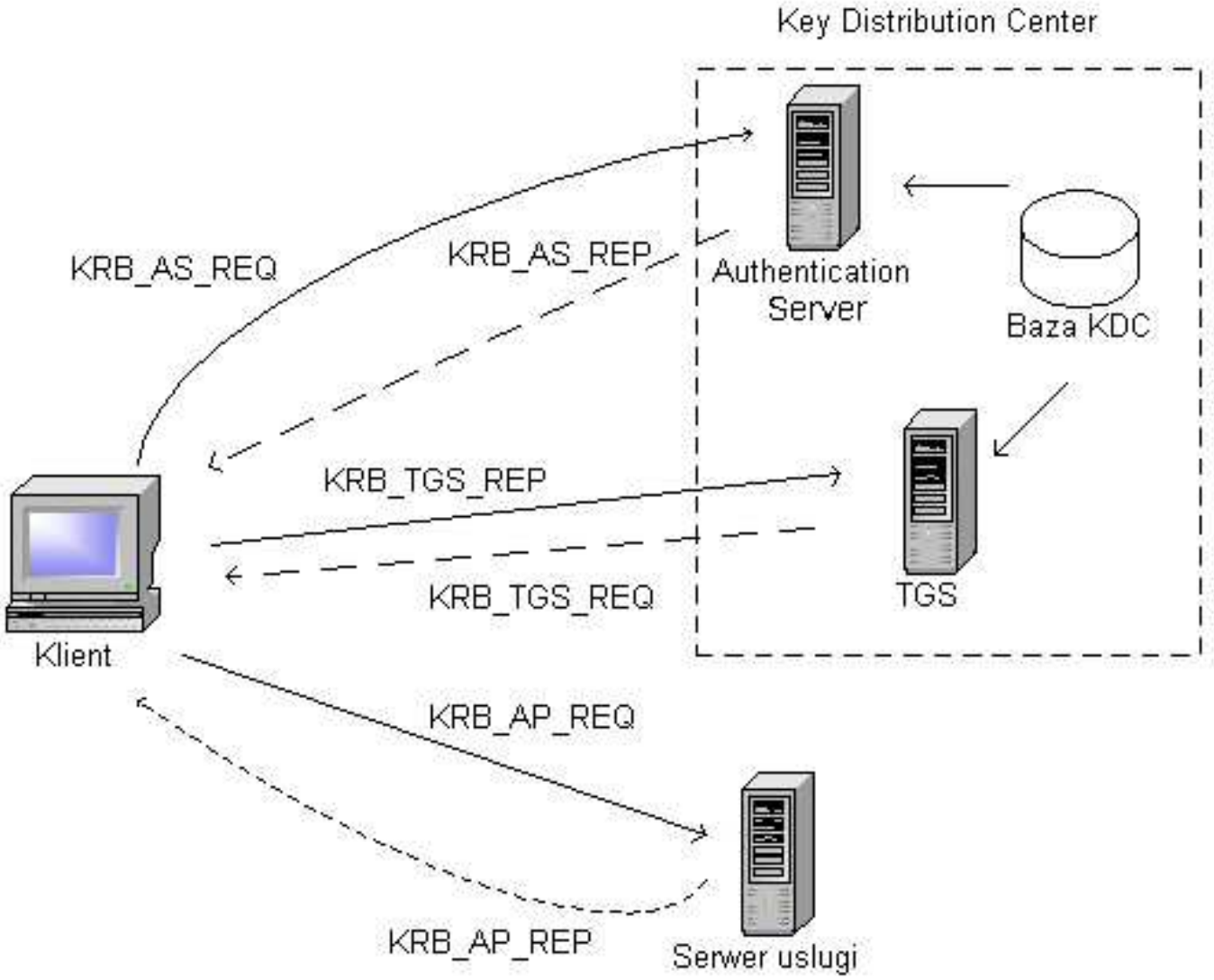
Serwer Kerberosa do klienta: KRB_TGS_REP, KRB_ERROR.

Wiadomości podprotokołu CS:

Klient do serwer usługi: KRB_AP_REQ.

Serwer usługi do klient (opcjonalnie): KRB_AP_REP, KRB_ERROR.

Wymiana wiadomości w systemie Kerberos



Pole	Struktura komunikatów KRB_AS_REQ i KRB_TGS_REQ
pvno	Numer wersji protokołu Kerberos (5).
msg-type	Identyfikuje komunikat KRB_AS_REQ lub KRB_TGS_REQ
padata-type	Identyfikuje mechanizm wstępnego potwierdzania tożsamości (PA_AS_REQ)
padata-value	Zaszyfrowany znacznik czasu.
ap-options	Opcje żądanego biletu: 1 — może zostać przesłany pod podane adresy nadawcy, 2 — jest przesyłany dalej, 3 — może zostać przesłany ze wskazanych adresów, 4 — jest przesyłany w cudzym imieniu, 5 — może być postdatowany, 6 — jest postdatowany, 8 — może być odnawiany, 14 — jest wysłany przez użytkownika anonimowego, 26 — wyłączenie śledzenia domen przez który bilet jest przesyłany, 27 — zezwolenie na tworzenie na jego podatnie kolejnych biletów, 28 — zaszyfrowany kluczem sesji, 30 — wymagający odnowienia, 31 — używany do sprawdzenia poprawności postdatowanych biletów.
crealm	Nazwa domeny klienta
cname	Nazwa klienta
sname	Nazwa KDC
from	Czas od którego żądany bilet ma być ważny (postdatowanie biletu)
till	Czas wygaśnięcia ważności biletu
Rtime	Czas do kiedy możliwe będzie odnowienie ważności biletu
Nonce	Pseudolosowa liczba
Etype	Użyty algorytm szyfrowania.
Addresses	Adresy nadawcy. Bilet wysłany z tych adresów będzie poprawny.
enc-authorization-data	Zaszyfrowane dane aplikacji. Pole używane tylko w komunikatach KRB_TGS_REQ.
additional-tickets	W komunikatach KRB_TGS_REQ możne przesyłać więcej niż jeden bilet.

Pole	Struktura komunikatów KRB_AS_REP i KRB_TGS_REP
pvno	Numer wersji protokołu Kerberos (5).
msg-type	dentyfikuje komunikat (KRB_AS_REP lub KRB_TGS_REP)
padata	Odebrane dane wstępnego uwierzytelnienia
crealm	Nazwa domeny klienta
cname	Nazwa klienta
ticket	Bilet
key	Klucz sesji
last-req	Czas ostatniego żądanie biletu
nonce	Pseudolosowa liczba
key-expiration	Czas wygaśnięcia ważności klucza
flags	Opcje biletu
authtime	Czas wystawienia biletu
starttime	Czas od którego bilet jest ważny
endtime	Czas wygaśnięcia ważności biletu
renew-till	Czas przez który bilet będzie mógł być odnawiany
srealm	Domena serwera
sname	Nazwa serwera
caddr	Poprawne adresy nadawcy

Pole	Struktura komunikatu KRB_AP_REQ
pvno	Numer wersji protokołu Kerberos (5).
msg-type	dentyfikuje komunikat (KRB_AP_REQ)
ap-options	Używane są następujące bity: 1 — bilet został zaszyfrowany kluczem sesji a nie kluczem długoterminowym serwera, 2 — wymagane jest wzajemne uwierzytelnienie.
ticket	Bilet usługi
authenticator	Dane uwierzytelniające użytkownika

Pole	Struktura komunikatu KRB_AP_REP
pvno	Numer wersji protokołu Kerberos (5).
msg-type	dentyfikuje komunikat (KRB_AP_REP)
ctime	Czas zegara systemowego komputer klienckiego odczytany z danych uwierzytelniających
cusec	Liczba milisekund zegara systemowego komputer klienckiego odczytana z danych uwierzytelniających
subkey	Klucz sesji

Identyfikator

W celu **uwierzytelnienia się**, klient wysyła na serwer bilet.

Aby uniknąć przechwycenia i użycia biletu przez innego użytkownika sieci (tylko cześć biletu jest przesyłana w formie szyfrowanej), klient wysyła drugą wiadomość szyfrowaną kluczem sesji zawierająca **znacznik czasu**, (ang.) timestamp.

Wiadomość ze znacznikiem czasu nazywa się **identyfikatorem**, (ang.) **authenticator**.

Identyfikator służy do potwierdzenia, że bilet został wysłany przez klienta dla którego został wydany.

Podprotokoły Kerberosa (exchanges)

Podprotokoły systemu Kerberos:

- podprotokół AS, (ang.) **Authentication Service Exchange**.

  Służy do realizacji funkcji serwera identyfikacyjnego (serwera AS), tzn. wydawania biletów do przydzielania biletów, (ticket-granting ticket) i kluczy sesji TGT.

- podprotokół TGS, (ang.) **Ticket-Granting Service Exchange**.

  Służy do realizacji funkcji serwera TGS, (ticket-granting server), tzn. wydawania biletów i kluczy sesji na żądanie klienta.

- podprotokół CS, (ang.) **Client/Server Authentication Exchange**.

  Służy do komunikacji między klientem a serwerem usługi w celu identyfikacji stron.

Struktura bazy KDC

Pole	Opis
name	nazwa użytkownika (principal's identifier)
key	klucz tajny użytkownika (principal's secret key)
p_kvno	numer wersji klucza (principal's key version)
max_life	termin ważności biletu (max. lifetime for tickets)
max_renewable_life	max. czas ważności biletu (max. total lifetime for renewable tickets)

Struktura biletu

```
Ticket ::=  [APPLICATION 1] SEQUENCE
{
        tkt-vno[0]        INTEGER,          // wersja biletu, np.5
        realm[1]           Realm,           // obszar dzialania serwera AS (domena AS)
        sname[2]           PrincipalName,   // nazwa serwera
        enc-part[3]        EncryptedData    // kod dla zaszyfrowanej czesci biletu
}
    -- Encrypted part of ticket
EncTicketPart ::=    [APPLICATION 3] SEQUENCE {
        flags[0]          TicketFlags,      // flagi biletu, szczegoly w RFC 1510
        key[1]            EncryptionKey,    // klucz sesji
        crealm[2]          Realm,           // nazwa obszaru w którym zarejestowany jest klient
        cname[3]           PrincipalName,   // nazwa klienta
        transited[4]      TransitedEncoding,  // lista nazw 'Kerberos realms'
        authtime[5]        KerberosTime,    // czas identyfikacji
        starttime[6]       KerberosTime OPTIONAL,  // czas od ktorego bilet jest wazny
        endtime[7]         KerberosTime,    // czas do ktorego bilet jest wazny
        renew-till[8]      KerberosTime OPTIONAL,  // max. czas na jaki mozna odnowic bilet
        caddr[9]           HostAddresses OPTIONAL,  // adres klienta
        authorization-data[10]  AuthorizationData OPTIONAL  }  // dane identyfikujace klienta
    -- encoded Transited field
TransitedEncoding ::=       SEQUENCE
{
        tr-type[0]  INTEGER, -- must be registered
        contents[1]         OCTET STRING
}
```

Algorytmy szyfrowania w systemie Kerberos

System Kerberos używa następujące tryby szyfrowania (encryption modes):

- null, The NULL Encryption System (brak szyfrowania)
- des-cbc-crc, (ang.) DES in Cipher-Block-Chaining mode with CRC-32 checksum
- des-cbc-md4, (ang.) DES in CBC mode with an MD4 checksum
- des-cbc-md5, (ang.) DES in CBC mode with an MD5 checksum.

W trybie des-cbc-crc suma kontrolna wyliczana jest algorytmem CRC-32.

W trybie des-cbc-md5 suma kontrolna wyliczana jest algorytmem MD5 (RFC 1321).

Metody wyliczania sum kontrolnych

Stosowane metody wyliczania sum kontrolnych:

- crc32, (CRC-32 Checksum)
- rsa-md4, (RSA MD4 Checksum)
- rsa-md4des, (RSA MD4 Cryptographic Checksum Using DES)
- rsa-md5, (RSA MD5 Checksum)
- rsa-md5des
- des-mac
- rsa-md4-des-k
- desmac-k.

Delegacja uwierzytelnienia

Mechanizm **delegacji uwierzytelnienia** - proces uwierzytelnienia jest przechodni, tzn. jeżeli host A ma zaufanie do hostów B i C to host B ma zaufanie do hosta C.

Specyfikacje RFC dla Kerberosa

RFC 1964 , The Kerberos Version 5 GSS-API Mechanism, J. Linn , 1996.

RFC 2712, Addition of Kerberos Cipher Suites to Transport Layer Security (TLS).

RFC 2942, Telnet Authentication: Kerberos Version 5, T. Ts'o, 2000.

RFC 2743, Generic Security Service Application Program Interface Version 2, Update 1 J. Linn , 2000.

RFC 3244, Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols, M. Swift, J.Trostle, J.Brezak,2002.

RFC 4120, The Kerberos Network Authentication Service (V5) C. Neuman, T.Yu, S. Hartman, K. Raeburn

RFC 4121, The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2 L. Zhu, K. Jaganathan, S. Hartman, 2005.

RFC 4178, The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism L. Zhu, P. Leach, K. Jaganathan, W. Ingersoll, 2005.

RFC 4422, Simple Authentication and Security Layer (SASL) A. Melnikov, K. Zeilenga ,2006.

IPSec

Standard opracowany przez Internet Engineering Task Force (IETF) IPSec working group.

IPsec jest mechanizmem służącym do bezpiecznej transmisji w **warstwie Sieci** modelu OSI.

IPSec został stworzony w celu:

- kontroli dostępu do danych,
- identyfikacji stron biorących udział w komunikacji,
- ochrony integralności datagramów IP,
- szyfrowania transmisji (ochrony poufność transmisji).

RFC 2401, Security Architecture for the Internet Protocol, S. Kent, R. Atkinson, 1998.
RFC 2402, IP Authentication Header, S. Kent, R. Atkinson [1998].
RFC 2406, IP Encapsulating Security Payload (ESP), S. Kent, R. Atkinson, 1998.
RFC 3168, The Addition of Explicit Congestion Notification (ECN) to IP, K. Ramakrishnan, S. Floyd, D. Black.

Typy nagłówków IPSec

IPsec wykorzystuje trzy protokoły do bezpiecznej transmisji datagramów IP:

- Authentication Header (AH), RFC 4302, 4305.
- Encapsulating Security Protocol (ESP), RFC 4303, 4305.
- Internet Key Exchange (IKE), RFC 2409, RFC 4306.

Nagłówek AH w datagramie IP służy do:

- zabezpieczeniu integralności przesyłanego datagramu,
- identyfikacji nadawcy,
- częściowej kontroli kolejności przesyłania datagramów (anti-replay service korzysta z pola 'Sequence Number' w nagłówku ESP).

Nagłówek ESP służy do:

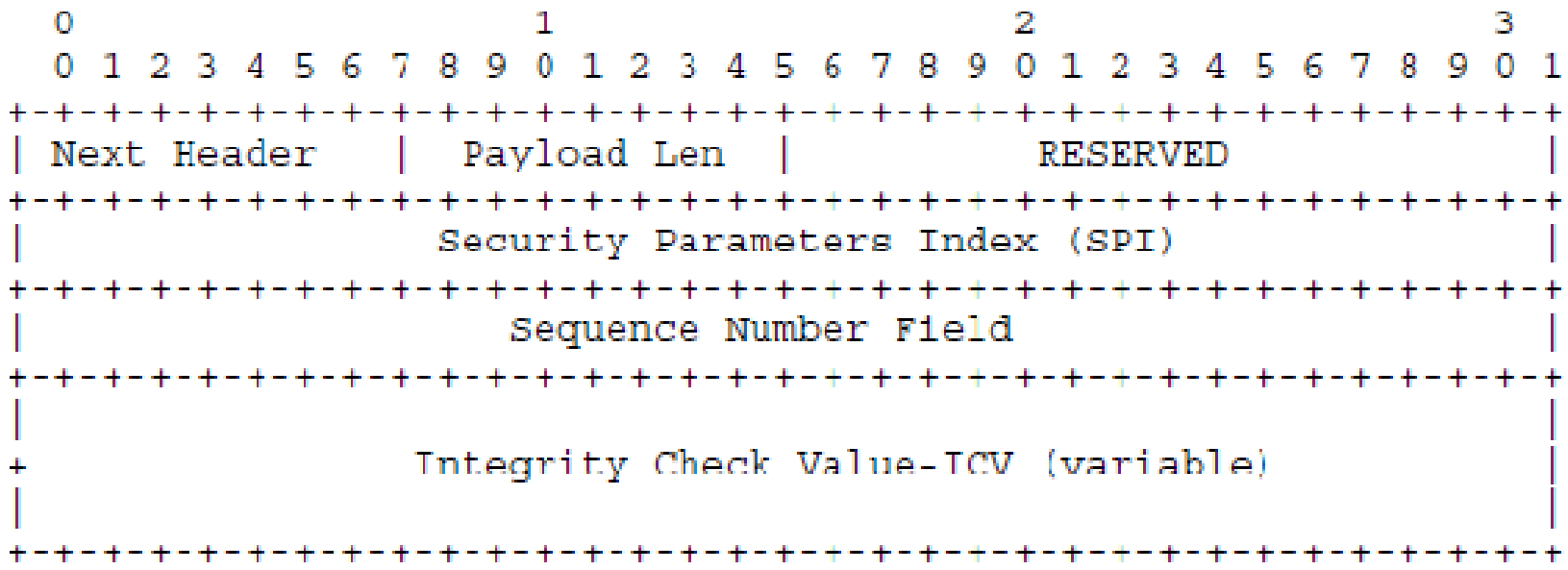
- szyfrowania pakietów IP (poufność transmisji),
- identyfikacji nadawcy,
- częściowej kontroli kolejności przesyłania datagramów.

Protokół Authentication Header

Protokół AH służy do ochrony integralności datagramów IP (nagłówka, danych).

Protokół AH wykorzystuje funkcje HMAC (keyed-Hash Message Authentication Code): MD5, SHA-1, RIPEMD-160 lub inne wynegocjowane.

Przy fragmentacji datagramu podpisywany jest każdy fragment oddzielnie.



Struktura nagłówka protokołu AH, źródło RFC 4302.

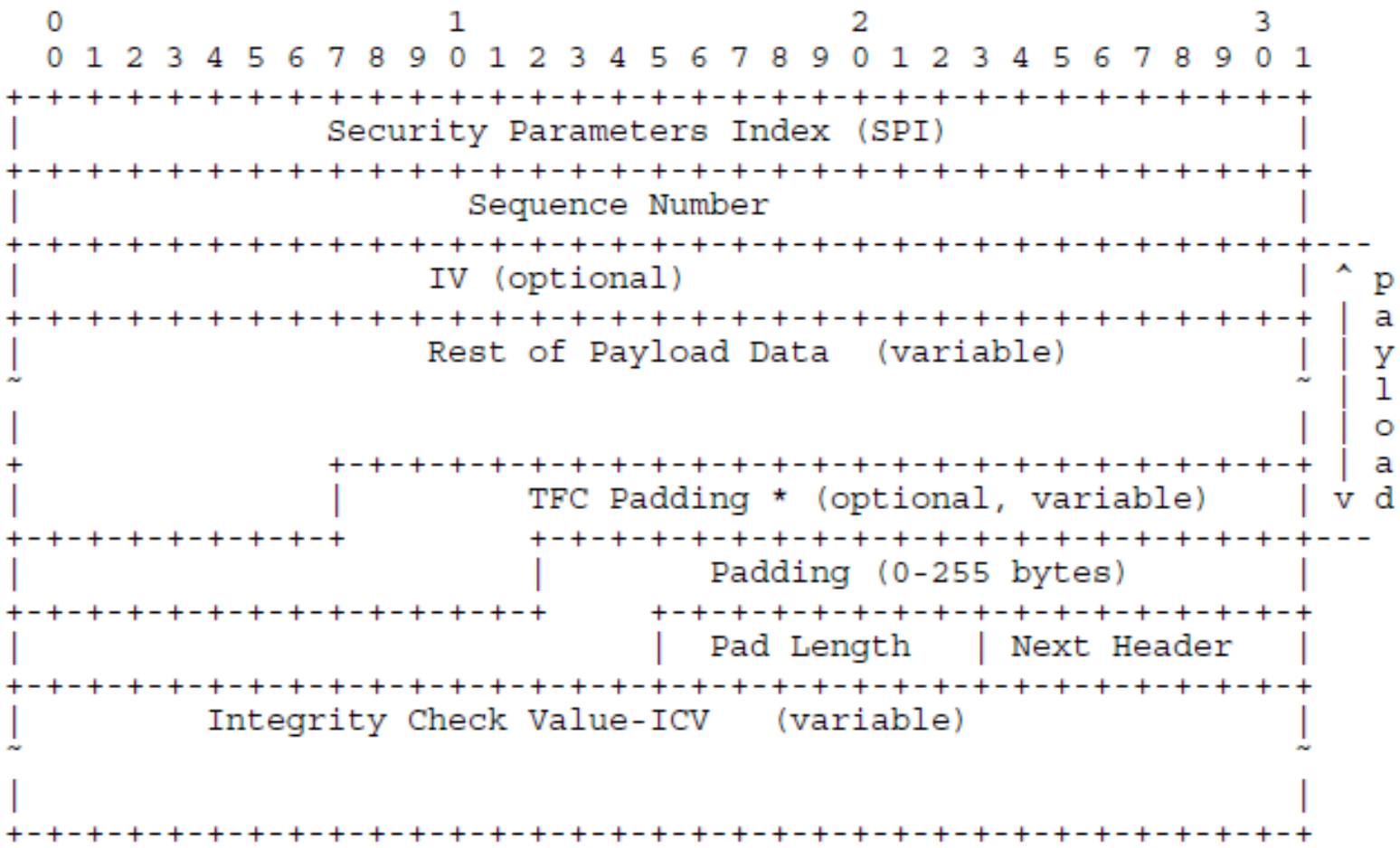
Protokół Encapsulating Security Payload

Protokół ESP (Encapsulating Security Payload) służy do zapewnienia poufności transmitowanych danych.

Protokół ESP do szyfrowanie datagramu wykorzystuje szyfry blokowe w trybie CBC,

- DES, 3DES,
- Blowfish,
- CAST-128
- Rijndael/AES,
- 3-IDEA.

Protokół Encapsulating Security Payload



Struktura nagłówka protokołu ESP, źródło RFC 4303.

Internet Key Exchange

IKE - protokół służy do negocjacji parametrów bezpiecznej transmisji, wymiany kluczy uwierzytelniających.

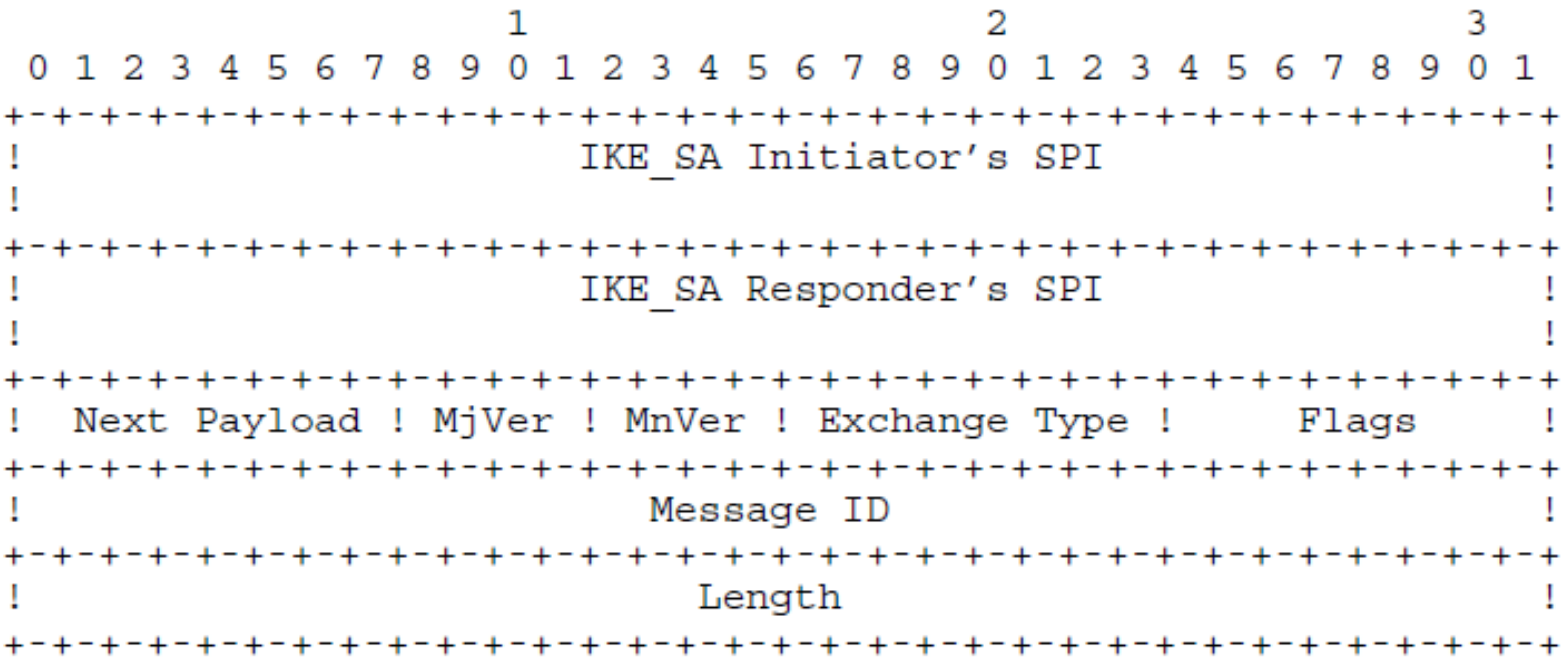
RFC 4306, C. Kaufman, Internet Key Exchange (IKEv2) Protocol, 2005.

IKE automatycznie negocjuje parametry bezpiecznej transmisji, ang. security association, tzn. nie jest wymagana wstępna konfiguracja IPsec.

IKE jest hybrydowym protokołem, który wykorzystuje w Internet Security Association and Key Management Protocol (ISAKMP):

- mechanizm wymiany kluczy Oakley (Oakley key exchange), kryptograficzny protokół wymiany kluczy za pomocą algorytmu Diffiego-Hellmana,
- mechanizm wymiany kluczy Skeme.

Struktura nagłówka IKE, źródło RFC 4306.



Struktura nagłówka IKE, źródło RFC 4306

Protokół ISAKMP

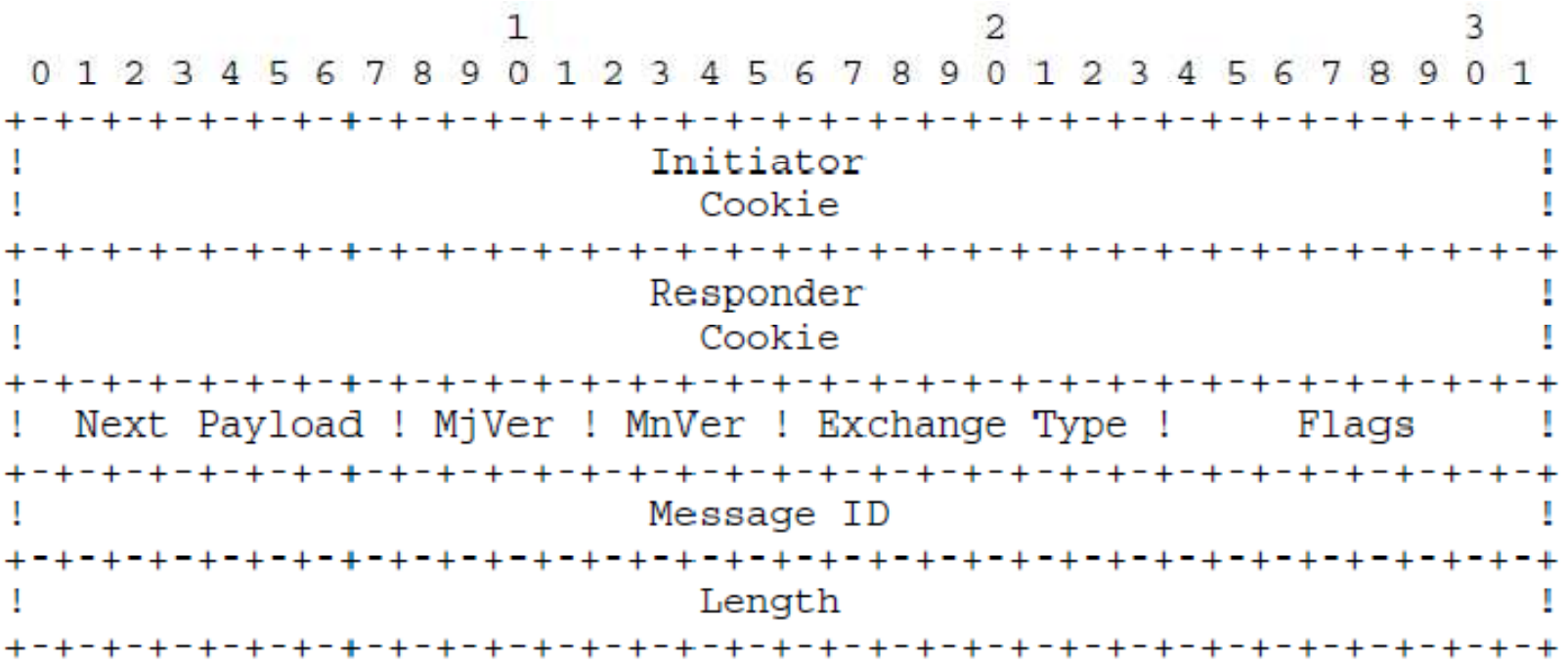
Protokół ISAKMP (Internet Security Association and Key Management Protocol) służy do uzgadniania kluczy w IPsec.

Protokół ISAKMP korzysta z portu 500 UDP.

RFC 2408, Maughan, D., Schertler, M., Schneider, M., and J. Turner, Internet Security Association and Key Management Protocol (ISAKMP), 1998.

RFC 4306, C. Kaufman, Internet Key Exchange (IKEv2) Protocol, 2005.

Struktura nagłówka ISAKMP



Struktura nagłówka ISAKMP

Security Association (SA)

Security Association (SA)

jest połączeniem w trybie simplex (połączeniem jednokierunkowym), które umożliwia realizacje usług bezpiecznej transmisji (security services).

Usługa bezpiecznej transmisji jest realizowana przez protokół AH lub ESP (nigdy jednocześnie).

Aby bezpieczna transmisja zachodziła w obu kierunkach należy utworzyć dwa połączenia SA.

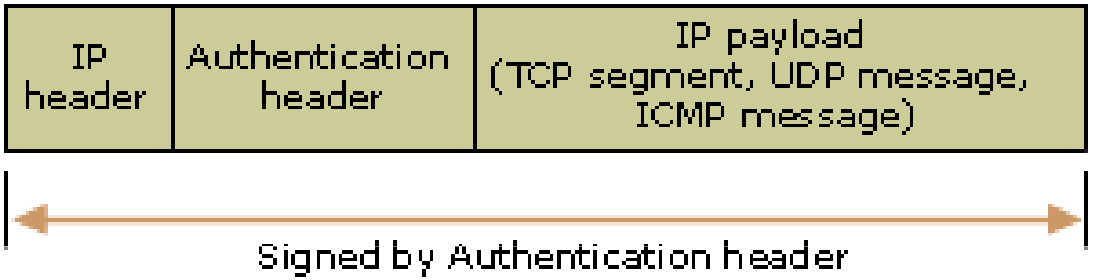
Połączenie SA jest określone gdy znane są parametry:

- parametr SPI (Security Parameter Index),
- adres IP odbiorcy (IP Destination Address),
- identyfikator nagłówka AH lub ESP (security protocol AH or ESP identifier).

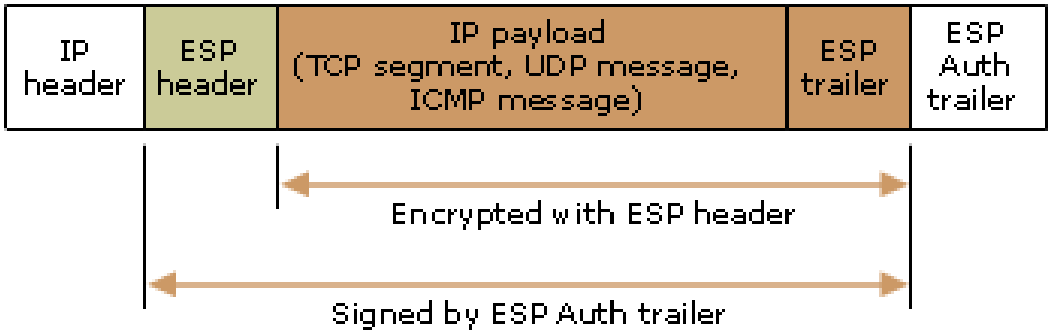
Tryb pracy IPsec

Połączenie SA może zachodzi w dwóch trybach:

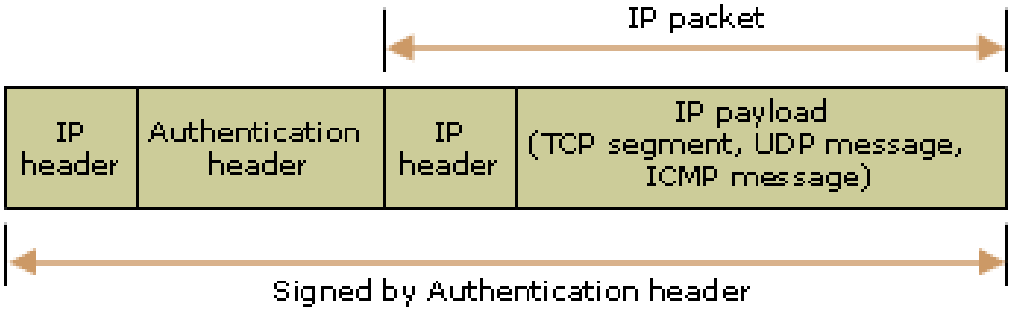
- tryb transportowy (transport mode). Tryb typowy dla transmisji host-host.
- w tunelu (tunnel mode). Tryb typowy dla transmisji gateway-gateway.



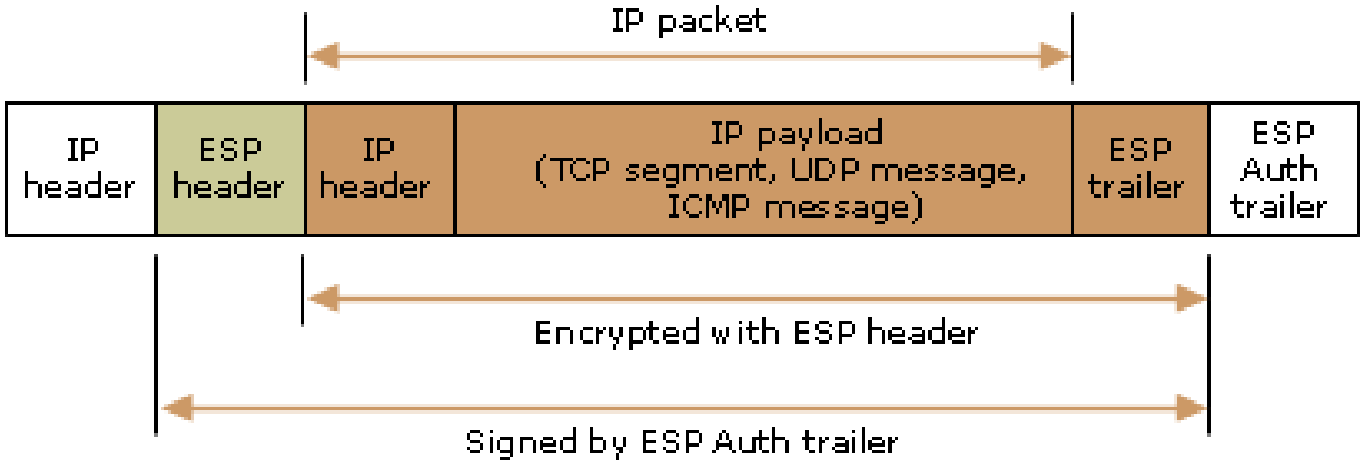
Struktura datagramu IP w trybie transportowym z nagłówkiem AH. Źródło MSDN.



Struktura datagramu IP w trybie transportowym z nagłówkiem ESP. Źródło MSDN.



Struktura datagramu IP w trybie tunelowym z nagłówkiem AH. Źródło MSDN.



Struktura datagramu IP w trybie tunelowym z nagłówkiem ESP. Źródło MSDN.

Architektura IPSec w systemie Windows

Implementacja IPSec w systemie Windows składa się z następujących komponentów:

- IPSec policy agent,
- ISAKMP Key Management Service,
- Oakley Key Determination,
- IPSec driver (IPSEC.SYS),
- IPSec model.

IPSec policy agent uruchamiany jest przy starcie systemu na podstawie danych zawartych w AD, konfiguruje parametry IPsec w rejestrach, **usłudze ISAKMP**.

ISAKMP, (ang.) Internet Security Association and Key Management Protocol, RFC 2408.

Protokół służący do uzgadniania kluczy w IPsec.

Przepływ danych IPSec w systemie Windows

1. Host A łączy się z hostem B.

   Na hoście A uruchamiana jest usługa ISAKMP/Oakley.

2. Usługa ISAKMP/Oakley tworzy klucz i SA, (Security Association).

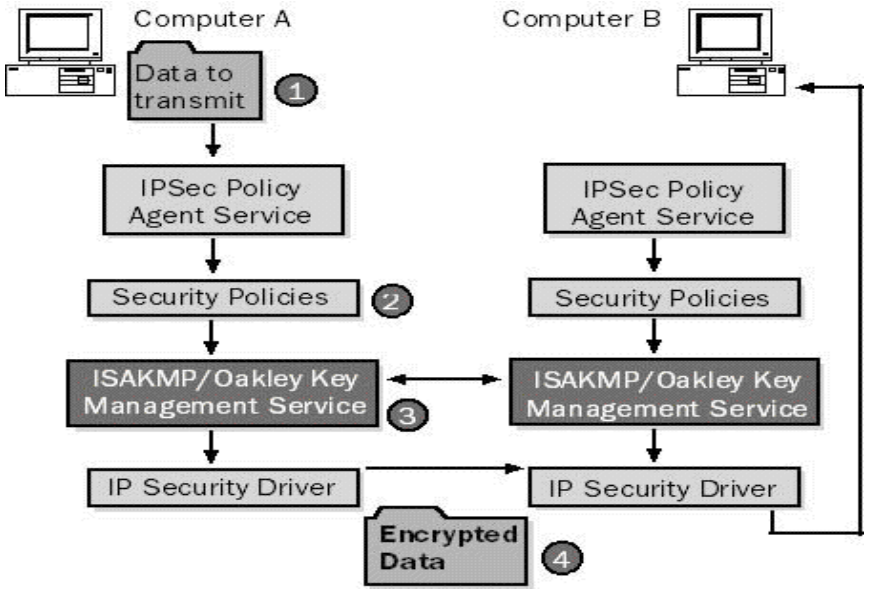
   Uzgadniane są algorytmy szyfrowania (np. 3DES), algorytmy służące do zabezpieczania integralności datagramow (MD5, SHA1), uzgadniana jest metoda uwierzytelnienia.

   Wymieniane są dane potrzebne do wygenerowania klucza.

3. Klucz i parametry SA zostają przekazane na hosty A i B.

4. Host A wykorzystuje klucz do szyfrowania danych wysyłanych na host B.

5. Host B wykorzystuje klucz do rozszyfrowania otrzymanych danych.



Źródło: MCSE Training Kit, Microsoft Windows 2000 Network Infrastructure Administration

Przykład: nagłówek ISAKMP

```
# ----- Naglowek Ethernetowy -----
 ETYPE = 0x0800 : Protocol = IP:  DOD Internet
      Protocol
 Destination address : 00600801D303
  .......0 = Individual address
  ......0. = Universally administered address
  Source address : 00104B6B1608
  .......0 = No routing information present
  ......0. = Universally administered address
  Frame Length : 506 (0x01FA)
  Ethernet Type : 0x0800 (IP:  DOD Internet Protocol)
 Ethernet Data: Number of data bytes remaining = 492
      (0x01EC)

 # ----- Naglowek IP -----
 ID = 0x345; Proto = UDP; Len: 492
 Version = 4 (0x4)
 Header Length = 20 (0x14)
 Precedence = Routine
 Type of Service = Normal Service
 Total Length = 492 (0x1EC)
 Identification = 837 (0x345)
 Flags Summary = 0 (0x0)
 Flags Summary = 0 (0x0)
       .......0 = Last fragment in datagram
       ......0. = May fragment datagram if necessary
  Fragment Offset = 0 (0x0) bytes
   Time to Live = 128 (0x80)
   Protocol = UDP - User Datagram
   Checksum = 0x1E67
   Source Address = 10.10.1.100
   Destination Address = 10.10.1.222
   Data: Number of data bytes remaining = 472 (0x01D8)

 # ----- Naglowek UDP -----
Src Port: ISAKMP, (500);
 Dst Port: ISAKMP (500); Length = 472 (0x1D8)
 Source Port = ISAKMP
  Destination Port = ISAKMP
  Total length = 472 (0x1D8) bytes
  UDP Checksum = 0xE4AD
  Data: Number of data bytes remaining = 464 (0x01D0)

   # ----- Naglowek ISAKMP -----
...
```

Przykład: nagłówek ISAKMP

```
  # ----- Naglowek ISAKMP -----
 Major Version: 1  Minor Version: 0  Length: 464
 Initiator cookie = A8 CF 87 E8 A6 72 5A 0F
 Responder cookie = 00 00 00 00 00 00 00 00
 Next payload = Security Association
 Major version = 1 (0x1)
 Minor version = 0 (0x0)
 Exchange type = Identity Protection
 Flags summary = 0 (0x0)
 .......0 = Payloads are not encrypted
 ......0. = Not Commit
 .....0.. = Not authentication only
 Message ID = 0 (0x0)
 Length = 464 (0x1D0)
 Payload type = Security Association
 Next payload = Vendor ID
 Reserved = 0 (0x0)
 Payload length = 412 (0x19C)
 DOI = 1 (0x1)
DOI = 1 (0x1)
 Situation = SIT_INDENTITY_ONLY
 Labeled domain identifier = 400 (0x190)
 Secrecy length = 257 (0x101)
 ERROR: Reserved = 6 (0x6), it should = 0 (0x0)
 Secrecy level = 03 00 00 24 01 01 00 00 80 01 00 01 80 02 00 01 80 04 00 01...
Secrecy length(in bits) = 12 (0xC)
ERROR: Reserved = 4 (0x4), it should = 0 (0x0)
Secrecy catagory = 00 00 70 80 7D 01 00 28 74 00 61 00 6C 00 6C 00 67 00 75...
```