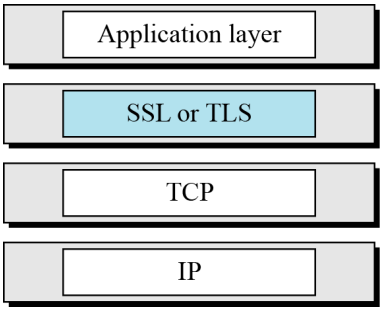
SSL/TLS

SSL/TLS

Rol: criptarea şi *autentificarea* legăturii de date între 2 aplicaţii, fără a face acestea la nivelul aplicaţie

Aplicaţie directă: ***https***
Alte aplicaţii: oricare: FTP, POP3, ...

Fără SSL: HTTP=80, SMTP=25, POP3=110
Cu SSL: HTTPS=443, SSMTP=465, SPOP3=995



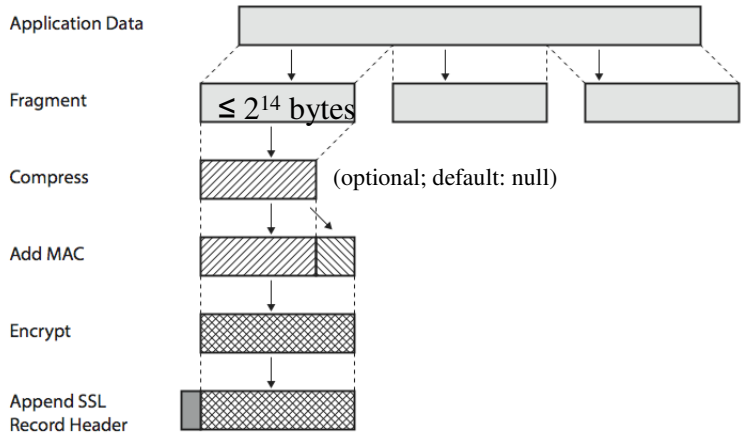
Istoric

- SSL 1.0
  - Standard proprietar Netscape la începutul anilor 90 (înainte de 1994)
- SSL 2.0
  - Prima versiune SSL publică (Netscape, 1994)
  - Diferite probleme; cheie foarte scurtă (40 biţi) în varianta de “export”
- SSL 3.0
  - Ultima versiune SSL (proiectat de Netscape, 1996)
- TLS 1.0
  - Standard Internet IETF, bazat pe SSL 3.0, 1999
  - Nu e interoperabil cu SSL 3.0

Partea 1: SSL

SSL – operaţii, pe scurt

Rol SSL: *autentificarea* (folosind MAC) şi *criptarea* cîmpului de date; opţional: compresie



Generarea parametrilor criptografici

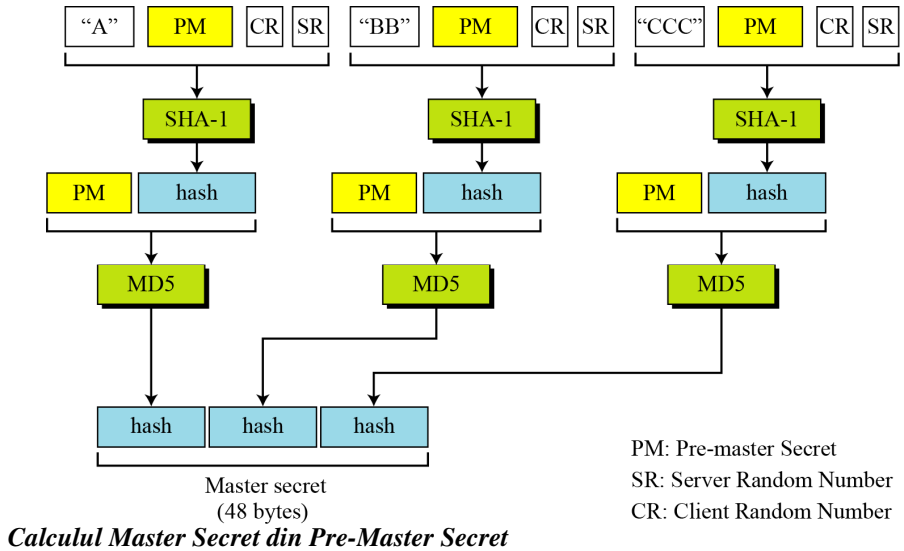
Sînt necesare *3 chei client* (autentificare + criptare + IV) *3 chei server* =6 chei în total (detalii în slides următoare)

• Clientul generează o valoare pseudoaleatoare numită *Pre-Master Secret* (PMS)
• O criptează cu cheia publică a serverului (dacă algoritmul pt. schimbul de chei nu este “NULL”)
• O trimite serverului

• Serverul decriptează PMS folosind cheia sa privată
• Din PMS generează MS *(Master Secret)* şi apoi cele 6 chei

• Detaliere !

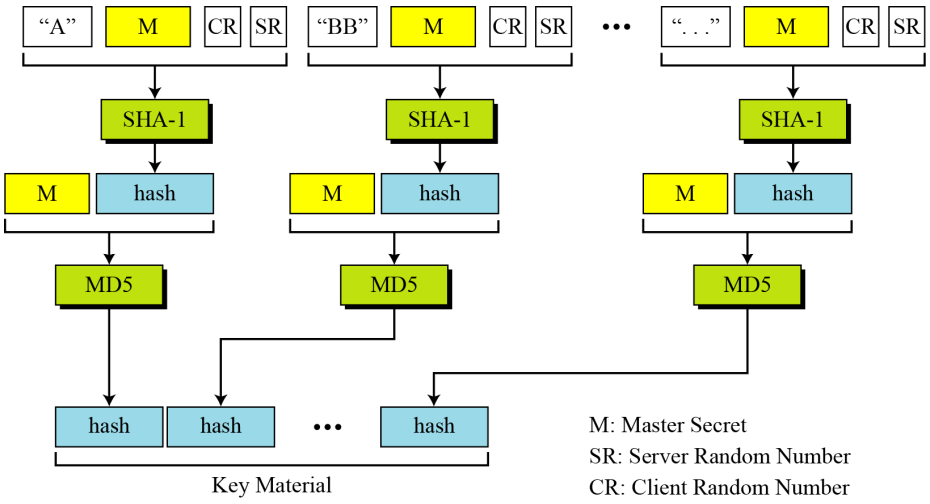
Generarea parametrilor criptografici (1)



Calculul Master Secret din Pre-Master Secret

"A", "BB", "CCC" sînt şirurile respective de caractere

Generarea parametrilor criptografici (2)

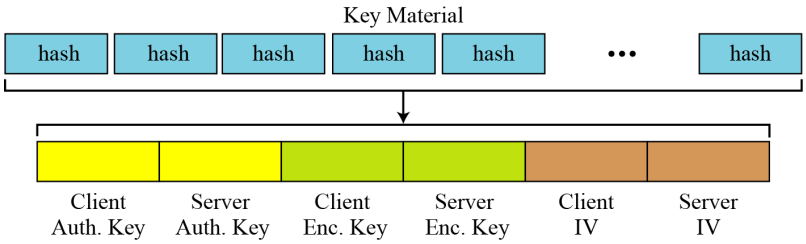


Calculul "key material" din Master secret

"key material" va fi folosit la generarea tuturor cheilor folosite în sesiune

Generarea parametrilor criptografici (3)

Auth. Key: Authentication Key
Enc. Key: Encryption Key
IV: Initialization Vector



Din "key material" precedent se extrag cele 6 chei necesare pe parcursul sesiunii:

- cheile de autentificare folosite de funcţiile MAC
- cheile de criptare folosite de "*record protocol*" pt criptarea datelor
- IV (*Initialization Vector*) pt CBC

Algoritmi pentru schimbul de chei

NULL

- *Nu se face schimb de chei;*
- *Clientul şi serverul trebuie să cunoască PMS (*Pre-Master Secret)

RSA

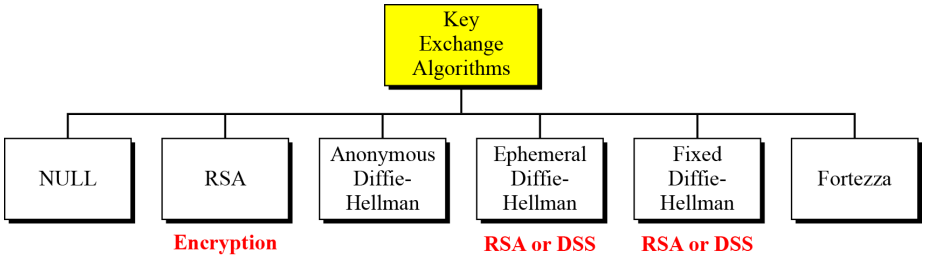
- *Schimb de chei RSA; cheie publică pe server*

Diffie-Hellman (3 variante)

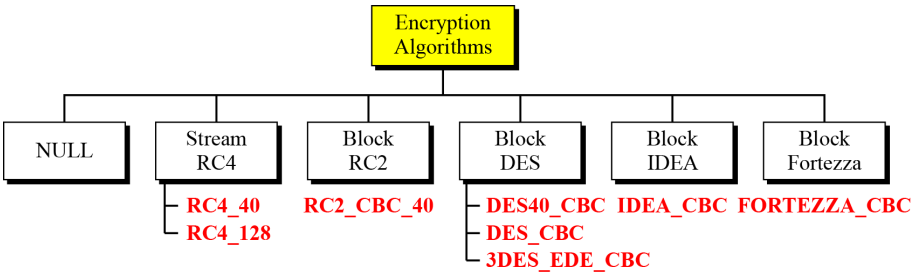
- *Se prepară 2 parametri (g şi p) specifici algoritmului DH (algoritm pentru chei publice)*

Fortezza

- *Algoritm dezvoltat de U.S. National Security Agency (NSA). Este o familie de protocoale de securitate dezvoltate pentru Department of Defense (DoD).*

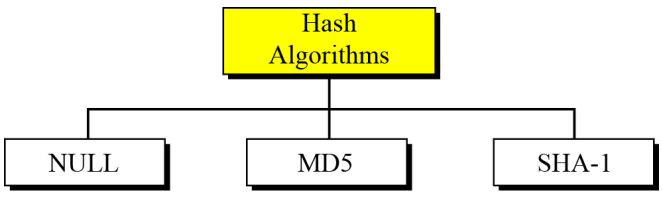


Algoritmi de cifrare/descifrare



Algoritmi de hash

Hash : algoritmi pentru integritatea mesajelor



NULL: Nu se face autentificarea mesajelor

Tipuri de cifru

- *Un tip de cifru: combinaţia de algoritmi de schimb de chei, hash şi algoritmi de criptare;*
- *este specific unei sesiuni SSL*

SSL_DHE_RSA_WITH_DES_CBC_SHA

Lista de tipuri de cifru (cipher suite)

Cipher suite	*Key Exchange*	*Encryption*	*Hash*
SSL_NULL_WITH_NULL_NULL	NULL	NULL	NULL
SSL_RSA_WITH_NULL_MD5	RSA	NULL	MD5
SSL_RSA_WITH_NULL_SHA	RSA	NULL	SHA-1
SSL_RSA_WITH_RC4_128_MD5	RSA	RC4	MD5
SSL_RSA_WITH_RC4_128_SHA	RSA	RC4	SHA-1
SSL_RSA_WITH_IDEA_CBC_SHA	RSA	IDEA	SHA-1
SSL_RSA_WITH_DES_CBC_SHA	RSA	DES	SHA-1
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES	SHA-1
SSL_DH_anon_WITH_RC4_128_MD5	DH_anon	RC4	MD5
SSL_DH_anon_WITH_DES_CBC_SHA	DH_anon	DES	SHA-1
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	DH_anon	3DES	SHA-1
SSL_DHE_RSA_WITH_DES_CBC_SHA	DHE_RSA	DES	SHA-1
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES	SHA-1
SSL_DHE_DSS_WITH_DES_CBC_SHA	DHE_DSS	DES	SHA-1
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE_DSS	3DES	SHA-1
SSL_DH_RSA_WITH_DES_CBC_SHA	DH_RSA	DES	SHA-1
SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH_RSA	3DES	SHA-1
SSL_DH_DSS_WITH_DES_CBC_SHA	DH_DSS	DES	SHA-1
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH_DSS	3DES	SHA-1
SSL_FORTEZZA_DMS_WITH_NULL_SHA	Fortezza	NULL	SHA-1
SSL_FORTEZZA_DMS_WITH_FORTEZZA_CBC_SHA	Fortezza	Fortezza	SHA-1
SSL_FORTEZZA_DMS_WITH_RC4_128_SHA	Fortezza	RC4	SHA-1

Algoritmi de compresie

în SSLv3 compresia este opţională.
compresia implicită este tipul NULL.

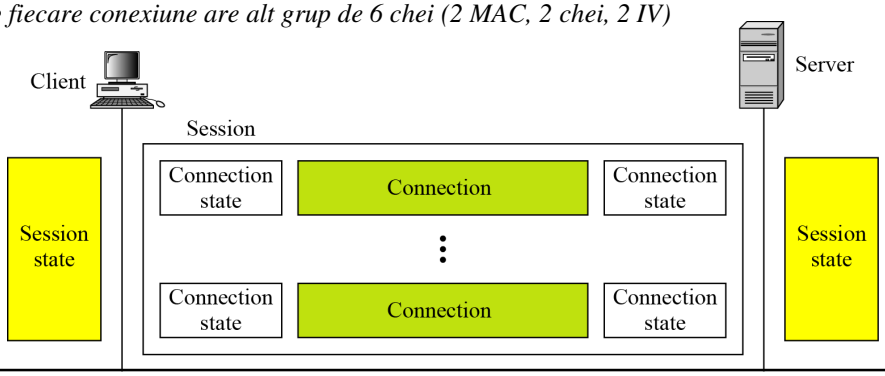
Sesiuni şi conexiuni

sesiune:

- *client + server negociază parametrii de securitate, o singură dată*
- *certificat, compresie, cipher_spec*
- *master secret unic la nivel de sesiune*
- *se alocă un "session state" pt sesiune*

conexiune:

- *mai multe conexiuni posibile pt o sesiune; folosesc acelaşi "session state" şi deci master secret; fiecare are propriul "connection state"*
- *fiecare conexiune are alt grup de 6 chei (2 MAC, 2 chei, 2 IV)*



Parametri de sesiune

Session State

Parameter	*Description*
Session ID	A server-chosen 8-bit number defining a session.
Peer Certificate	A certificate of type X509.v3. This parameter may by empty (null).
Compression Method	The compression method.
Cipher Suite	The agreed-upon cipher suite.
Master Secret	The 48-byte secret.
Is resumable	A yes-no flag that allows new connections in an old session.

Parametri de conexiune

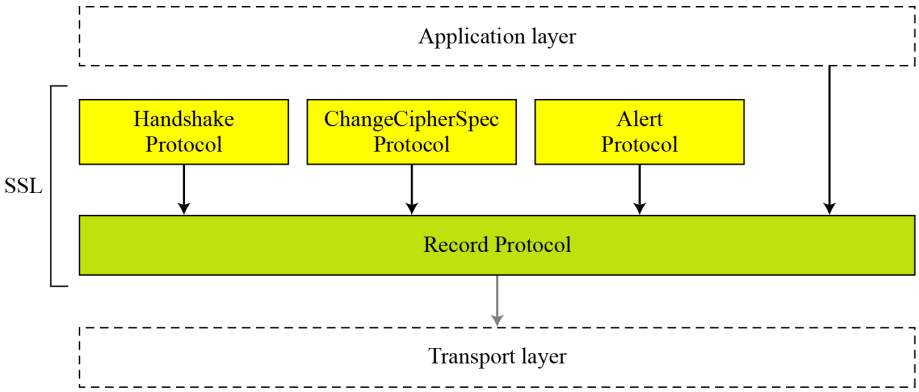
Connection State

Parameter	*Description*
Server and client random numbers	A sequence of bytes chosen by the server and client for each connection.
Server write MAC secret	The outbound server MAC key for message integrity. The server uses it to sign; the client uses it to verify.
Client write MAC secret	The outbound client MAC key for message integrity. The client uses it to sign; the server uses it to verify.
Server write secret	The outbound server encryption key for message integrity.
Client write secret	The outbound client encryption key for message integrity.
Initialization vectors	The block ciphers in CBC mode use initialization vectors (IVs). One initialization vector is defined for each cipher key during the negotiation, which is used for the first block exchange. The final cipher text from a block is used as the IV for the next block.
Sequence numbers	Each party has a sequence number. The sequence number starts from 0 and increments. It must not exceed $2^{64} - 1$.

Cele 4 protocoale SSL

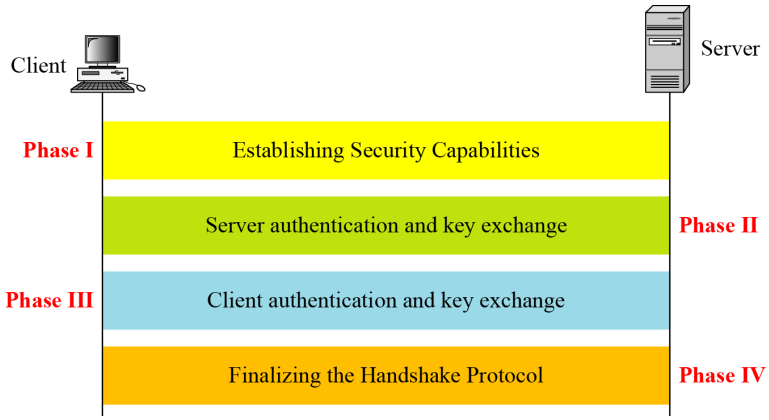
4 protocoale, în 2 nivele:

1. Handshake Protocol: stabilirea variabilelor de comunicatie, schimbul de chei între cele 2 entităţi
2. ChangeCipher Spec Protocol: schimbări în variabilele de comunicaţie, după ce sesiunea a început
3. Alert Protocol: mesaje importante pentru conexiunea SSL
4. Record Protocol: criptarea datelor (funcţia principală a SSL)



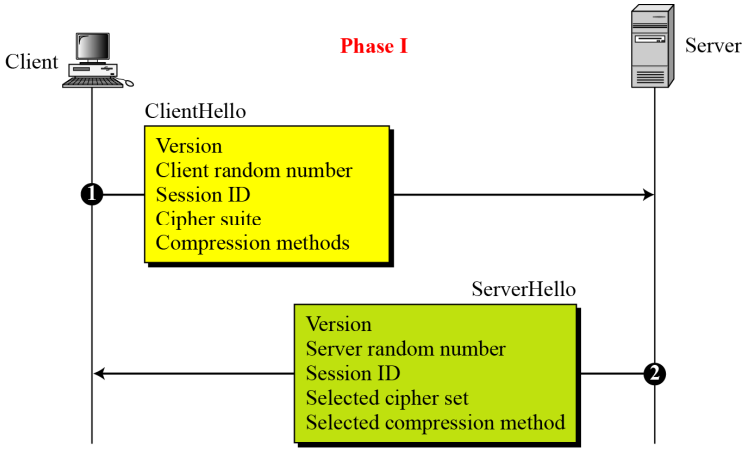
Protocolul SSL nr. 1: Handshake Protocol

Handshake Protocol - cele 4 faze



Handshake Protocol

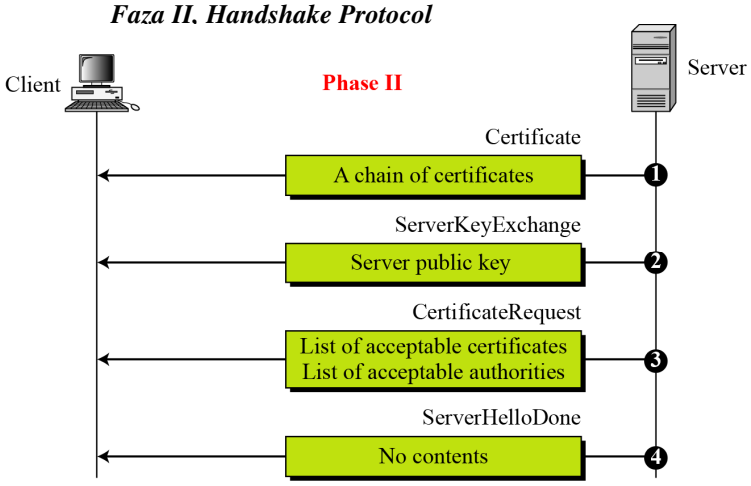
Faza I, Handshake Protocol



La sfîrşitul fazei I:
clientul şi serverul au negociat protocoalele de securitate disponibile

Handshake Protocol

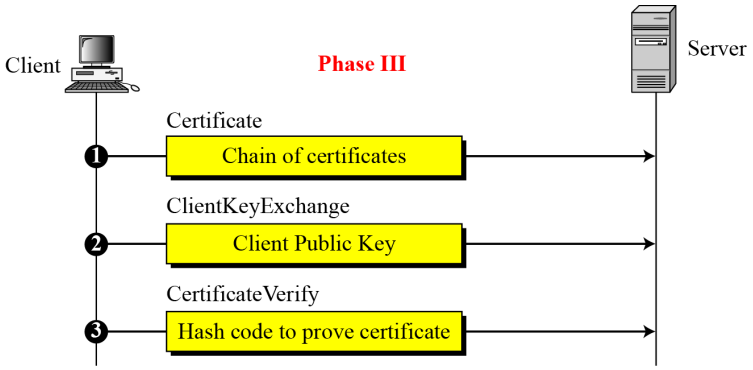
Faza II. Handshake Protocol



Certificatele conţin cheia publică a serverului, autentificată de un CA sau un lanţ - *chain* - de CA (vezi detalii la sfîrşit)
La sfîrşitul fazei II:
• serverul este autentificat clientului
• clientul cunoaşte cheia publică a serverului

Handshake Protocol

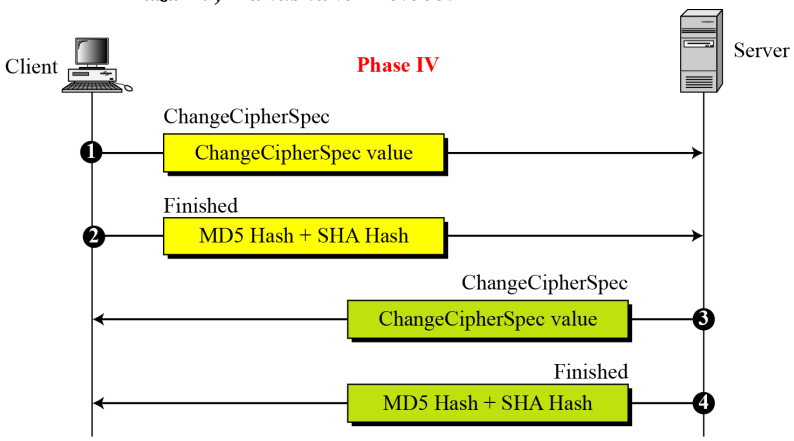
Faza III, Handshake Protocol



La sfîrşitul fazei III:
- clientul este autentificat serverului (*autentificarea clientului este opţională*)
- clientul şi serverul cunosc PMS (pre-master secret)

Handshake Protocol

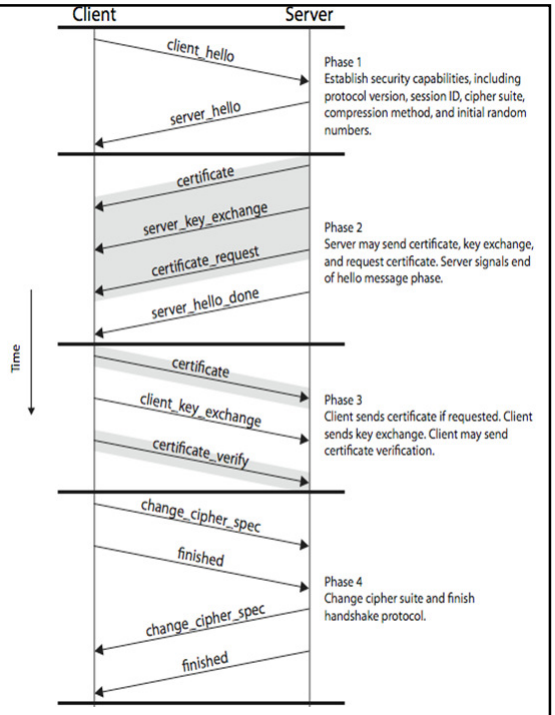
Faza IV, Handshake Protocol



La sfîrşitul fazei IV:
clientul şi serverul sînt gata să facă schimbul de date

Handshake Protocol

Handshake în rezumat
(Stallings Fig. 17.6)
cu gri: opţional

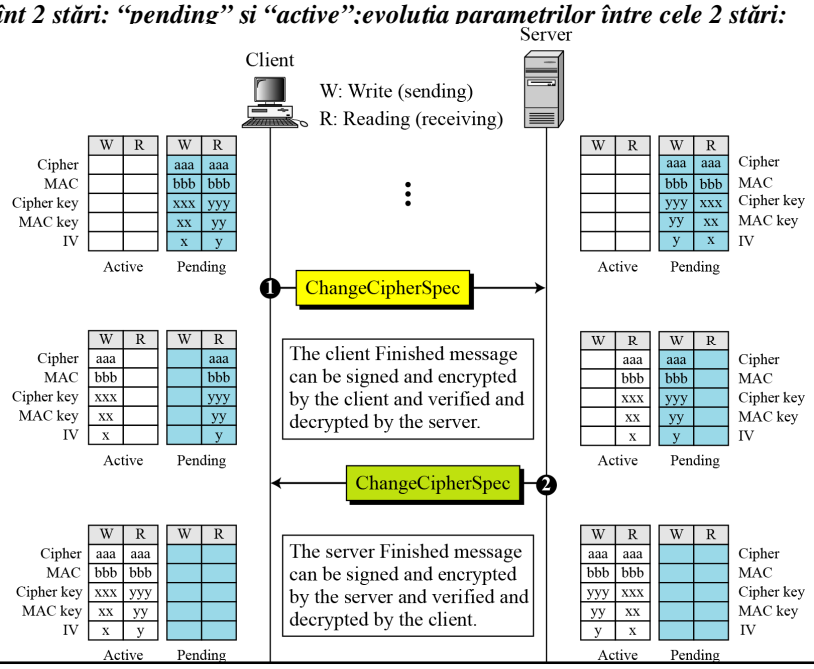


Tipuri de mesaje Handshake

Type	*Message*
0	HelloRequest
1	ClientHello
2	ServerHello
11	Certificate
12	ServerKeyExchange
13	CertificateRequest
14	ServerHelloDone
15	CertificateVerify
16	ClientKeyExchange
20	Finished

Protocolul SSL nr. 2: ChangeCipherSpec Protocol

Sînt 2 stări: "pending" si "active":evolutia parametrilor între cele 2 stări:



Protocolul SSL nr. 3: Alert Protocol

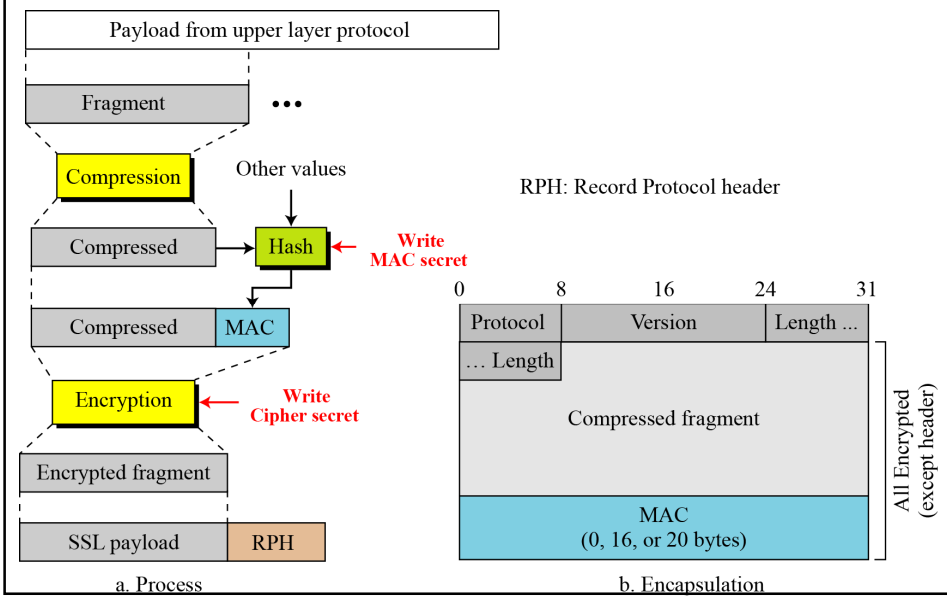
Mesaje de alertă defnite pt SSL

Value	*Description*	*Meaning*
0	*CloseNotify*	Sender will not send any more messages.
10	*UnexpectedMessage*	An inappropriate message received.
20	*BadRecordMAC*	An incorrect MAC received.
30	*DecompressionFailure*	Unable to decompress appropriately.
40	*HandshakeFailure*	Sender unable to finalize the handshake.
41	*NoCertificate*	Client has no certificate to send.
42	*BadCertificate*	Received certificate corrupted.
43	*UnsupportedCertificate*	Type of received certificate is not supported.
44	*CertificateRevoked*	Signer has revoked the certificate.
45	*CertificateExpired*	Certificate expired.
46	*CertificateUnknown*	Certificate unknown.
47	*IllegalParameter*	An out-of-range or inconsistent field.

Protocolul SSL nr. 4: Record Protocol

OBS: MAC=Message Authentication Code; MAC trebuie sa fie acelaşi la sursă şi destinaţie

Operaţiile efectuate de către Record Protocol:



Formatul mesajelor SSL

Mesajele Record Protocol conţin (prin încapsulare) mesajele de la 3 protocoale SSL şi datele de la nivelul aplicaţie

ChangeCipherSpec Protocol
Alert Protocol
Handshake Protocol
+ Application Data

Partea 2: TLS

Transport Layer Security (TLS)

• TLS este versiunea IETF de SSL.
• Ultima versiune de SSL v. 3.0 stă la baza TLS v. 1.0
• În *handshake* versiunea transmisă este v. 3.1

•TLS este foarte similar SSL dar micile diferenţe le fac incompatibile:

- Se foloseşte HMAC în loc de MAC
- Fortezza nu mai există în lista de cifuri
- Schimbări în calculul hash-ului
- Alerte adiţionale

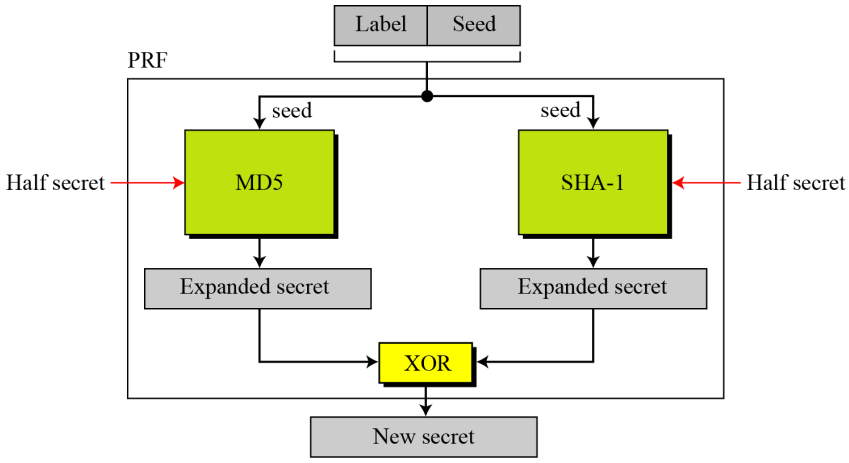
Cipher Suite

Cipher Suite TLS

Cipher suite	*Key Exchange*	*Encryption*	*Hash*
TLS_NULL_WITH_NULL_NULL	NULL	NULL	NULL
TLS_RSA_WITH_NULL_MD5	RSA	NULL	MD5
TLS_RSA_WITH_NULL_SHA	RSA	NULL	SHA-1
TLS_RSA_WITH_RC4_128_MD5	RSA	RC4	MD5
TLS_RSA_WITH_RC4_128_SHA	RSA	RC4	SHA-1
TLS_RSA_WITH_IDEA_CBC_SHA	RSA	IDEA	SHA-1
TLS_RSA_WITH_DES_CBC_SHA	RSA	DES	SHA-1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	RSA	3DES	SHA-1
TLS_DH_anon_WITH_RC4_128_MD5	DH_anon	RC4	MD5
TLS_DH_anon_WITH_DES_CBC_SHA	DH_anon	DES	SHA-1
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA	DH_anon	3DES	SHA-1
TLS_DHE_RSA_WITH_DES_CBC_SHA	DHE_RSA	DES	SHA-1
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES	SHA-1
TLS_DHE_DSS_WITH_DES_CBC_SHA	DHE_DSS	DES	SHA-1
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE_DSS	3DES	SHA-1
TLS_DH_RSA_WITH_DES_CBC_SHA	DH_RSA	DES	SHA-1
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH_RSA	3DES	SHA-1
TLS_DH_DSS_WITH_DES_CBC_SHA	DH_DSS	DES	SHA-1
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH_DSS	3DES	SHA-1

Pseudo-Random Function (PRF)

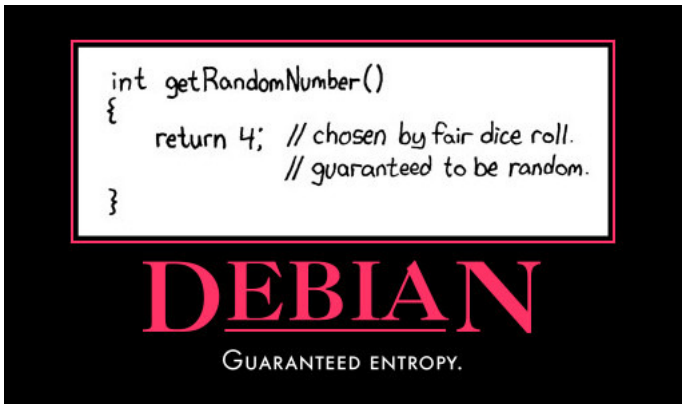
PRF în TLS:
se generează din tripletul (secret,label,seed)



Pseudo-Random Function

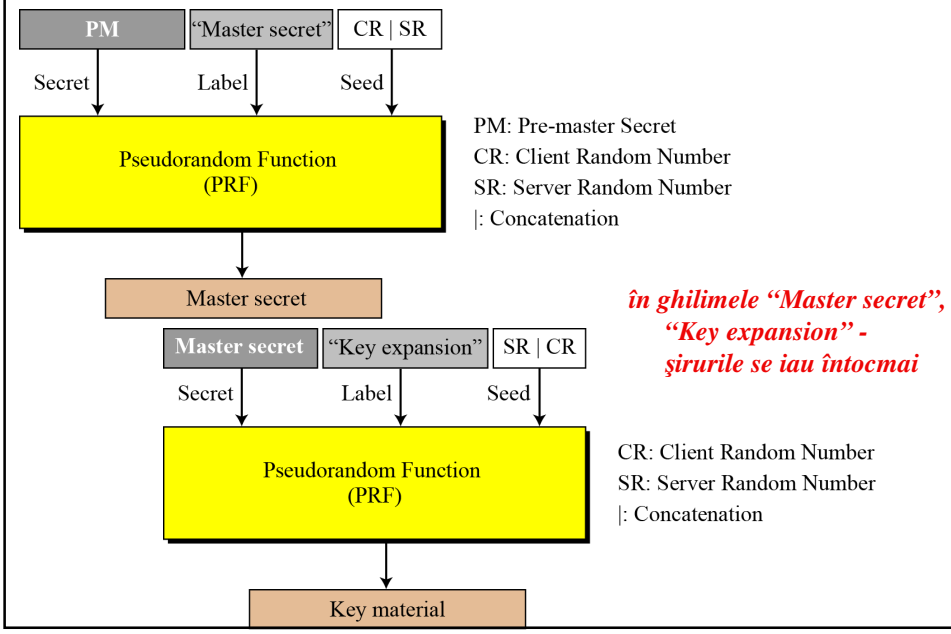
Ce înseamnă pseudo-random ?

Pseudo-Random Function



Master Secret

PMS se calculează la fel ca în SSL



în ghilimele "Master secret", "Key expansion" - şirurile se iau întocmai

Alert Protocol

Schimbări în TLS:

- *nu există alerta NoCertificate.*
- *câteva alerte noi*

Tipuri de alerte:

Value	*Description*	*Meaning*
0	*CloseNotify*	Sender will not send any more messages.
10	*UnexpectedMessage*	An inappropriate message received.
20	*BadRecordMAC*	An incorrect MAC received.
21	*DecryptionFailed*	Decrypted message is invalid.
22	*RecordOverflow*	Message size is more than 2^{14} + 2048.
30	*DecompressionFailure*	Unable to decompress appropriately.
40	*HandshakeFailure*	Sender unable to finalize the handshake.
42	*BadCertificate*	Received certificate corrupted.
43	*UnsupportedCertificate*	Type of received certificate is not supported.
44	*CertificateRevoked*	Signer has revoked the certificate.
45	*CertificateExpired*	Certificate has expired.
46	*CertificateUnknown*	Certificate unknown.
47	*IllegalParameter*	A field out of range or inconsistent with others.
48	*UnknownCA*	CA could not be identified.

HTTPS: HTTP+ TLS

HTTPS: cea mai populară aplicaţie a TLS

URL: https://

Necesită un server şi un browser web *TLS-capable*

Certificate: detaliere

- Sînt soluţia la problemele:
  - cine garantează *autenticitatea* unei chei publice ?
  - cum realizăm acest lucru în mod *scalabil* în Internet ?

- Folosite în fazele II, III ale *SSL Handshake Protocol*
- Certificatul conţine cheia publică
- X.509: standard de certificate; asociază o cheie publică cu o identitate (identificarea serverului)
  - Certificatele **sînt create de un CA (*Certificate Authority)***
  - Permit verificarea identităţii
  - **CA semnează certificatele** folosind cheia sa privată; oricine poate verifica, nimeni nu poate altera
  - problemă: certificatele *self-signed* din considerente de cost
  - "*certificate chain*": un CA poate semna certif. pt. un alt CA
  - un certificat poate fi revocat prin publicarea sa într-un CRL (*Certificate Revocation List)*

Conţinutul unui certificat X.509

Cu bold: cîmpurile esenţiale

Versiune (1,2,3)
Serial Number
Signature Algorithm Identifier
    Object Identifier (OID)
    e.g. id-dsa: {iso(1) member-body(2) us(840) x9-57 (10040) x9algorithm(4) 1}
Emis de CA: nume X.500
Perioada de validitate (Start,End)
Subject X.500 name (*standard ISO; începînd cu ver.3, se acceptă inclusiv folosirea unei adrese de e-mail standard*)
Subject Public Key
    Algoritm
    Valoare
Issuer (CA) Unique Id (Ver. 2 ,3)
Subject Unique Id (Ver. 2,3)
Extensii (ver. 3)
    optional
CA digital Signature

Subject Name X.500

numit şi DN = *Distinguished Name*

- Nivelul 1: ţara de origine (e.g. US)
- Nivelul 2: organizaţia care emite certificatul (e.g. CertCo)
- Nivelul 3: "numele comun" (e.g. *Common Name* "Elizabeth" cu Id = 1)

- Exemplu:
  DN = {C=US/O=CertCo/CN=Elizabeth, ID=1}

Începînd cu v3 se acceptă variante gen elisabeth@adresa.com

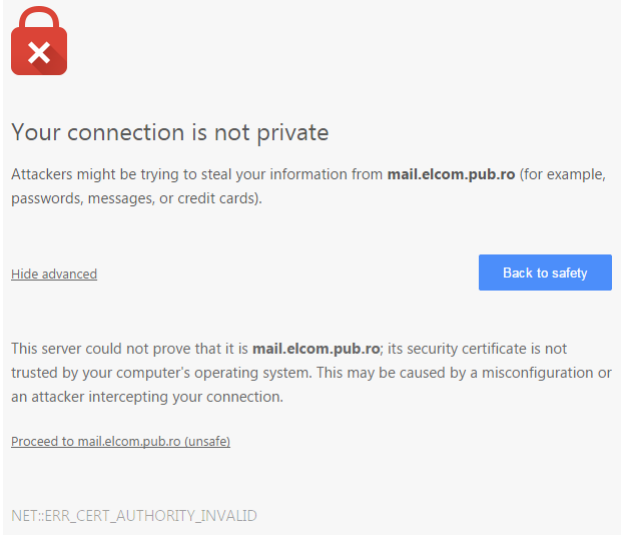
Semnarea certificatelor

- Serverul crează o semnătură RSA
  - crează hash al certificatului
  - îl criptează folosind cheia privată a CA, publică rezultatul
- Verificarea semnăturii în client (browser), echivalentă cu verificarea autenticităţii serverului
  - Decriptează hash-ul primit, folosind cheia publică a CA
  - Verifică egalitatea hash-ului primit cu cel calculat local
- Cheile publice ale CA sînt menţinute de obicei de către OS într-o listă de "*root certificates*"
  - Ele se pot actualiza periodic (ex: Windows updates - *root certificates updates*)
  - Peste 100 de CA: VeriSign, Equifax, GlobalSign, etc

Semnarea certificatelor

- *Self-signed certificates*: atunci cînd nu se foloseşte un CA; sesiunea *SSL* **este criptată** dar nu se realizează **autentificarea** serverului; metodă simplă folosită mai ales în intraneturi
- clientul va genera un warning
- semnarea unui certificat de către un CA costă ! (ex: $250/an, Thawte Inc.)

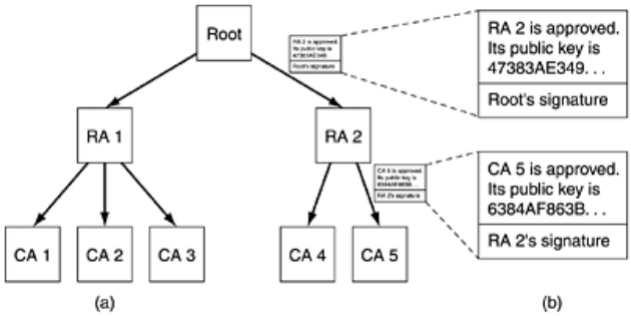
Semnarea certificatelor



Exemplu de *Self-signed certificate* în Google Chrome

← eroarea raportată: nu există un CA valid

Ierarhia CA: PKI (Public-Key Infrastructure)



a) ierarhia CA
b) lanţ de "încredere" = *certification path*

- CA de nivelul 1: *root* (mai multe *root* în paralel, ca la DNS)
- CA de nivelul 2: RA (*Regional Authority)*
- CA de nivelul 3 folosite pentru certificarea propriu-zisă; nivelele superioare pentru autentificarea acestora.
- *Exemplu*
  - Clientul obţine/autentifică cheia publică a serverului de la CA5
  - Autentificarea lui CA5 este asigurată de RA2, care furnizează cheia publică a lui CA5
  - Autentificarea lui RA2 este asigurată de *root*, care furnizează cheia publică a lui RA2
  - Autentificarea *root*: listă de chei publice ţinută/actualizată de OS şi/sau de browser

Modele de încredere

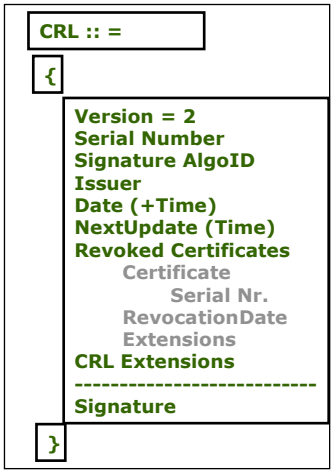
- *"trust models"*
- Modelul de încredere distribuit
- Se crează un *chain of trust* din certificatele CA5, RA2, root

- xkcd.com:

Revocarea certificatelor

- un certificat trebuie revocat dacă:
  - cheia privată a proprietatului (serverului) a fost compromisă
  - schimbări de nume, afiliere, ...
  - încălcarea unor politici ale CA
  - etc

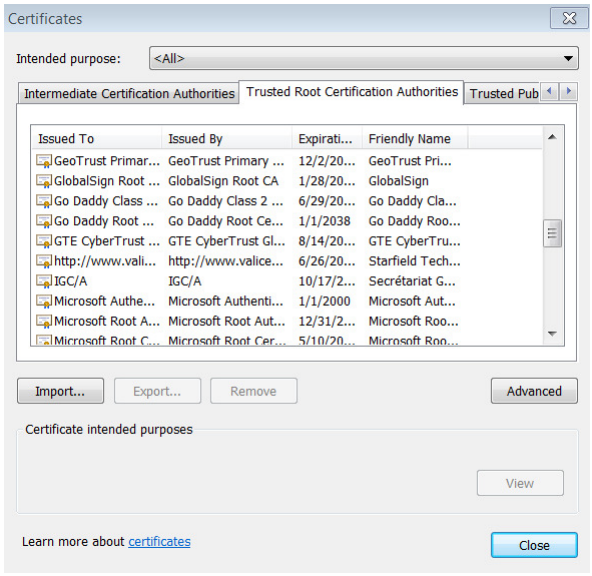
- CRL: *Certificate Revocation List*
- Este semnat de CA
- Este stocat pe acelaşi CA care a emis certificatul, sau în alte locaţii
- CRL actualizate de N ori pe zi



Certificate: exemplu

Google Chrome -> settings
-> HTTPS/SSL
Manage Certificates

1) Trusted Root Certification Authorities



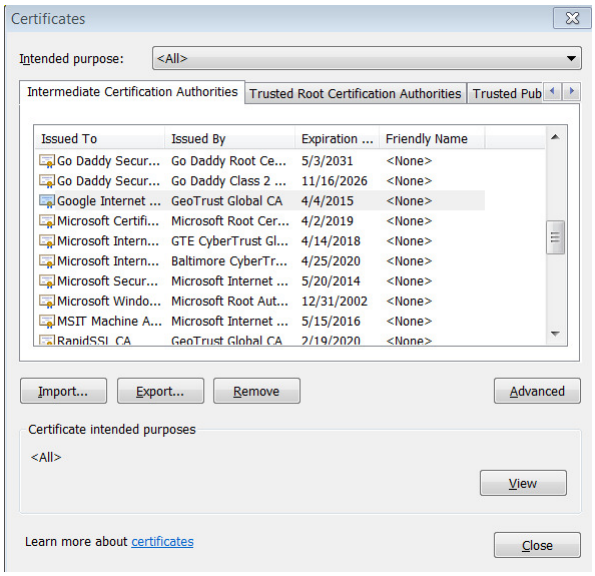
Certificate: exemplu (2)

Google Chrome -> settings
-> HTTPS/SSL
Manage Certificates

2) Intermediate Certification Authorities

Export →

Fişierul exportat *googlecert.cer* analizat cu ***certutil***



Certificate: exemplu (3)

```
c:\> certutil -dump googlecert.cer
X509 Certificate:
Version: 3
Serial Number: 023a69
Signature Algorithm:
    Algorithm ObjectId: 1.2.840.113549.1.1.5 sha1RSA
    Algorithm Parameters:
    05 00
Issuer:
    CN=GeoTrust Global CA
    O=GeoTrust Inc.
    C=US

 NotBefore: 4/5/2013 5:15 PM
 NotAfter: 4/4/2015 5:15 PM

Subject:
    CN=Google Internet Authority G2
    O=Google Inc
    C=US

Public Key Algorithm:
    Algorithm ObjectId: 1.2.840.113549.1.1.1 RSA (RSA_SIGN)
    Algorithm Parameters:
    05 00
Public Key Length: 2048 bits
```

Certificate: exemplu (4)

```
Public Key: UnusedBits = 0
    0000  30 82 01 0a 02 82 01 01  00 9c 2a 04 77 5c d8 50
    0010  91 3a 06 a3 82 e0 d8 50  48 bc 89 3f f1 19 70 1a
    0020  88 46 7e e0 8f c5 f1 89  ce 21 ee 5a fe 61 0d b7
    0030  32 44 89 a0 74 0b 53 4f  55 a4 ce 82 62 95 ee eb
    0040  59 5f c6 e1 05 80 12 c4  5e 94 3f bc 5b 48 38 f4
    0050  53 f7 24 e6 fb 91 e9 15  c4 cf f4 53 0d f4 4a fc
    0060  9f 54 de 7d be a0 6b 6f  87 c0 d0 50 1f 28 30 03
    0070  40 da 08 73 51 6c 7f ff  3a 3c a7 37 06 8e bd 4b
    0080  11 04 eb 7d 24 de e6 f9  fc 31 71 fb 94 d5 60 f3
    0090  2e 4a af 42 d2 cb ea c4  6a 1a b2 cc 53 dd 15 4b
    00a0  8b 1f c8 19 61 1f cd 9d  a8 3e 63 2b 84 35 69 65
    00b0  84 c8 19 c5 46 22 f8 53  95 be e3 80 4a 10 c6 2a
    00c0  ec ba 97 20 11 c7 39 99  10 04 a0 f0 61 7a 95 25
    00d0  8c 4e 52 75 e2 b6 ed 08  ca 14 fc ce 22 6a b3 4e
    00e0  cf 46 03 97 97 03 7e c0  b1 de 7b af 45 33 cf ba
    00f0  3e 71 b7 de f4 25 25 c2  0d 35 89 9d 9d fb 0e 11
    0100  79 89 1e 37 c5 af 8e 72  69 02 03 01 00 01
Certificate Extensions: 7
    2.5.29.35: Flags = 0, Length = 18
    Authority Key Identifier
        KeyID=c0 7a 98 68 8d 89 fb ab 05 64 0c 11 7d aa 7d 65 b8 ca cc 4e

    2.5.29.14: Flags = 0, Length = 16
    Subject Key Identifier
        4a dd 06 16 1b bc f6 68 b5 76 f5 81 b6 bb 62 1a ba 5a 81 2f
```

Certificate: exemplu (5)

```
 2.5.29.19: Flags = 1(Critical), Length = 8
    Basic Constraints
        Subject Type=CA
        Path Length Constraint=0

    2.5.29.15: Flags = 1(Critical), Length = 4
    Key Usage
        Certificate Signing, Off-line CRL Signing, CRL Signing (06)

    2.5.29.31: Flags = 0, Length = 33
    CRL Distribution Points
        [1]CRL Distribution Point
             Distribution Point Name:
                  Full Name:
                       URL=http://crl.geotrust.com/crls/gtglobal.crl

    1.3.6.1.5.5.7.1.1: Flags = 0, Length = 31
    Authority Information Access
        [1]Authority Info Access
             Access Method=On-line Certificate Status Protocol (1.3.6.1.5.5.7.48.1)
             Alternative Name:
                  URL=http://gtglobal-ocsp.geotrust.com

    2.5.29.32: Flags = 0, Length = 10
    Certificate Policies
        [1]Certificate Policy:
             Policy Identifier=1.3.6.1.4.1.11129.2.5.1
```

Certificate: exemplu (6)

```
Signature Algorithm:
    Algorithm ObjectId: 1.2.840.113549.1.1.5 sha1RSA
    Algorithm Parameters:
    05 00
Signature: UnusedBits=0
    0000  71 33 54 24 20 50 a0 c8  79 16 7e 3d 08 2c d0 94
    0010  da ca fc 78 05 a7 e1 ca  3a 05 cd 6b 83 6b 8f 51
    0020  c0 61 5d 40 87 fa 67 f9  b2 dc 38 70 c8 a0 ee d4
    0030  81 3a 50 5d 78 4c 2e 90  03 78 bd 33 a9 0d 9d 35
    0040  06 3b 62 24 a8 db 73 58  68 cb c2 e3 3b 95 b3 33
    0050  fb f0 c6 02 8f e8 fb 73  cc 8d 10 b2 ac 3c f0 17
    0060  ff 13 95 6e f0 be 95 c2  74 8c dc 96 aa a2 db 2a
    0070  57 6f 7b 51 96 bd 56 1c  da 3c 5e 06 fe 55 fc 40
    0080  d0 be 4f 9d 12 1d d1 46  f6 7a 38 9e a3 09 44 4f
    0090  ed 59 e0 bd a3 56 f0 35  99 3d 79 f7 c8 fd cd 13
    00a0  51 68 c3 58 03 85 6f cc  bf f6 7f 5b 8c 04 ca 6a
    00b0  dc dc 86 95 32 56 ba a1  80 6f 86 8e da 66 95 ed
    00c0  0d 78 c8 14 da 90 40 46  76 67 5e eb 12 6e 25 3b
    00d0  d1 5e 38 bd 6b c8 22 fe  58 42 1b fd 37 3b 95 f0
    00e0  e0 97 06 36 8e d8 1e da  72 ba 42 83 7e b1 bb 58
    00f0  75 a0 23 b3 77 38 9b 14  2a ad 27 11 80 06 d7 36
Non-root Certificate
Key Id Hash(rfc-sha1): 4a dd 06 16 1b bc f6 68 b5 76 f5 81 b6 bb 62 1a ba 5a 81 2f
Key Id Hash(sha1): 43 da d6 30 ee 53 f8 a9 80 ca 6e fd 85 f4 6a a3 79 90 e0 ea
Cert Hash(md5): 9e 4a c9 64 74 24 51 29 d9 76 67 00 41 2a 1f 89
Cert Hash(sha1): d8 3c 1a 7f 4d 04 46 bb 20 81 b8 1a 16 70 f8 18 34 51 ca 24
CertUtil: -dump command completed successfully.
```

Bibliografie

- Behrouz Forouzan, *Cryptography and network security*, McGraw-Hill
- Tanembaum, *Computer Networks 4th ed.,* Prentice Hall