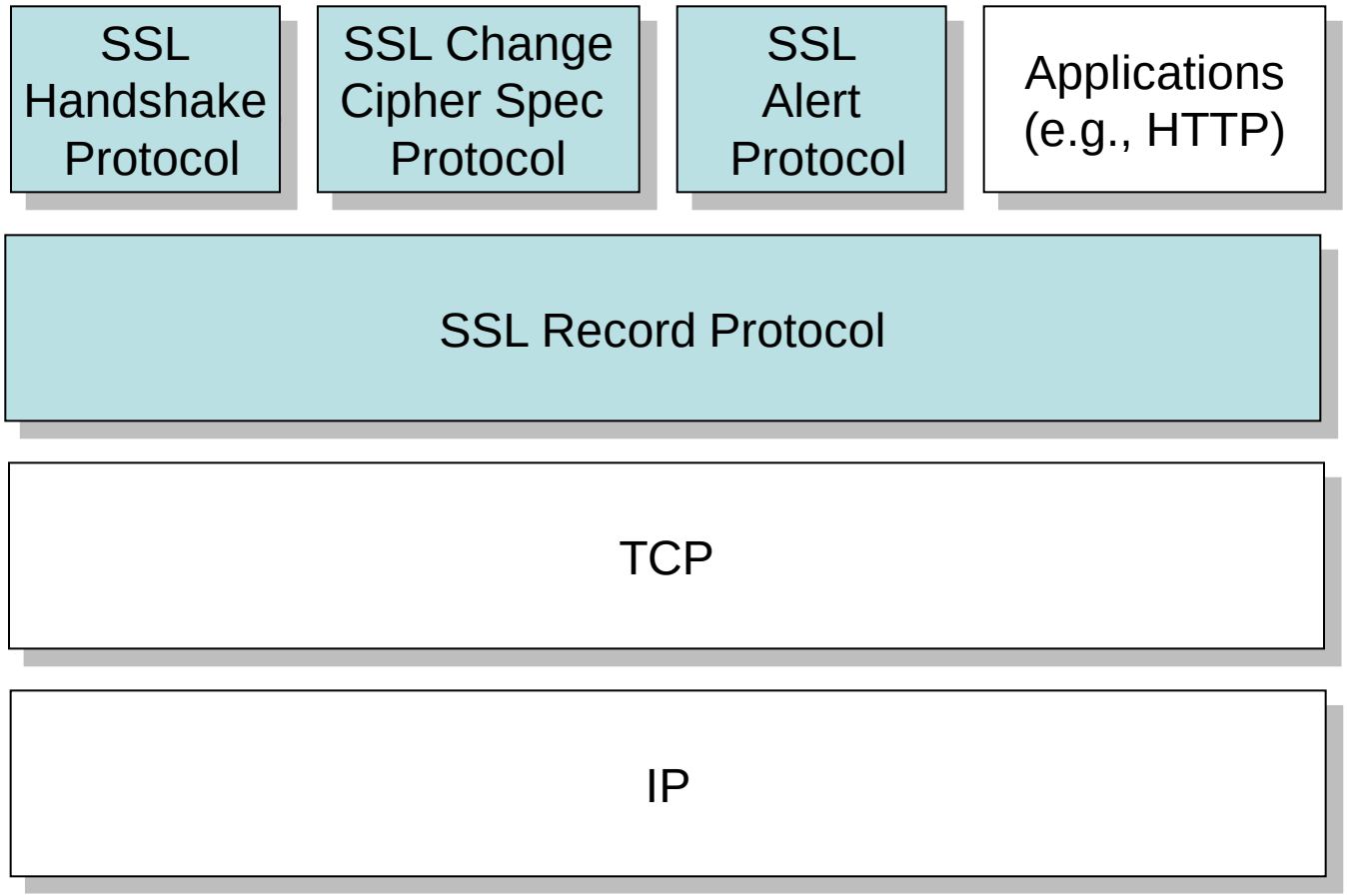
3. Protocoale de Securitate la Nivel Transport

Protocolul SSL

- Propus de firma Netscape
  - http://wp.netscape.com/eng/ssl3/ssl-toc.html
- Protocol de Nivel 4+
  - se bazează pe mecanismele de comunicaţie existente în TCP
  - pune la dispoziţia aplicaţiilor socket-uri de comunicaţie securizate
  - aplicaţiile trebuie modificate să suporte acest lucru – “SSLizare”
  - stiva TCP/IP de pe client/server nu trebuie modificată!
- SSL v3.0
- TLS v1.0 - RFC 2246
- TLS v1.1 - RFC 4346
- TLS v1.2 - RFC 5246
- TLS v1.3 - RFC 8446

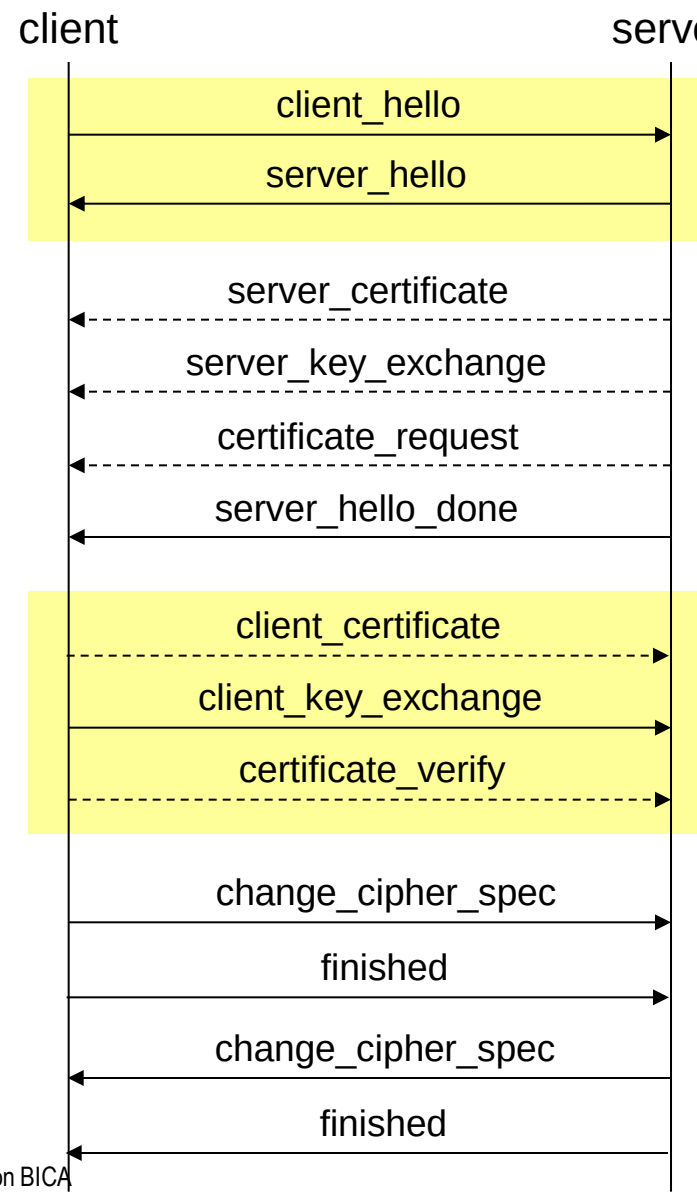
Protocolul SSL (cont.)



Protocolul SSL (cont.)

- SSL Handshake Protocol (Protocolul de Negociere a Conexiunii)
  - negocierea algoritmilor şi parametrilor criptografici
  - schimbul de chei
  - autentificare server şi opţional client
- SSL Record Protocol (Protocolul de Transfer al Datelor)
  - fragmentarea datelor
  - compresia datelor
  - autenticitatea şi integritatea datelor (folosind MAC)
  - criptarea datelor (folosind algoritmi simetrici)
- SSL Change Cipher Spec Protocol
  - un singur mesaj ce indică sfârşitul fazei de handshake
- SSL Alert Protocol
  - mesaje de eroare

SSL Handshake Protocol



Phase 1: Negotiation of the session ID, key exchange algorithm, MAC algorithm, encryption algorithm, and exchange of initial random numbers

Phase 2: Server may send its certificate and key exchange message, and it may request the client to send a certificate. Server signals end of hello phase.

Phase 3: Client sends certificate if requested and may send an explicit certificate verification message. Client always sends its key exchange message.

Phase 4: Change cipher spec and finish handshake

SSL Handshake Protocol (cont.)

- **Autentificarea entităţilor**
  - certificate digitale
  - obligatoriu pentru server, opţional pentru client
- **Mecanisme pentru schimbul de chei**
  - RSA based
    - cheia de secretă este criptată cu cheia publică a serverului
  - Fixed Diffie-Hellman
    - certificat digital în care sunt stabiliţi parametrii DH
  - Ephemeral Diffie-Hellman
    - parametrii DH sunt generaţi aleator şi semnaţi folosind RSA / DSA
  - Anonymous Diffie-Hellman
    - parametrii DH sunt generaţi aleator şi transmişi fără autentificare
  - Fortezza
    - schemă proprietară

SSL Handshake Protocol (cont.)

1. client_hello
   - client_version
     - versiunea de protocol suportă de client
   - client_random
     - timpul curent (4 octeţi) + date aleatoare (28 octeţi)
   - session_id
   - cipher_suites (lista algoritmilor criptografici suportaţi de client)
     - exemplu: SSL_RSA_with_3DES_EDE_CBC_SHA, …
   - compression_methods (lista algoritmilor de compresie suportaţi de client)
2. server_hello
   - server_version
     - min (versiunea suportă de client, versiunea suportă de server)
   - server_random
     - timpul curent (4 octeţi) + date aleatoare (28 octeţi)
   - session_id
   - cipher_suite (algoritmul criptografic selectat de server din lista propusă de client)
   - compression_method (algoritmul de compresie selectat de server din lista propusă de client)

SSL Handshake Protocol (cont.)

3. server certificate
   - certificat X.509 (RSA / DSA / DH)
4. server_key_exchange
   - cheia publică RSA / parametrii DH
5. certificate_request
   - transmis numai dacă se cere autentificarea clientului
   - specifică tipul de certificat cerut
6. server_hello_done
   - indică faptul că serverul a încheiat schimbul de chei

SSL Handshake Protocol (cont.)

7. client certificate
   - transmis numai dacă serverul cere clientului să se autentifice
   - certificat X.509 (RSA / DSA / DH)
8. client_key_exchange
   - cheia secretă criptată cu cheia publică a serverului / parametrii DH
9. certificate_verify
   - transmis numai dacă serverul cere clientului să se autentifice
   - mesaj semnat digital pentru autentificarea clientului

SSL Handshake Protocol (cont.)

10. client change_cipher_spec
    - clientul anunţă serverul că a terminat de încărcat noile configuraţii criptografice
11. client finished
12. server change_cipher_spec
    - serverul anunţă clientul că a terminat de încărcat noile configuraţii criptografice
13. server finished

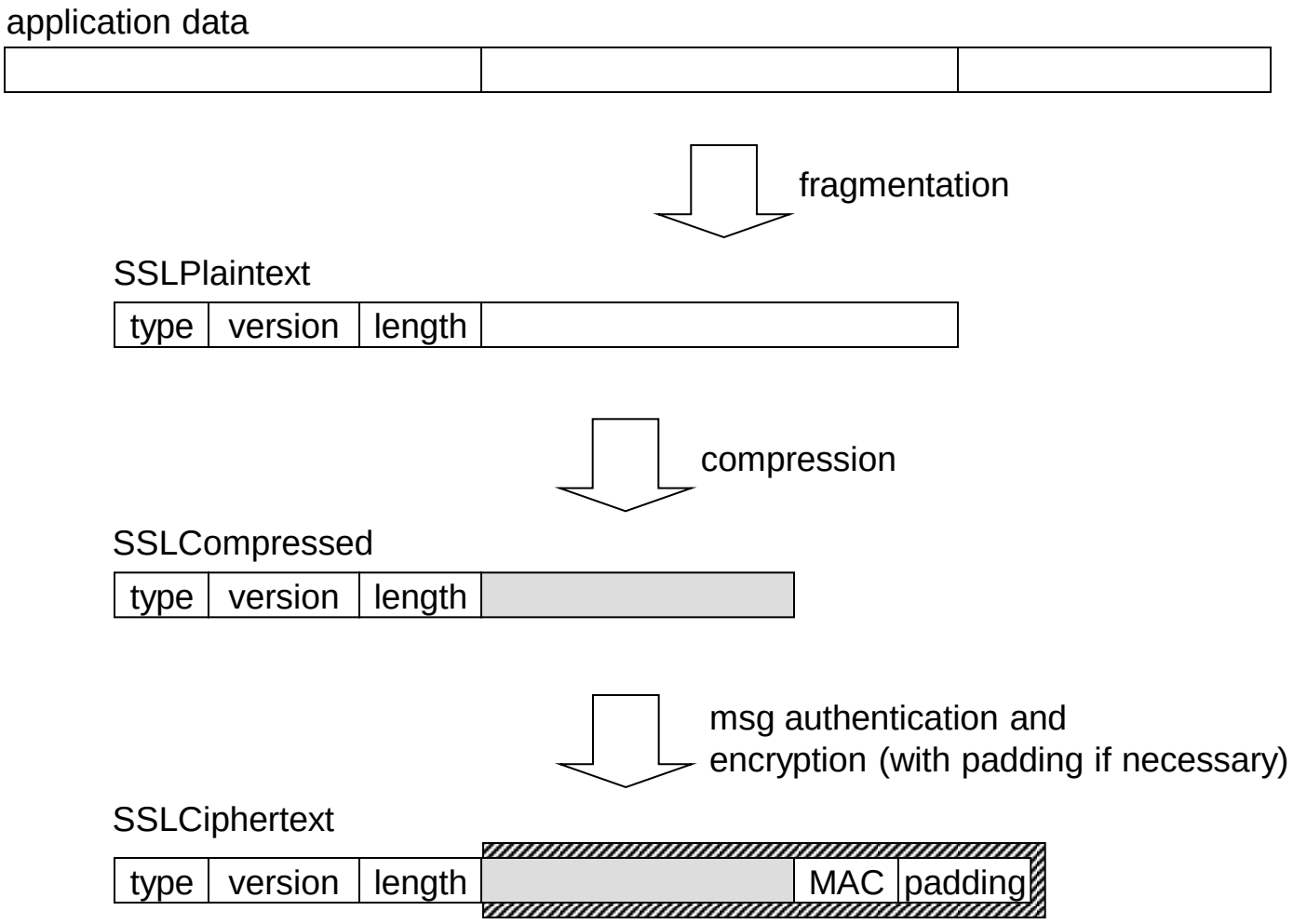
Generarea cheilor criptografice

master_secret = MD5(pre_master_secret + SHA('A' + pre_master_secret + ClientHello.random + ServerHello.random)) + MD5(pre_master_secret + SHA('BB' + pre_master_secret + ClientHello.random + ServerHello.random)) + MD5(pre_master_secret + SHA('CCC' + pre_master_secret + ClientHello.random + ServerHello.random));

key_block = MD5(master_secret + SHA('A' + master_secret + ServerHello.random + ClientHello.random)) + MD5(master_secret + SHA('BB' + master_secret + ServerHello.random + ClientHello.random)) + MD5(master_secret + SHA('CCC' + master_secret + ServerHello.random + ClientHello.random)) + [...];

client_write_MAC_secret[CipherSpec.hash_size]
server_write_MAC_secret[CipherSpec.hash_size]
client_write_key[CipherSpec.key_material]
server_write_key[CipherSPec.key_material]
client_write_IV[CipherSpec.IV_size]
server_write_IV[CipherSpec.IV_size]

SSL Record Protocol



Message Authentication Code

- sumă de control securizată
  - poate fi calculată numai cu ajutorul unei chei secrete
- IETF RFC 2104

  HMAC = hash(MAC_write_secret + pad_2 +

  hash (MAC_write_secret + pad_1 +

  seq_num + length + content));
- hash: MD5, SHA-1
- pad_1 = 0x36 (repetată de 48 de ori în cazul MD5 şi de 40 de ori în cazul SHA-1)
- pad_2 = 0x5C (repetată de 48 de ori în cazul MD5 şi de 40 de ori în cazul SHA-1)

Criptarea datelor

- Cifruri block
  - RC2_40
  - DES_40
  - DES_56
  - 3DES_168
  - IDEA_128
  - Fortezza_80
- Cifruri şir
  - RC4_40
  - RC4_128

Protocolul TLS

- Rezultatul standardizării SSL de către IETF
  - IETF RFC 2246
- Diferenţe faţă de SSL
  - Numărul versiunii
  - Modul de calcul al MAC (XOR în loc de ||)
  - Modul de generare a cheilor criptografice (PRF)
  - Mesaje de alertare
  - Formatul mesajelor (certificate_verify, finished)
  - Modul de completare a datelor (padding)
- Suport pentru algoritmi criptografici de ultimă generație
  - AES, Camellia, GOST
  - SHA-2

Implementare SSL/TLS

- Certificate digitale
  - Server Authentication
  - Client Authentication
- Securizarea tranzacţiilor Web (https)
- “SSLizare” aplicaţii: librării criptografice
  - BSAFE (RSA Security)
  - OpenSSL (www.openssl.org)

Achiziţionare certificat de server

- Autorități de Certificare recunoscute la nivel global (preinstalate în browsere): Verisign, Symantec, Thawte, etc
- Domain validation
  - se verifică doar existenţa domeniului
  - emis foarte repede
- Organisation validation
  - se verifică existenţa organizaţiei şi a domeniului
  - emis în câteva zile
- Extended validation
  - verificări amănunţite cu privire la organizaţie şi domeniu
  - emis în câteva săptămâni

Porturi SSL standard

Serviciu	Port	Descriere
https	443/tcp	http protocol over TLS/SSL
smtps	465/tcp	smtp protocol over TLS/SSL
nntps	563/tcp	nntp protocol over TLS/SSL
sshell	614/tcp	SSLshell
ldaps	636/tcp	ldap protocol over TLS/SSL
ftps-data	989/tcp	ftp protocol, data, over TLS/SSL
ftps	990/tcp	ftp, control, over TLS/SSL
telnets	992/tcp	telnet protocol over TLS/SSL
imaps	993/tcp	imap4 protocol over TLS/SSL
ircs	994/tcp	irc protocol over TLS/SSL
pop3s	995/tcp	pop3 protocol over TLS/SSL

Atacuri împotriva SSL/TLS

- Renegotiation attack
- RC4 attacks
- BEAST attack
- CRIME attack
- Heartbleed
- ChangeCipherSpec injection attack
- POODLE attack against TLS
- Protocol downgrade

➢ Din motive de securitate, se recomandă să se dezactiveze suportul pentru SSL și să se folosească numai TLS