CSE509: (Intro to) Systems Security

Fall 2012

Invited Lecture by Vyas Sekar

SSL

Real Life^2: Secure Communication

- Secure Socket Layer (SSL)
  - Transport layer protocol
- IP Security (IPSec)
  - Network layer protocol

SSL

- Transport layer security
  - Provides confidentiality, integrity, authentication of endpoints
  - Developed by Netscape for WWW browsers and servers
- Internet protocol version: TLS
  - Almost identical to SSL
  - RFC 4346 (ver. 1.1)

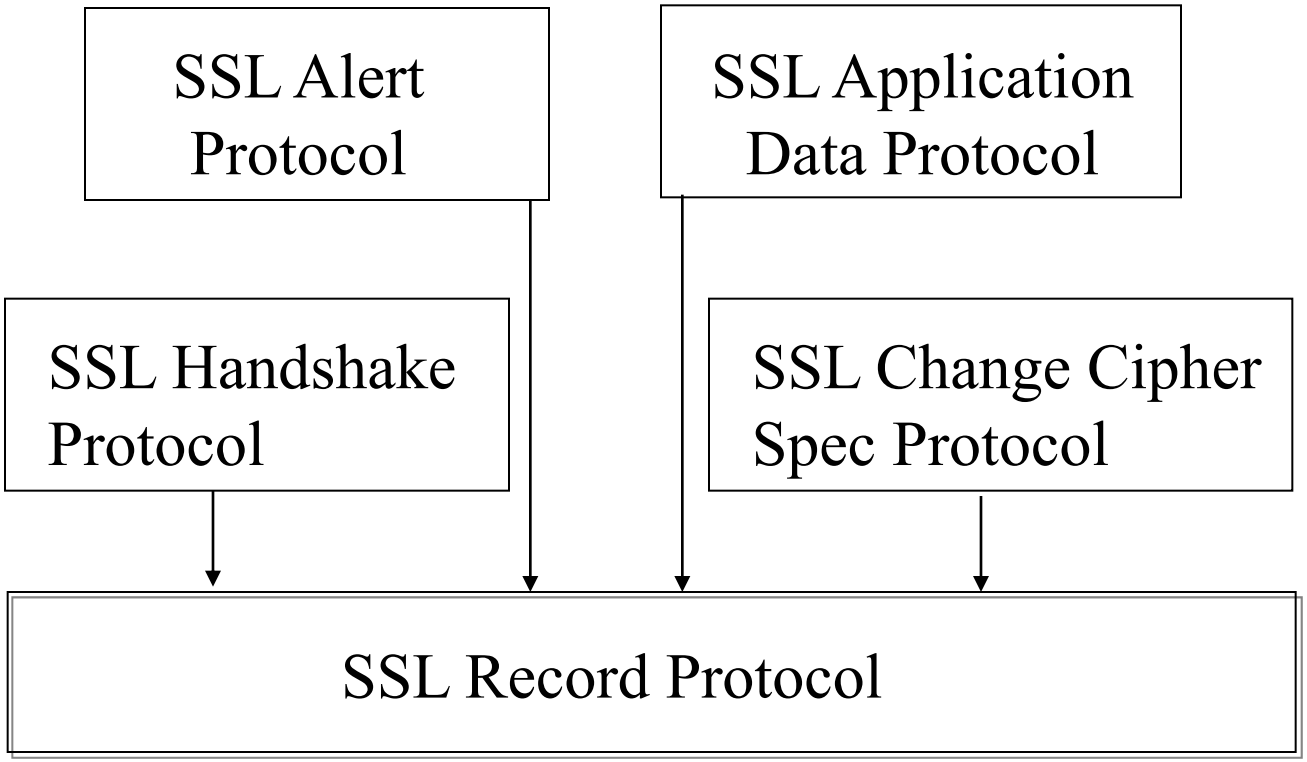
SSL Session

- Association between two peers
  - May have many associated connections
  - Information for each association:
    - Unique session identifier
    - Peer's X.509v3 certificate, if needed
    - Compression method
    - Cipher spec for cipher and MAC
    - "Master secret" shared with peer (384 bits)

SSL Connection

- Describes how data exchanged with peer
- Information for each connection
  - Random data
  - Write keys (used to encipher data)
  - Write MAC key (used to compute MAC)
  - Initialization vectors for ciphers, if needed
  - Sequence numbers

Structure of SSL



Supporting Crypto

- Initial phase: PK system exchanges keys
  - Messages enciphered using classical ciphers, check-summed using cryptographic checksums
  - Only certain combinations allowed
    - Depends on algorithm for interchange cipher
  - Interchange algorithms: e.g., RSA, DH, Fortezza (D.O.D)

RSA: Cipher, MAC Algorithms

Interchange cipher	*Classical cipher*	*MAC Algorithm*
RSA, key ≤ 512 bits	*none*	MD5, SHA
	RC4, 40-bit key	MD5
	RC2, 40-bit key, CBC mode	MD5
	DES, 40-bit key, CBC mode	SHA
RSA	*None*	MD5, SHA
	RC4, 128-bit key	MD5, SHA
	IDEA, CBC mode	SHA
	DES, CBC mode	SHA
	DES, EDE mode, CBC mode	SHA

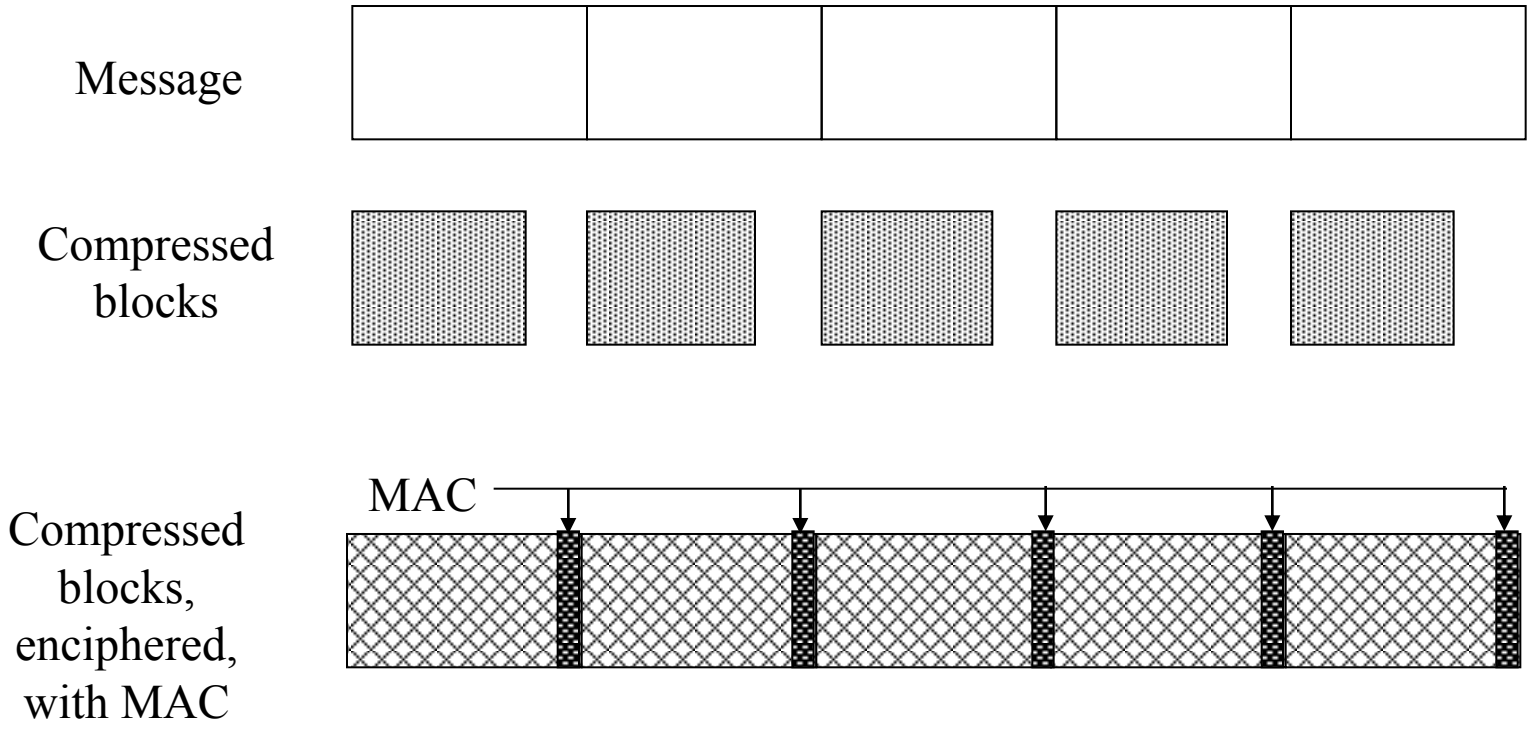
Fortezza: Cipher, MAC Algorithms

Interchange cipher	*Classical cipher*	*MAC Algorithm*
Fortezza key exchange	*none*	SHA
	RC4, 128-bit key	MD5
	Fortezza, CBC mode	SHA

Digital Signatures

- RSA
  - Concatenate MD5 and SHA hashes
- Fortezza
  - Compute SHA hash

SSL Record Layer



Record Protocol Overview

- Lowest layer, taking messages from higher
  - Max block size 16,384 bytes
  - Bigger messages split into multiple blocks
- Construction
  - Block b compressed; call it b_c
  - MAC computed for b_c
    - If MAC key not selected, no MAC computed
  - b_c, MAC enciphered
    - If enciphering key not selected, no enciphering done
  - SSL record header pre-pended

SSL MAC Computation

- Symbols
  - h hash function (MD5 or SHA)
  - k_w write MAC key of entity
  - *ipad* = 0x36, *opad* = 0x5C
    - Repeated to block length (from HMAC)
  - *seq* sequence number
  - *SSL_comp* message type
  - *SSL_len* block length
- MAC: $h(k_w||opad||h(k_w||ipad||seq||SSL\_comp||SSL\_len||block))$

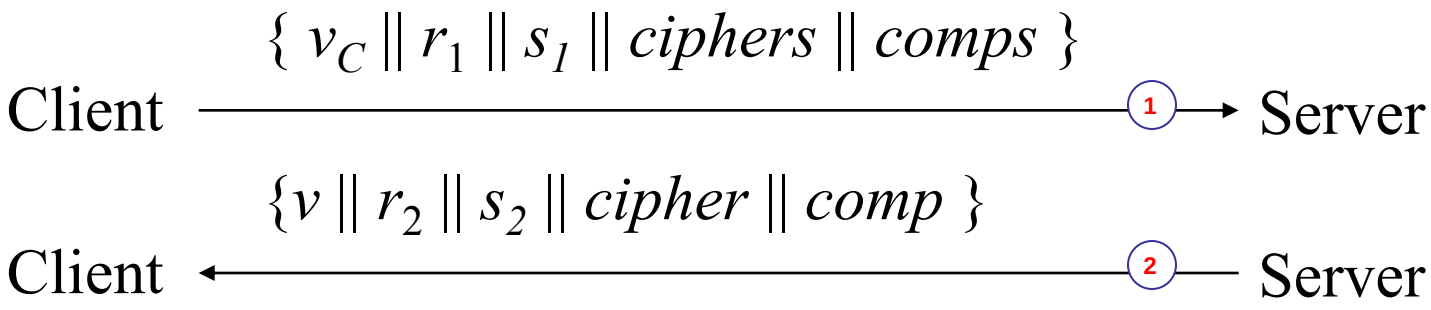
SSL Handshake Protocol

- Used to initiate connection
  - Sets up parameters for record protocol
  - 4 rounds
- Upper layer protocol
  - Invokes Record Protocol
- Note: what follows assumes client, server use RSA as interchange cryptosystem

Overview of Rounds

1. Create SSL client-server connection
2. Server authenticates itself
3. Client validates server, begins key exchange
4. Acknowledgments all around

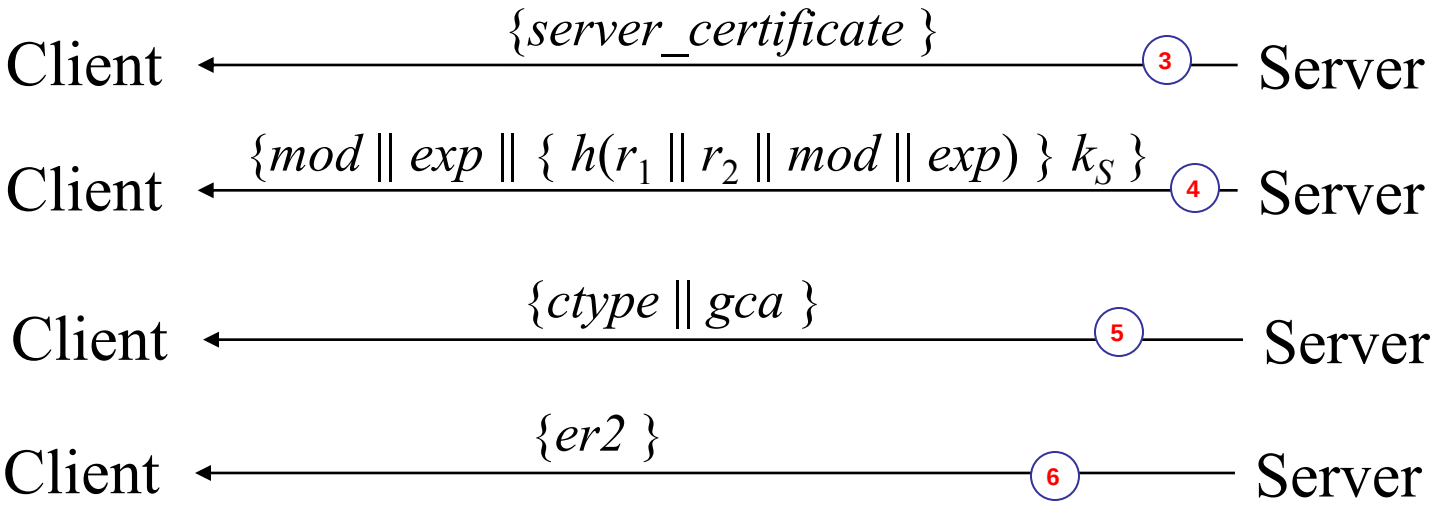
Handshake Round 1: connection



v_C	Client's version of SSL
v	Highest version of SSL that Client, Server both understand
r_1, r_2	nonces (timestamp and 28 random bytes)
s_1	Current session id (0 if new session)
s_2	Current session id (if $s_1 = 0$, new session id)
ciphers	Ciphers that client understands
comps	Compression algorithms that client understand
cipher	Cipher to be used
comp	Compression algorithm to be used

Note: we assume client and server use RSA as interchange cryptosystem

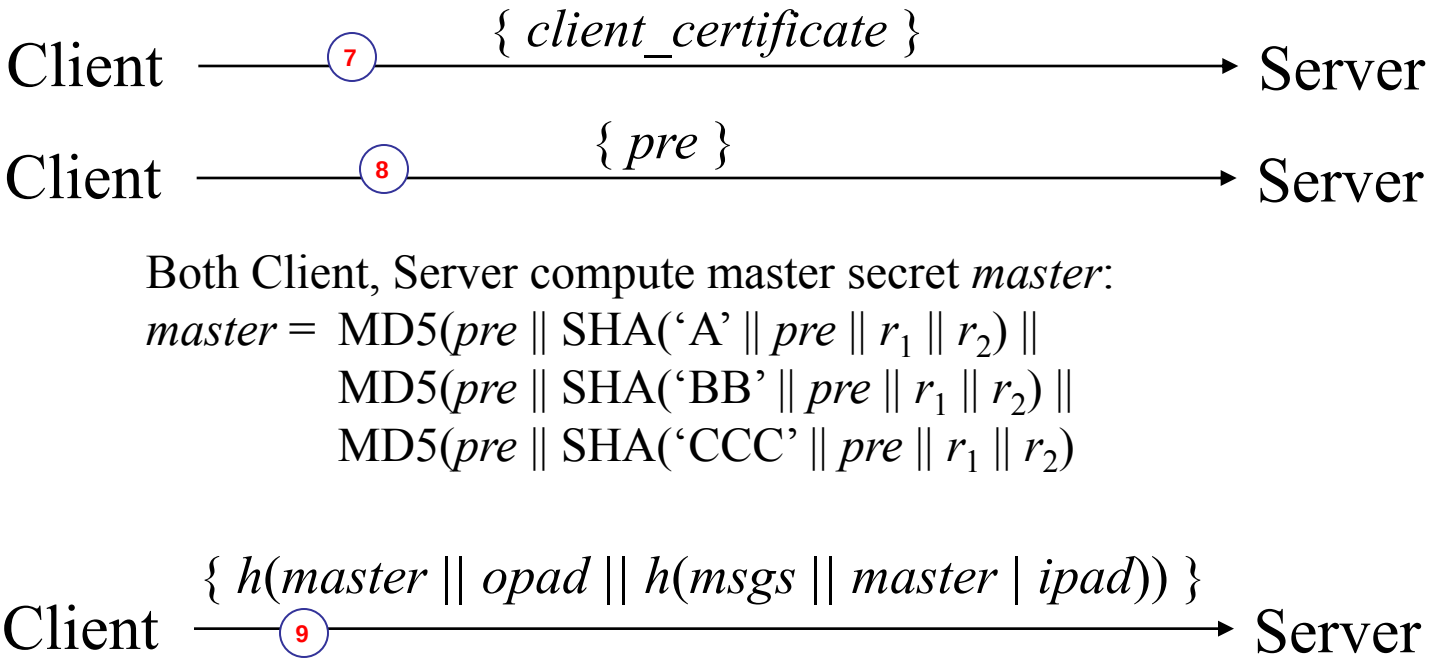
Handshake Round 2: server authentication



Note: if Server not to authenticate itself, only last message sent; third step omitted if Server does not need Client certificate (**mutual auth not default !!!**)

k_S	Server's private key
ctype	Certificate type requested (by cryptosystem)
gca	Acceptable certification authorities
er2	End round 2 message
mod,exp	For a new temporary key pair (not the one associated with certificate)

Handshake Round 3: key exchange



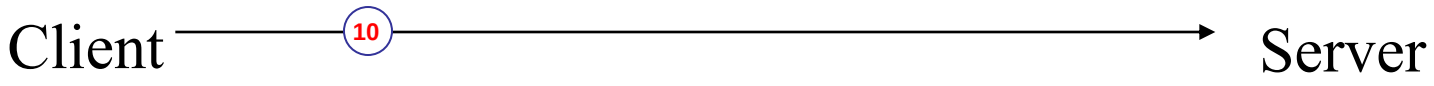
Both Client, Server compute master secret *master*:

master = MD5(*pre* || SHA('A' || *pre* || r_1 || r_2) ||
MD5(*pre* || SHA('BB' || *pre* || r_1 || r_2) ||
MD5(*pre* || SHA('CCC' || *pre* || r_1 || r_2)

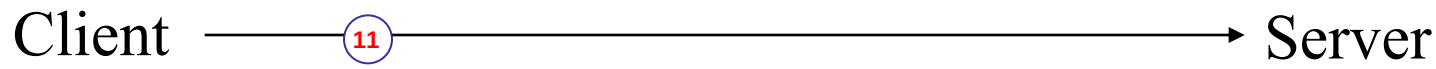
msgs	Concatenation of previous messages sent/received in this handshake
opad, *ipad*	As above
{*pre*}	Encrypted with mod/exp from previous slide

Handshake Round 4: acknowledgements

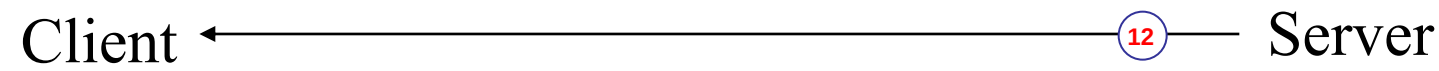
Client sends "change cipher spec" message using that protocol



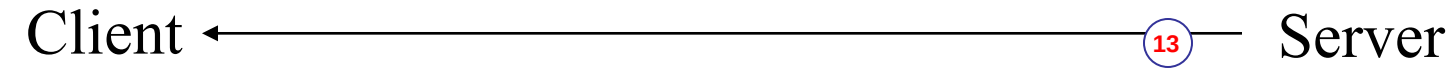
$\{ h(master \,||\, opad \,||\, h(msgs \,||\, 0x434C4E54 \,||\, master \,||\, ipad\,)) \}k_{cipher}$



Server sends "change cipher spec" message using that protocol



$\{ h(master \,||\, opad \,||\, h(msgs \,||\, 0x53525652 \,||\, master \,||\, ipad)) \}\, k_{cipher}$



msgs Concatenation of messages sent/received this handshake in *previous* rounds (does not include these messages)

opad, *ipad*, *master* As above

SSL Change Cipher Spec Protocol

- Send single byte
- In handshake, new parameters considered "pending" until this byte received
  - Old parameters in use, so cannot just switch to new ones

SSL Alert Protocol

- Closure alert
  - Sender will send no more messages
  - Pending data delivered; new messages ignored
- Error alerts
  - Warning: connection remains open
  - Fatal error: connection torn down as soon as sent or received

SSL Alert Protocol Errors

- Always fatal errors:
  - unexpected_message, bad_record_mac, decompression_failure, handshake_failure, illegal_parameter
- May be warnings or fatal errors:
  - no_certificate, bad_certificate, unsupported_certificate, certificate_revoked, certificate_expired, certificate_unknown

SSL Application Data Protocol

- Passes data from application to SSL Record Protocol layer

Real World

- Toolkits
  - http://www.openssl.org
- Certificate Authorities (300+)
  - http://www.verisign.com
  - http://www.thawte.com
  - http://www.instantssl.com
  - http://www.entrust.com

Vulnerabilities

- Virtual server issues
- Rogue CAs
- Useless warning messages
- etc.