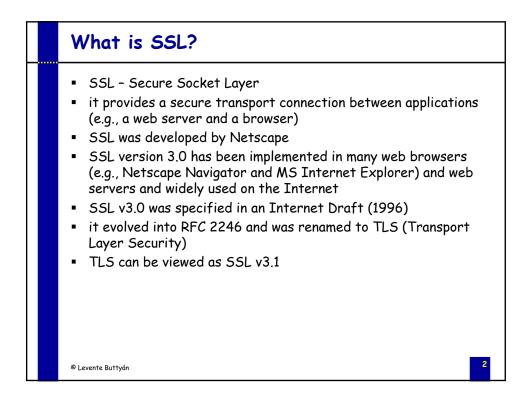
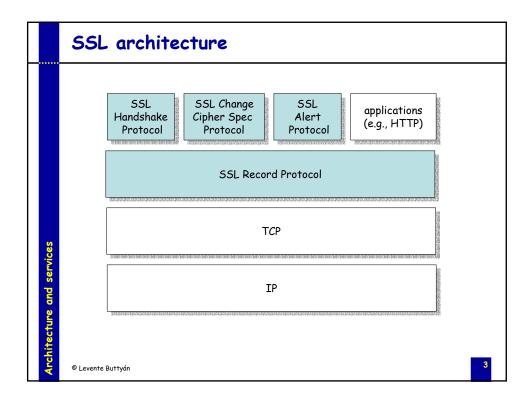
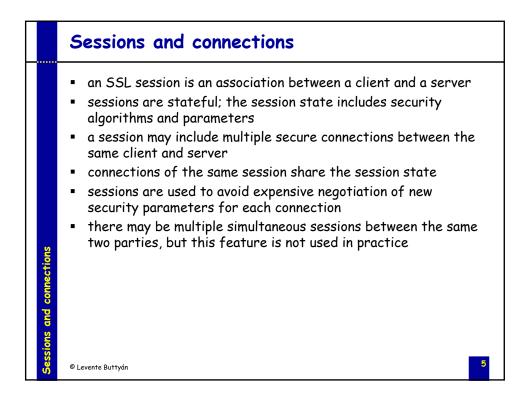
## SSL - Secure Socket Layer

- architecture and services
- sessions and connections
- SSL Record Protocol
- SSL Handshake Protocol
- key exchange alternatives
- analysis of the SSL Record and Handshake Protocols
- SSL vs. TLS



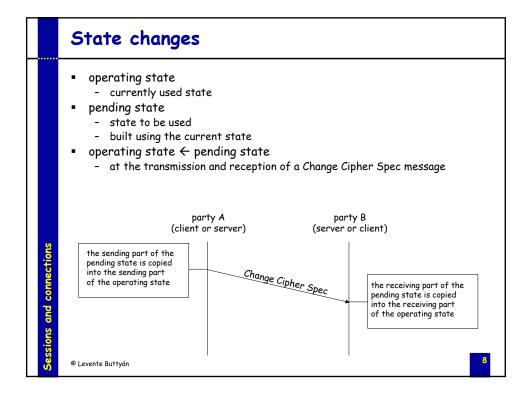


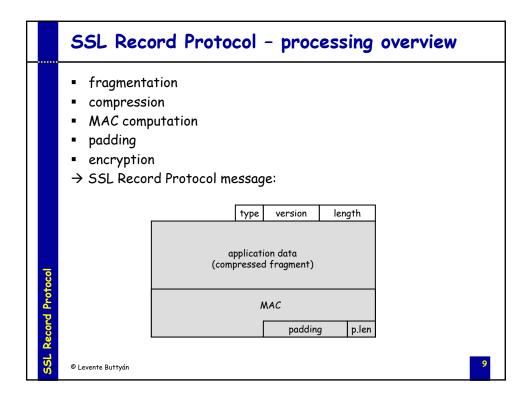
	SSL components
Architecture and services	<ul> <li>SSL Handshake Protocol <ul> <li>negotiation of security algorithms and parameters</li> <li>key exchange</li> <li>server authentication and optionally client authentication</li> </ul> </li> <li>SSL Record Protocol <ul> <li>fragmentation</li> <li>compression</li> <li>message authentication and integrity protection</li> <li>encryption</li> </ul> </li> <li>SSL Alert Protocol <ul> <li>error messages (fatal alerts and warnings)</li> </ul> </li> <li>SSL Change Cipher Spec Protocol <ul> <li>a single message that indicates the end of the SSL handshake</li> </ul> </li> </ul>



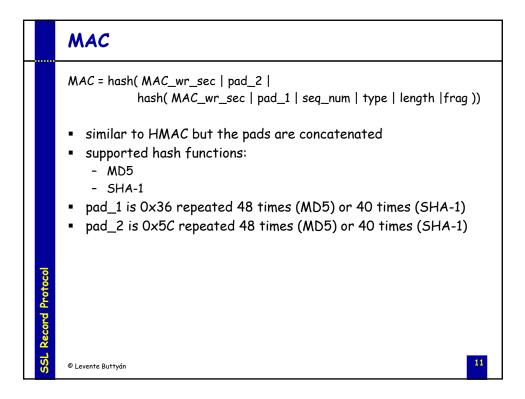
	session identifier
	<ul> <li>arbitrary byte sequence chosen by the server to identify the session</li> </ul>
•	peer certificate
	- X509 certificate of the peer
	- may be null
•	compression method
•	cipher spec
	<ul> <li>bulk data encryption algorithm (e.g., null, DES, 3DES,)</li> <li>MAC algorithm (e.g., MD5, SHA-1)</li> </ul>
	<ul> <li>cryptographic attributes (e.g., hash size, IV size,)</li> </ul>
-	master secret
	- 48-byte secret shared between the client and the server
•	is resumable
	<ul> <li>a flag indicating whether the session can be used to initiate new connections</li> </ul>
-	connection states

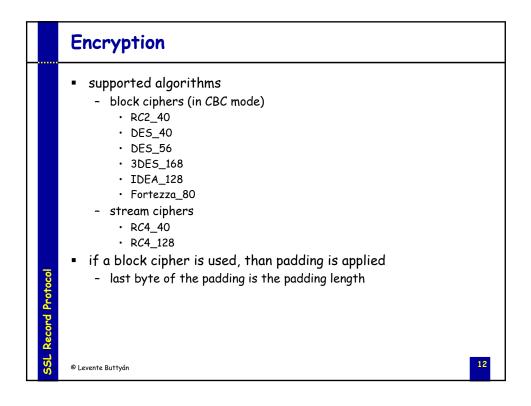
	Connection state
essions and connections	<ul> <li>server and client random <ul> <li>random byte sequences chosen by the server and the client for every connection</li> </ul> </li> <li>server write MAC secret <ul> <li>secret key used in MAC operations on data sent by the server</li> </ul> </li> <li>client write MAC secret <ul> <li>secret key used in MAC operations on data sent by the client</li> </ul> </li> <li>server write key <ul> <li>secret encryption key for data encrypted by the server</li> </ul> </li> <li>client write key <ul> <li>secret encryption key for data encrypted by the client</li> </ul> </li> <li>secret encryption key for data encrypted by the client</li> </ul> <li>initialization vectors <ul> <li>an IV is maintained for each encryption key if CBC mode is used</li> <li>initialized by the SSL Handshake Protocol</li> <li>final ciphertext block from each record is used as IV with the following record</li> </ul> </li> <li>sending and receiving sequence numbers <ul> <li>sequence numbers are 64 bits long</li> <li>reset to zero after each Change Cipher Spec message</li> </ul> </li>
Ses	© Levente Buttyán 7

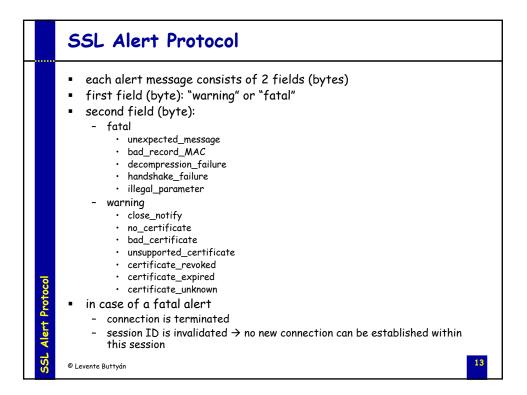




	Header
Record Protocol	<ul> <li>type         <ul> <li>the higher level protocol used to process the enclosed fragment</li> <li>possible types:                 <ul> <li>change_cipher_spec</li> <ul> <li>alert</li> <li>handshake</li> <ul> <li>application_data</li></ul></ul></ul></li></ul></li></ul>
SSL	© Levente Buttyán



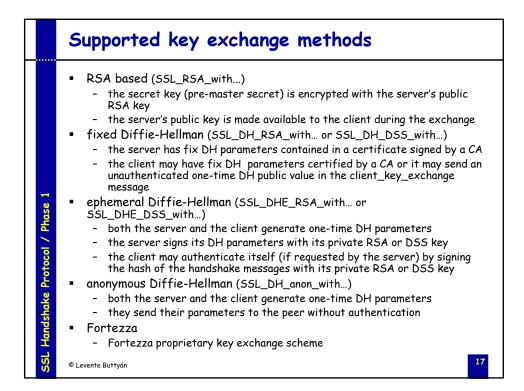




client	se	rver
	client_hello	Phase 1: Negotiation of the session ID, key
-	server_hello	exchange algorithm, MAC algorithm, encryption algorithm, and exchange of initial random numbers
	certificate	<u>Phase 2</u> : Server may send its certificate and key exchange message, and it may request the client to send a certificate. Server signals end of hello phase.
	server_key_exchange	
	certificate_request	
-	server_hello_done	
	certificate	Phase 3: Client sends certificate if requested and
	client_key_exchange	may send an explicit certificate verification
	certificate_verify	<ul> <li>message. Client always sends its key exchange message.</li> </ul>
	change_cipher_spec	
	finished	
-	change_cipher_spec	<u>Phase 4</u> : Change cipher spec and finish handshake
	finished	

	Client hello
L Handshake Protocol / Phase 1	<ul> <li>client_version <ul> <li>the highest version supported by the client</li> </ul> </li> <li>client_random <ul> <li>current time (4 bytes) + pseudo random bytes (28 bytes)</li> </ul> </li> <li>session_id <ul> <li>empty if the client wants to create a new session, or</li> <li>the session ID of an old session within which the client wants to create the new connection</li> </ul> </li> <li>cipher_suites <ul> <li>list of cryptographic options supported by the client ordered by preference</li> <li>a cipher suite contains the specification of the <ul> <li>key exchange method, the encryption and the MAC algorithm</li> <li>the algorithms implicitly specify the hash_size, IV_size, and key_material parameters (part of the Cipher Spec of the session state)</li> <li>exmaple: SSL_RSA_with_3DES_EDE_CBC_SHA</li> </ul> </li> <li>compression_methods <ul> <li>list of compression methods supported by the client</li> </ul> </li> </ul></li></ul>
SSL	© Levente Buttyán 15

	Server hello
Handshake Protocol / Phase 1	<ul> <li>server_version <ul> <li>min(highest version supported by client, highest version supported by server)</li> </ul> </li> <li>server_random <ul> <li>current time + random bytes</li> <li>random bytes must be independent of the client random</li> </ul> </li> <li>session_id <ul> <li>session ID chosen by the server</li> <li>if the client wanted to resume an old session: <ul> <li>server checks if the session is resumable</li> <li>if so, it responds with the session ID and the parties proceed to the finished messages</li> </ul> </li> <li>if the client wanted a new session <ul> <li>server generates a new session ID</li> </ul> </li> <li>cipher_suite</li> <li>single cipher suite selected by the server from the list given by the client</li> </ul> </li> </ul>
landsh	<ul> <li>compression_method</li> <li>single compression method selected by the server</li> </ul>
SSL F	© Levente Buttyán



•	<ul> <li>certificate</li> <li>required for every key exchange method except for anonymous DH</li> <li>contains one or a chain of X.509 certificates (up to a known root CA)</li> <li>may contain <ul> <li>public RSA key suitable for encryption, or</li> <li>public RSA or DSS key suitable for signing only, or</li> <li>fix DH parameters</li> </ul> </li> </ul>
•	<ul> <li>server_key_exchange</li> <li>sent only if the certificate does not contain enough information to complete the key exchange (e.g., the certificate contains an RSA signing key only)</li> <li>may contain <ul> <li>public RSA key (exponent and modulus), or</li> <li>DH parameters (p, g, public DH value), or</li> <li>Fortezza parameters</li> </ul> </li> <li>digitally signed <ul> <li>if DSS: SHA-1 hash of (client_random   server_random   server_params) is signed</li> <li>if RSA: MD5 hash and SHA-1 hash of (client_random   server_random   server_random   server_params) are concatenated and encrypted with the private RSA key</li> </ul> </li> </ul>

