IK2206 – Internet Security and Privacy
Chapter 19 – SSL/TLS

19.2 Using TCP

- What is the benefit of running on TCP?

  TCP is a reliable layer-4 protocol that takes over the part of resending lost packets. SSL/TLS is much simpler because it doesn't have to worry about timing out and retransmitting data.

19.3 Quick History

19.4 SSL/TLS Basic Protocol

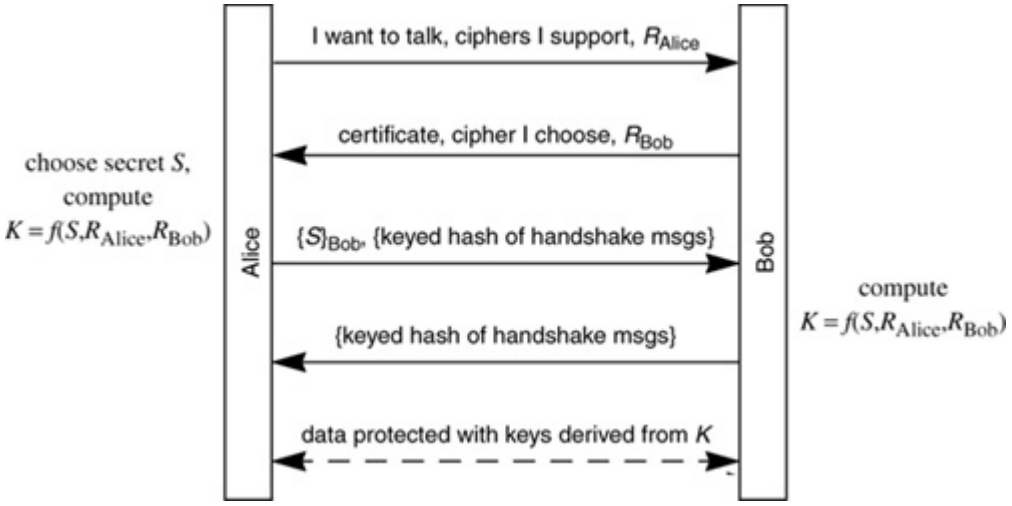
- How is the basic handshake performed?

  Message 1: Alice informs Bob that she wants to talk with him. She gives him a list of cryptographic algorithms she supports, along with a random number R_{Alice}.

  Message 2: Bob sends Alice his public key certificate and a random number R_{Bob}. Bob responds with one of the crypto algorithms he also supports.

  Message 3: Alice calculates the master key from S, R_A, R_B and chooses a random number S (=pre-master secret) and sends it - encrypted with Bob's private key - to Bob. She also sends a hash of the master secret K and hand-shake messages.

  Message 4: Bob calculates the master key from S, R_A , R_B and proves he knows the session keys.



- Why are so many keys used?

  SSL/TLS uses a total of six keys: One for encryption, integrity and IV in each communication direction. The reason for taking different keys for different security tasks is that if one key gets compromised, an attacker is limited to a particular security boundary. **(???)**

19.5 Session Resumption

- How can a session be resumed?

  The client can transmit a session-id with the initialization message. If the server remembers the session, it continues with the negotiation of the cipher suites. In case the server doesn't remember the session-id, it sends a new one (or none) back to indicate that a new session has to be initialized.

- Why do we want to resume a session, rather than just creating a new?

  The resumption of an existing session brings a performance advantage. If a session can be resumed, the public key portion of the handshake can be avoided. (Public key operations are very costly). If we had no session resumption in SSL/TLS, each connection would calculate six new RSA keys…

19.6 Computing the Keys

- Why are the keys shuffled with the two Rs?

  The master key is shuffled with the two R's to produce the six keys.

19.7 Client Authentication

- What is the problem that makes client authentication hard?

  Clients identify servers using public key certificates that were issued for a particular server. If we wanted to authenticate clients against servers, each client needs to get a certificate as well. Technically, this isn't hard to realize – but it is a huge administrative effort to deploy/maintain client certificates.

19.8 PKI as Deployed by SSL

- How did you get the trusted root CAs for your web client?

  The vendor of the browser (e.g. Microsoft IE) has defined a list with trustworthy CAs.

19.9 Version Numbers (READ BRIEFLY)

- Why is the lack of integrity protection of the client-hello an issue with SSLv2?

  Since SSLv2 doesn't integrity-protect the client-hello message, it's possible for an active attacker to modify a v3 client-hello by changing the version number from 768 (SSLv3) or 769 (TLS) to 2. SSLv2 shouldn't be used today since it has some security flaws.

19.10 Negotiating Cipher Suites

- Who decides what cipher suite to use?

  Alice sends a list of cipher suites to Bob. Bob chooses all suites that he also supports and sends them back to Alice. Alice finally decides about the cipher suite to use. (This negation is silly; Bob could decide which cipher suite to use).

  Indeed one of the enhancements in SSLv3 is that Bob does make the choice, from the list Alice sent.

19.11 Negotiating Compression Method

19.12 Attacks Fixed in v3

- What is a downgrade attack?

  An attacker can remove cipher suites with strong encryption from the list of requested cipher suites, causing Alice and Bob to agree upon a weaker cipher.

- What is a truncation attack?

  SSLv3 introduced a *finished* message which is sent to close the connection. SSLv2 used TCP closing methods to disconnect. TCP close message are not cryptographically protected and can be used by attackers to close SSL connections.

19.13 Exportability (READ BRIEFLY)

- How has exportability affected the design of SSL/TLS?

  The designers of SSL had to find a way to 1) keep SSL secure and 2) comply with the US export regulations. They implemented several tricks to make SSL secure even though they just use 40bit symmetric with 512bit asymmetric keys.

- How has the exportability provided perfect forward secrecy in v3?

  In an authenticated key-agreement protocol that uses public key cryptography, perfect forward secrecy (or PFS) is the property that ensures that a session key derived from a set of long-term public and private keys will not be compromised if one of the (long-term) private keys is compromised in the future. (http://en.wikipedia.org/wiki/Perfect_forward_secrecy)

  In SSL, once Bob forgets the ephemeral private key, it would require breaking the 512bit ephemeral public key or the 40bit encryption key in order to decrypt the conversation. With non-exportable cipher suites, someone obtaining Bob's long-term private key would be able to directly decrypt previous conversations.

  A cryptographic key is called ephemeral if it is generated for each execution of a key establishment process. (http://en.wikipedia.org/wiki/Ephemeral_key)

- What is Step-Up, and how does it work?

???

19.14 Encoding (READ BRIEFLY)

You should have an idea of what goes into these messages, but you are not required to remember the exact formats of every message.

19.14.1 Encrypted Records

- How are message records protected?

  Integrity-protected using HMAC (based on MD5 or SHA1) and encrypted using a block cipher in CBC mode.