Working principle, attacks and defenses of SSL/TLS

Tom Petersen

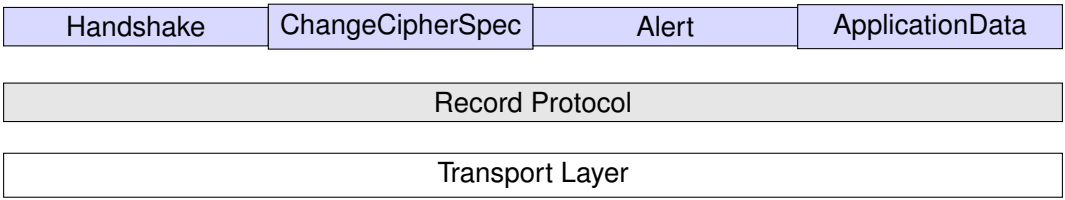
University of Hamburg
Department of Informatics

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Why is it important to know TLS?

- security protocol above the transport layer of the OSI model
- authentication of communication partners, encryption and integrity check of sent messages
- used by HTTPS, SFTP, OpenVPN...
- "most used security protocol today"

Protocol hierarchie



- consists of 5 subprotocols
- on the lower layer -> Record protocol which is responsible for encrypting and authenticating messages and sending these over the transport layer e.g. TCP protocol
- on the upper layer -> - handshake protocol: choosing ciphers and exchanging keys when starting a connection - change cipher spec protocol: telling the partner to use the ciphers agreed on from now on - alert protocol: sending error messages when an error occurs, for example when a message has a bad MAC - application data protocol: sending the data from higher layer protocols or applications, when the connection has been established

Didactic reduction (?)

- concrete methods (like protocols) should be abstracted
- long-lasting principles, which are used in multiple fields
  - hybrid cryptosystems: using symmetric ciphers for encrypting the communication (faster than asymmetric ciphers) and asymmetric cryptography to exchange a symmetric key
  - authenticated key exchange: authenticating the public key of the receiver to prevent man-in-the-middle-attacks
  - side-channel attacks: getting information about the encrypted data or key through other channels like power consumption or time measurement
  - cryptographic principles: just as an example: in the first TLS version the initialisation vector for AES in CBC mode was not unpredictable like it should be. Instead the last block of the previous message was used as IV which led to a chosen-plaintext-attack against TLS.

-> didactic reduction is a general recommendation for teaching, one concrete approach for learning is discovery learning

Discovery learning with simulations

Discovery learning (sometimes referred to as exploratory learning)
= learning by interacting with the world by exploring and manipulating objects and performing experiments. One approach for discovery learning is simulation-based learning.
- useful for complex topics, which are hard to understand with other learning materials
Simulations = interactive computer programs modelling activities, which enable us to observe normally hidden processes.
example ?

Developed application

- Simulating TLS
- extendable (requirement) -> with this in mind i developed the application plugin-ready
- general application for simulating two party protocol flows (usually client and server)
- observable internal party states to understand the processes happening during a protocol flow
- interactive to see consequences of messages or changes of messages

Live demo

- cipher suite choosing: cipher suite tells you which cryptographic primitives (key exchange algorithm, symmetric cipher and hash function) is used for securing the communication
- server/client views: show the values of internal state fields of client and server
- start connection
- message and message details view: starts with a client hello message, you can see the sent bytes and there is a detailed view with the values of the single fields of the message
- info view: maybe this little information-icons have catched your eyes. They provide brief explanations for every field, mostly just citing the specification. Show an example.
- finish handshake -> connection established (client server agreed on crypto algorithms and have the same key material)
- send data -> successful
- edit bytes and watch error occur
- echo plugin (?)

Conclusion

—TLS is great for using it in university teaching

Unimplemented in TLS plugin

- certificate validation
- session resumption
- client authentication
- TLS extensions

Implement other protocols