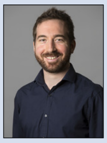
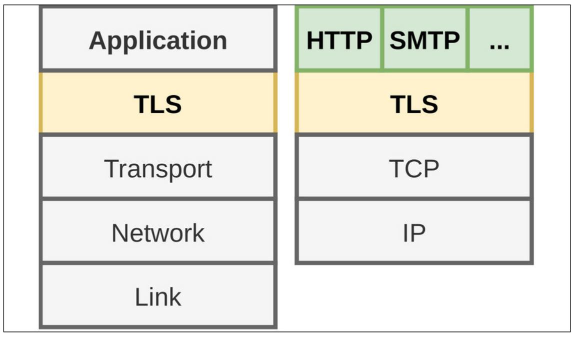


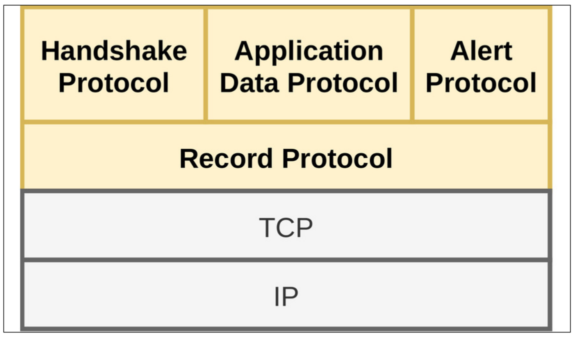
Pascal<br>Knecht

| | |
|---|---|
| Graduate Candidate | Pascal Knecht |
| Examiner | Prof. Dr. Andreas Steffen |
| Co-Advisor | Dr. Ralf Hauser, PrivaSphere AG, Zürich, ZH |
| Subject Area | Sicherheit |

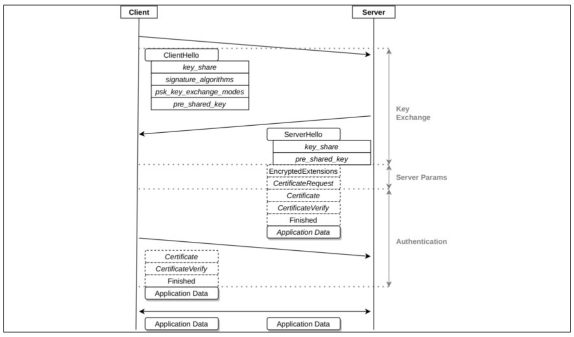
# TLS 1.3 Stack for strongSwan



Network Protocol Stack
Own presentment



TLS Protocols
Own presentment



TLS 1.3 Handshake
Own presentment

Introduction: The Transport Layer Security (TLS) protocol secures network connections between a client and a server. It encrypts and authenticates data from higher-level protocols such as the Hypertext Transfer Protocol (HTTP), and guarantees that the information transmitted remains confidential and keeps its integrity. The most widely used TLS version today still is version 1.2 released in 2008 (RFC 5246), even though 1.3 was released in 2018 (RFC 8446). The strongSwan project maintained by the University of Applied Sciences Rapperswil (HSR) is an open-source IPsec implementation written in C. strongSwan features its own TLS stack encapsulated in the library libtls. It enables communication-authentication via various EAP authentication methods (TLS, TTLS, PEAP) used to establish an IKEv2 connection. A client-side TLS 1.3 prototype stack was implemented in the preliminary work before this thesis. However, libtls does not yet fully support TLS 1.3 in the sense strongSwan requires.

Objective: The goal of this bachelor thesis is to implement the TLS 1.3 server-side protocol stack, add support for mutual authentication to enable client authentication in a TLS 1.3 handshake and lastly add support for PSK-based session resumption with TLS 1.3. The former two tasks are mandatory features to make the new TLS 1.3 implementation useful for the strongSwan project and the latter is an optional feature. To achieve these goals, it is necessary to integrate new or adapt existing messages that are exchanged between client and server. In addition, TLS 1.3 requires fundamental changes to the cryptographic mechanisms that enable a secure and authenticated encryption.
Until a connection is established, the handshake passes through various states in a state machine. The state machine has considerably changed in the new version, which also implies that the handshake flow and state machine must be adapted. Moreover, the way cryptographic keys are generated and derived by each peer has fundamentally changed, this also needs to be addressed in this work.

Result: TLS 1.3 was successfully implemented and provides a server-side stack and mutual authentication. The TLS 1.3 client-side stack, which was implemented already in the preliminary study term project, was improved significantly. Additionally, smaller but important features such as support for KeyUpdate or HelloRetryRequest messages were implemented. The client and server implementations have been extensively tested against each other and also with external servers and tools such as OpenSSL. The mandatory goals were achieved. The optional goal, the PSK-based session resumption, was not implemented fully due to time constraints but the foundation has been laid: The cryptographic logic encapsulated in the HKDF implementation is able to provide all the necessary secrets. However, the communication protocol and logic implementation remains open to further work. Nevertheless, the current implementation is usable and provides TLS 1.3 secured communication for the strongSwan project.