Universität Hamburg

Fachbereich Informatik

Entwurf vom
11. Mai 2015

Exposé

Bachelorarbeit über TLS (1.3)

vorgelegt von

Tom Petersen

geb. am 13. Dezember 1990 in Hannover

Matrikelnummer 6359640

Studiengang Informatik

eingereicht am 11. Mai 2015

1 SSL und TLS - ein Überblick

SSL (Secure Socket Layer) bzw. TLS[1] (Transport Layer Security) ist ein zustandsbehaftetes Protokoll, das auf dem TCP-Protokoll der Transportschicht des TCP/IP-Protokollstapels aufbaut[2]. Es bildet also eine Schicht zwischen Transport- und Anwendungsschicht. Viele Protokolle der Anwendungsschicht nutzen TLS zur sicheren Datenübertragung, so beispielsweise HTTPS oder FTPS.

SSL wurde von der Firma Netscape entwickelt und nachdem es starke Verbreitung gefunden hatte, durch die IETF als TLS 1.0 in RFC 2246 standardisiert (TLS 1.0 entspricht hierbei SSL 3.1). Aktuell ist die TLS-Version 1.2 und an Version 1.3 wird gearbeitet. [Sch09]

Hauptaufgaben von TLS sind Authentifikation der Kommunikationspartner, symmetrische Verschlüsselung der Kommunikation sowie die Sicherstellung der Integrität der übertragenen Nachrichten. Die hierbei verwendeten kryptographischen Verfahren werden erst zu Beginn der Kommunikation festgelegt. [Eck13]

TLS 1.0 RFC 2246 - http://tools.ietf.org/html/rfc2246

TLS 1.1 RFC 4346 - http://tools.ietf.org/html/rfc4346

TLS 1.2 RFC 5246 - http://tools.ietf.org/html/rfc5246

TLS Extensions RFC 3546 - http://tools.ietf.org/html/rfc3546,
RFC 3466 - http://tools.ietf.org/html/rfc4366,
RFC 6066 - http://tools.ietf.org/html/rfc6066

TLS 1.3 Draft - https://tools.ietf.org/html/draft-ietf-tls-tls13-05

1. Im weiteren Verlauf dieser Arbeit wird der Einfachheit lediglich von TLS gesprochen, es ist jedoch ebenso SSL gemeint. Bei etwaigen Unterschieden wird explizit auf diese eingegangen werden.
2. Es gibt auch DTLS (Datagram Transport Layer Security), ein auf TLS basierendes Protokoll, dass auch per UDP Daten übertragen kann

2 Funktionsweise und Teilprotokolle

Grafik der TLS-Protokolle

TLS besteht selbst aus zwei Schichten. In der unteren Schicht befindet sich das *Record-Protokoll*, das die Daten von den Teilprotokollen der oberen Schicht entgegennimmt und je nach aktuell verhandelten krypographischen Funktionen verschlüsselt und signiert, sowie Anwendungsdaten fragmentiert.

HMAC oder was oder wie

In der oberen Schicht sind vier Teilprotokolle spezifiziert: *Handshake-, ChangeCipherSpec-, Alert-* und *ApplicationData-Protokoll*.

?

Das *Handshake-Protokoll* dient zur Vereinbarung kryptographischer Verfahren (CipherSuites) und zur Aushandlung eines Schlüssels für die symmetrische Verschlüsselung der später gesendeten Daten. Der Handshake kann folgendermaßen ablaufen (abhängig von optionalem Clientzertifikat, Preshared Key, erneute Verbindung, ...):

- –> client_hello mit (verfügbare symmetrische Verschlüsselungsverfahren/kryptographische Hashfunktionen/ Schlüsselaustauschverfahren) und Zufallszahl_Client
- <– server_hello mit gewählten Verfahren und Zufallszahl_Server
- <– Server Zertifikat (inklusive öffentlichem Schlüssel des Servers, meist nach X.509v3)
- Client: Zertifikatverifikation
- –> Generierung und Senden von PreMasterSecret verschlüsselt mit öffent. Schlüssel des Servers (bei RSA) oder Diffie-Hellman-Verfahren
- Client und Server: aus Zufallszahl_Client, Zufallszahl_Server und PreMasterSecret wird das MasterSecret berechnet
- –> Handshakeabschluss
- <– Handshakeabschluss

Das *ChangeCipherSpec-Protokoll* dient dazu, die vereinbarten kryptographischen Verfahren zu ändern. Es enthält lediglich eine Nachricht mit dem Wert 1, die für das Übernehmen der während des Handshakes ausgehandelten Verfahren steht. [Eck13]

Das *Alert-Protokoll* dient dazu, auftretende Fehler oder Warnungen zu versenden, die während des Datenaustausches auftreten.

Übersicht

Das *ApplicationData-Protokoll* ist zuständig für das Durchreichen von Anwendungsdaten, die von der Anwendungsschicht gesendet werden sollen.[Sch06]

3 Ciphersuites

4 Angriffe

Literaturverzeichnis

[Eck13] Claudia Eckert. *IT-Sicherheit - Konzepte, Verfahren, Protokolle (8. Aufl.)*. Oldenbourg Verlag, München, 2013.

[Sch06] Bruce Schneier. *Angewandte Kryptographie - Der Klassiker. Protokolle, Algorithmen und Sourcecode in C*. Pearson Studium, München, 2006.

[Sch09] Klaus Schmeh. *Kryptographie - Verfahren, Protokolle, Infrastrukturen*. dpunkt.verlag, Heidelberg, 2009.