

TLS

Transport Layer Security

Autor: Prof. Dr.-Ing.
Anatol Badach



Auszug aus dem
Werk:



Heinz Schulte
WEKA-Verlag

ISBN 978-
3824540662

<http://www.weka.at/bestellen/protokolle-und-dienste-der-informationstechnologie>

Transport Layer Security (TLS) bezeichnet ein wichtiges Sicherheitsprotokoll in IP-Netzen, das von mehreren Anwendungsprotokollen (Application Protocol) benutzt wird, um über TCP¹-Verbindungen transportierte Daten vor böswilligen Angriffen zu schützen. Beim Einsatz von TLS müssen die kommunizierenden Rechner vor dem eigentlichen Datenaustausch aber vereinbaren, wie die zu transportierenden Daten geschützt werden sollen. TLS wurde somit entwickelt, um die Kommunikation über TCP-Verbindungen sichern zu können. Mit TLS können transportierte Daten sowohl vor dem Abhören geschützt werden, um ihre Vertraulichkeit (Privacy) zu garantieren, als auch gegen gezielte Verfälschungen abgesichert werden, um Datenintegrität (Data Integrity) gewährleisten zu können.

Die Konzepte von TLS wurden in mehreren Standardisierungsdokumenten² der Internet Engineering Task Force (IETF) spezifiziert. Die Version 1.0 von TLS – in RFC 2246 (1999) dargestellt – entspricht weitgehend der Version 3.0 des von der Firma Netscape entwickelten Protokolls *Secure Socket Layer (SSL)* zur Sicherung der Kommunikation zwischen Webbrowser und Webserver. Seit der Spezifikation von TLS im Jahr 1999 wurde das Protokoll weiterentwickelt und um neue Sicherheitsverfahren erweitert, sodass inzwischen neue Versionen von TLS – und zwar TLS 1.1 in RFC 4346 (2006) und TLS 1.2 in RFC 5246 (2008) – spezifiziert wurden.

Bedeutung von TLS

TLS wird im Schichtenmodell der OSI³-Referenzarchitektur in der Anwendungsschicht quasi als TLS-Teilschicht zwischen Anwendungsprotokollen, die das verbindungsorientierte Transportprotokoll TCP nutzen (wie z.B. HTTP⁴, SMTP⁵, FTP⁶ und SIP⁷), und dem

¹ Transmission Control Protocol

² <http://www.ietf.org/dyn/wg/charter/tls-charter.html>

³ Open Systems Interconnection

⁴ Hypertext Transfer Protocol

T

TLS

TCP angesiedelt. Bild 004132 illustriert dies. Nutzt ein Anwendungsprotokoll das TLS, so wird sein Name mit dem Buchstaben „S“ beendet – HTTP wird beispielsweise zu HTTPS. Damit wird zum Ausdruck gebracht, dass es sich um die gesicherte Version eines Anwendungsprotokolls handelt. Demzufolge kann ein Anwendungsprotokoll zusammen mit TLS – aus der Sicht des Transportprotokolls TCP – als ein eigenständiges Anwendungsprotokoll angesehen werden, dem seitens des TCP auch ein Well Known Port (WKP) zugewiesen werden muss.

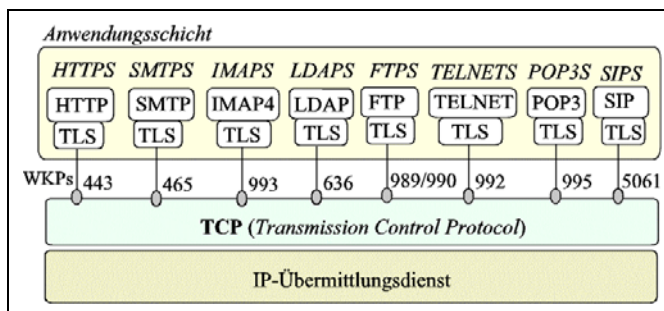


Bild 004132: TLS im Schichtenmodell und seine Bedeutung

IMAP:	Internet Message Access Protocol
LDAP:	Lightweight Directory Access Protocol
POP3:	Post Office Protocol, Vers. 3
TELNET:	Telecommunication Network
WKP:	Well Known Port

Wie in Bild 004132 dargestellt, werden mithilfe von TLS folgende Anwendungsprotokolle vor böswilligen Angriffen geschützt:

- HTTP over TLS – *Secure HTTP (HTTPS)*
Das Hypertext Transfer Protocol (HTTP) wird zur Kommunikation zwischen Webserver und Browser verwendet. Um diese Kommunikation zu sichern, wird HTTPS eingesetzt.

⁵ Simple Mail Transfer Protocol

⁶ File Transfer Protocol

⁷ Session Initiation Protocol

- **SMTP over TLS – *Secure SMTP (SMTPS)***
Das Simple Mail Transfer Protocol (SMTP) dient dem Transport von E-Mails. SMTPS ermöglicht eine gegenseitige Authentifizierung der Kommunikationspartner und garantiert sowohl die Integrität als auch die Vertraulichkeit von E-Mails.
- **IMAP4 over TLS – *Secure IMAP (IMAPS)***
Das Internet Message Access Protocol (IMAP) ermöglicht den Zugriff auf empfangene E-Mails, die sich in einem Postfach auf einem Mailserver befinden. Die aktuelle Version von IMAP ist IMAP4. Um den Abruf von E-Mails zu sichern, wird IMAPS verwendet.
- **LDAP over TLS – *Secure LDAP (LDAPS)***
Das Lightweight Directory Access Protocol (LDAP) wird bei sog. Verzeichnisdiensten (Directory Services) eingesetzt. Es beschreibt die Kommunikation zwischen einem LDAP-Client und einem Directory Server. Zur Absicherung dieser Kommunikation wird LDAPS eingesetzt.
- **FTP over TLS – *Secure FTP (FTPS)***
Das File Transfer Protocol (FTP) ist ein Anwendungsprotokoll zur Übermittlung großer Dateien (Files) in IP-Netzen und wird genutzt, um Dateien sowohl vom Server zum Client (Herunterladen) als auch um diese vom Client zum Server (Hochladen) zu übertragen. Mit FTPS können sich die beiden kommunizierenden Rechner gegenseitig authentifizieren und untereinander verschlüsselte Dateien übermitteln.
- **TELNET over TLS – *Secure TELNET (TELNETS)***
Telecommunication Network (TELNET) ist ein Client-Server-Protokoll für den Zugriff auf einen entfernten Rechner (Remote Computer) und ermöglicht eine bidirektionale, Byte-orientierte Kommunikation – quasi als *Remote Computing*. Um den Zugriff mit TELNET auf einen entfernten Rechner zu sichern, wird TELNETS verwendet.
- **POP3 over TLS – *Secure POP3 (POP3S)***
Das Post Office Protocol (POP) – in der Version 3 als POP3 bezeichnet – dient dem Transport von E-Mails von einem E-Mail-Server – bei POP auch als POP-Server bezeichnet – auf den Rechner eines Benutzers. Zur Absicherung der Kommunikation zum POP-Server kann POP3S verwendet werden.

- **SIP over TLS – *Secure SIP (SIPS)***
Das Session Initiation Protocol (SIP) gehört zu den wichtigsten Protokollen in IP-Netzen und wird u.a. zum Auf- und Abbau von Verbindungen für multimediale Kommunikation – insbesondere beim Voice over IP (VoIP) als Signalisierungsprotokoll – verwendet. Normalerweise nutzt SIP das User Datagram Protocol (UDP), ein verbindungsloses Transportprotokoll. Zur Übermittlung von SIP-Nachrichten zwischen sog. SIP-Proxies bzw. zwischen SIP-Servern kann, um die Übermittlung zu sichern, SIPS zum Einsatz kommen.

Besonderheiten von TLS

Die wichtigsten Besonderheiten von TLS sind:

- ***TLS als Client-Server-Protokoll:*** Bei TLS unterscheidet man zwischen TLS-Client und TLS-Server. Sie werden im Weiteren meist kurz *Client* und *Server* genannt. TLS fordert von ihnen verschiedene Verhaltensweisen. Beim Einsatz von HTTPS zur Sicherung von Webanwendungen übernimmt z.B. der Webbrowser die Funktion eines TLS-Clients und der Webserver die Funktion eines TLS-Servers.
- Vor dem eigentlichen Datenaustausch vereinbaren Client und Server mithilfe des sog. *Handshake-Protokolls* eine *Sicherheits-Suite* (Cipher Suite), die festlegt, wie der zwischen ihnen über eine TCP-Verbindung verlaufende Datentransport abgesichert werden soll. Dies könnte man sich so vorstellen, als ob zwischen Client und Server eine gesicherte – als *TLS-Verbindung* (TLS Connection) bezeichnete – TCP-Verbindung aufgebaut würde (Bild 004134).
- Durch das Festlegen einer Cipher Suite beim Aufbau einer TLS-Verbindung werden folgende „Punkte“ bezüglich der Sicherheitsgewährleistung geklärt:
 - *Schlüsselaustausch-* und *Authentifizierungsverfahren*. Die Art und Weise, wie Client und Server bestimmte Schlüsselmaterialien austauschen, um sich gegenseitig in die Lage zu versetzen, eigenständig einen gemeinsamen und geheimen Schlüssel gene-

rieren zu können, legt das Schlüsselaustauschverfahren⁸ (Key Exchange Method) fest. Zusätzlich wird geklärt, wie sich Client und Server authentifizieren können. – Dies beschreibt ein Authentifizierungsverfahren (Authentication Method). Bild 004140 zeigt eine Auflistung von bei TLS einsetzbaren Verfahren zum Schlüsselaustausch und zur Authentifizierung.

- Das *symmetrische Krypto- oder Verschlüsselungsverfahren* (Symmetric Encryption), nach dem die über eine TLS-Verbindung zu übertragenden Daten verschlüsselt werden. Bild 004140 zeigt, welche Verfahren zu diesem Zweck eingesetzt werden können.

- Die *Hashfunktion* (Hash Function) zur Authentifizierung transportierter Daten – insbesondere, um die Integrität von empfangenen Daten überprüfen zu können.

- Das *Datenkompressionsverfahren* (Data Compression Method) zur Reduzierung (Komprimierung) des zu übertragenden Datenvolumens.

Komponenten von TLS

TLS stellt eigentlich ein Rahmenwerk dar, das die Zusammenarbeit mehrerer Protokolle beschreibt. Wie Bild 004133 zeigt, werden diese – als Komponenten von TLS – zwei Schichten zugeordnet.

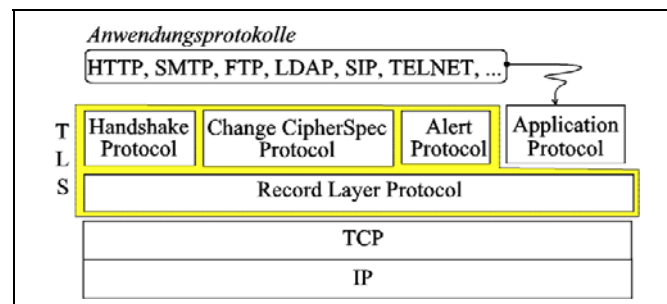


Bild 004133: TLS im Schichtenmodell und seine Komponenten

Die obere Schicht von TLS bilden folgende Protokolle:

⁸ Es handelt es sich hier eigentlich um den Austausch von Schlüsselmaterialien. Der geheime Schlüssel wird und darf nie übertragen werden.

- *Handshake-Protokoll*
Dieses Protokoll definiert die Prinzipien, nach denen Client und Server all ihre Sicherheitsvereinbarungen treffen können. Eine Vereinbarung zwischen Client und Server im Hinblick auf die Gewährleistung der Sicherheit beim Datenaustausch zwischen ihnen bezeichnet man als *TLS-Verbindung*. Mithilfe des Handshake-Protokolls wird die sog. *Sicherheits-Suite*, die bei TLS als *Cipher Suite* bezeichnet wird, vereinbart (Bild 004140).
- *Change Cipher Spec Protocol (CCS)*
Hier handelt es sich um ein primitives Protokoll, das nur die Nachricht `ChangeCipherSpec` definiert – siehe dazu Bilder 004134 und 004135. Mit dieser Nachricht signalisiert jeder Kommunikationspartner dem anderen lediglich, dass er seinen Status gewechselt hat.
- *Alert-Protokoll*
Dieses Protokoll wird zum Abbau von TLS-Verbindungen und zur Signalisierung von fehlerhaften Situationen in Form von Warn- und Fehlermeldungen verwendet.

Logisch gesehen lässt sich jedes selbstständige Anwendungsprotokoll, das die Funktionalität von TLS – *nur* – nutzt (wie z.B. HTTP, SMTP, LDAP) der oberen TLS-Schicht zuordnen. Bild 004133 bringt dies zum Ausdruck.

Die untere Schicht bei TLS funktioniert nach dem *Record Layer Protocol* und stellt die Rahmen (Frames) zur Verfügung, in denen die Nachrichten aller Protokolle der oberen Schicht, d.h. sowohl von TLS-Systemkomponenten als auch von Anwendungsprotokollen, die den TLS-Dienst nutzen, übermittelt werden. Dies bedeutet, dass die Anwendungsprotokolle ohne Modifikation auf die TLS-Dienste zugreifen können. Für die Transportschicht erscheint TLS dagegen als separates Anwendungsprotokoll.

Allgemeiner Ablauf des Handshake-Protokolls

Die erste Aufgabe, die ein Client und ein Server zusammen ausführen müssen, ist die Vereinbarung von Prinzipien, nach denen die Sicherheit der Kommunikation zwischen ihnen garantiert werden soll. Eine solche Vereinbarung bezeichnet man als *TLS-Verbindung*. Zu diesem Zweck wird das Handshake-Protokoll verwendet. Den

allgemeinen Ablauf von TLS beim Aufbau einer TLS-Verbindung zeigt Bild 004134.

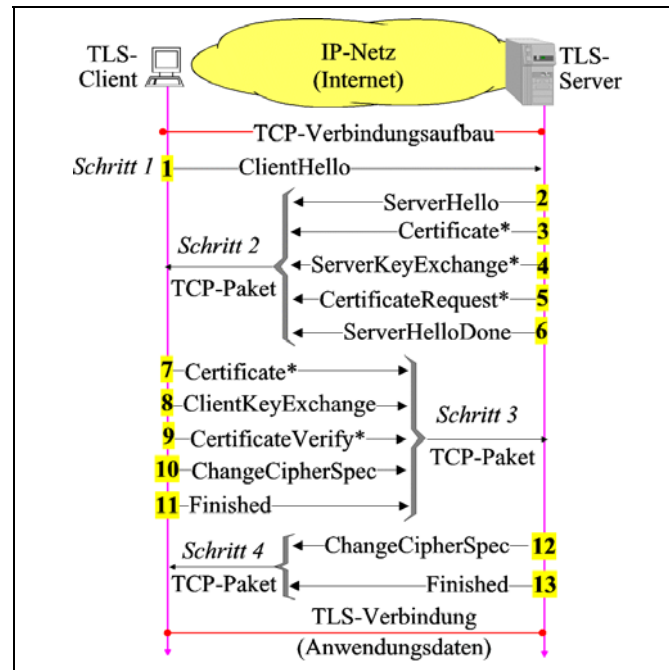


Bild 004134: TLS-Handshake – Allgemeine Schritte beim Aufbau einer TLS-Verbindung

*: optionale Nachricht

Beim Aufbau einer TLS-Verbindung müssen sich Client und Server eventuell authentifizieren. Oft wird lediglich der Server seitens des Clients authentifiziert. In solch einem Fall findet nur eine *einseitige Authentifizierung* (Unilateral Authentication)⁹ statt. Demzufolge ist nur der Client – also der Benutzer – sicher, mit „wem“ er kommuniziert. TLS ermöglicht aber auch eine *gegenseitige Authentifizierung*

⁹ auch One-Way Authentication

(Mutual Authentication)¹⁰. Bei dieser Art der Authentifizierung wird eine *Public Key Infrastructure (PKI)*¹¹ – und werden folglich auch Zertifikate (nach dem Standard ITU-T X.509) – vorausgesetzt. Bevor der Client mit dem Aufbau einer TLS-Verbindung beginnt, muss er eine TCP-Verbindung zum Server einrichten. Beim Aufbau einer TLS-Verbindung tauschen Client und Server bestimmte Nachrichten nach dem Handshake-Protokoll aus. Mehrere Nachrichten vom Client bzw. vom Server können als Nachrichtenblock in einem Frame vom Record Layer und damit als ein TCP-Paket übermittelt werden. Bild 004134 bringt dies zum Ausdruck.

Wie in Bild 004134 dargestellt, sind beim Aufbau einer TLS-Verbindung folgende Schritte zu unterscheiden:

- *Schritt 1:* Der Client gibt dem Server einige Cipher Suites sowie Datenkompressionsverfahren zur Auswahl.
Mit der Nachricht `ClientHello` (1) initiiert der Client eine TLS-Verbindung zum Server. Diese Nachricht enthält u.a.: `Session ID`, eine Zufallszahl als `client random` und eine Auflistung kryptografischer Algorithmen – sog. `Cipher Suites` –, die der Client unterstützen kann. Zusätzlich schlägt der Client dem Server eine Liste von Verfahren zur Datenkompression von zu übertragenden Daten vor.
- *Schritt 2:* Der Server teilt dem Client die ausgewählte Cipher Suite und das ausgewählte Datenkompressionsverfahren mit; der Server kann dem Client auch sein Zertifikat übergeben.
Der Server antwortet mit der Nachricht `ServerHello` (2). In dieser Nachricht, die u.a. `Session ID` (Identifikation) und eine bei ihm erzeugte Zufallszahl als `server random` enthält, teilt der Server dem Client mit, welche `Cipher Suite` und welches Datenkompressionsverfahren er ausgewählt hat.
In Schritt 2 kann der Server dem Client `optional` – wenn die Situation es erfordert – noch folgende Nachrichten senden:

¹⁰ auch Two-Way Authentication

¹¹ Mit der PKI werden asymmetrische Kryptoverfahren – auch als Public-Key-Verfahren bezeichnet – realisiert. In sog. Trust Centers werden bei der PKI Zertifikate aufbewahrt. Weil das Zertifikat eines Benutzers bzw. eines Rechners u.a. seinen öffentlichen Schlüssel (Public Key) enthält, kann der Empfänger eines Zertifikats bei dem zuständigen und vertrauenswürdigen Trust Center überprüfen, ob der öffentliche Schlüssel dem Absender gehört, der zu sein er vorgibt.

- `Certificate` (3) mit seinem Zertifikat, um dem Client mithilfe der PKI seine Authentifizierung zu ermöglichen. In diesem Zertifikat teilt der Server dem Client auch seinen öffentlichen Schlüssel mit.
 - `ServerKeyExchange` (4) mit seinem Schlüsselmaterial, um den gemeinsamen und geheimen Sitzungsschlüssel (Session Key) zu generieren, falls der Server eine sog. Pre-Shared Key Suite ausgewählt hat.
 - `CertificateRequest` (5), um das Zertifikat des Clients anzufordern.
 - Mit der Nachricht `ServerHelloDone` (6) teilt der Server dem Client mit, dass er seine Angaben beendet hat. Der Server wartet nun auf die Antwort des Clients.
- *Schritt 3:* Der Client teilt dem Server mit, dass er die ausgewählte Cipher Suite aktiviert hat und bereits zur gesicherten Sitzung übergegangen ist; er Client kann dem Server auch sein Zertifikat übergeben.

Als Antwort auf `ServerHelloDone` vom Server sendet der Client die Nachricht `ClientKeyExchange` (8). Ihr Inhalt ist von der – seitens des Servers – ausgewählten Cipher Suite abhängig. Sollte eine Pre-Shared Key Suite eingesetzt werden, kann der Client dem Server in `ClientKeyExchange` sein Schlüsselmaterial senden, damit die beiden einen gemeinsamen und geheimen Sitzungsschlüssel generieren können.

In diesem Schritt kann der Client dem Server optional, das heißt, falls die Situation es erfordert, folgende Nachrichten senden:

 - `Certificate` (7) mit seinem Zertifikat, falls dieses in Schritt 2 vom Server angefordert wurde.
 - `CertificateVerify` (9) mit der digitalen Signatur (Digital Signature), sodass der Server das in der Nachricht `Certificate` vom Client empfangene Zertifikat überprüfen kann. Die digitale Signatur wird aus der Nachricht `Certificate` mit dem privaten Schlüssel (Private Key) des Clients berechnet. Somit kann der Server diese Signatur mit dem – ihm aus dem Zertifikat des Clients bekannten – öffentlichen Schlüssel des Clients entschlüsseln.
 - Mit `ChangeCipherSpec` (10)¹² teilt der Client dem Server

¹² Es handelt sich hier eigentlich um eine einzige Nachricht des Change Cipher Spec Protocol (Bild 004133).

mit, dass er die ausgehandelten kryptografischen Verfahren aktiviert und seinen Status gewechselt hat – d.h. zur gesicherten Sitzung¹³ übergegangen ist. Anschließend signalisiert er dem Server noch mit der Nachricht `Finished` (11), dass der Aufbau der initiierten TLS-Verbindung seinerseits beendet wurde.

- *Schritt 4:* Der Server teilt dem Client mit, dass auch er zur gesicherten Sitzung übergegangen ist. Mit der Nachricht `ChangeCipherSpec` (12) teilt der Server dem Client mit, dass auch er die ausgehandelte Cipher Suite aktiviert und seinen Status gewechselt hat – d.h. auch er ist zur gesicherten Sitzung übergegangen. Anschließend signalisiert der Server dem Client noch mit `Finished` (13), dass er den Aufbau der TLS-Verbindung seinerseits beendet hat.

In den eben dargestellten vier Schritten kommt zwischen Client und Server eine Vereinbarung hinsichtlich der Unterstützung der Sicherheit zustande, die eine virtuelle TLS-Verbindung darstellt. Folglich können nun die eigentlichen Daten geschützt zwischen Client und Server übertragen werden.

Für die Fortsetzung siehe: Fachkompendium
Protokolle und Dienste der Informationstechnologie,
WEKA-Verlag, ISBN-13: 978-3824540662

¹³ Alles, was der Client von nun an sendet, wird nach dem Verschlüsselungsverfahren aus der ausgewählten Cipher Suite gesichert (verschlüsselt) (Bild 004140).