Kryptografie II

IT-Security

Andreas Unterweger

Studiengang Web Business & Technology
FH Kufstein

Sommersemester 2020

Wiederholung I

- CIA-Schutzziele
  - Vertraulichkeit (engl. *confidentiality*)
  - Integrität (engl. *integrity*)
  - Verfügbarkeit (engl. *availability*)
- Weitere Schutzziele
  - Authentifizierung (engl. *authentication*): Echtheit des Absenders (der Absender ist der, für den er sich ausgibt)
  - Nichtabstreitbarkeit (engl. *non-repudiation*): Nachricht wurde gesendet (nachträgliches Abstreiten, dass gesendet wurde, ist nicht möglich)
- Kryptografische Verfahren
  - Symmetrische Verschlüsselung (alle CIA-Schutzziele)
  - Asymmetrische Verschlüsselung (zusätzlich Authentifizierungsschutzziel)
  - Hashes (nur Integritätsschutzziel)

Wiederholung II

- Symmetrische Verschlüsselung
  - Sender und Empfänger verwenden den gleichen Schlüssel
  - Notation[1]: $c = E(k; p)$ und $p = D(k; c)$
  - Schlüsselaustausch zwischen Sender und Empfänger notwendig
- Asymmetrische Verschlüsselung
  - Sender und Empfänger verwenden unterschiedliche Schlüssel
  - Notation[1]: $c = E(k_1; p)$ und $p = D(k_2; c)$
  - Schlüsselaustausch unproblematisch (ein Schlüssel öffentlich)
- Hashes („Fingerabdrücke“, Prüfsummen)
  - Notation[2]: $h = H(p)$
  - Ungleiche Eingabedaten führen fast immer zu ungleichen Hashes

  $\rightarrow$ Erzeugung von alternativen Eingabedaten mit gleichem Hash schwierig

[1] Verschlüsselungsalgorithmus E bzw. Entschlüsselungsalgorithmus D: $E(x; y)$ bzw. $D(x; y)$, wobei x Schlüssel und y Klartext p bzw. Geheimtext c

[2] Hashalgorithmus H: $H(x)$, wobei x Nachricht

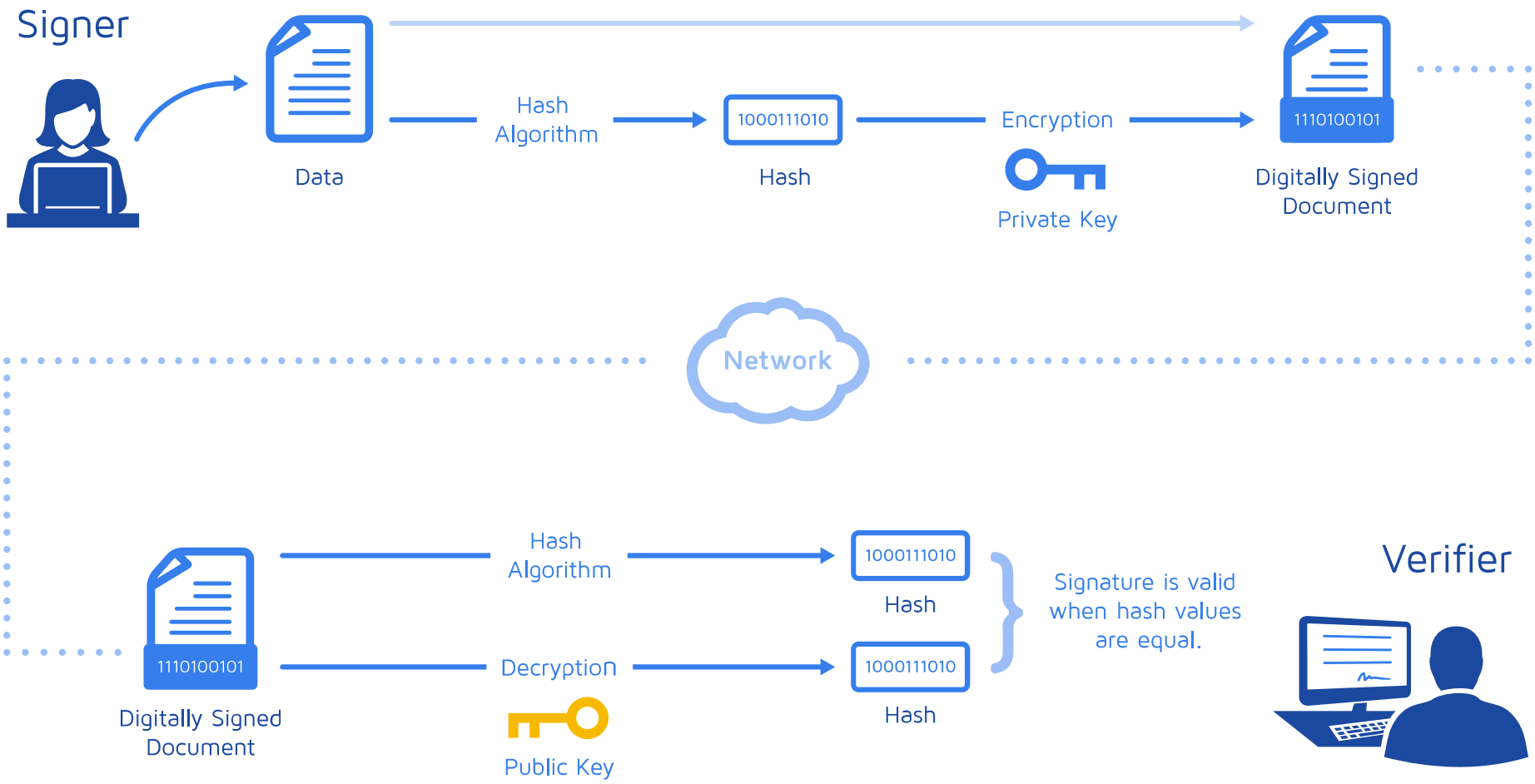
Themenüberblick

- Authentifizierungstechniken
  - Digitale Signaturen
  - Digitale Zertifikate
  - Weitere Authentifizierungstechniken
- Man-in-the-Middle-Angriffe
  - Sniffing
  - Spoofing
- Kryptografische Übertragungsprotokolle
  - Transport Layer Security (TLS)
  - HTTP Secure (HTTPS)
  - Weitere Protokolle

Digitale Signaturen (engl. *digital signatures*) I

- Analog zu Unterschriften auf Dokumenten
- Erlauben die Überprüfung der Unverändertheit des signierten Dokumentes (Schutzziel Integrität)
- Erlauben Identitätsüberprüfung des Senders durch den Empfänger (erweitertes Schutzziel Authentifizierung)
- Machen es dem Sender unmöglich, die Signatur im Nachhinein für ungültig zu erklären (erweitertes Schutzziel Nichtabstreitbarkeit)
- Verwenden asymmetrische Kryptografie, um Schutzziele zu erreichen
- Notation: Signatur $s = S(k_{private}; p) := E(k_{private}; H(p))$ mit Signaturalgorithmus $S(x, y)$, wobei x privater Schlüssel $k_{private}$ und y Klartext p mit Hashalgorithmus H und Verschlüsselungsalgorithmus E

Digitale Signaturen (engl. *digital signatures*) II



Quelle: DocuSign: Understanding digital signatures. `https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq` (Zugriff am 4.3.2017), 2017.

Digitale Signaturen (engl. *digital signatures*) III

- Empfänger erhält Nachricht p' mit Signatur s
- Überprüfung der Signatur s:
  1. Berechnung des Hashes der Nachricht $h' = H(p')$
  2. Entschlüsselung der Signatur mit dem öffentlichen Schlüssel: $h = D(k_{public}; s) = D(k_{public}; E(k_{private}; H(p))) = H(p)$
  3. Hashvergleich: $h' \stackrel{?}{=} h$ (nur gleich, wenn $p = p'$)
- Bei Hashgleichheit: Nachricht ist unverändert **und** vom Sender
- Ansonsten: Nachricht wurde verändert **oder** ist nicht vom Sender
- Annahme: Verschlüsselungs- und Hashalgorithmus sicher
  - Andere Nachricht führt zu anderem Hash
  - Privater Schlüssel ist nur dem Sender bekannt $\rightarrow$ niemand sonst kann andere Nachrichten signieren

Digitale Zertifikate (engl. *digital certificates*) I

- Digitale Signaturen erlauben **nur** Identitätsüberprüfung auf Ebene der Schlüssel (Angriffsszenario: veröffentlichten Schlüssel austauschen)

$\rightarrow$ Nicht überprüfbar, welche **Person** hinter dem Schlüssel steckt

$\rightarrow$ Vertrauenswürdige dritte Partei (engl. *trusted third party*) notwendig

- Dritte Partei bescheinigt Zusammenhang zwischen Identität des Senders und öffentlichem Schlüssel per digitaler Signatur
- Zertifikat: Signierte Information über Name (des Schlüsselbesitzers), Schlüssel, Ablaufdatum etc.

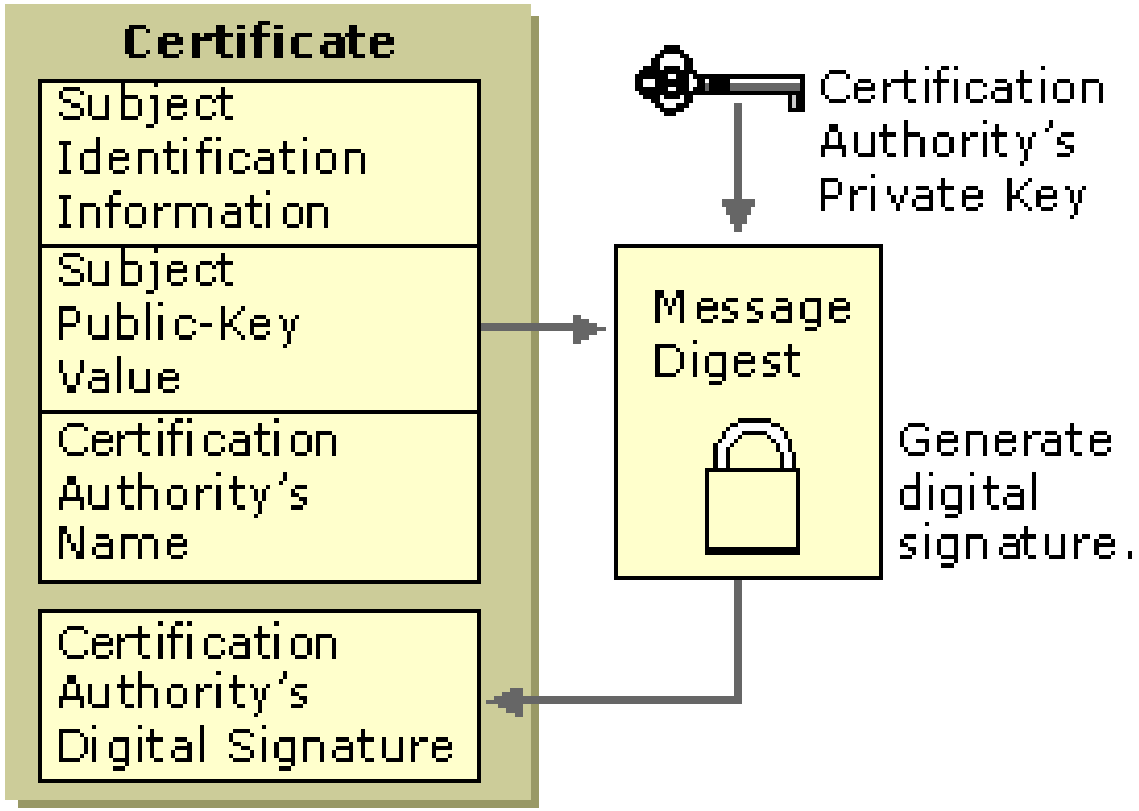
$\rightarrow$ Identität überprüfbar (Sender und Empfänger vertrauen dritter Partei)

- Bezeichnung der dritten Partei: Certificate Autority (CA)

Digitale Zertifikate (engl. *digital certificates*) II

- Zertifikat ist mit privatem Schlüssel der CA signiert

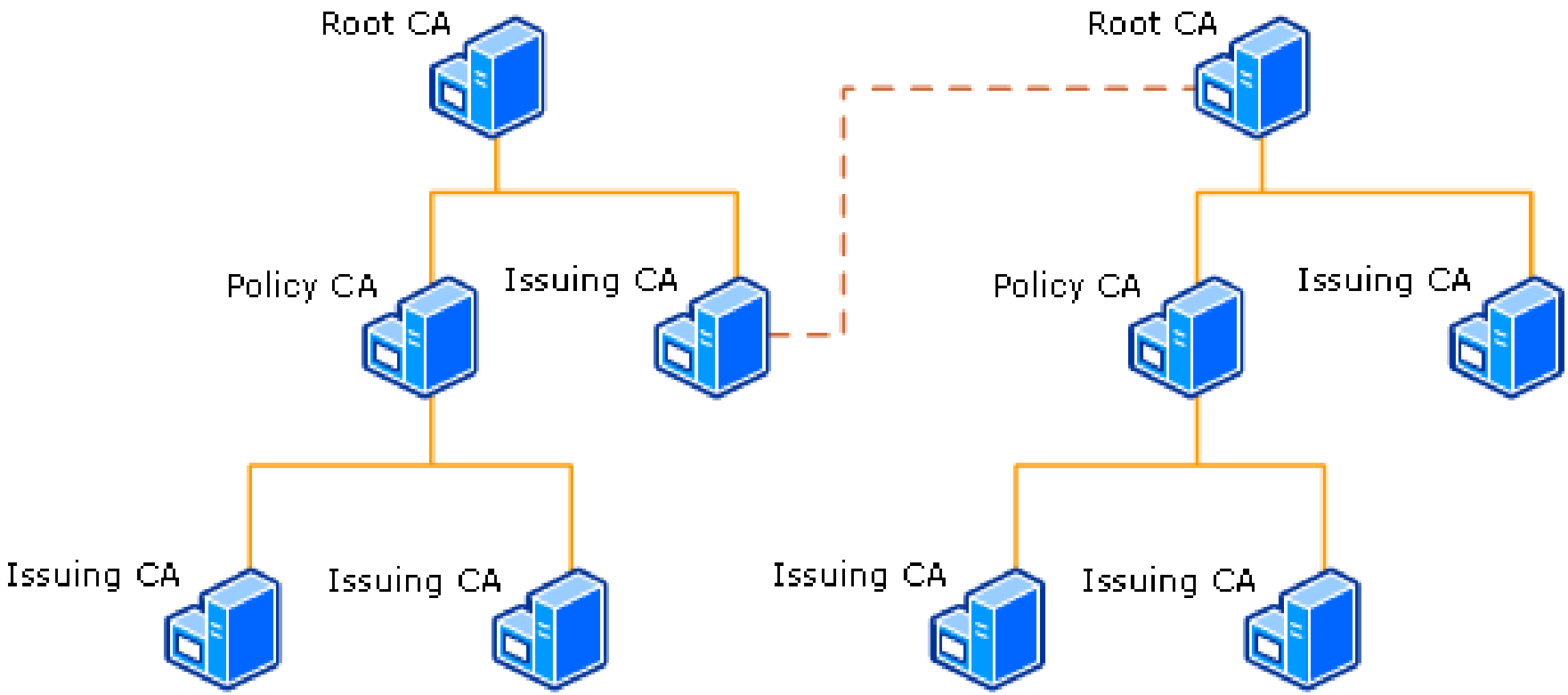
→ Wer zertifiziert die öffentlichen Schlüssel der CA?



Quelle: Microsoft: Digital Certificates. `https://technet.microsoft.com/en-us/library/cc962029.aspx` (Zugriff am 4.3.2017), 2017.

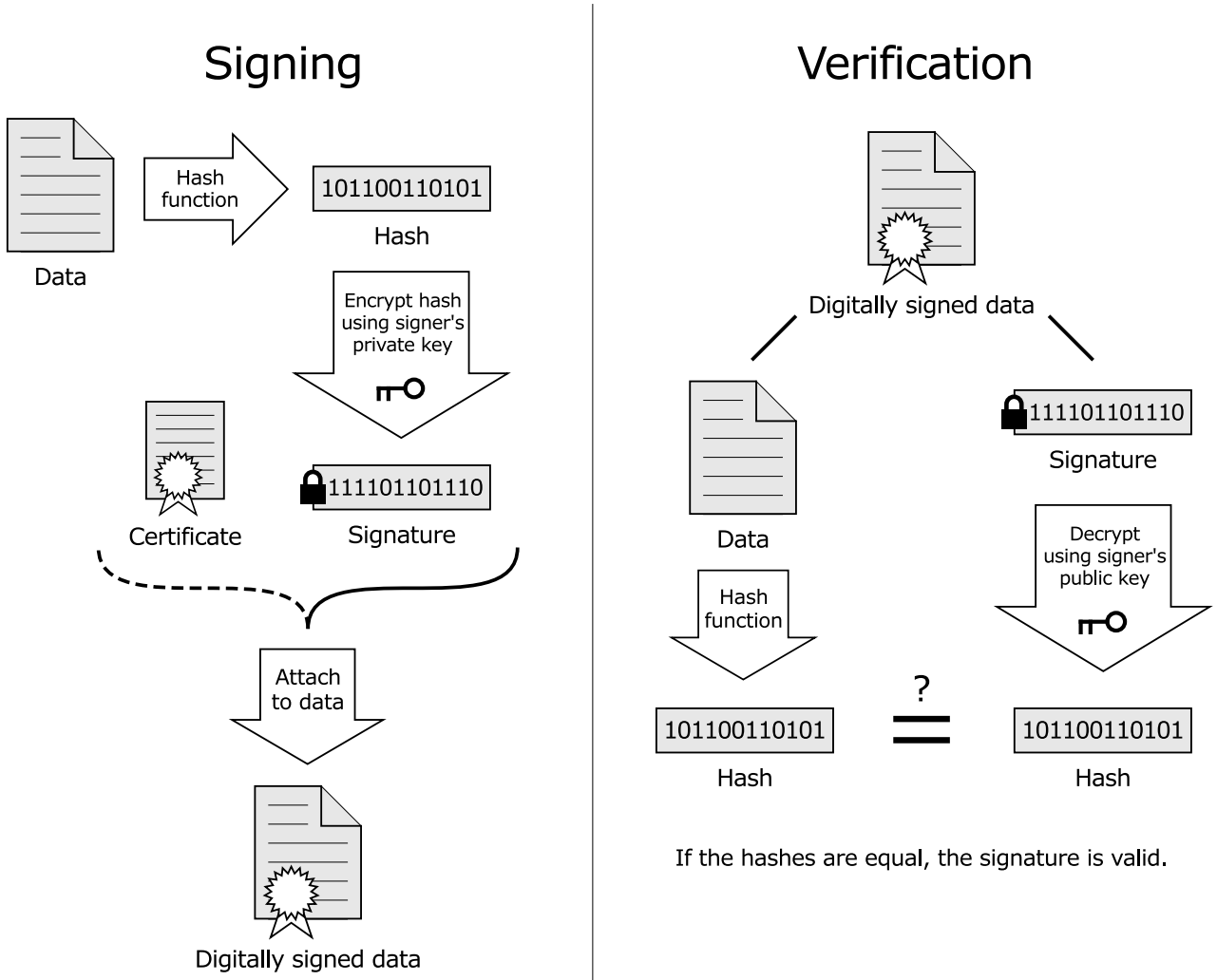
Publik Key Infrastructure (PKI)

- Hierarchische CA mit hierarchischem Vertrauensmodell
- Zusätzliche Organisationen zur Zertifikatverteilung und -stornierung
- Öffentliche Schlüssel **in** Zertifikaten gespeichert $\rightarrow$ private?



Quelle: Microsoft: PKI Technologies. `https://technet.microsoft.com/en-us/library/cc779826(v=ws.10).aspx` (Zugriff am 4.3.2017), 2003.

Digitale Signaturen mit Zertifikaten



Quelle: Acdx: Digital Signature diagram. `https://commons.wikimedia.org/wiki/File:Digital_Signature_diagram.svg` (Zugriff am 4.3.2017), 2012.

Weitere Authentifizierungstechniken

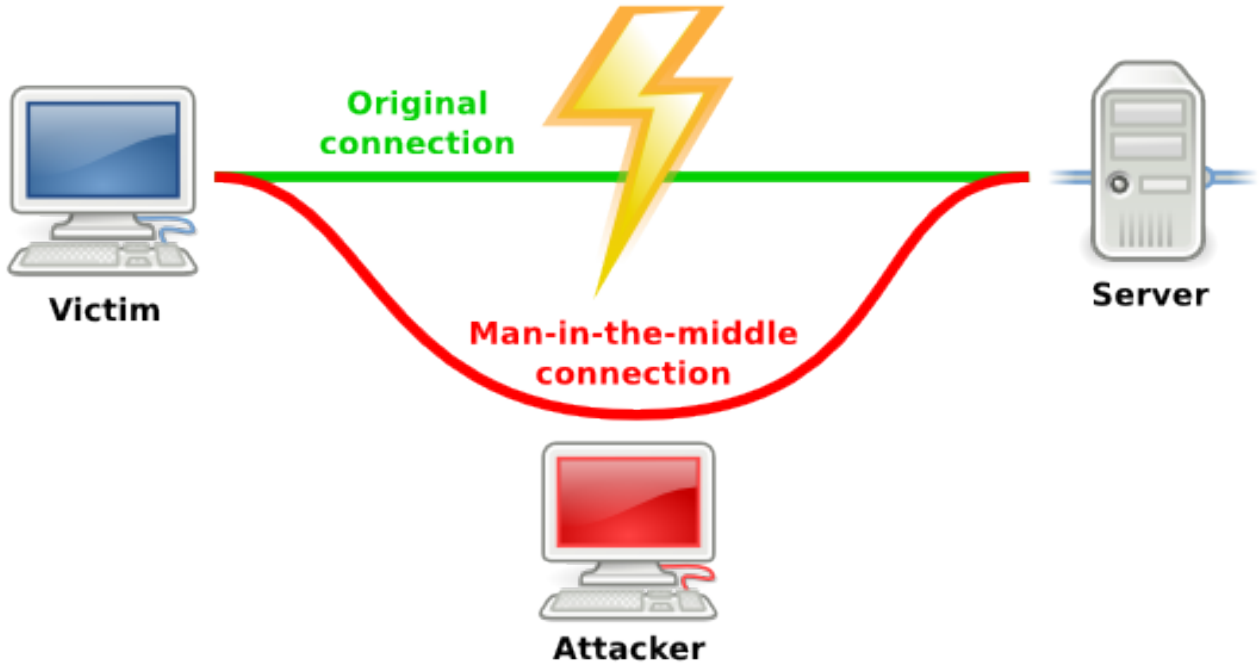
- Verschiedene Arten (Kategorisierung)
  - Was man hat, z.B. Zutrittskarte, (Raum-)Schlüssel
  - Was man ist, z.B. Fingerabdruck, Stimmencharakteristikum
  - Was man weiß, z.B. Passwort, Schlosskombination
  - Wo man ist, z.B. geografische Koordinaten, zutrittsgesicherter Raum
- Authentifizierung auf mehrere Arten möglich (Mehrfaktor-Authentifizierung)

$\rightarrow$ Höhere Echtheitswahrscheinlichkeit, höherer Betrugsaufwand

$\rightarrow$ Erhöhte Sicherheit

Definition Man-in-the-Middle-Angriffe

- Man in the Middle (MITM): Angreifer schaltet sich zwischen Sender und Empfänger (Übertragungsstrecke bzw. Kommunikationsprotokoll)
- Ziel: (Authentifizierungs-)Informationen $\rightarrow$ weitere Angriffe



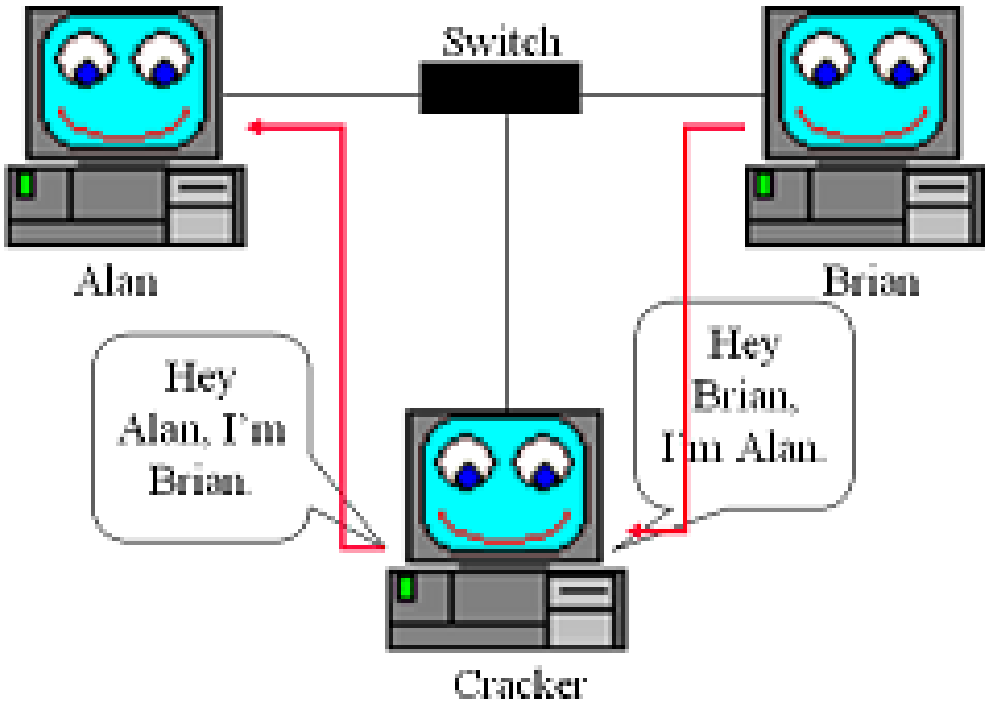
Quelle: Nawaz, A.: Man in The Middle Attack Explained (MITM). `http://viru5hax.blogspot.co.at/2014/12/man-in-middle-attack-explained-mitm.html` (Zugriff am 5.3.2017), 2014.

Sniffing

- Sniffing: Mitschneiden von Informationen als MITM
- Arten (Auswahl):
  - Passives Sniffing: „nur mithören“
  - Aktives Sniffing: Direkte Interaktion mit Sender und/oder Empfänger
  - Replay-Attacke (Kombination): Abgegriffene Daten erneut senden
- Voraussetzung: Zugang zu Kommunikationskanal bzw. Paketen
  - Trivial in Netzwerken mit Hub (kabelgebunden unüblich; WLAN!)
  - Nicht in geswitchten Netzwerken (außer mit **aktiven** Angriffen)
  - Zusätzlich: Promiscuous Mode aktiviert
- **Un**verschlüsselte Protokolle vereinfachen Mitlesen
- Verschlüsselte Protokolle hindern am Mitlesen, aber nicht immer an Replay-Attacken (vgl. später)
- Beispiele für Sniffing-Software: *Wireshark*, *Tcpdump*

ARP Poisoning (auch ARP Spoofing) I

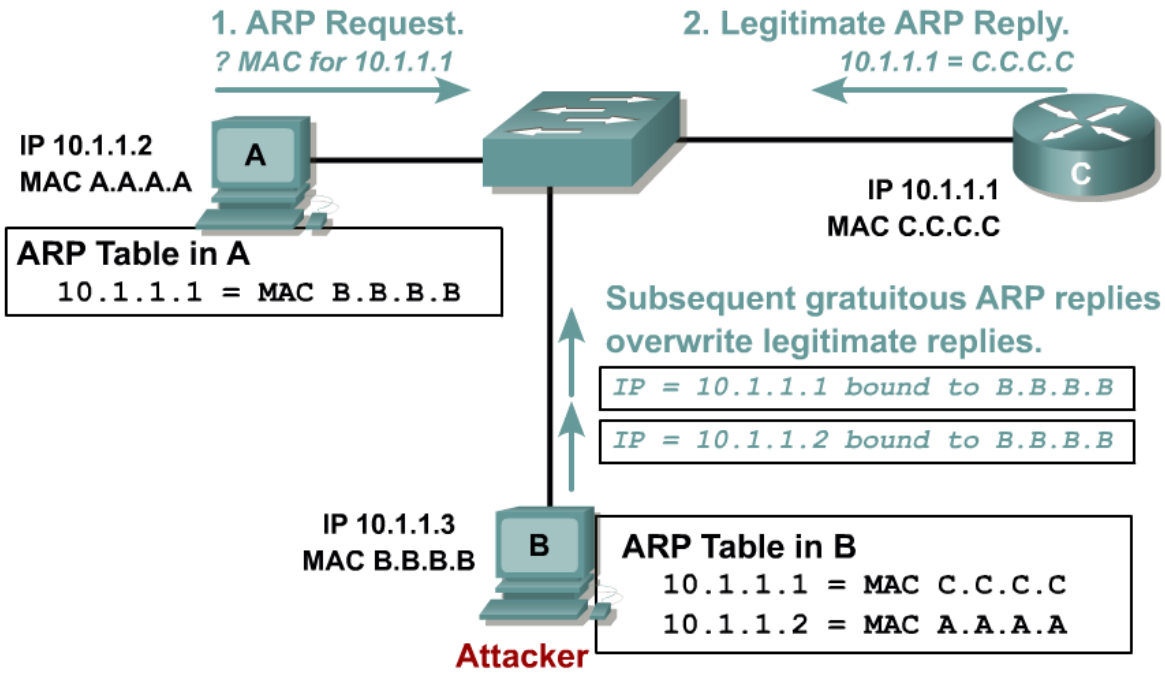
- Ziel: ARP Caches von Sender und Empfänger manipulieren (Empfänger glaubt, dass Angreifer Sender ist; Sender glaubt, dass Angreifer Empfänger ist) $\rightarrow$ Pakete mitlesen und weiterleiten



Quelle: Tangient: HOWTO - ARP Poisoning in Linux. `https://cyberang3l.wikispaces.com/HOWTO+-+ARP+Poisoning+in+Linux` (Zugriff am 5.3.2017), 2017.

ARP Poisoning (auch ARP Spoofing) II

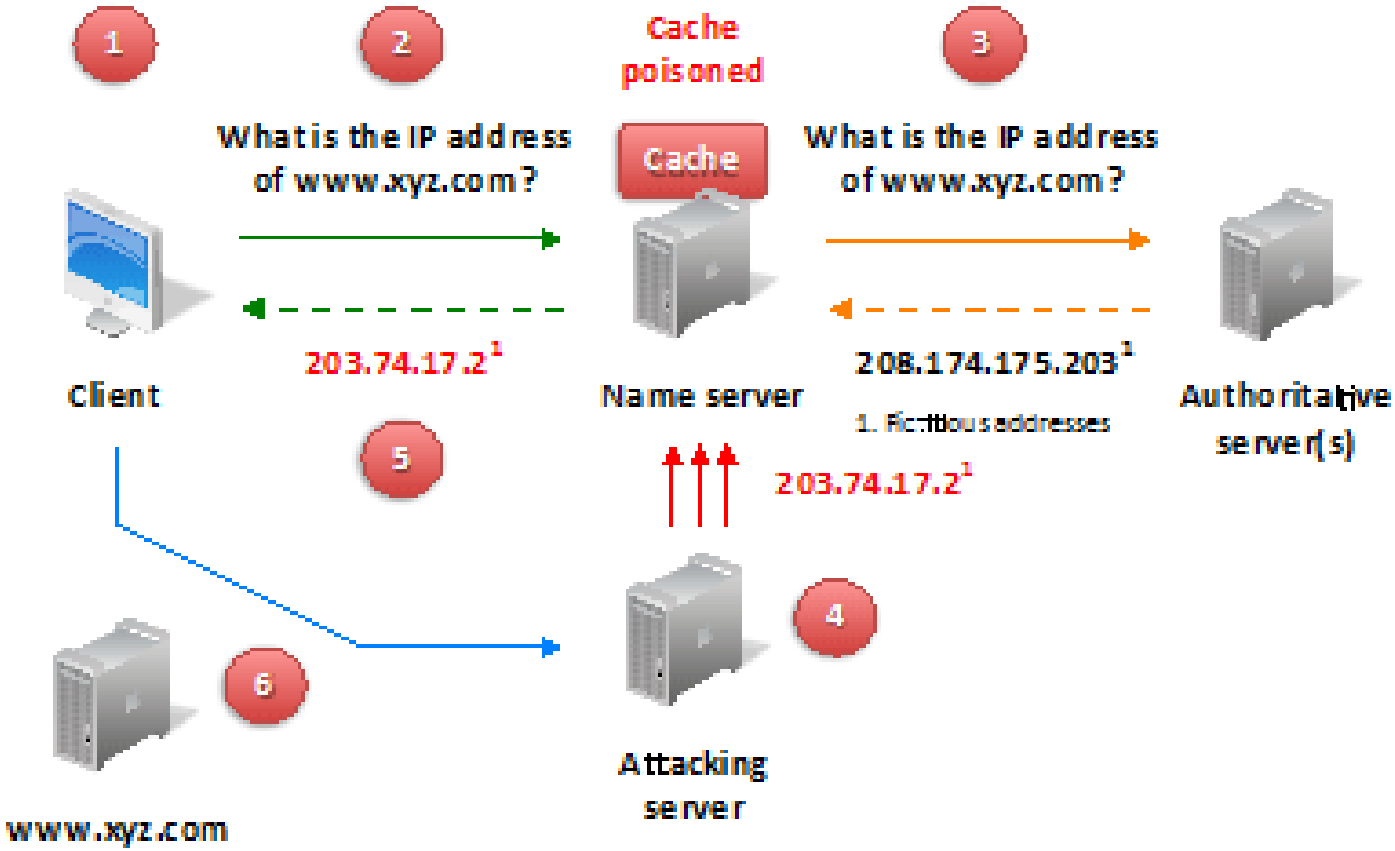
- Schwachstelle in ARP: Antworten auf Anfragen werden nicht geprüft
- Angreifer antwortet auf ARP Requests mit seiner MAC-Adresse
- Zusätzliche ARP Requests $\rightarrow$ Rechner aktualisieren ARP Cache



Quelle: Bhai, H.: ARP spoofing explained. `http://huzeifabhai.blogspot.co.at/2011/08/arp-spoofing-explained.html` (Zugriff am 5.3.2017), 2011.

DNS Poisoning (auch DNS Spoofing)

- Wie ARP Poisoning, aber bei Layer-3- statt Layer-2-Adressauflösung
- Angriffsziel: DNS-Server und/oder Schwachstellen in DNS-Protokoll



Adaptiert von NRZA Web Security: What Is DNS Cache Poisoning.
`https://nrza.net/index.php/2016/05/03/dns-cache-poisoning/` (Zugriff am 5.3.2017), 2016.

Überblick kryptografische Übertragungsprotokolle

- MITM-Attacken sind für die Sicherheit problematisch
- Verschlüsselung macht (mitgeschnittene) Daten unlesbar, aber verhindert für sich alleine u.a. keine Replay-Attacken

$\rightarrow$ Zusätzliche Absicherung auf Protokollebene (Layer 4)

- Kryptografische Transportprotokolle (Auswahl)
  - IP Security (IPsec): Verschlüsselte und authentifizierte[3] IP-Pakete
  - SSH: Verschlüsselung und Authentifizierung mit Zertifikaten
  - Secure Sockets Layer (SSL) – nicht mehr aktuell
  - **Transport Layer Security (TLS)** – Nachfolger von SSL
  - **HTTP Secure (HTTPS)** – HTTP über SSL oder TLS

[3] Verwendet keine Signaturen, sondern Hashed Message Authentication Codes (HMACs) – ähnlich wie Signaturen, aber mit symmetrischen Schlüsseln $\rightarrow$ keine Nichtabstreitbarkeit!

Überblick TLS

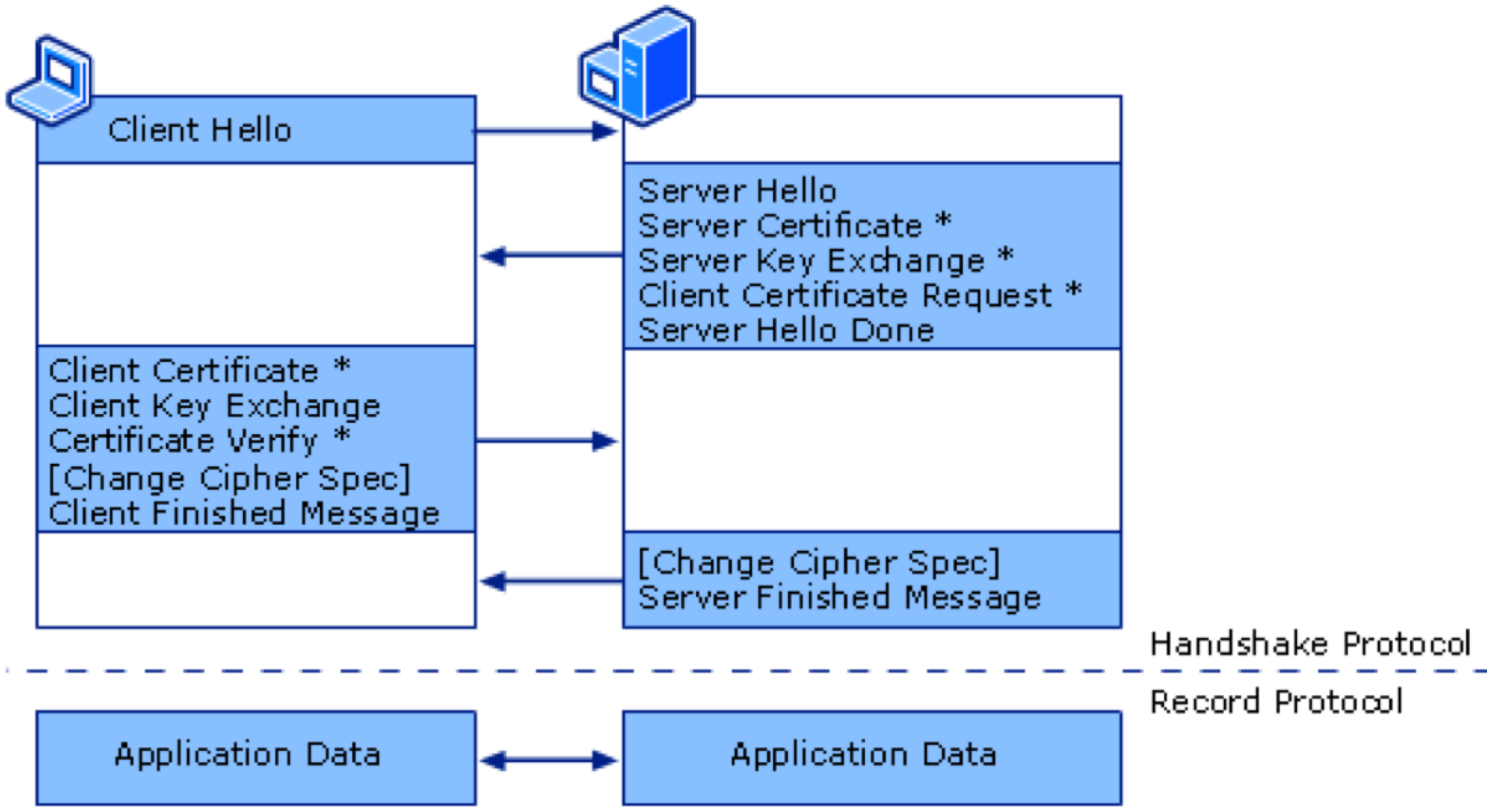
- Baut auf verbindungsorientiertem Transportprotokoll (TCP) auf
- Verwendet symmetrische Verschlüsselung für übertragene Daten
- Verwendet Authentifizierung (HMAC) für übertragene Daten
- Authentifiziert Kommunikationspartner über PKI
- Bietet Forward Secrecy
  - Generiert symmetrischen Schlüssel für jede Verbindung neu
  - MITM kann symmetrischen Schlüssel nicht aus abgefangenen Daten ermitteln, selbst wenn alle Pakete abgefangen werden
- Erkennt MITM auch während des Verbindungsaufbaus

→ Erfüllt alle Schutzziele außer Nichtabstreitbarkeit

- Jedes Protokoll der Anwendungsschicht kann auf TLS aufsetzen

TLS Handshake

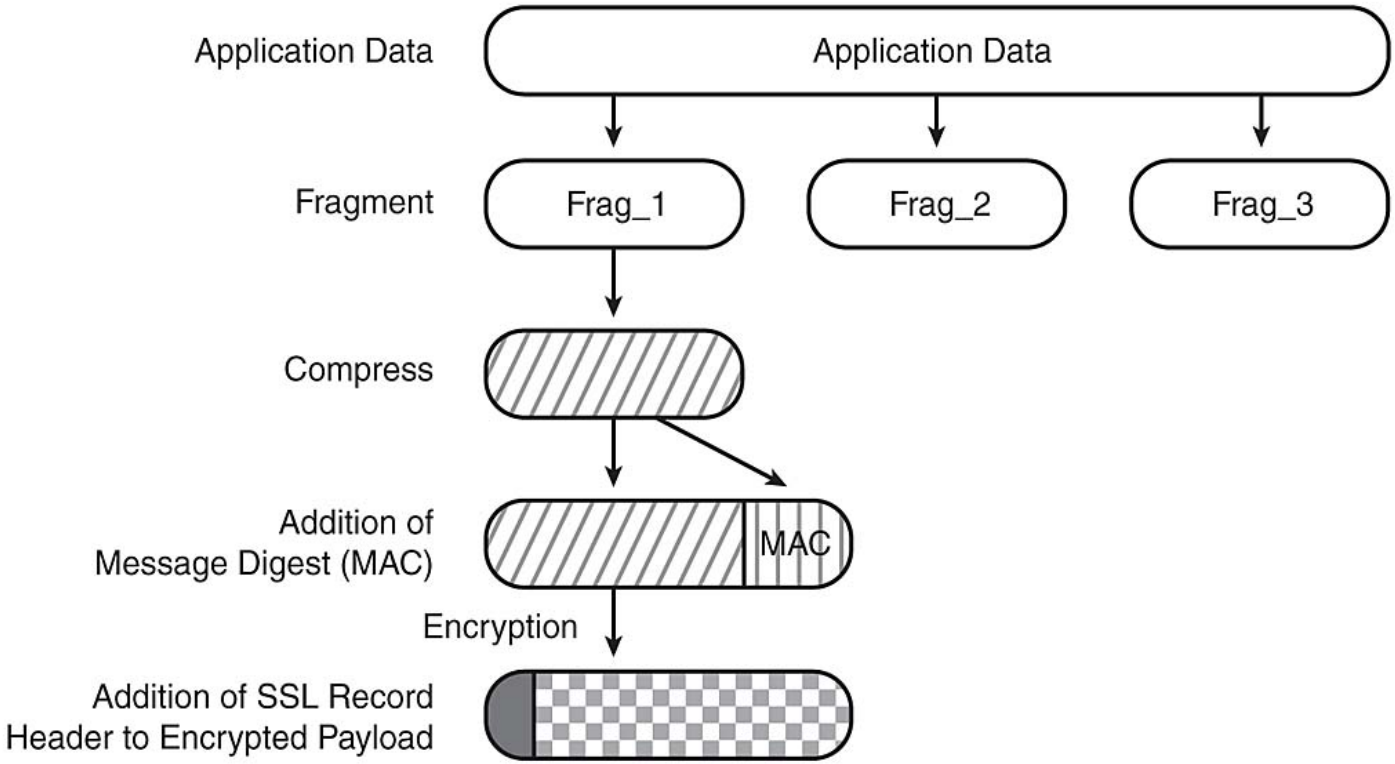
- Verbindungsaufbau mit Authentifizierung über Zertifikate
- Aushandlung von verwendeten kryptografischen Algorithmen
- Schlüsselaustausch und Vorberechnungen für symmetrischen Schlüssel



Quelle: Hoffman, B.: SSL Performance Diary #4: Optimizing the TLS Handshake. `https://zoompf.com/blog/2014/12/optimizing-tls-handshake` (Zugriff am 5.3.2017), 2014.

TLS Record Protocol

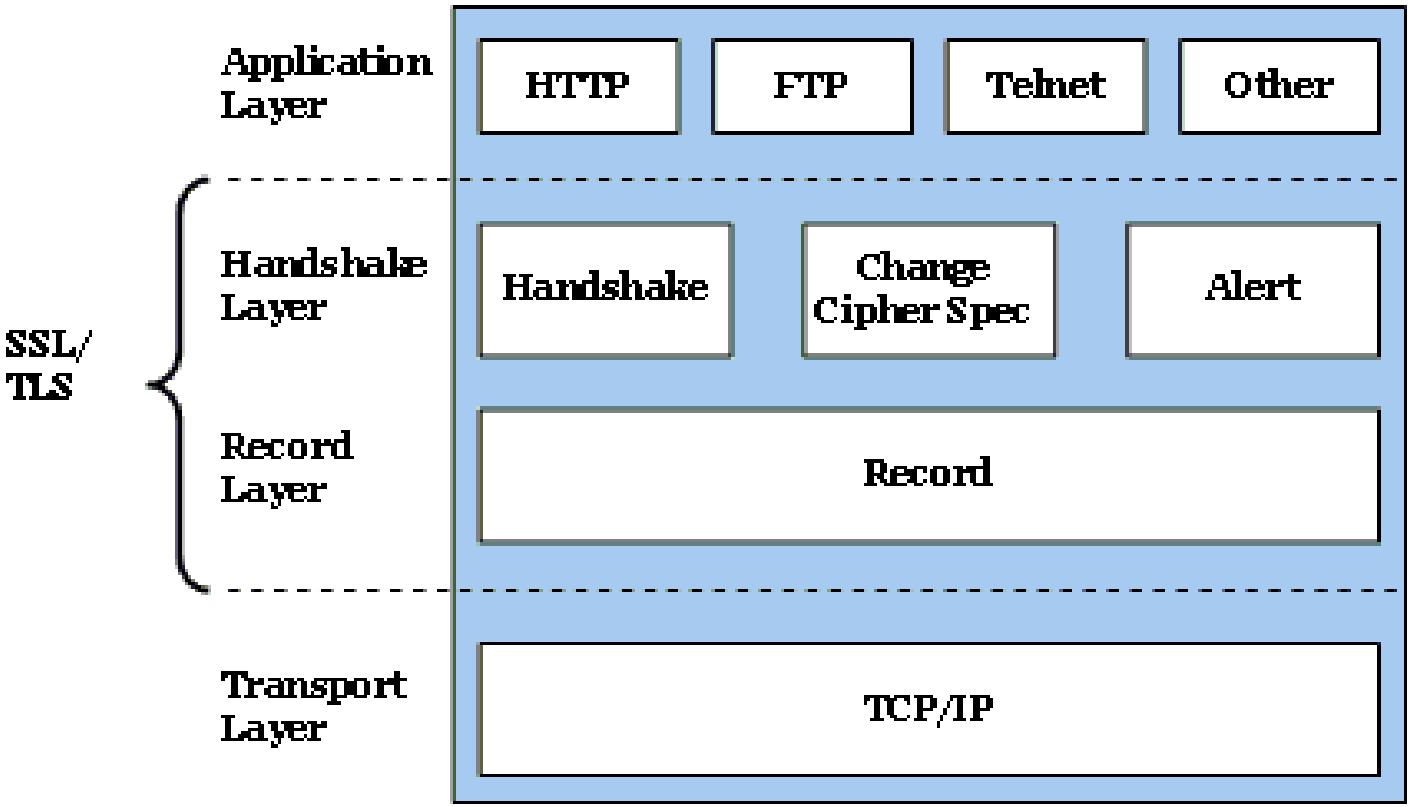
- Mehrschichtige Datenverarbeitung: Fragmentierung, Komprimierung (optional), Authentifizierung und Verschlüsselung (CBC-Modus)
- Analoge, aber umgekehrte Verarbeitung auf Empfängerseite



Quelle: Ghosh, A.: Optimize Nginx TLS Record Size For HTTP/2 TLS (Ubuntu, CentOS). `https://thecustomizewindows.com/2016/10/optimize-nginx-tls-record-size-http2-tls-ubuntu-centos/` (Zugriff am 5.3.2017), 2016.

HTTPS

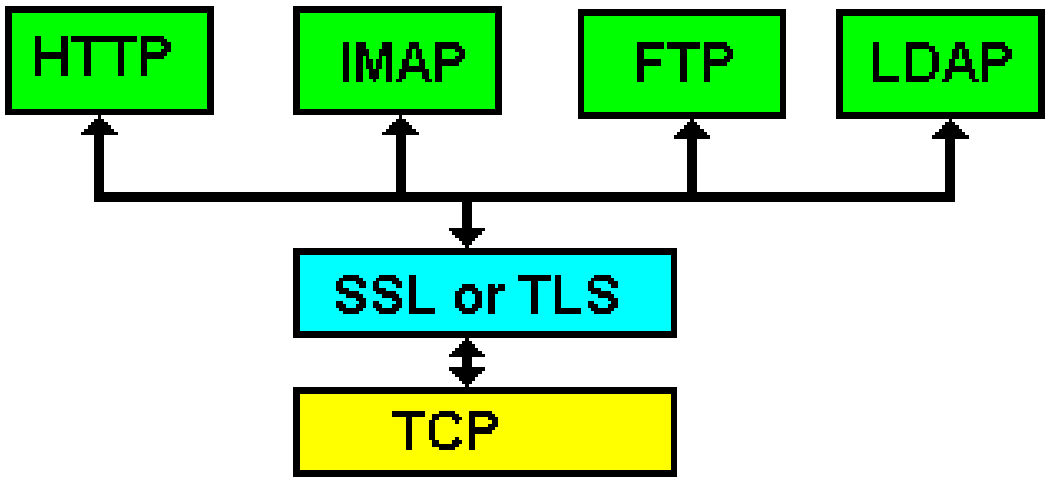
- (Reguläres) HTTP über SSL (obsolet) bzw. HTTP über TLS
- TCP-Port 443 statt 80 (https:// statt http://)



Adaptiert von Laveti, G.: New D/TLS attacks published. `http://laveti-infosec.blogspot.co.at/2013/02/new-dtls-attacks-published.html` (Zugriff am 5.3.2017), 2013.

Weitere Protokolle auf Basis von TLS

- Verwendung von SSL bzw. TLS durch andere Protokolle möglich, z.B.
  - IMAP über SSL/TLS $\rightarrow$ IMAPS (TCP-Port 993)
  - SMTP über SSL/TLS (TCP-Port 465 wie SMTP ohne SSL/TLS)
  - ...
- STARTTLS: Kommando, um bestehende unsichere Verbindung (z.B. SMTP) zu TLS-Verbindung zu machen (kein eigener Port notwendig)



Quelle: ZyTrax: Survival guides - TLS/SSL and SSL (X.509) Certificates. `http://www.zytrax.com/tech/survival/ssl.html` (Zugriff am 5.3.2017), 2017.

Danke für die Aufmerksamkeit!

Fragen?