IT-Sicherheit

Anwendungen Kryptographischer Protokolle (Teil 1)

Prof. Dr. Dominik Merli, Prof. Dr. Lothar Braun

Sommersemester 2020

Hochschule Augsburg - Fakultät für Informatik

Protokolle

Welche Protokolle kennen Sie?

Vielzahl an Sicherheitsprotokollen

- **Internet**
  - Transport Layer Security (TLS)
  - Internet Protocol Security (IPsec)
  - Internet Key Exchange (IKE)
  - ...

Vielzahl an Sicherheitsprotokollen

- **Internet**
  - Transport Layer Security (TLS)
  - Internet Protocol Security (IPsec)
  - Internet Key Exchange (IKE)
  - ...
- **Anwendungen**
  - Secure Shell (SSH)
  - Hypertext Transfer Protocol over TLS (HTTPS)
  - ...

Vielzahl an Sicherheitsprotokollen

- **Internet**
  - Transport Layer Security (TLS)
  - Internet Protocol Security (IPsec)
  - Internet Key Exchange (IKE)
  - ...
- **Anwendungen**
  - Secure Shell (SSH)
  - Hypertext Transfer Protocol over TLS (HTTPS)
  - ...
- **Messaging**
  - Off-The-Record (OTR)
  - Signal Protocol
  - ...

Transport Layer Security (TLS)

Überblick

- **Einsatzzweck**
  - Absicherung von Kommunikationskanälen
  - ... auf der Transportschicht (engl. transport layer)
  - ... gegen unberechtigtes Mitlesen
  - ... gegen unberechtigte Manipulation
  - ... zur (gegenseitigen) Authentifizierung

Überblick

- **Einsatzzweck**
  - Absicherung von Kommunikationskanälen
  - ... auf der Transportschicht (engl. transport layer)
  - ... gegen unberechtigtes Mitlesen
  - ... gegen unberechtigte Manipulation
  - ... zur (gegenseitigen) Authentifizierung

- **Geschichte**
  - 1994: Netscape veröffentlicht Secure Socket Layer (SSL) v2
  - 1996: TLS Arbeitsgruppe von Netscape zu IETF
  - 1999: TLS v1.0 wird als RFC 2246 veröffentlicht
  - 2006: TLS v1.1 mit TLS Extensions
  - 2008: TLS v1.2 mit Authenticated Encryption
  - 2018: TLS v1.3 Performance-Verbesserungen und Entfernung unsicherer Betriebsmodi

Bausteine für TLS

- **Symmetrische Kryptographie**
  - Verschlüsselung von Nutzdaten
  - Authenticated Encryption

Bausteine für TLS

- **Symmetrische Kryptographie**
  - Verschlüsselung von Nutzdaten
  - Authenticated Encryption
- **Message Authentication Codes**
  - Integritätsschutz von Nutzdaten

Bausteine für TLS

- **Symmetrische Kryptographie**
  - Verschlüsselung von Nutzdaten
  - Authenticated Encryption
- **Message Authentication Codes**
  - Integritätsschutz von Nutzdaten
- **Hash Funktionen**
  - Abbildung größerer Datenmengen auf kurzes Datum

Bausteine für TLS

- **Symmetrische Kryptographie**
  - Verschlüsselung von Nutzdaten
  - Authenticated Encryption
- **Message Authentication Codes**
  - Integritätsschutz von Nutzdaten
- **Hash Funktionen**
  - Abbildung größerer Datenmengen auf kurzes Datum
- **Asymmetrische Kryptographie**
  - Schlüsselaustausch
  - Authentifizierung
  - PKI

System-Voraussetzungen für TLS

- **Schlüsselspeicher**
  - Sichere Ablage für private Schlüssel

System-Voraussetzungen für TLS

- **Schlüsselspeicher**
  - Sichere Ablage für private Schlüssel
- **"Gute" Zufallszahlen**
  - Unvorhersehbarkeit in kryptographischen Protokollen

System-Voraussetzungen für TLS

- **Schlüsselspeicher**
  - Sichere Ablage für private Schlüssel
- **"Gute" Zufallszahlen**
  - Unvorhersehbarkeit in kryptographischen Protokollen
- **Zuverlässige Zeit-Quelle**
  - Zur korrekten Validierung von Zertifikaten

System-Voraussetzungen für TLS

- **Schlüsselspeicher**
  - Sichere Ablage für private Schlüssel
- **"Gute" Zufallszahlen**
  - Unvorhersehbarkeit in kryptographischen Protokollen
- **Zuverlässige Zeit-Quelle**
  - Zur korrekten Validierung von Zertifikaten
- **Genügend Leistung**
  - z.B. für asymmetrische Krypto-Algorithmen

QUIZ: TLS (twbk.de)

Welcher Angriff soll durch TLS hauptsächlich verhindert werden?

A) Phishing

B) Evil Maid

C) Man-in-the-Middle

QUIZ: TLS (twbk.de)

Die Generierung guter Zufallszahlen benötigt üblicherweise den Großteil der für TLS nötigen Performance. Richtig oder falsch?

A) Richtig!

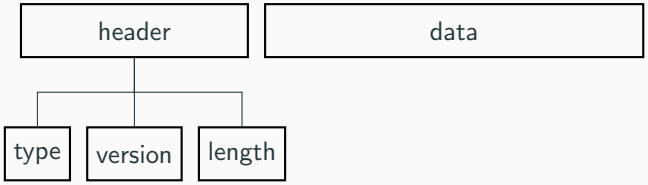
B) Falsch!

TLS v1.2

Record Protocol

- **Regelt Übertragung von (verschlüsselten) Nachrichten**
  - Maximale Länge: $2^{14} = 16384$ Bytes
  - Jeder Record bekommt einzigartig 64-bit Sequenznummer

- **TLS Record**
  - Header (Type, Version, Length)
  - Data



Sub-Protokolle

- **Handshake Protocol**
  - Schlüsselaustausch, Authentifizierung, Cipher Suites, etc.
  - Hauptbestandteil von TLS
- **Change Cipher Spec Protocol**
  - Nachricht, die Wechsel der Verbindungsparameter anzeigt
- **Alert Protocol**
  - Nachrichten, die Warnungen anzeigen
- **Application Data Protocol**
  - Übertragung der Nutzdaten auf Basis des Record Protocols

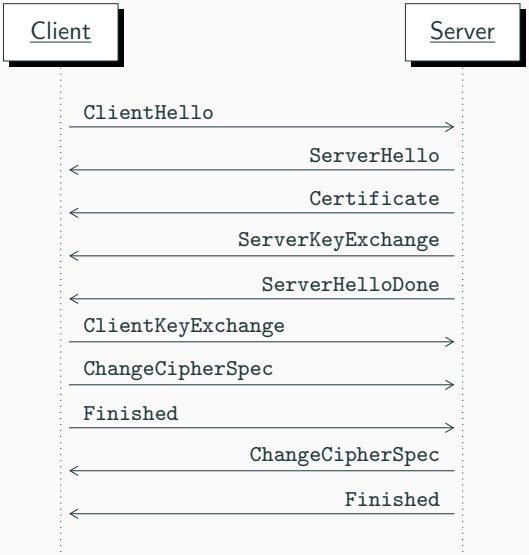
Handshake Protocol

- **Hauptbestandteil von TLS**
  - 6-10 Nachrichten

Handshake Protocol

- **Hauptbestandteil von TLS**
  - 6-10 Nachrichten
- **Drei typische Abläufe**
  - Kompletter Handshake mit Server-Authentifizierung
  - Kompletter Handshake mit Client-/Server-Authentifizierung
  - Abgekürzter Handshake mit Session Resumption

Kompletter Handshake mit Server-Authentifizierung



- **Erste Nachricht im Handshake**

- **Inhalt**
  - Protocol Version
    - Beste unterstützte Protokoll-Version
    - Kann von Extensions überschrieben werden
  - Random (32 Bytes: 28 Bytes zufällig, 4 Bytes zeitabhängig)
  - Session ID (leer für erste Verbindung)
  - Cipher Suites (Liste unterstützter Cipher Suites, nach Priorität)
  - Compression (üblicherweise `null`)
  - Extensions (Liste von Erweiterungen)

Cipher Suites

- **Auswahl kryptographischer Primitive**

- **Besteht aus ...**
  - Schlüsselaustausch
  - Authentifizierung
  - Verschlüsselung
  - Schlüssellänge für Verschlüsselung
  - evtl. Betriebsmodus
  - evtl. MAC Algorithmus
  - evtl. PRF Algorithmus (nur TLS 1.2)
  - evtl. Hashfunktion für `Finished` Nachricht (nur TLS 1.2)
  - evtl. Länge der `verify_data` Struktur (nur TLS 1.2)

Cipher Suites - Beispiele

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

Cipher Suites - Beispiele

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

Cipher Suites - Beispiele

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

Cipher Suites - Beispiele

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_PSK_WITH_AES_256_CBC_SHA384

Cipher Suites - Beispiele

- `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`
- `TLS_RSA_WITH_AES_128_CBC_SHA`
- `TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA`
- `TLS_PSK_WITH_AES_256_CBC_SHA384`
- OpenSSL Cipher Suites: `openssl ciphers ALL -V`
- Cipher Suites in RFC 5246: https://tools.ietf.org/html/rfc5246#appendix-A.5

- **Antwort auf** `ClientHello`
  - Ähnlicher Aufbau wie `ClientHello`

- **Inhalt**
  - Protocol Version
    - Muss nicht mit Client übereinstimmen
    - Kann durch Extension überschrieben werden
  - Random (32 Bytes: 28 Bytes zufällig, 4 Bytes zeitabhängig)
  - Session ID (32 einzigartige Bytes)
  - Cipher Suites (ausgewählte Cipher Suite)
  - Compression (üblicherweise `null`)
  - Extensions (ausgewählte Erweiterungen)

- **Zertifikatskette des Servers**
  - Aneinanderreihung von Zertifikaten
  - ASN-1 DER Format
  - Root-Zertifikat wird nicht mitgeliefert

- **Zertifikatskette des Servers**
  - Aneinanderreihung von Zertifikaten
  - ASN-1 DER Format
  - Root-Zertifikat wird nicht mitgeliefert

- **Muss zur gewählten Cipher Suite passen**
  - Client muss Zertifikats-Algorithmen unterstützen
  - Server muss evtl. mehrere Zertifikate haben

- **Zertifikatskette des Servers**
  - Aneinanderreihung von Zertifikaten
  - ASN-1 DER Format
  - Root-Zertifikat wird nicht mitgeliefert
- **Muss zur gewählten Cipher Suite passen**
  - Client muss Zertifikats-Algorithmen unterstützen
  - Server muss evtl. mehrere Zertifikate haben
- **Nachricht optional, da Authentifizierung nicht immer nötig**

- **Daten für Schlüsselaustausch**
  - Hängen von gewählter Cipher Suite ab

- **Daten für Schlüsselaustausch**
  - Hängen von gewählter Cipher Suite ab
- **Nachricht nicht immer nötig**
  - z.B. für Pre-Shared-Key (PSK)

ServerHelloDone **und** ClientKeyExchange

- ServerHelloDone
  - Zeigt an, dass der Server alle Daten gesendet hat

ServerHelloDone und ClientKeyExchange

- ServerHelloDone
  - Zeigt an, dass der Server alle Daten gesendet hat
- ClientKeyExchange
  - Schlüsselaustausch-Daten des Clients
  - Hängt von ausgewählter Cipher Suite ab

ChangeCipherSpec **und** Finished

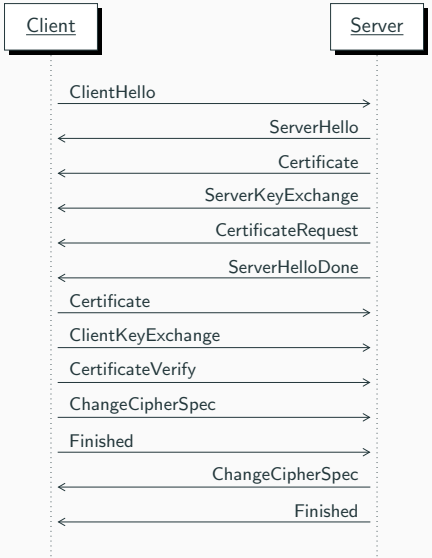
- ChangeCipherSpec
  - Absender hat Schlüssel generiert
  - Zeigt Wechsel zur verschlüsselter Verbindung an

- ChangeCipherSpec
  - Absender hat Schlüssel generiert
  - Zeigt Wechsel zur verschlüsselter Verbindung an
- Finished
  - Schließt Handshake ab
  - Verschlüsselter Inhalt
  - Enthält verify_data Feld (Hash aller Handshake-Nachrichten)

Client-Authentifizierung

- **Server wird oft authentifiziert**
- **Manchmal gegenseitige Authentifizierung nötig**
  - z.B. Machine-to-Machine (M2M) Kommunikation
  - z.B. Business-to-Business (B2B) Kommunikation
  - Aufwendiger auf Client-Seite

Kompletter Handshake mit Client-/Server-Authentifizierung



`CertificateRequest` **und** `CertificateVerify`

- `CertificateRequest`
  - Fordert Client-Authentifizierung

`CertificateRequest` und `CertificateVerify`

- `CertificateRequest`
  - Fordert Client-Authentifizierung
- `CertificateVerify`
  - Client beweist Besitz des privaten Schlüssels
  - Enthält Signatur aller bisher gesendeten Nachrichten

Session Resumption

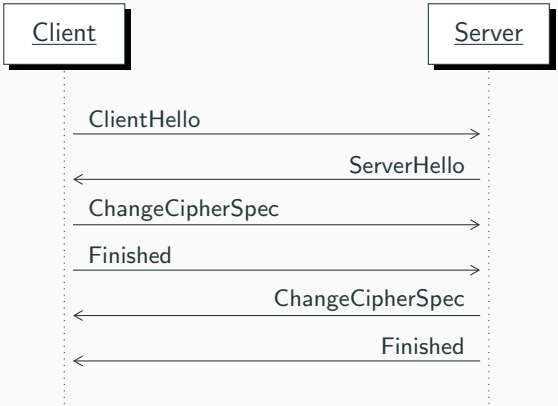
- **Kompletter Handshake aufwendig**
  - Viele Handshake-Nachrichten
  - Zwei Netzwerk-Round-Trips
  - Aufwendige asymmetrische Krypto-Operationen

Session Resumption

- **Kompletter Handshake aufwendig**
  - Viele Handshake-Nachrichten
  - Zwei Netzwerk-Round-Trips
  - Aufwendige asymmetrische Krypto-Operationen

- **Wiederaufnahme einer alten Session**
  - Client und Server müssen Parameter speichern
  - Client sendet Session ID in `ClientHello`
  - Server sendet Session ID in `ServerHello`

Abgekürzter Handshake mit Session Resumption



QUIZ: TLS (twbk.de)

Die Cipher Suite `TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA` gibt unter anderem an, dass RSA zum Schlüsselaustausch verwendet wird. Richtig oder falsch?

A) Richtig, klarer Fall!

B) Falsch, das stimmt nicht!

QUIZ: TLS (twbk.de)

Stellen Sie sich vor, Sie sind dafür verantwortlich, einen Server zu optimieren. Dieser kommuniziert ausschließlich per TLS, schafft daher aber nur 100 Verbindungen pro Minute. Nach kurzer Analyse stellen Sie fest, dass die asymmetrischen Krypto-Operationen die meiste Zeit verschlingen. Welche Überlegung stellen Sie an?

A) Evtl. lässt sich das Problem auch rein symmetrisch lösen.

B) Evtl. sollte ich die AES-Schlüssellänge von 256-bit auf 128-bit reduzieren.

C) Evtl. könnte das Session Resumption Feature das Verfahren beschleunigen.

D) Evtl. sollte ich eine andere Cipher Suite wählen.

TLS v1.3

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 1

- **Entfernung veralteter symmetrischer Algorithmen**
  - Nur noch Authenticated Encryption with Associated Data (AEAD)

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 1

- **Entfernung veralteter symmetrischer Algorithmen**
  - Nur noch Authenticated Encryption with Associated Data (AEAD)

- **Cipher Suites nur noch für symmetrische Algorithmen**
  - z.B. TLS_AES_256_GCM_SHA384

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 1

- **Entfernung veralteter symmetrischer Algorithmen**
  - Nur noch Authenticated Encryption with Associated Data (AEAD)
- **Cipher Suites nur noch für symmetrische Algorithmen**
  - z.B. TLS_AES_256_GCM_SHA384
- **Einführung eines 0-RTT Modus**
  - Anwendungen können direkt verschlüsselt kommunizieren, falls PSK vorhanden
  - Verzicht auf Sicherheitsmechanismen, z.B. Replay-Schutz

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 1

- **Entfernung veralteter symmetrischer Algorithmen**
  - Nur noch Authenticated Encryption with Associated Data (AEAD)
- **Cipher Suites nur noch für symmetrische Algorithmen**
  - z.B. TLS_AES_256_GCM_SHA384
- **Einführung eines 0-RTT Modus**
  - Anwendungen können direkt verschlüsselt kommunizieren, falls PSK vorhanden
  - Verzicht auf Sicherheitsmechanismen, z.B. Replay-Schutz
- **Entfernung von statischem RSA und Diffie-Hellman**
  - Asymmetrischer Schlüsselaustausch immer mit Forward Secrecy

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 2

- **Alle Nachrichten nach `ServerHello` sind verschlüsselt**
  - Wichtig für Handshake-Nachrichten und Extensions

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 2

- **Alle Nachrichten nach `ServerHello` sind verschlüsselt**
  - Wichtig für Handshake-Nachrichten und Extensions
- **Handshake-Nachrichten wurden vereinfacht**
  - `ClientHello` startet Key-Exchange mit vermutlichem Cipher
  - Nur ein Round-Trip notwendig statt zwei
  - `ChangeCipherSpec` nicht mehr notwendig (wird aber aus Gründen der Kompatibilität oft geschickt)

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 2

- **Alle Nachrichten nach `ServerHello` sind verschlüsselt**
  - Wichtig für Handshake-Nachrichten und Extensions
- **Handshake-Nachrichten wurden vereinfacht**
  - `ClientHello` startet Key-Exchange mit vermutlichem Cipher
  - Nur ein Round-Trip notwendig statt zwei
  - `ChangeCipherSpec` nicht mehr notwendig (wird aber aus Gründen der Kompatibilität oft geschickt)
- **Elliptische Kurven sind Teil der Spezifikation**
  - Auch neue Kurven wie z.B. `ed25519` und `ed448`

Änderungen von TLS v1.2 zu TLS v1.3 - Teil 2

- **Alle Nachrichten nach `ServerHello` sind verschlüsselt**
  - Wichtig für Handshake-Nachrichten und Extensions
- **Handshake-Nachrichten wurden vereinfacht**
  - `ClientHello` startet Key-Exchange mit vermutlichem Cipher
  - Nur ein Round-Trip notwendig statt zwei
  - `ChangeCipherSpec` nicht mehr notwendig (wird aber aus Gründen der Kompatibilität oft geschickt)
- **Elliptische Kurven sind Teil der Spezifikation**
  - Auch neue Kurven wie z.B. `ed25519` und `ed448`
- **Weitere kryptographische Verbesserungen**
  - z.B. Entfernung von DSA und der optionalen Komprimierung

Fazit zu TLS v1.3

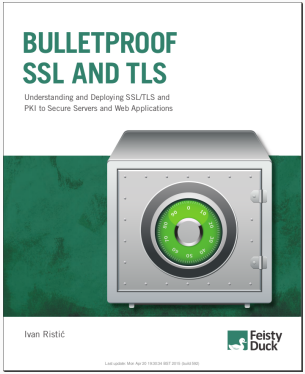
- **Einige Neuerungen**
  - Generell schlanker und mit Performance-Fokus

Fazit zu TLS v1.3

- **Einige Neuerungen**
  - Generell schlanker und mit Performance-Fokus
- **Server und Clients müssen Neuerungen implementieren**
  - Unterstützung/Einführung von TLS v1.3 muss beobachtet werden
  - Link: https://www.ssllabs.com/ssl-pulse/

Hilfreiche Unterlagen

Buch: Bulletproof SSL and TLS



- Themen: Protokolle, Angriffe, Implementierung, und vieles mehr...
- Verfügbar in der HSA Bibliothek

RFCs

- **TLS v1.2**
  - https://tools.ietf.org/html/rfc5246
- **TLS v1.3**
  - https://tools.ietf.org/html/rfc8446

Bekannte TLS Implementierungen

- **OpenSSL**
  - https://github.com/openssl/openssl
- **LibreSSL**
  - https://www.libressl.org/
- **GnuTLS**
  - https://www.gnutls.org/
- **mbed TLS**
  - https://github.com/ARMmbed/mbedtls
- **wolfSSL**
  - https://github.com/wolfSSL/wolfssl

Gibt es noch Fragen?