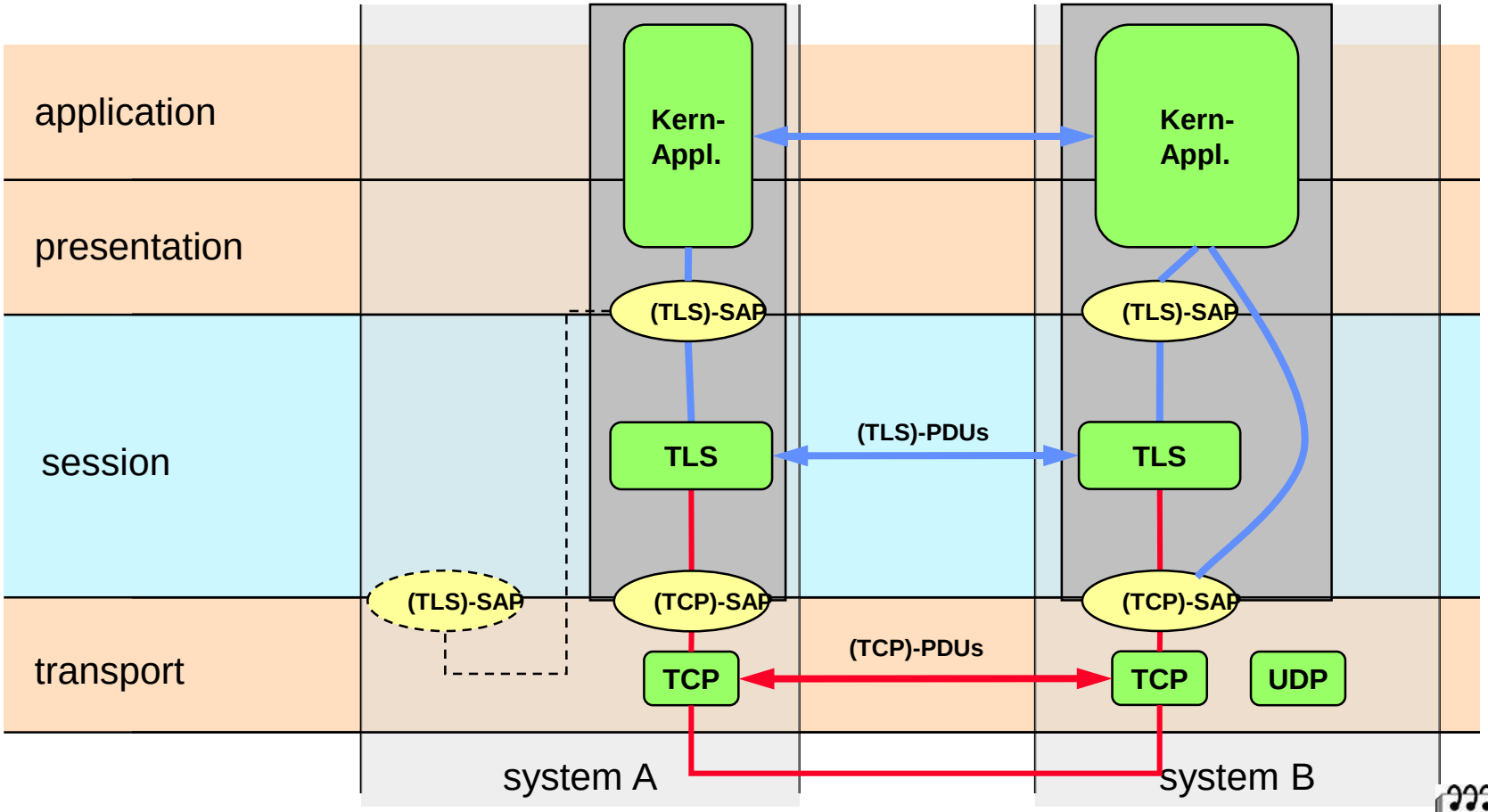
Hochschule
Bonn-Rhein-Sieg

Prof. Dr. Martin Leischner
Netzwerksysteme und TK

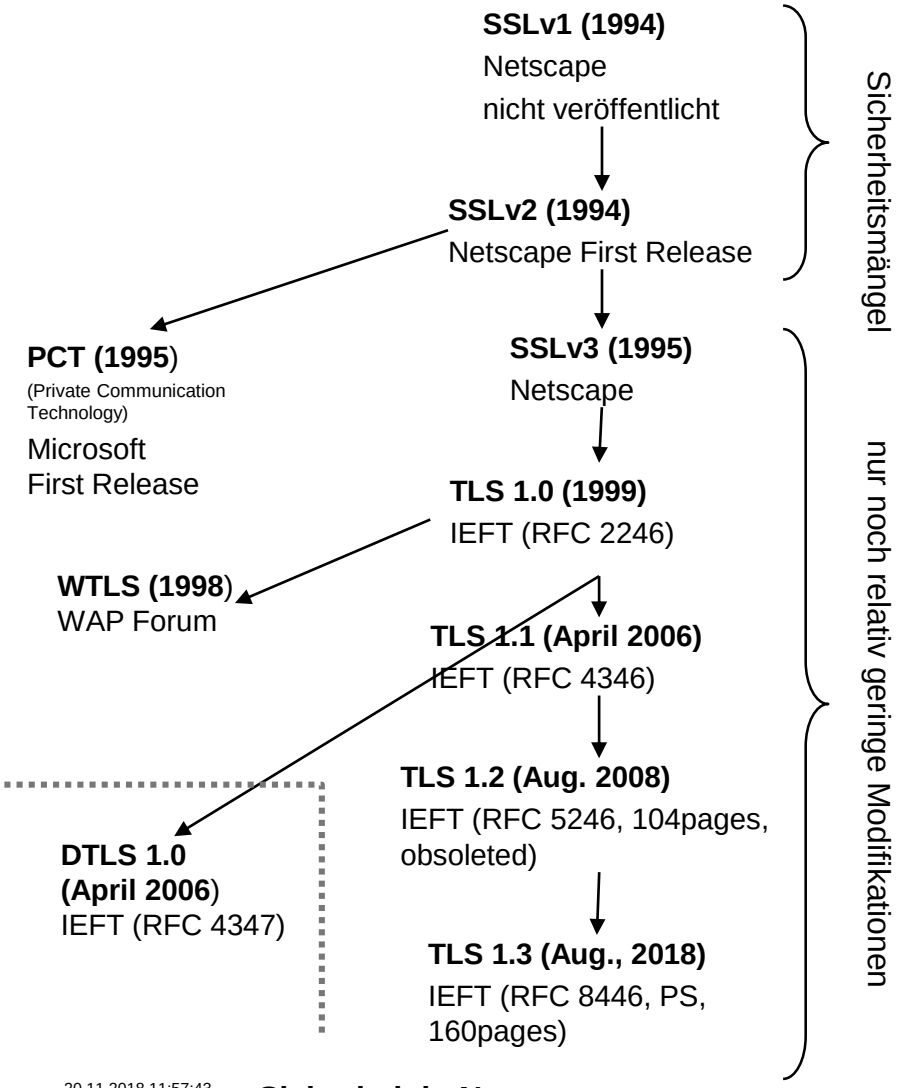
Sicherheit in Netzen
Modul 5: TLS – Transport Layer Security
Teil 1

1. TLS-Einordnung (OSI-Referenzmodell, Geschichte)
2. TLS-Datenübertragung
3. TLS Schlüssel und Algorithmen
4. TLS-Handshake
5. TLS-Implementierung (PDUs, TLS-Sockets)
6. Ausgewählte Angriffe gegen TLS

Einordnung von TLS in die TCP/IP-Architektur und das OSI-Modell



Historie: Entwicklung von SSL zu TLS



Designentscheidungen und Ziele:

- Ende-zu-Ende-Sicherheit
- Einfachheit
- Integrität und Vertraulichkeit (von Datenströmen)
- (Starke, gegenseitige) Authentifizierung möglich
- Schutz vor Replay-Attacken
- Mehrere Verbindungen in einer Sitzung
- SSL/TLS setzt auf TCP auf (→ DTLS)
- Oft keine Integration von SSL/TLS in das Betriebssystem (Kernel)

TLS: Sicherheitsdienst / Grenzen und Probleme

Sicherheitsdienst:

- **Flexible Instanzen-Authentisierung**
  → Client prüft Server, Server prüft Client, Client/Server gegenseitig
- **Vertraulichkeit ausgetauschter Nutzdaten**
  → Nur wenn vereinbart, versch. Verschlüsselungverf. (RC4, DES, Triple-DES, AES, …)
- **Nachrichtenauthentisierung und Datenintegrität**
  → Hashing + Verschlüsselung
- **Schutz gegen Replay-Attacken**

Grenzen und Probleme:

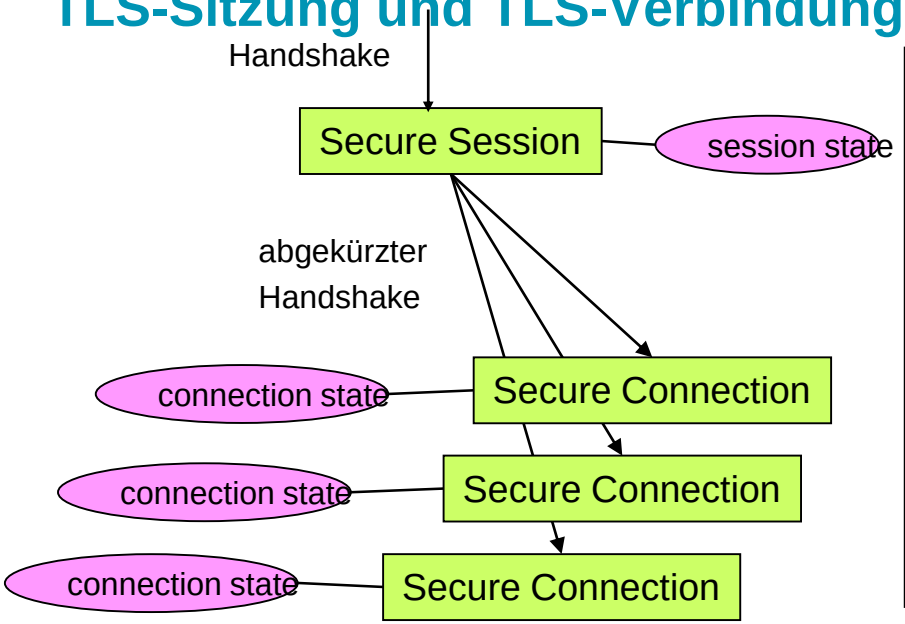
- **Verbindlichkeit von Transaktionen nur aufwändig realisierbar**
- **keine Anonymität**
- **keine (etablierte) Unterstützung UDP-basierter Dienste**
- **kein automatischer Rückruf von Zertifikaten, manuelles Trust-Management**

Prof. Dr. Martin Leischner
Netzwerksysteme und TK

Sicherheit in Netzen
Modul 6: TLS – Transport Layer Security

1. TLS-Einordnung (Geschichte, OSI-Referenzmodell)
2. **TLS-Datenübertragung**
3. TLS Schlüssel und Algorithmen
4. TLS-Handshake
5. TLS-Implementierung (PDUs, SSL-Sockets)
6. Ausgewählte Angriffe gegen TLS

TLS-Sitzung und TLS-Verbindung

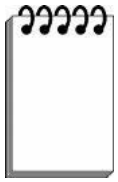


session state

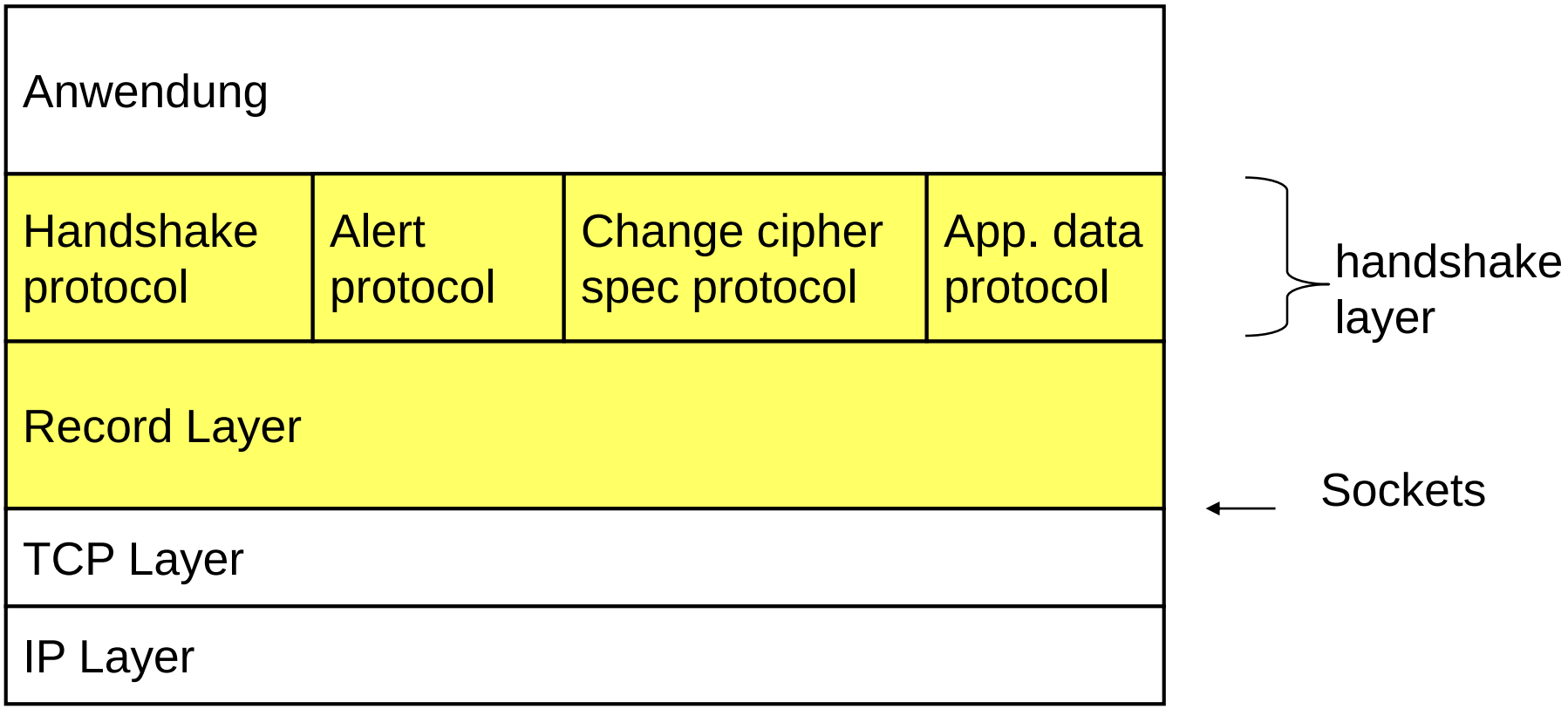
- session identifier: Identifikator der Session
- peer certificate: X509.v3[X509] Zertifikat des Partners (optional).
- compression method: Kompressionsalgorithmus
- cipher spec: definiert Sicherheitsalgorithmen für Schlüsselaustausch, Verschlüsselung, Hashwertbildung
- master secret: 48-Byte Geheimzahl
- is resumable: ein Flag, das anzeigt, ob aus der Session heraus neue Verbindungen aufgebaut werden können.

connection state

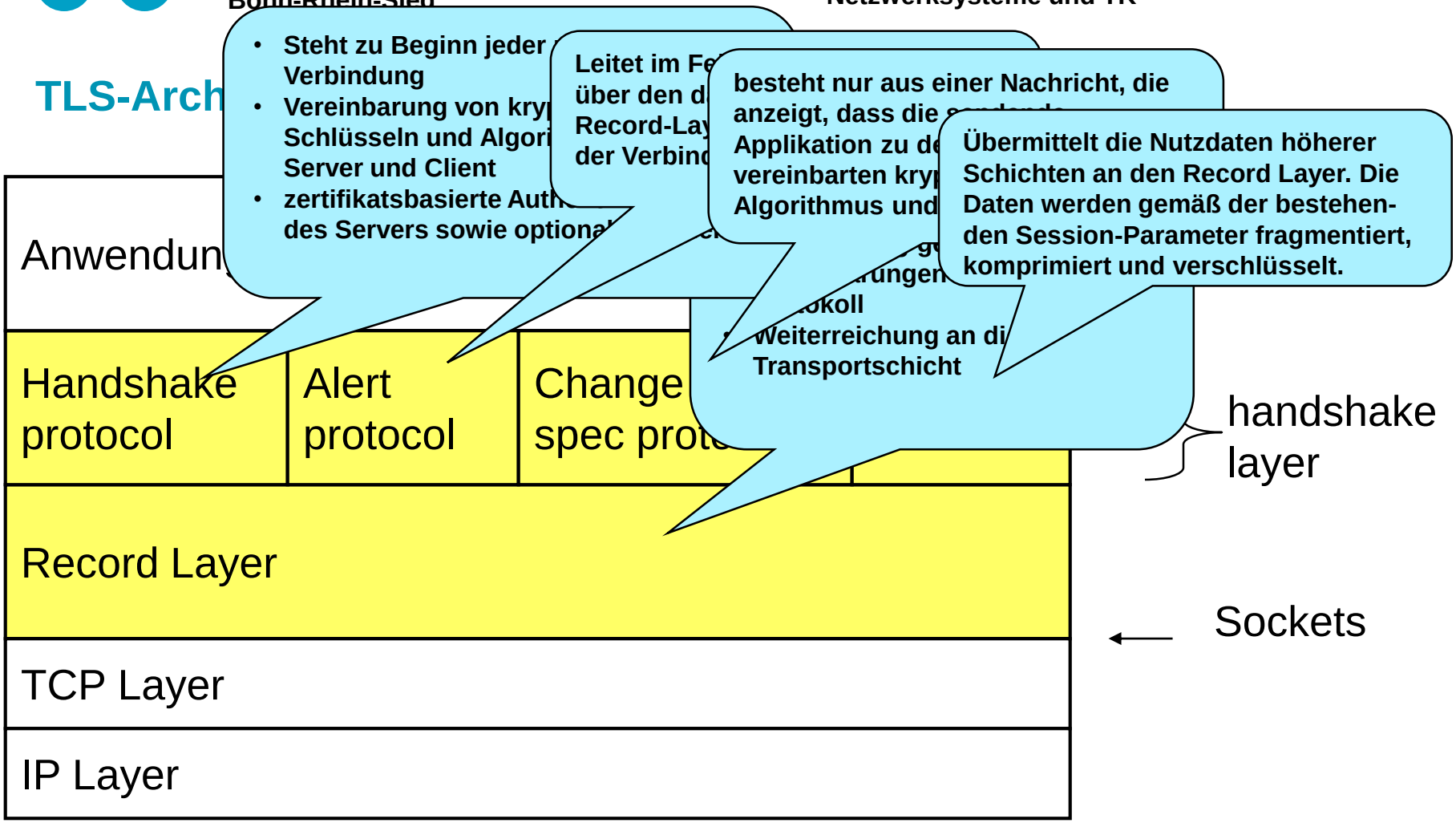
- server and client random: Zufallszahlen, die von Client und Server zu Beginn der Verbindung gewählt werden.
- server / client write MAC secret: die Geheimzahl, die bei MAC bzw. HMAC Operationen auf Daten vom Server angewandt wird.
- server / client write key: Schlüssel für die symmetrische Datenverschlüsselung
- initialization vector: Initialisierungsvektor im CBC-Mode
- sequence numbers: zählen der TLS-Records durch (64-Bit lang)



TLS-Architektur: Feinstruktur TLS



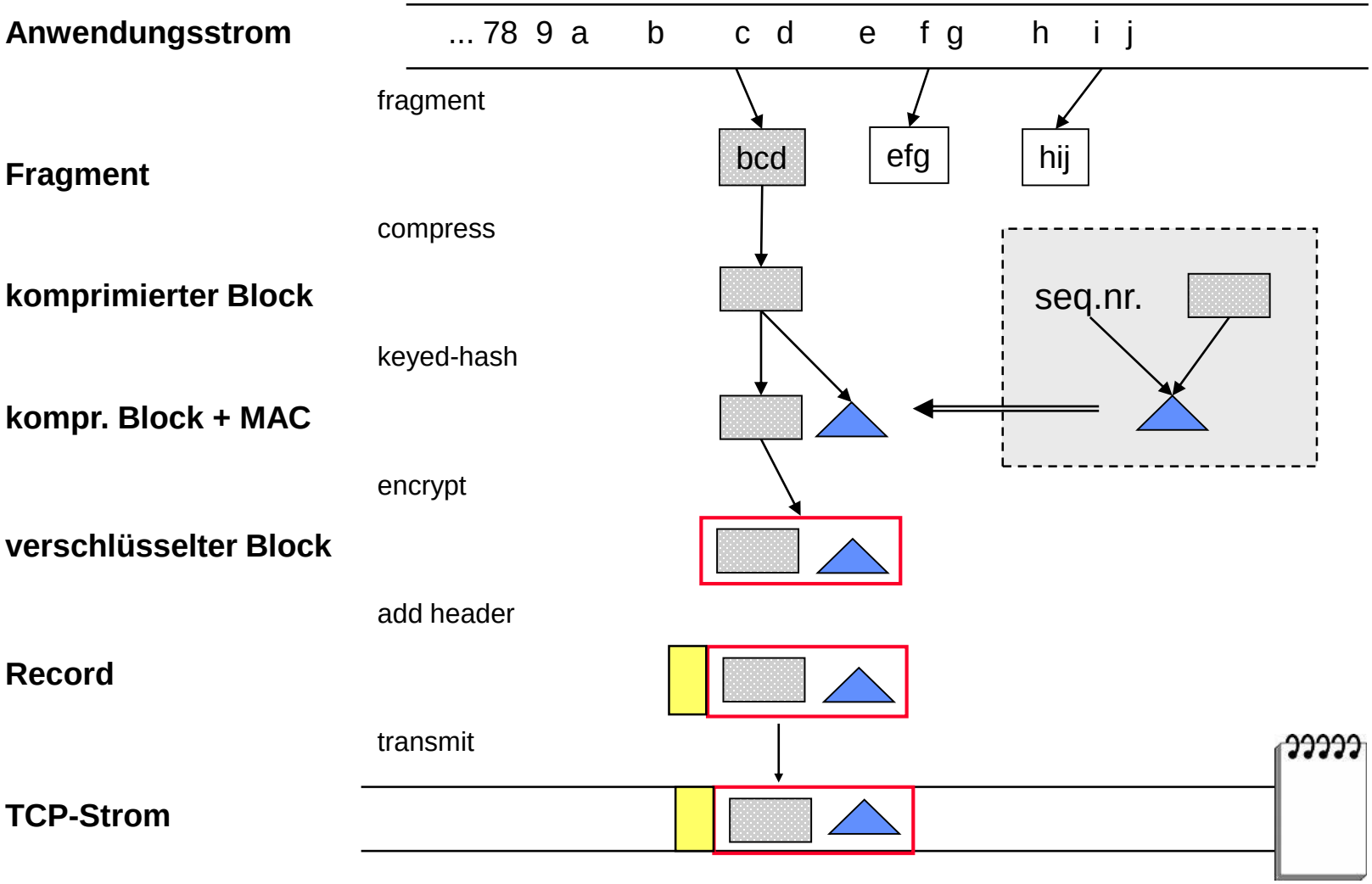
TLS-Arch


• Steht zu Beginn jeder
Verbindung
• Vereinbarung von kry
Schlüsseln und Algori
Server und Client
• zertifikatsbasierte Auth
des Servers sowie optiona
Leitet im Fe
über den d
Record-Lay
der Verbind
besteht nur aus einer Nachricht, die
anzeigt, dass die
Applikation zu d
vereinbarten kryp
Algorithmus und
rungen
okoll
Weiterreichung an di
Transportschicht
Übermittelt die Nutzdaten höherer Schichten an den Record Layer. Die Daten werden gemäß der bestehenden Session-Parameter fragmentiert, komprimiert und verschlüsselt.
Anwendun
Handshake protocol
Alert protocol
Change spec prot
Record Layer
TCP Layer
IP Layer
handshake layer
Sockets

TLS-Architektur: Feinstruktur (blasenfreier Text)

- **Handshake protocol**
  - Steht zu Beginn jeder neuen Verbindung.
  - Vereinbarung von kryptographischen Schlüsseln und Algorithmen zwischen Server und Client
  - zertifikatsbasierte Authentifizierung des Servers sowie optional des Clients
- **Alert protocol:**
  Leitet im Fehlerfall eine Meldung über den darunterliegenden Record-Layer an das andere Ende der Verbindung weiter.
- **Change cipher spec protocol**
  besteht nur aus einer Nachricht, die anzeigt, dass die sendende Applikation zu dem im Handshake vereinbarten kryptographischen Algorithmus und Schlüssel wechselt.
- **App. data protocol:**
  Übermittelt die Nutzdaten höherer Schichten an den Record Layer. Die Daten werden gemäß der bestehen-den Session-Parameter fragmentiert, komprimiert und verschlüsselt.
- **Record Layer:**
  - Fragmentierung der Anwendungsdaten
  - Komprimierung (optional)
  - Absicherung gemäß Vereinbarungen im Handshake-Protokoll
  - Weiterreichung an die Transportschicht

TLS-Architektur: Record-Protokoll



Prof. Dr. Martin Leischner
Netzwerksysteme und TK

Sicherheit in Netzen
Modul 6: TLS – Transport Layer Security

1. TLS-Einordnung (Geschichte, OSI-Referenzmodell)
2. TLS-Datenübertragung
3. **TLS Schlüssel und Algorithmen**
4. TLS-Handshake
5. TLS-Implementierung (PDUs, SSL-Sockets)
6. Ausgewählte Angriffe gegen TLS

Definition von TLS-Cipher-Suites

Eine Cipher-Suite definiert Sicherheitsalgorithmen für

- **Schlüsselaustausch**
- **Verschlüsselung beim Datenaustausch (Record Protocol)**
- **Hashwertbildung**

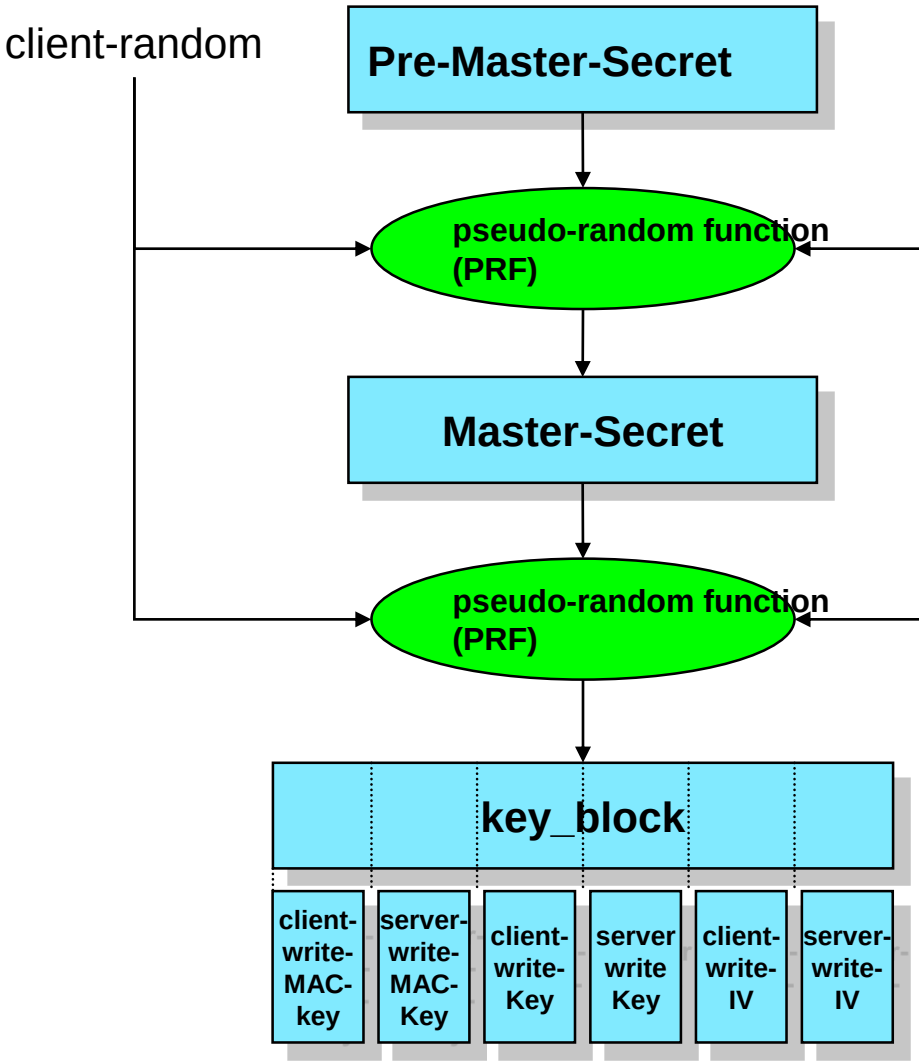
Beispiele:

- TLS_NULL_WITH_NULL_NULL
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA

bedeutet:

- Schlüsselaustausch mit ECDH_ECDSA (= Fixed ECDH (= elliptic Curve Diffie-Hellman) with ECDSA-signed certificates)
- AES mit 128 Bit Blocklänge im Cipher Block Chaining Mode
- Der Secure Hash Algorithm SHA-1 wird zur Berechnung des MAC (message authentication code) benutzt.
- Der Zahlencode für TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA ist 0xC0, 0x04, siehe RFC 4492 Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)

Schlüsselhierarchie und Schlüsselgenerierung

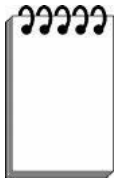


hängt von Key-Exchange-Algorithmus ab

Die recht komplexe Bildung der PRF (basierend auf SHA-256) ist in RFC5246, 5. HMAC and the Pseudorandom Function beschrieben

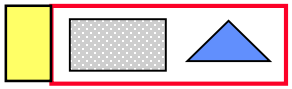
Immer 48 Byte lang

wird für jede Verbindung in der Session neu erzeugt



Beispiel: Berechnung des Record-MAC-Werts

HMAC_hash (MAC_write_key,
seq_num + type + version + length + fragment)



- **hash** ist der gekeyed MAC-Algorithmus, auf den sich beide Partner geeinigt haben, z.B. HMAC-SHA
- **seq_num** ist die Sequenz Nummer für diesen Rekord
  - Die Sequenznummer (64 bit) wird nicht explizit übertragen
  - Diese Sequenz Nummer ist in keiner Weise mit der Sequenznummer von TCP korreliert
  - Die Sequenznummer wird auf Null gesetzt, wenn die TLS-Verbindung gestartet wird
  - die Sequenznummer wird nach jedem Rekord erhöht
  - nach einem Overflow muss das Schlüsselmaterial neu verhandelt werden