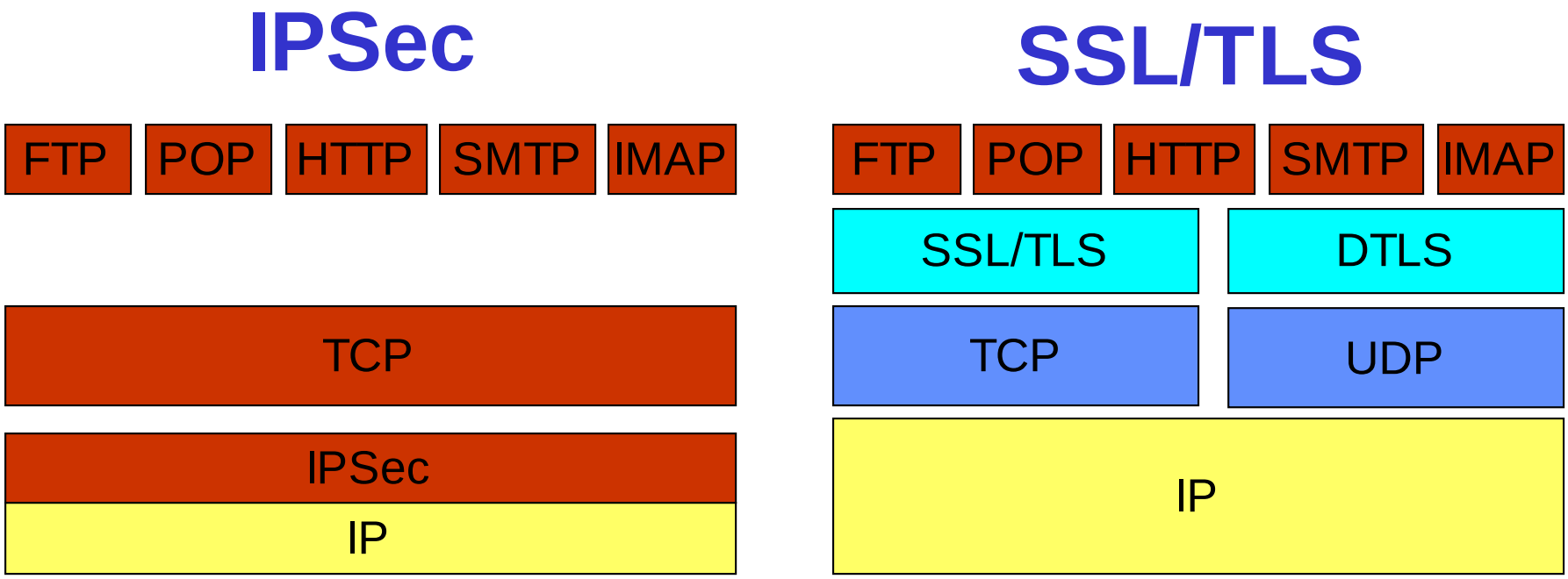
TLS ALS BEISPIEL FÜR EIN SICHERHEITSPROTOKOLL

Kleine Auswahl bekannter Sicherheitsprotokolle

- **X.509 Zertifikate / PKIX**
  - Standardisierte, häufig verwendete Datenstruktur zur Bindung von kryptographischen Schlüsseln an eine Person
  - Eventuell gekoppelt mit zusätzlichen Attributen
  - Hierarchische Baum-Struktur des “Vertrauens”
- **SSL / TLS / DTLS**
  - Secure Socket Layer / Transport Layer Security / Datagram TLS
  - symmetrisch verschlüsselte und integere Verbindung oberhalb TCP/IP / UDP/IP
  - Tunnelung von Anwendungsprotokollen wie HTTP, FTP SMTP, IMAP etc.
  - Identifizierung von Server oder Client mittels X.509 Zertifikaten
  - Aushandelbare Algorithmen
- **SSH**
  - gedacht für sicheren Remote-Zugriff
  - Verschiedene Authentifizierungsmöglichkeiten z.B. X.509 Zertifikate, Paßwort
  - aber z.B. auch Port-Forwarding möglich --> sicherer Tunnel
- **IPSec (VPN)**
  - Sicherheitsarchitektur oberhalb von IP
  - sicherer Tunnel für IP-Verkehr --> IP-Pakete werden verschlüsselt und dann als Payload verschickt
  - völlig transparent für Anwendungsprogramme
  - gesicherte Anbindung von Rechnern an Netz oder Verbindung von Netzen

Layering: IPSec vs. SSL/TLS



Eine kurze Geschichte von SSL/TSL

- ursprünglich für gesicherte Web-Kommunikation entwickelt:
  - 1994: SSL 1.0 von Netscape Communications (für Mosaic)
  - wenig später: SSL 2.0 für Netscape Navigator
  - 1996: SSL 3.0
    - behebt Sicherheitsprobleme von SSL 2.0
    - nachzulesen in RFC 6101
  - 1999: TLS 1.0 standardisiert als IETF RFC 2246
    - im Wesentlichen SSL 3.0
  - 2006: TLS 1.1 in RFC 4346
    - behebt Sicherheitsproblem von TLS 1.0 im Zusammenhang mit Betriebsart CBC
  - 2008: TLS 1.2 in RFC 5246
    - Verbesserungen bzgl. verwendbarer Hash-Funktionen
- mittlerweile in sehr vielen Protokollen verwendet, die sichere Ende-zu-Ende Kommunikation erfordern

TLS: Prinzipieller Ablauf

1. Aushandlung von Sicherheitsparametern (Handshake):
   - kryptographische Algorithmen
   - symmetrische Schlüssel
   - Authentifizierung von Server und Client (optional)

2. Aktivierung & Überprüfung der Sicherheit
   - Aktivierung durch Change Cipher Spec Nachricht
   - Überprüfung auf korrekte (gemeinsame) Sicherheitsparameter mit Finished Nachricht

3. sichere Übertragung der Anwendungsdaten

4. Beendigung der gesicherten Kommunikation

TLS Protokollstapel

TLS Handshake Protocol

TLS Change Cipher Spec. Protocol

TLS Alert Protocol

TLS Application Data Protocol

TLS Record Protocol

Content type

Length

Version

Payload

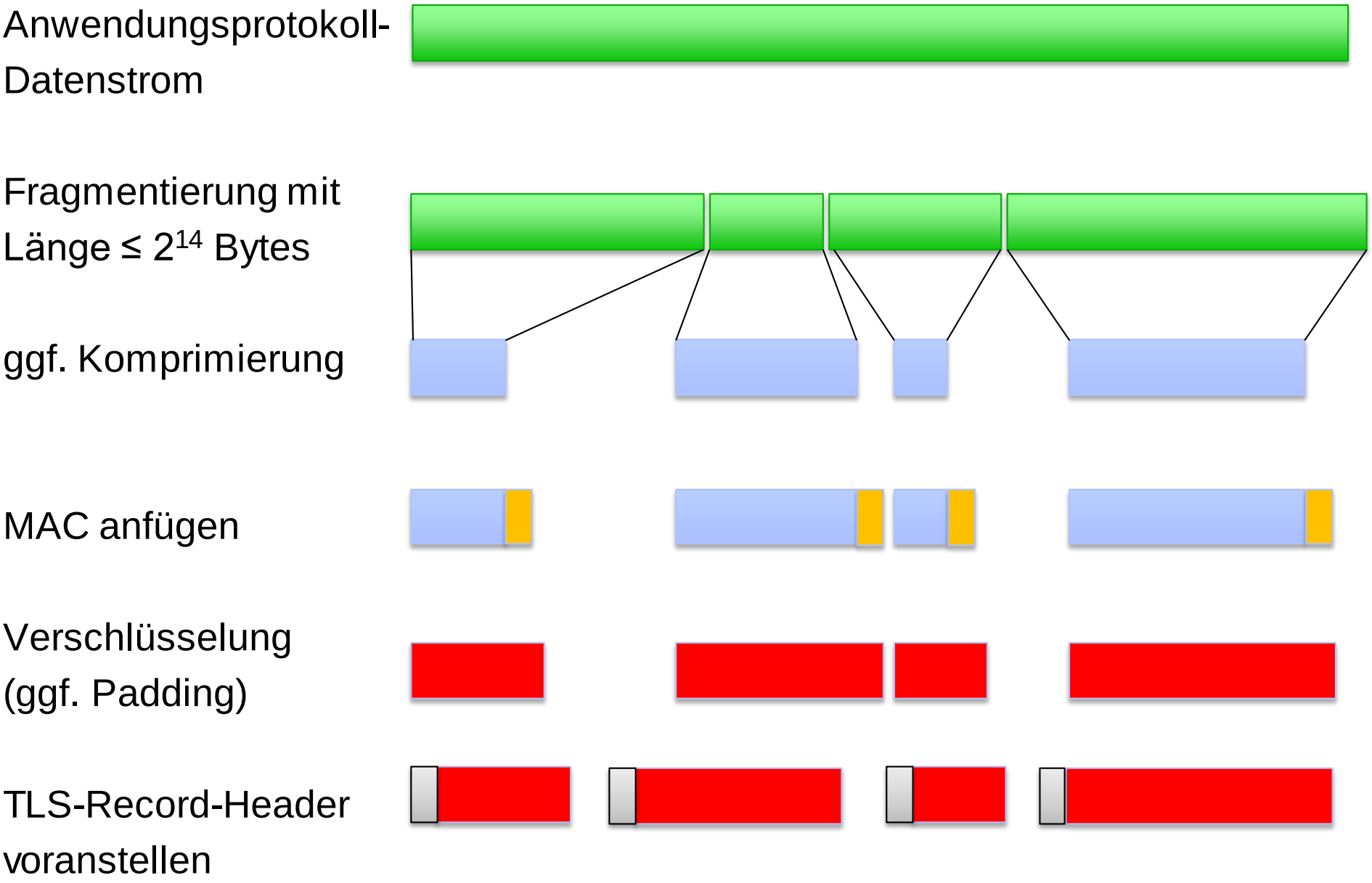
MAC

Padding

Ziel: gesicherte Ende-zu-Ende Kommunikation

- **Vertraulichkeit: symmetrische Verschlüsselung**
- **Integrität: MAC**

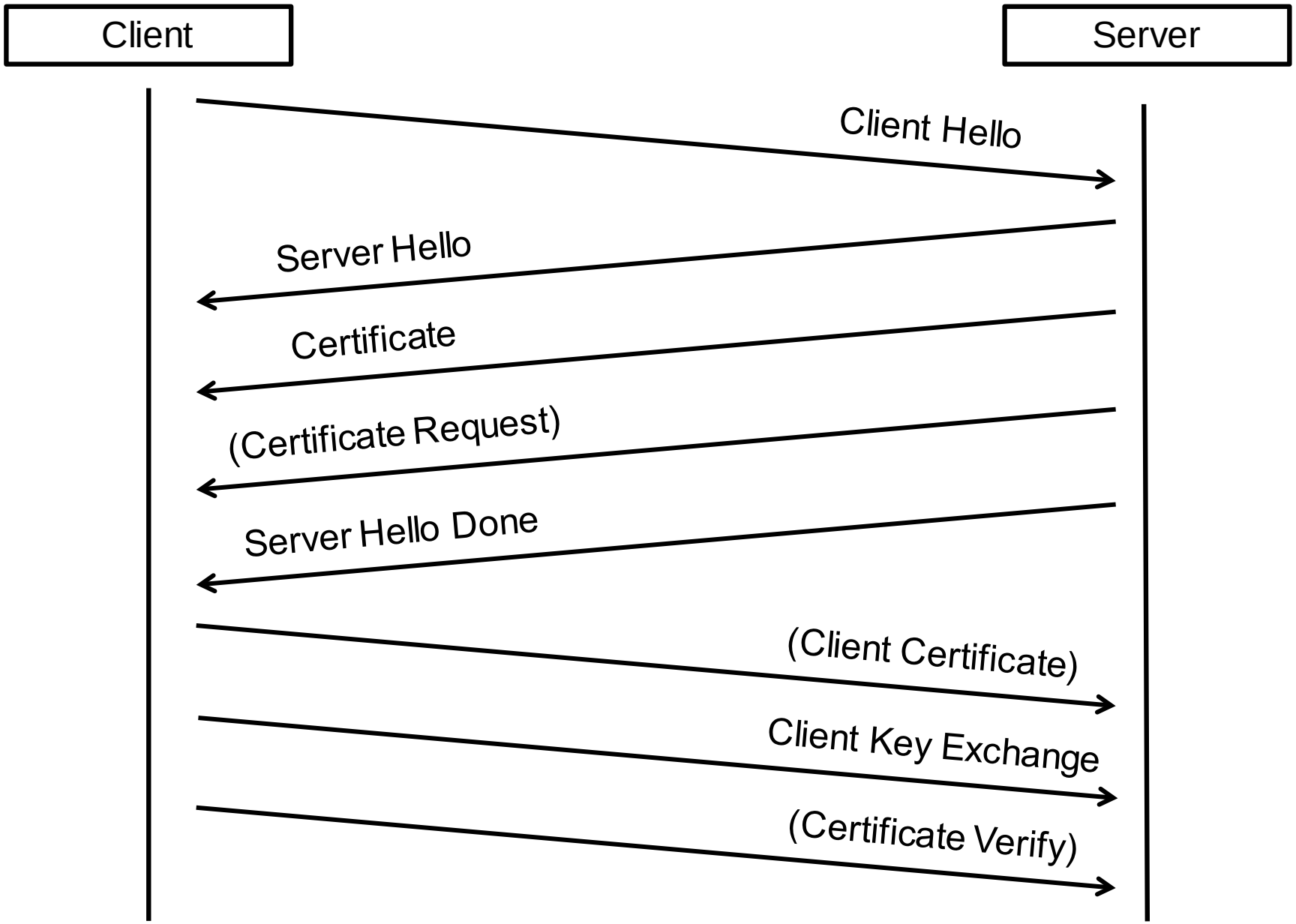
TLS: Sicherung des Anwendungsprotokolls



TLS Handshake: Ziele

- Aushandlung der zu verwendenden kryptographischen Algorithmen:
  - Client schickt Liste aller unterstützten Algorithmen
  - Server wählt aus und informiert den Client
- Gegenseitige Authentifizierung (optional)
  - für sichere Kommunikation notwendig
    - andernfalls: Man-In-The-Middle-Angriffe leicht möglich
  - üblich: Server authentifiziert sich durch Übermittlung von X.509-Zertifikat
    - ggf. zusätzlich: nur (vor-)definierte Zertifikate zulässig (*certificate pinning*)
  - möglich: Server fordert Client-Zertifikat an
  - alternativ: Verwendung von Pre-shared-Keys
- Schlüsselaustausch

TLS Handshake: Ablauf



TLS Handshake: Nachrichten

- Client Hello, Server Hello:
  - Aushandlung der Sicherheitsmechanismen
  - Austausch von Zufallszahlen (nonce)
    - Challenge-Response Authentifizierung
    - Freshness (Verhinderung Replay-Angriffe)
- Certificate:
  - Authentifizierung
- Client Key Exchange:
  - Übermittlung des Pre-Master-Secret an den Server (verschlüsselt)
  - Berechnung des Master-Secret aus Pre-Master-Secret & Zufallszahlen
    - Ableitung der kryptographischen Schlüssel mit Hilfe von Key Derivation Function (KDF)
- Abschluß: Change Cipher Spec
  - „Aktivierung“ der Sicherheitsmechanismen

TLS Anmerkungen

- TLS ist erweiterbar mit:
  - neuen symmetrischen Verschlüsselungsverfahren
  - neuen Schlüsselaustauschverfahren
  - neuen MAC-Verfahren
- Erweiterungen meist in eigenen RFCs spezifiziert

TLS Risiken & Probleme

- technische Risiken:
  - kryptographische Schwächen im Protokoll / den verwendeten Algorithmen
  - Implementierungsfehler
  - (Serverseitige) Konfigurationsfehler
    - unsichere Algorithmen (RC4)

➔ vergleichsweise leicht zu beheben

- menschliche Risiken / Schwächen
  - fehlendes bewußtes Benutzen von TLS
  - Überprüfung der Authentizität

➔ schwer zu lösen

- zusätzliche (organisatorische) Risiken:
  - Kompromittierung der Zertifikate / geheimen Schlüssel

  ➔ erhebliches Risiko