LEHRSTUHL 14
SOFTWARE ENGINEERING

Willkommen zur Vorlesung

Methodische Grundlagen des Software-Engineering

im Sommersemester 2011

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

21. Kryptographische Protokolle

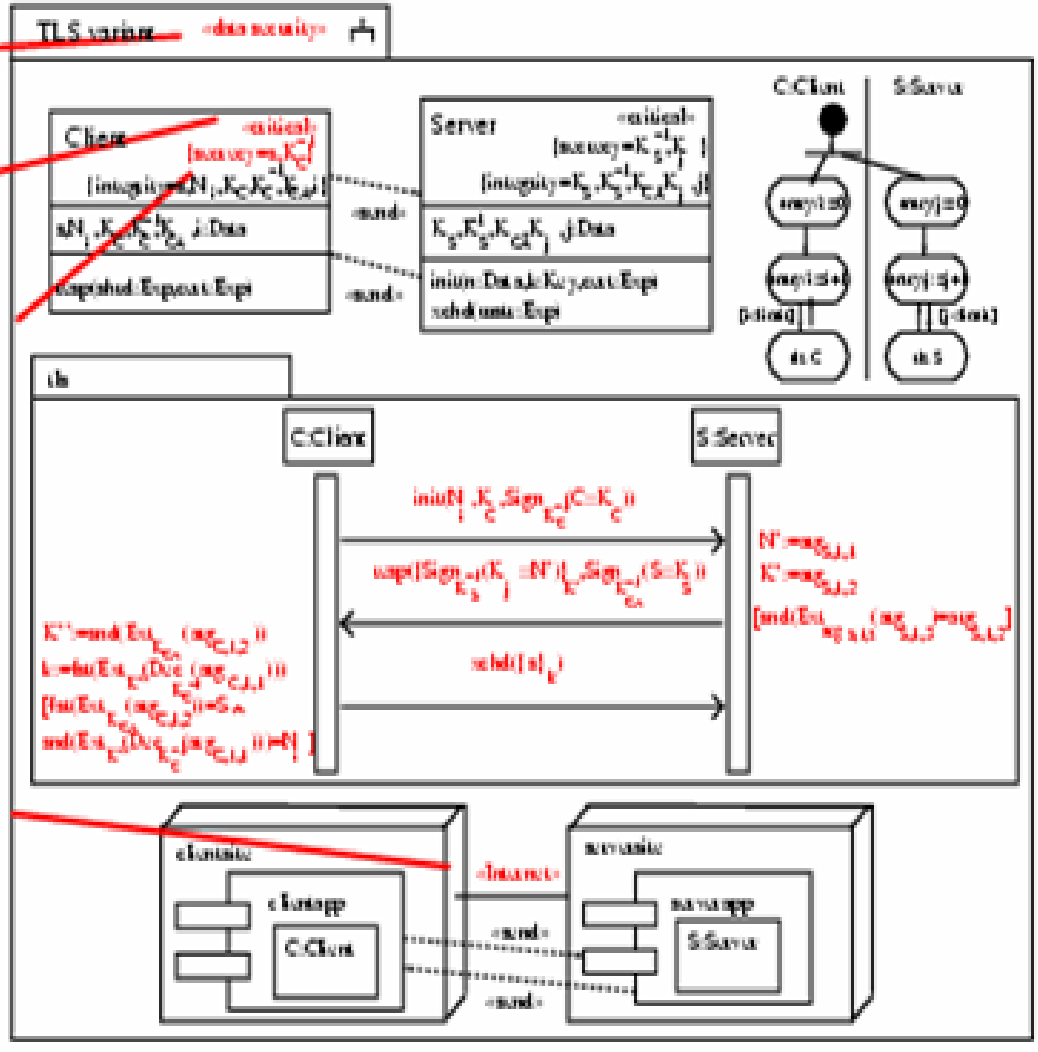
Einordnung
Protokolle mit UMLsec

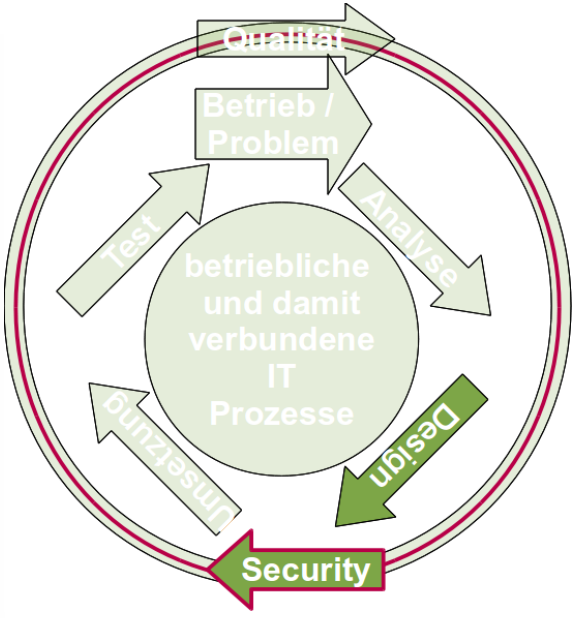
«data security»

«critical»

$\{secrecy = \{s, K_C^{-1}\}\}$

«Internet»





Einordnung
Einführung UML / UMLsec

- Business Prozesse
- Qualitätsmanagement
- Testen
- Sicherheit
- Sicheres Software Design
  - Einführung UML / UMLsec
  - Kryptographische Protokolle
  - Biometrische Authentifizierung
  - Elektronische Geldbörse
  - Weitere Anwendungsbeispiele

Sicherheitsprotokolle

System über nicht vertrauenswürdiges Netzwerk verteilt.

Angreifer kann Nachrichten abfangen, modifizieren, löschen und einfügen.

Kryptographie ermöglicht Sicherheit.

Kryptographisches Protokoll: Austausch von Nachrichten für verteilte Sitzungsschlüssel, authentisierenden Auftragsgebern etc. durch Benutzung von kryptographischen Algorithmen.

Korrekter Entwurf sehr schwierig.

Beispiel: Authentisierungs-Protokolle

Ziel: Sichere Authentisierung von Kommunikationspartnern.

Bedrohungen:

- Fälschung von Identitäten
- Unautorisierte Verwendung von Identitäten

Weitere Ziele von Sicherheitsprotokollen:

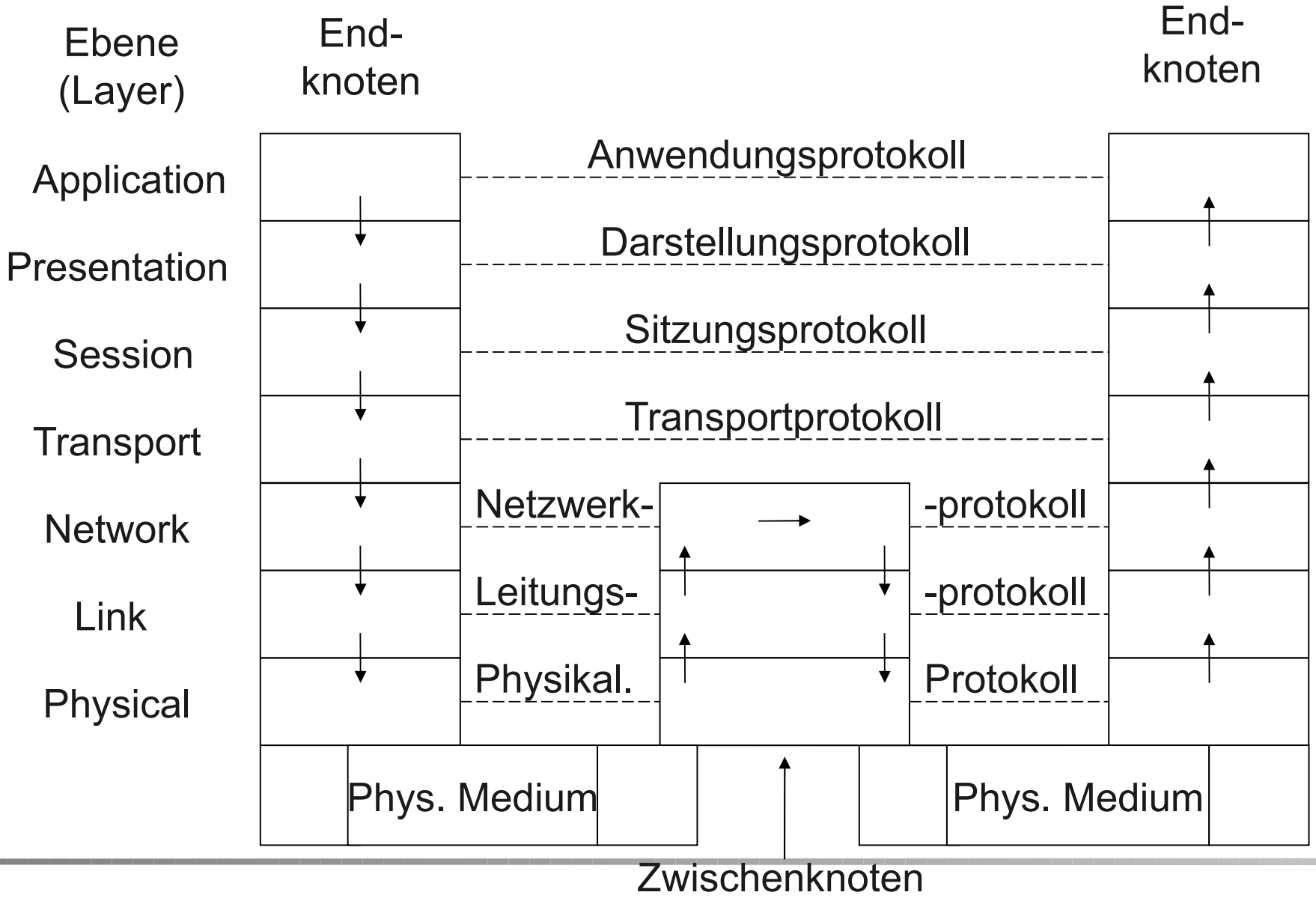
Schlüsselmanagement, elektronische Transaktionen, ...

Sicherheitsprotokolle: Probleme

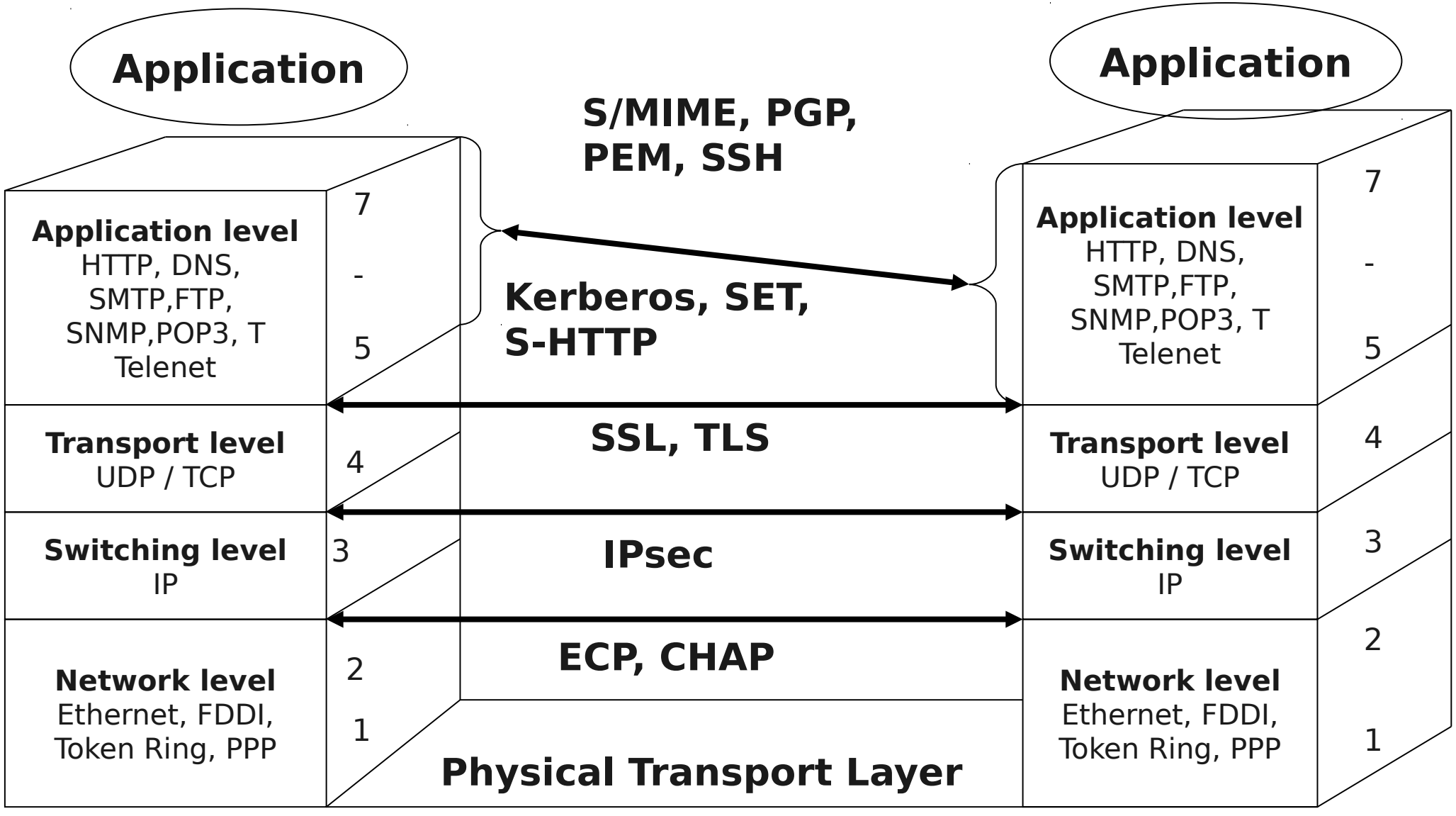
Viele Protokolle haben Schwachstellen aus unterschiedlichen Gründen:

- schwache Kryptographie
- Zentraler Nachrichten-Austausch
- Schnittstellen, Prologe, Epiloge
- Verwendung
- Implementierungsfehler

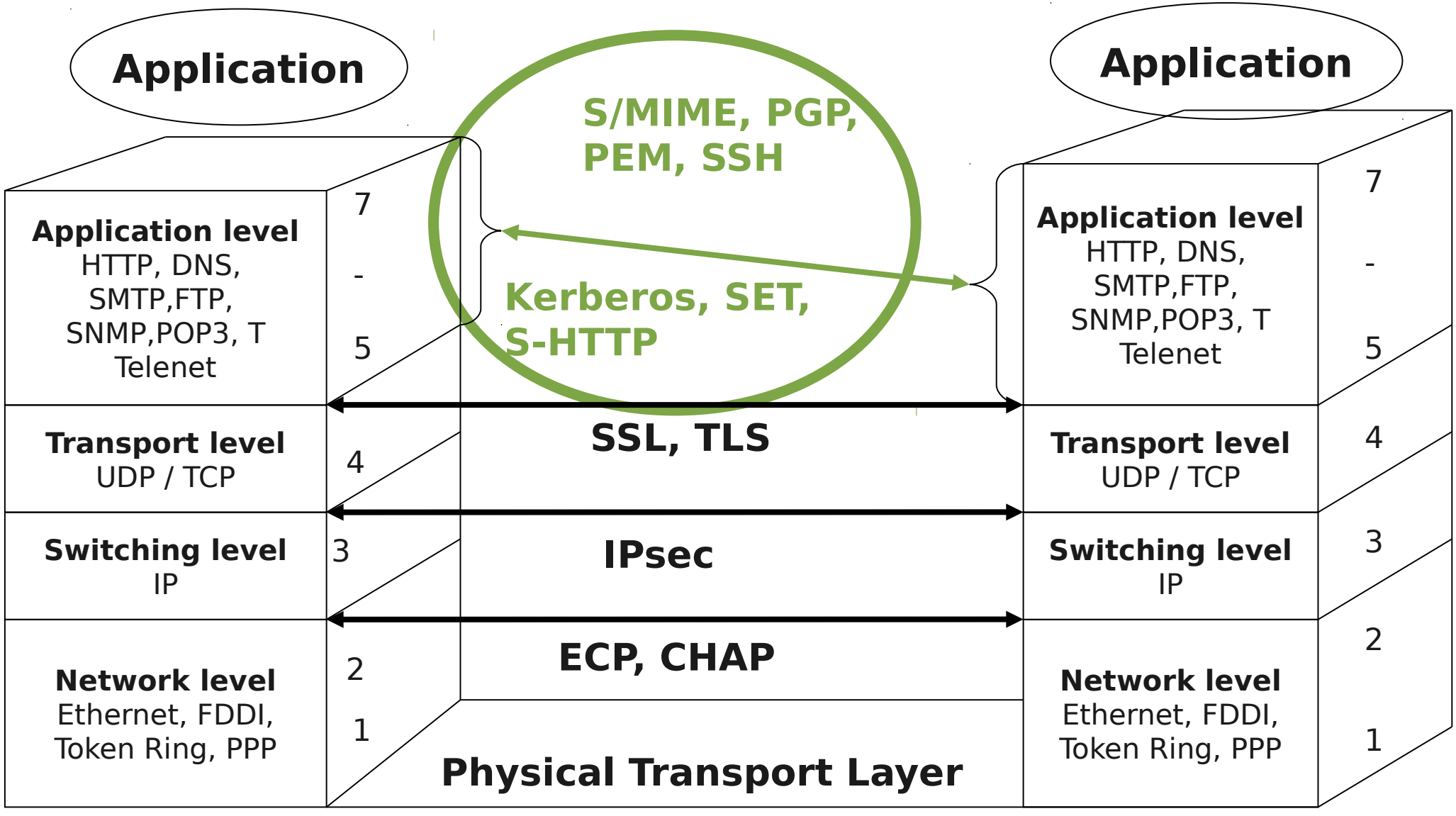
ISO OSI 7-Schichten-Modell



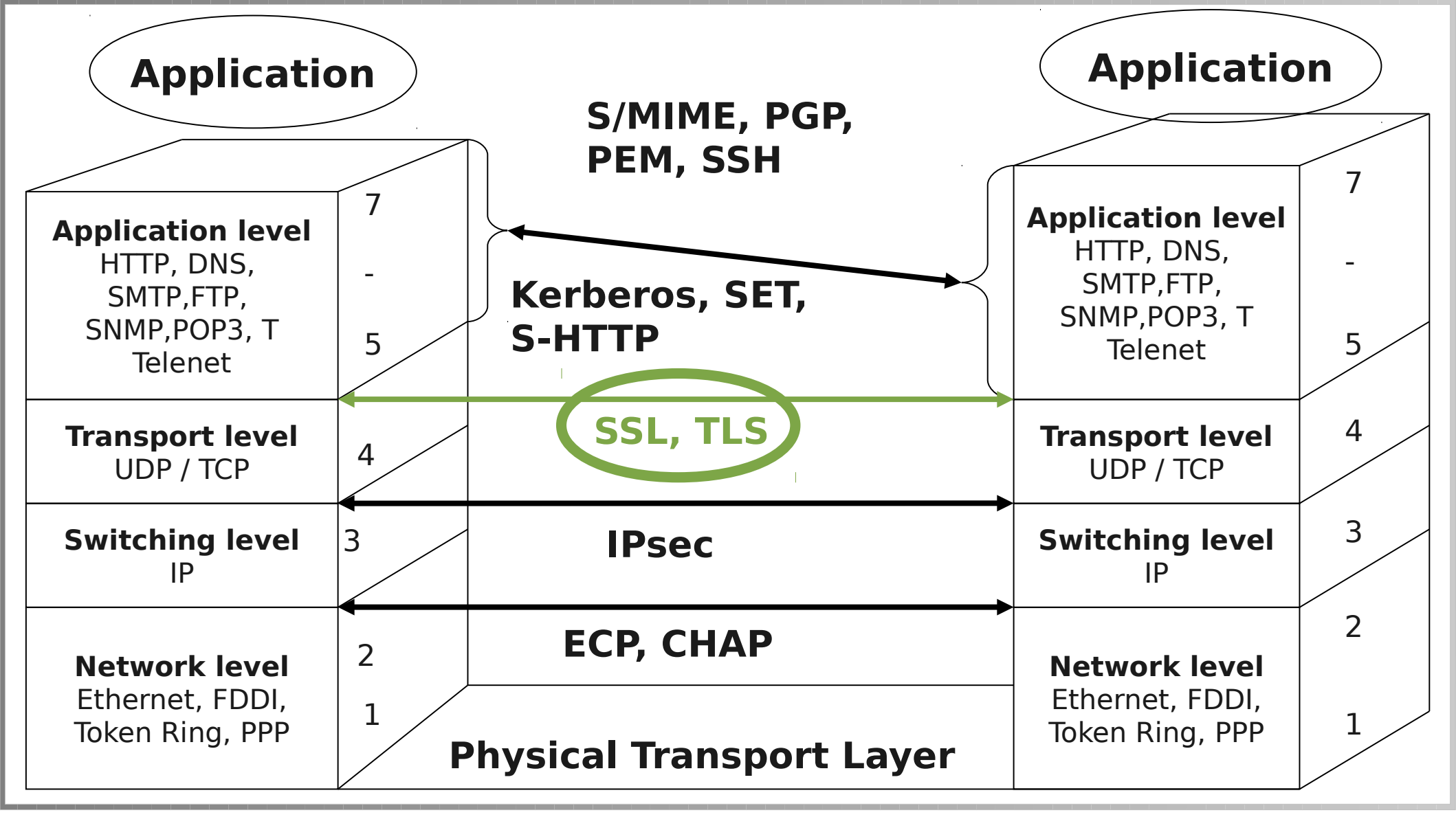
Verschlüsselung und Protokollebenen



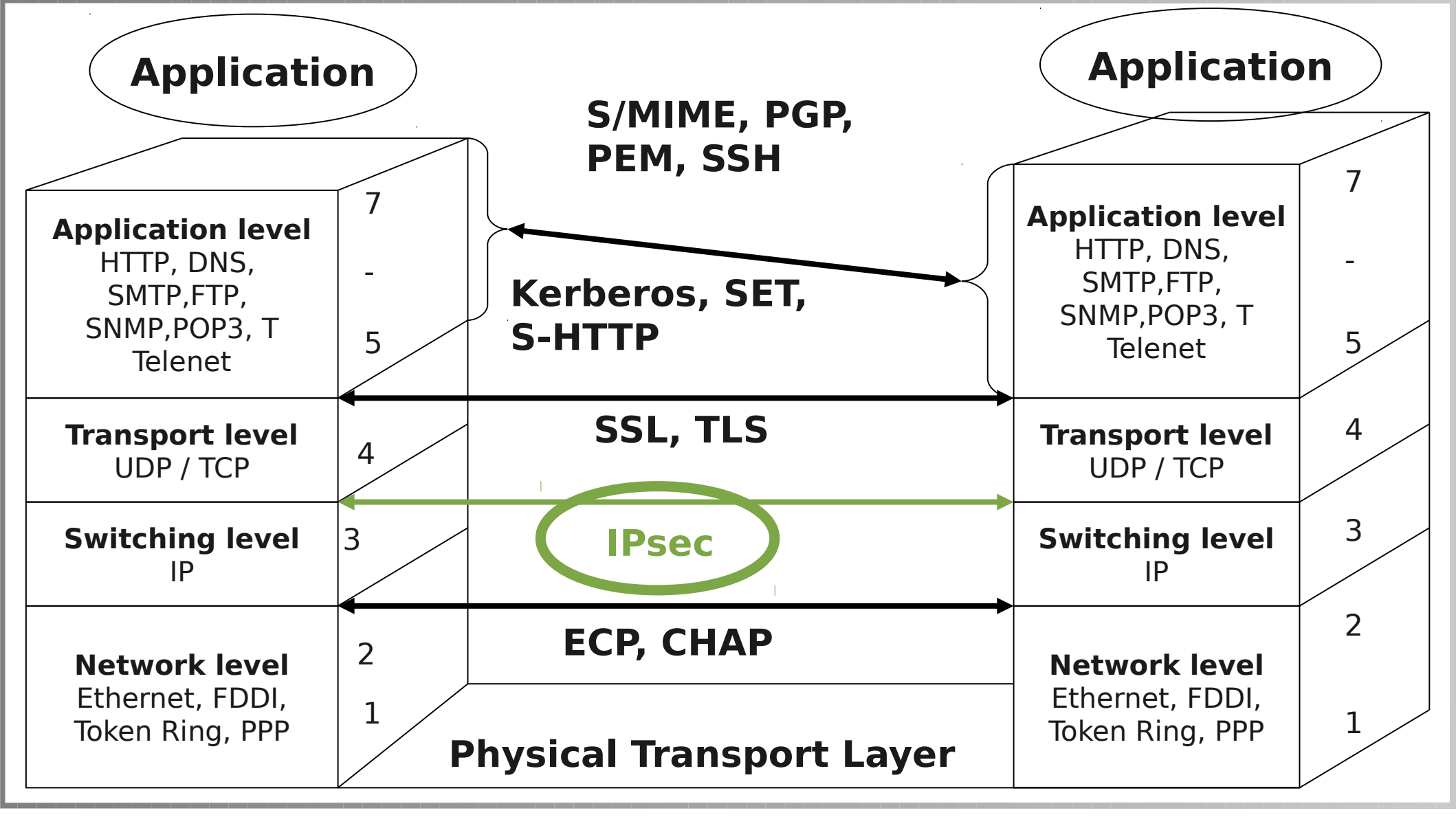
Verschlüsselung und Protokollebenen



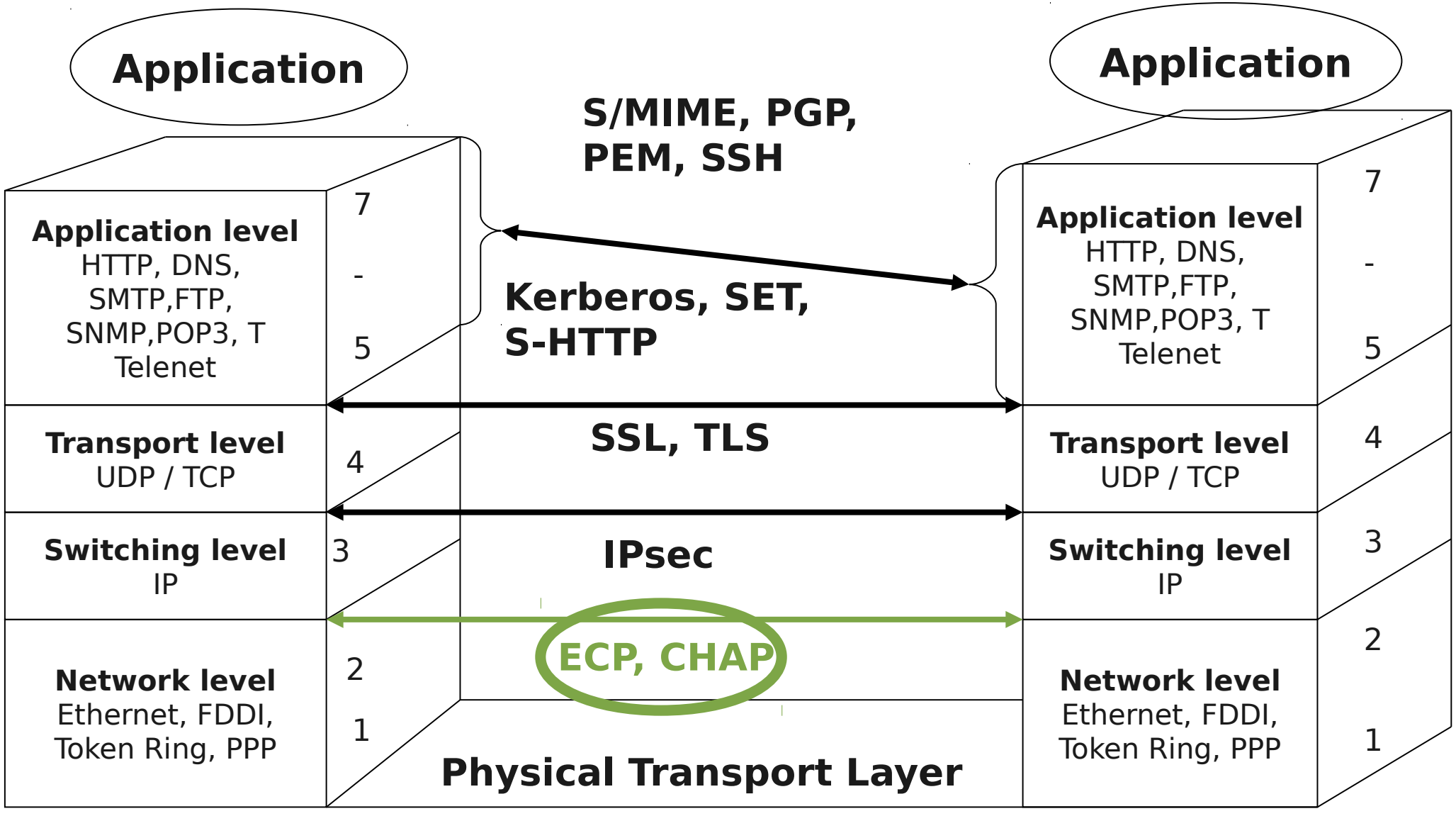
Verschlüsselung und Protokollebenen



Verschlüsselung und Protokollebenen



Verschlüsselung und Protokollebenen

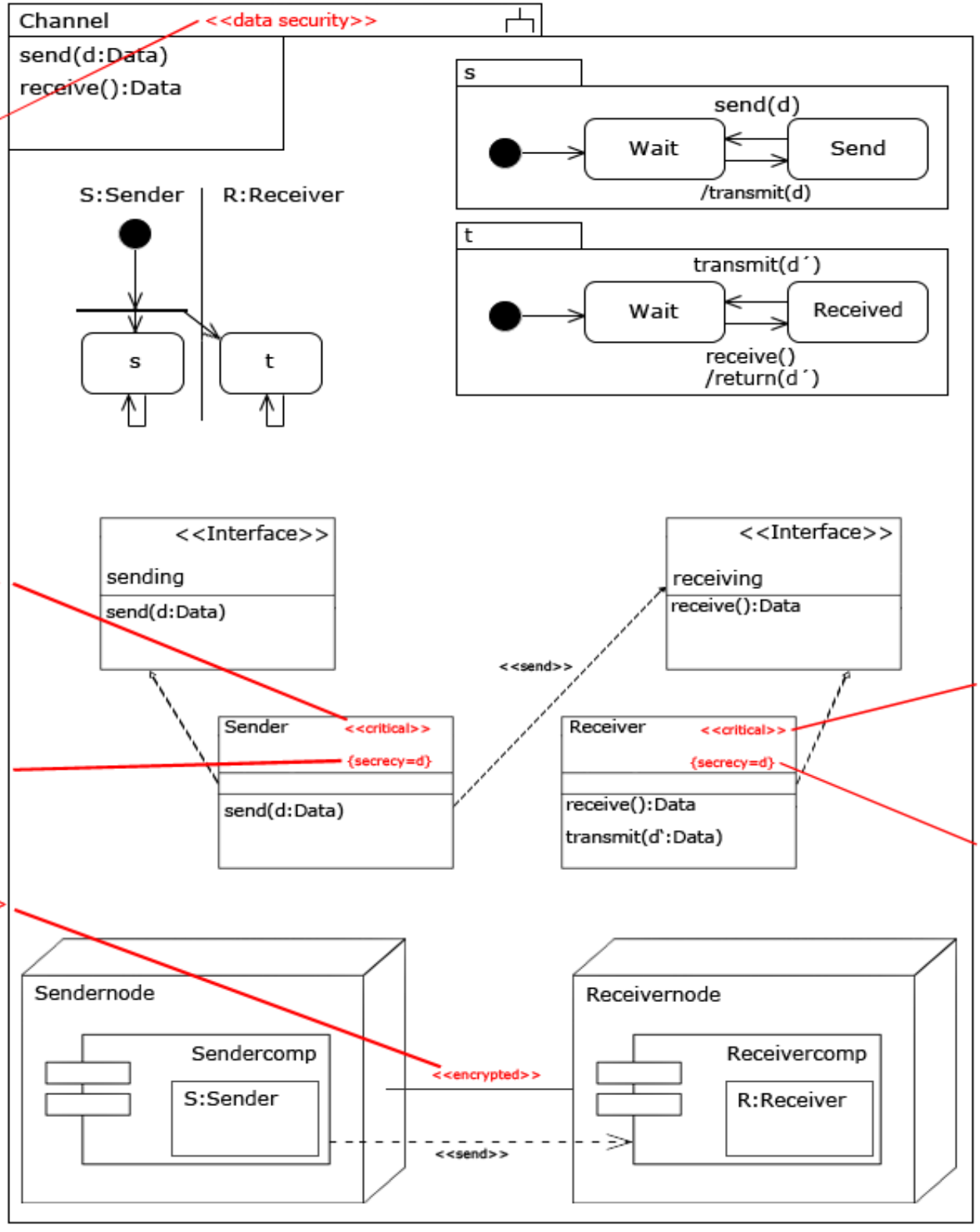


Beispiel:

Sicherer Kanal

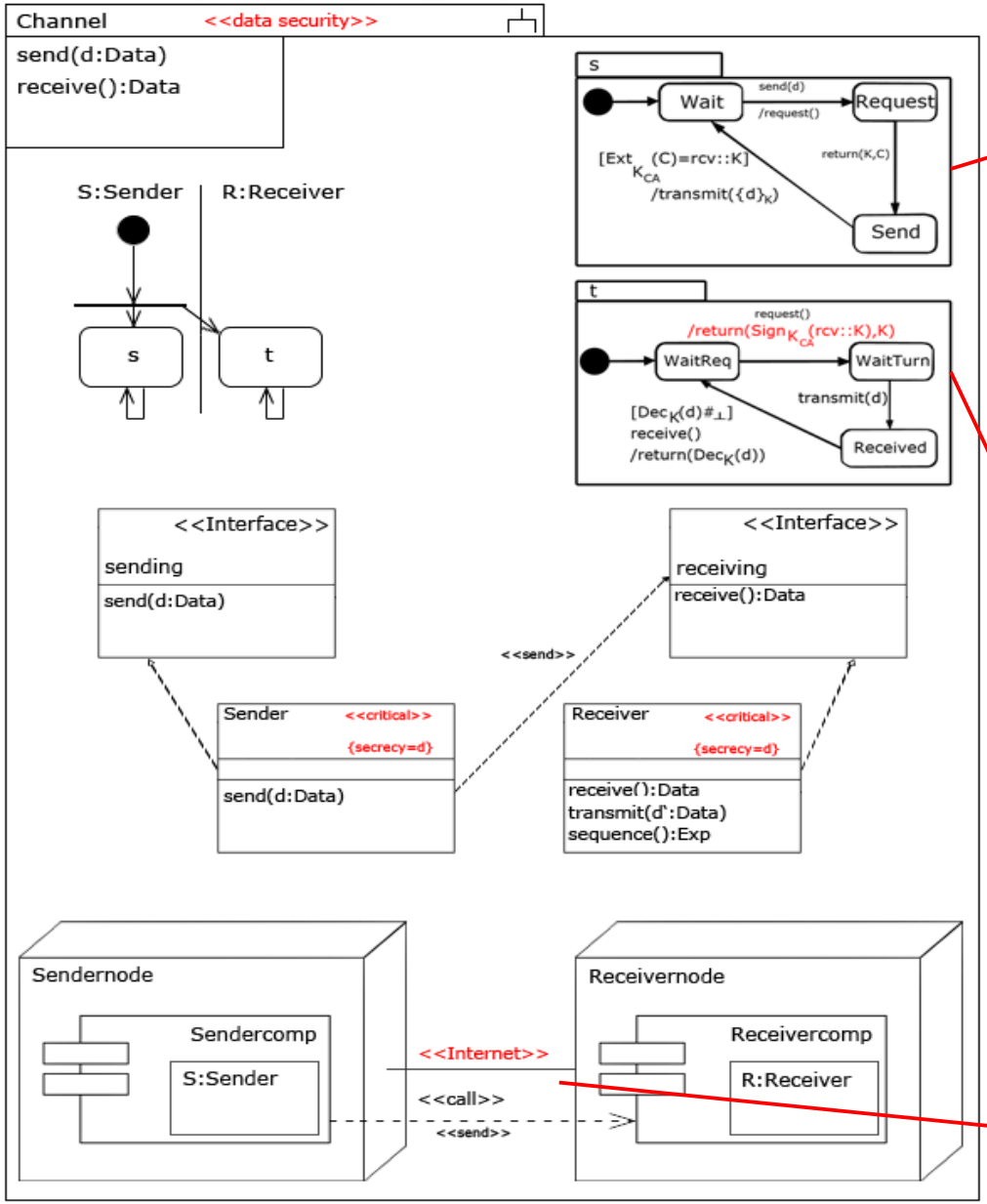
Ziel:

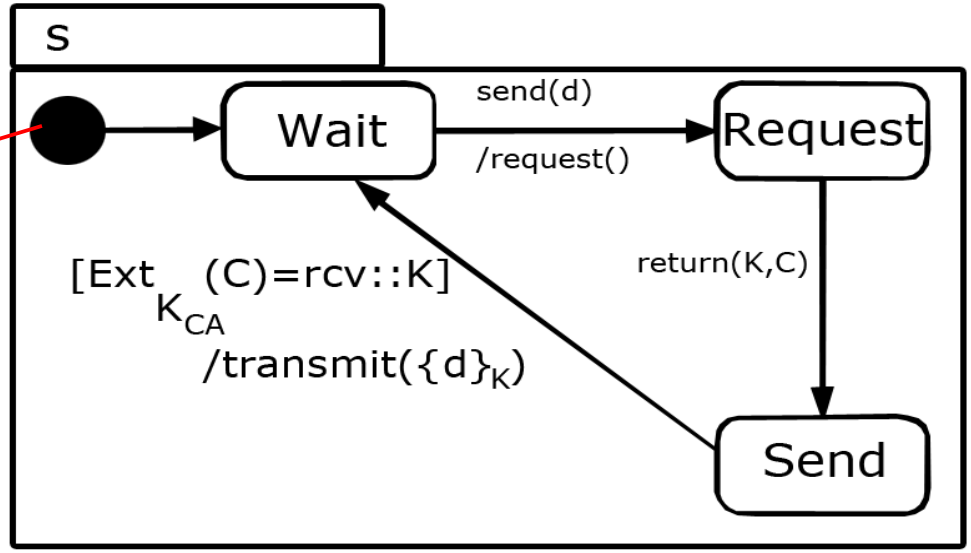
vertrauliche Übertragung von Daten über ungeschützte Kommuni-Kations-verbindung

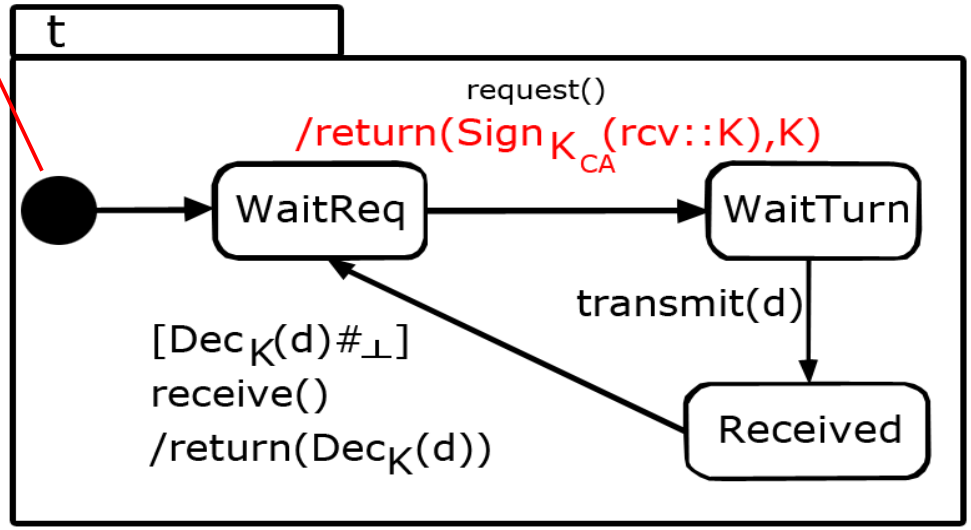


Sicherer Kanal: Protokoll

Verschlüsseln unter Sitzungsschlüssel nach Austausch eines Zertifikates.







<<Internet>>

hisch

Sicherheitsanalyse

Nach Dolev, Yao (1982): Für Sicherheitsanalyse, verifiziere Systemmodell gegen Angreifermodel auf Basis der Bedrohungsszenarien in den Verteilungsdiagrammen, das:

- an Protokolläufen teilnehmen kann,
- bestimmte Daten im Voraus kennt,
- Nachrichten von bestimmten Kommunikationsverbindungen abfangen kann,
- Nachrichten in bestimmte Kommunikationsverbindungen einfügen kann
- auf bestimmte System-Knoten zugreifen kann.

Angreifer

Verschiedene Angreifer-Klassen können unterschiedliche Stellen des Systems entsprechend der Gefährdungsszenarien angreifen.

Beispiel: Insider-Angreifer kann Kommunikationsverbindungen im LAN kontrollieren.

Für Sicherheitsanalyse der Spezifikation wird sie zusammen mit dem gegebenen Angreifermodell simuliert.

Kryptographie

Im Kontext der Sicherheitsanalye betrachten wir Schlüssel als Symbole und Kryptoalgorithmen als abstrakte Operationen:

- Kann nur mit richtigen Schlüsseln entschlüsseln.
- Kann keine statistischen Angriffe ausführen.

Kryptographische Ausdrücke I

Exp: Menge der *Krypto-Terme*, die aus den Symbolen in den Mengen *Data*, *Keys*, *Var* gebildet werden unter Verwendung der Operationen:

- *_::_* (Konkatenation), *head(_)*, *tail(_)*,
- *(_)*$^{-1}$ (K^{-1}: zum Verschlüsselungsschlüssel K gehöriger Entschlüsselungsschlüssel)
- *{ _ }_* ($\{M\}_K$: Verschlüsselung der Nachricht M mit Schlüssel *K*)
- *Dec_( )* ($Dec_K(C)$: Entschlüsselung der Daten C mit dem Schlüssel *K*)
- *Sign_( )* ($Sign_K(M)$: Signatur der Nachricht M mit dem Schlüssel *K*)
- *Ext_( )* ($Ext_K(S)$: Extrahieren der Signatur S mit dem Schlüssel *K*)

Kryptographische Ausdrücke II

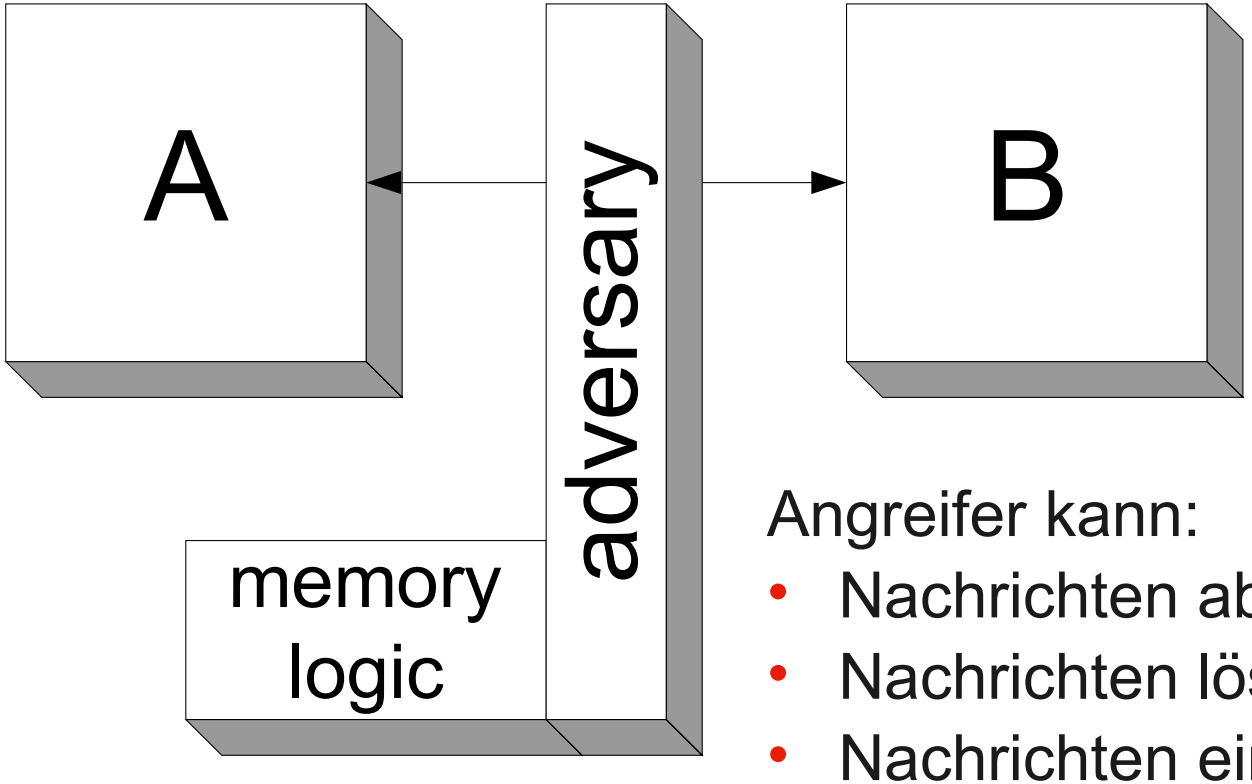
unter Berücksichtigung der folgenden Gleichungen:

- $\forall E,K.\ Dec_K^{-1}(\{E\}_K)=E$
- $\forall E,K.\ Ext_K(Sign_K^{-1}(E))=E$
- $\forall E_1,E_2.\ head(E_1::E_2)=E_1$
- $\forall E_1,E_2.\ tail(E_1::E_2)=E_2$
- Assoziativität für $::$.

Zur besseren Lesbarkeit, schreibe $E_1::E_2::E_3$ für $E_1::(E_2::E_3)$ und $fst(E_1::E_2)$ für $head(E_1::E_2)$ etc.

[NB: Die o.g. Gleichungen können bei Bedarf um weitere krypto-spezifische Eigenschaften erweitert werden (z.B. bei XOR).]

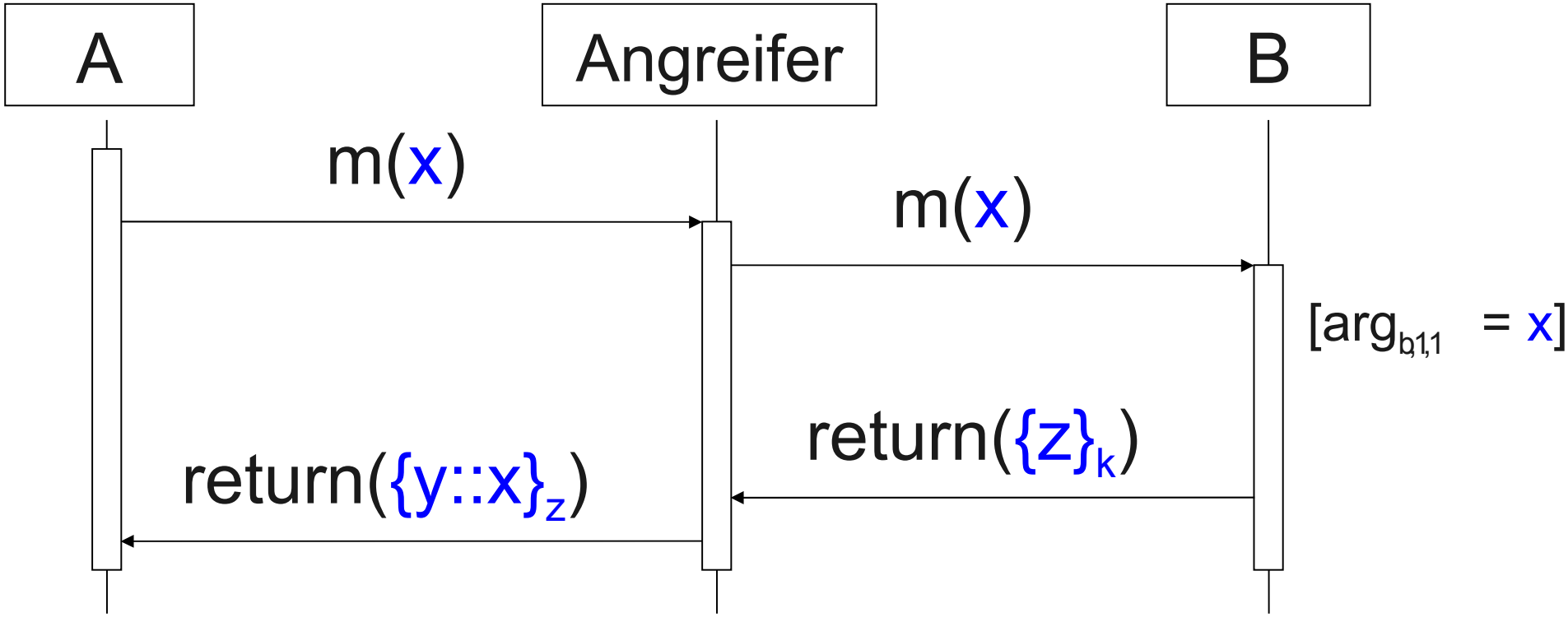
Angreifer-Modell



Angreifer kann:

- Nachrichten abspeichern
- Nachrichten löschen
- Nachrichten einfügen
- Nachrichten erstellen
- Kryptographische Funktionen benutzen
- Bestimmte Systemknoten kontrollieren
- Bestimmte Daten vorab kennen

Kryptobasierte Software (z.B Protokolle)



Angreifer-Wissen:

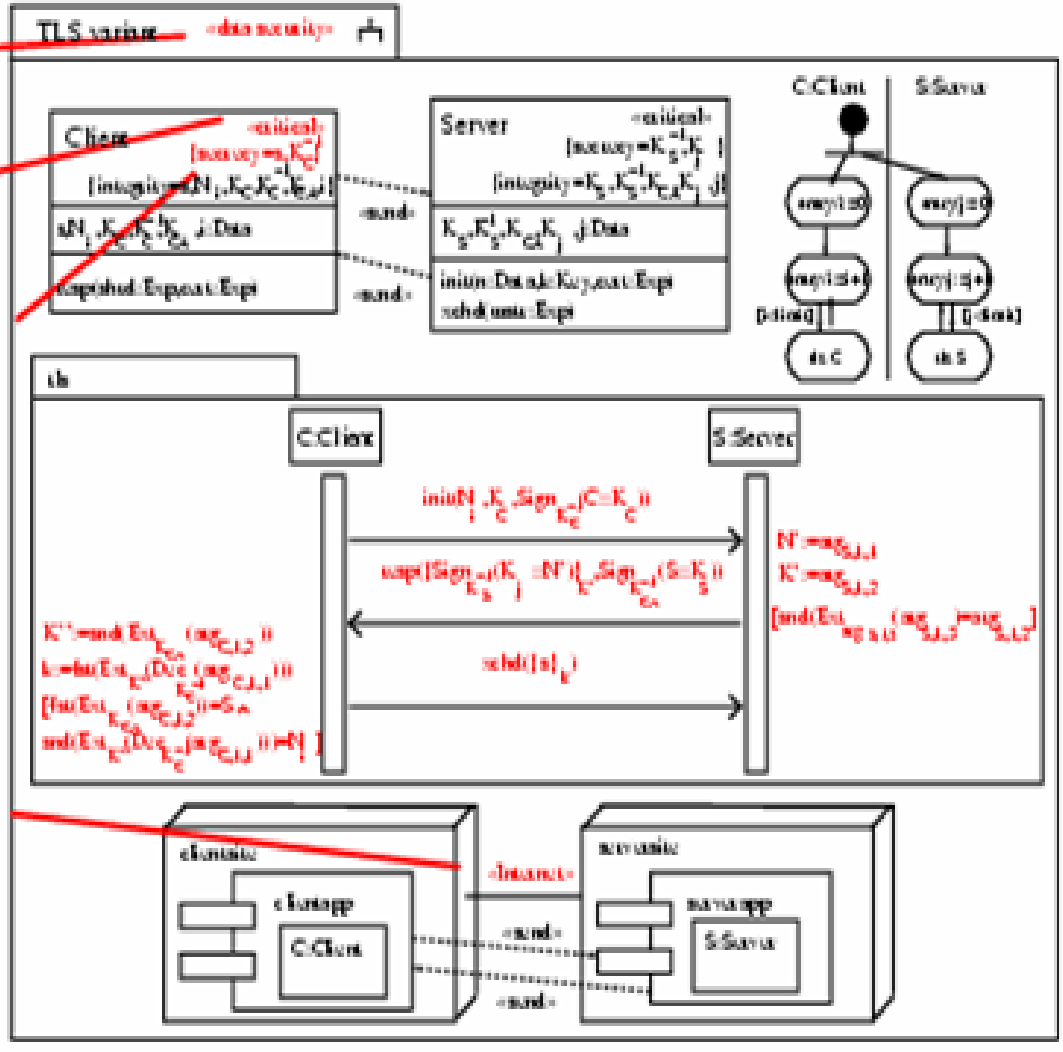
k^{-1}, y, x
$\{z\}_k$, z

Beispiel: Variante von TLS (SSL)

IEEE Infocom 1999.

Ziel: Vertrauliche Daten verschlüsselt unter Sitzungsschlüssel.

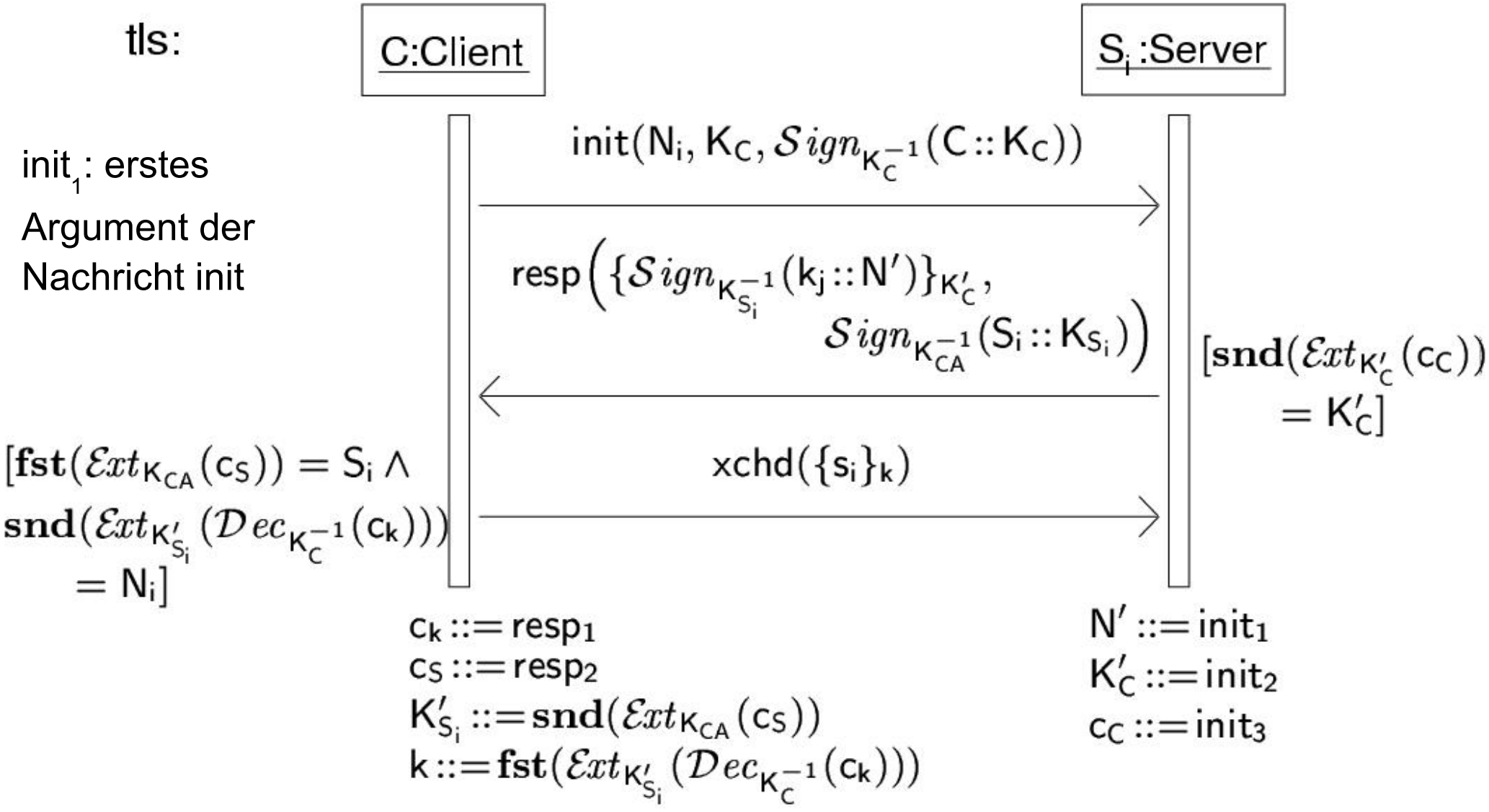
Weniger Server-belastung als bei TLS.



Protokollablauf

tls:

$init_1$: erstes Argument der Nachricht init



Man-in-the-Middle Angriff

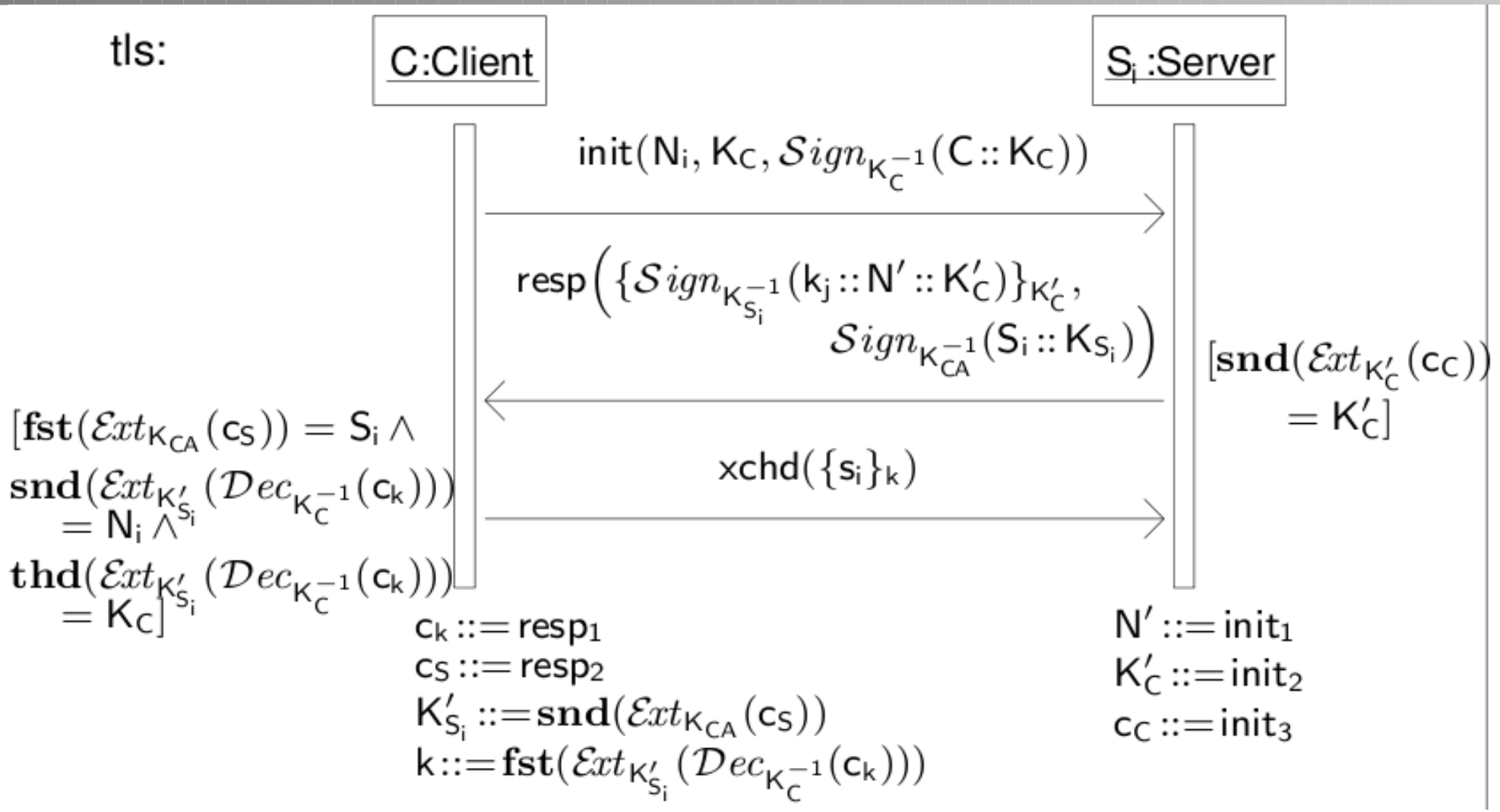
$$C \xrightarrow{N_i::K_C::Sign_{K_C^{-1}}(C::K_C)} A \xrightarrow{N_i::K_A::Sign_{K_A^{-1}}(C::K_A)} S$$

$$A \xleftarrow{\{Sign_{K_S^{-1}}(K_j::N_i)\}_{K_A}::Sign_{K_{CA}^{-1}}(S::K_S)} S$$

$$C \xleftarrow{\{Sign_{K_S^{-1}}(K_j::N_i)\}_{K_C}::Sign_{K_{CA}^{-1}}(S::K_S)} A$$

$$C \xrightarrow{\{s\}_{K_j}} A \xrightarrow{\{s\}_{K_j}} S$$

Korrigiertes Protokoll



Sicher laut automatischer Sicherheitsanalyse.

Frage

Angenommen, der Angreifer kommt in Besitz des zum Schlüssel K_{CA} gehörenden vertraulichen Signaturschlüssels der Certification Authority. Wie kann er damit in Besitz des zu übertragenden Geheimnisses d kommen ?

Zusammenfassung

- ISO Schichtenmodell und Sicherheit
- Angreifer-Modell
- Kryptographische Ausdrücke
- Kryptographische Protokolle