LEHRSTUHL 14
SOFTWARE ENGINEERING

Willkommen zur Vorlesung
Softwarearchitekturen im Finanz- und Versicherungsbereich
im Sommersemester 2010

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

6.Kryptographische Protokolle

Kryptographische Protokolle

Ziel: Sichere Identifizierung von Kommunikationspartnern.
Bedrohungen:

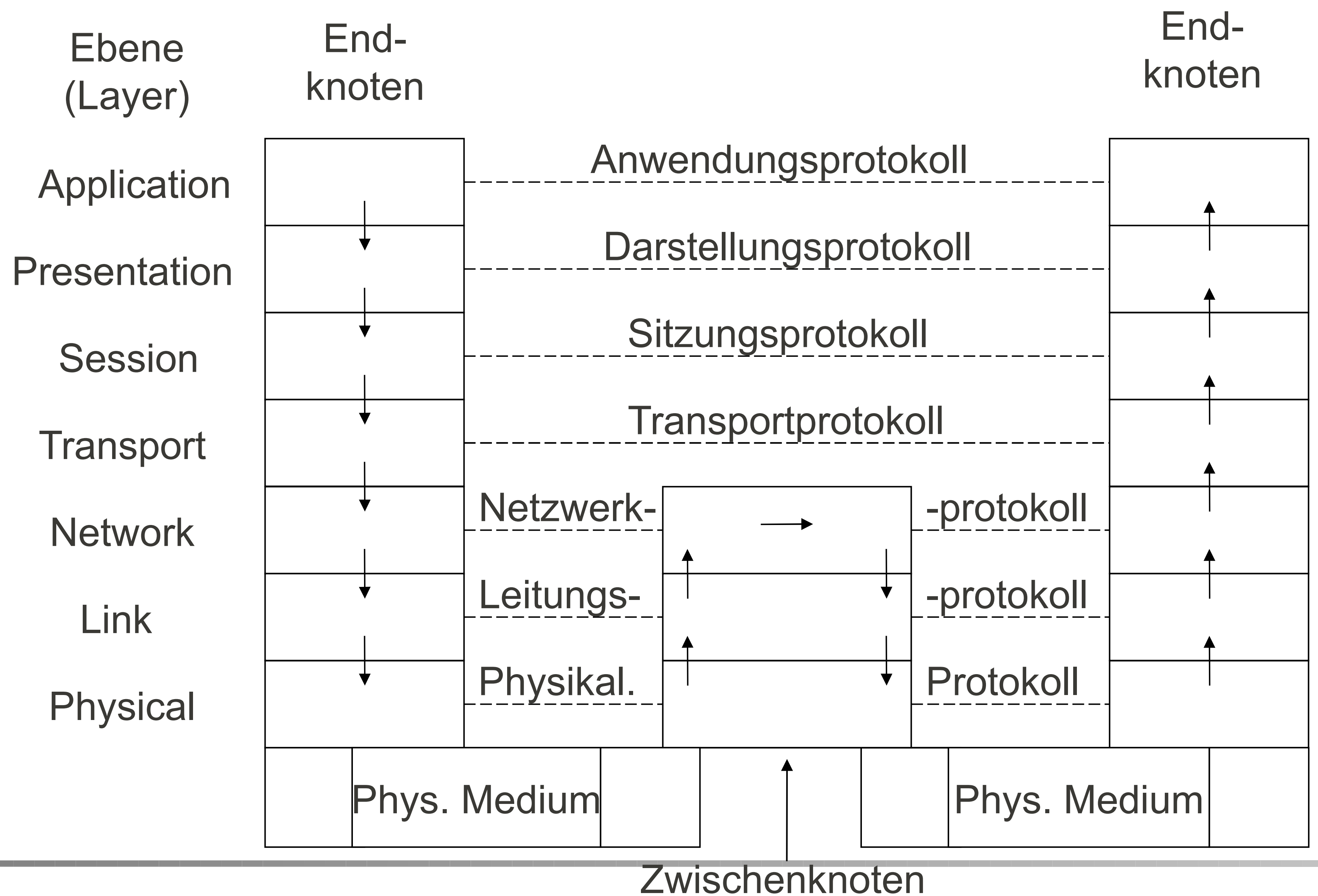
- Fälschung von Identitäten
- Unautorisierte Verwendung von Identitäten

Weitere Ziele: Schlüsselmanagement, elektronische Transaktionen, ...

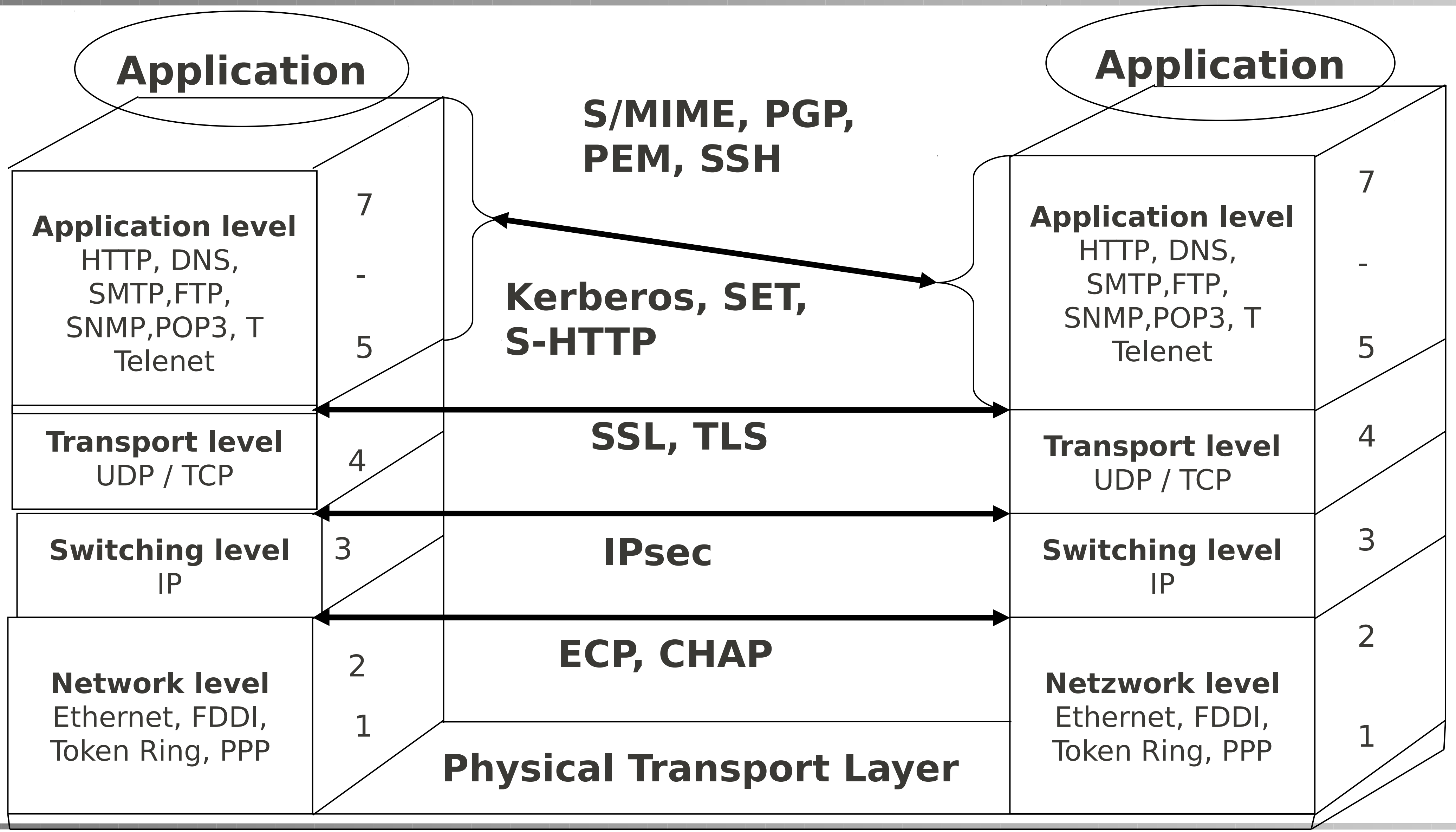
➔ kryptographische Protokolle: Austausch von Nachrichten zur Verteilung von Schlüsseln, Authentisierung etc.

Korrekter Entwurf sehr schwierig.

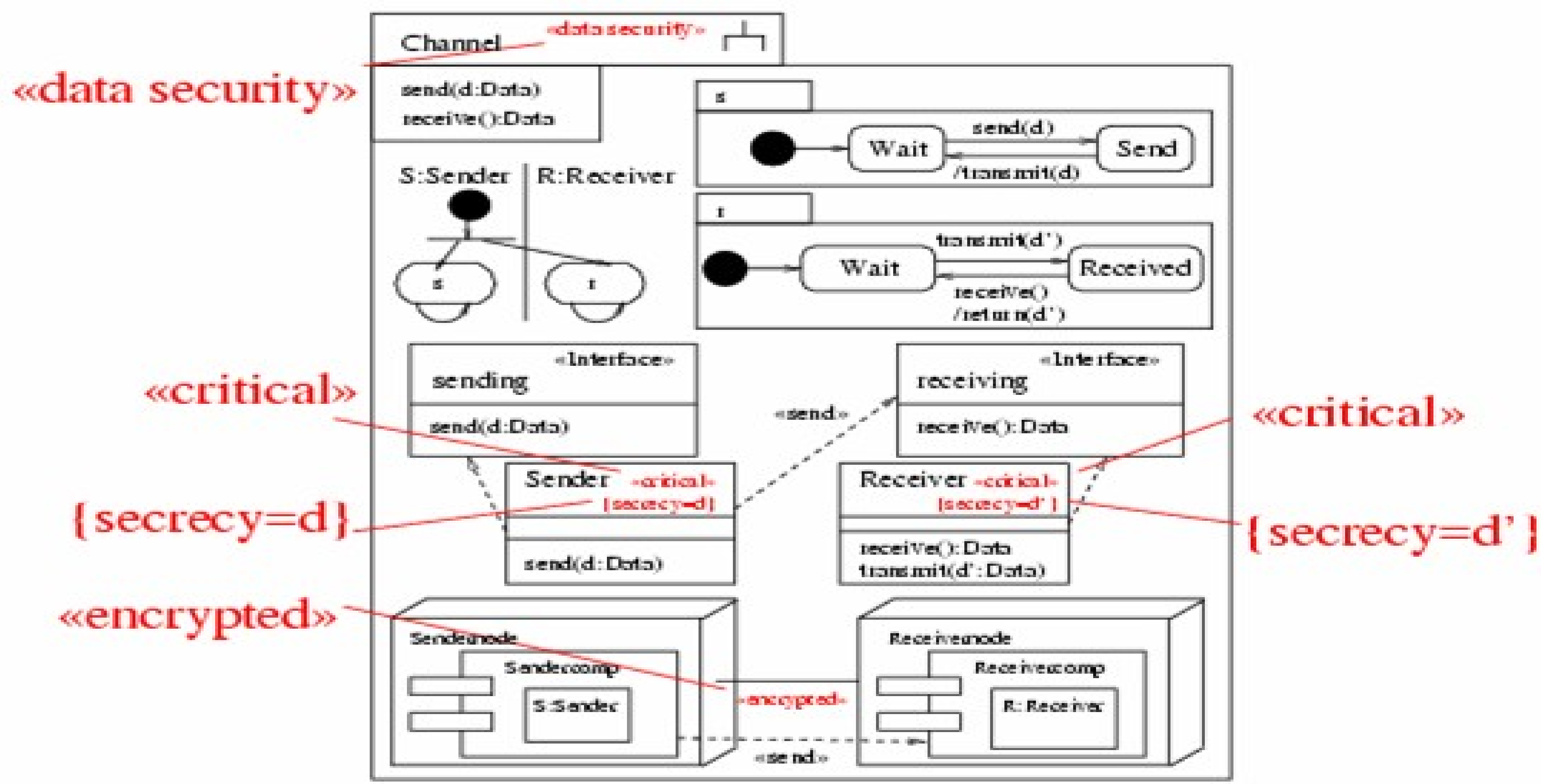
ISO OSI 7-Schichten-Modell



Verschlüsselung und Protokollebenen

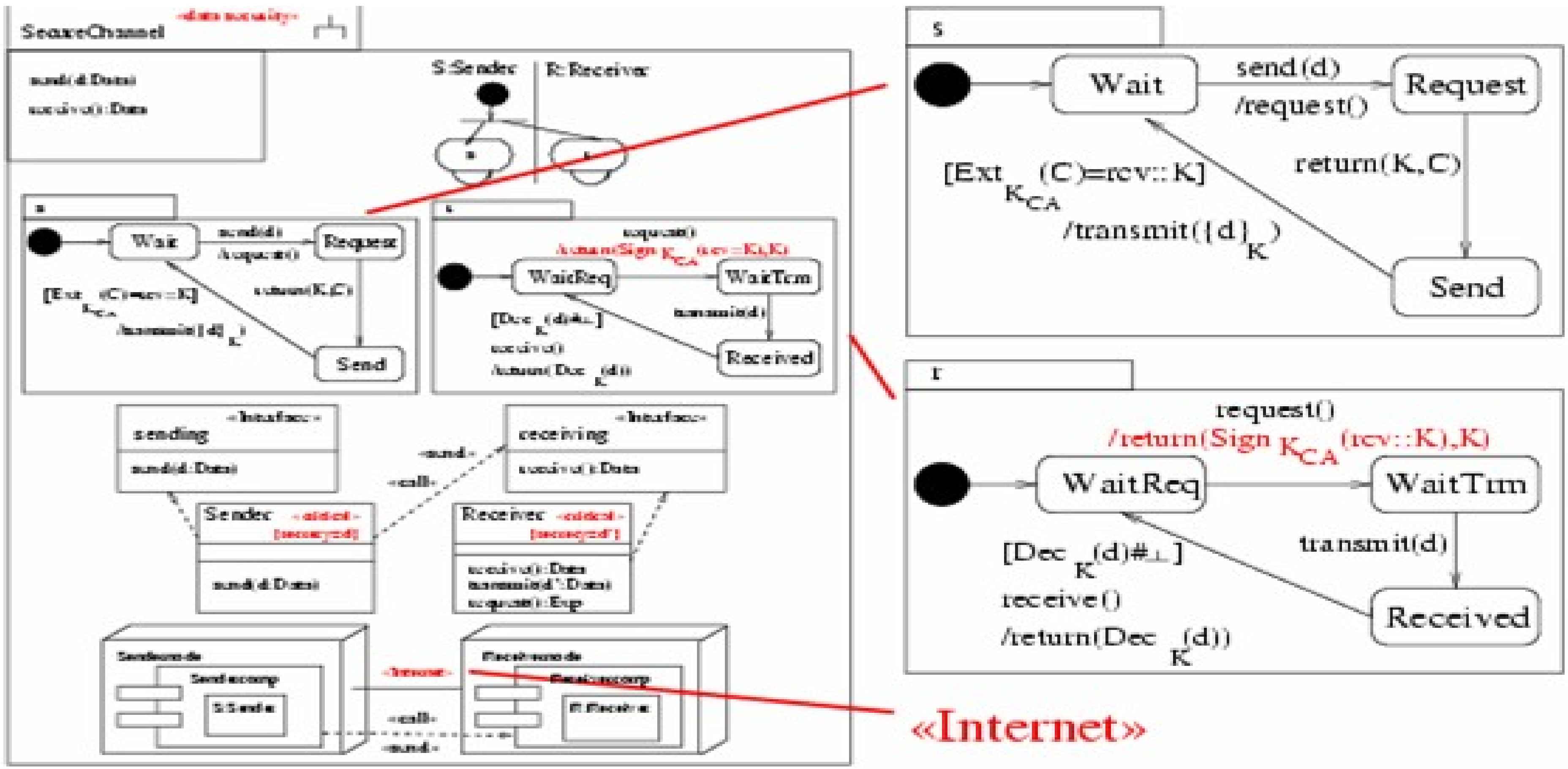


Beispiel: sicherer Kanal



Ziel: vertrauliche Übertragung von Daten über ungeschützte Kommunikationsverbindung.

Sicherer Kanal: Protokoll



Verschlüsseln unter Sitzungsschlüssel nach Austausch eines Zertifikates.

Security Protocols

System distributed over untrusted networks.

„Adversary“ intercepts, modifies, deletes, inserts messages.

Cryptography provides security.

Cryptographic Protocol: Exchange of messages for distributing session keys, authenticating principals etc. using cryptographic algorithms

Security Protocols: Problems

Many protocols have vulnerabilities or subtleties for various reasons

- weak cryptography
- core message exchange
- interfaces, prologues, epilogues
- deployment
- implementation bugs

Security Analysis

Following Dolev, Yao (1982): To analyze system, verify against attacker model from threat scenarios in deployment diagrams who

- may participate in some protocol runs,
- knows some data in advance,
- may intercept messages on some links,
- injects messages that it can produce in some links
- may access certain nodes.

Adversaries

Model classes of adversaries.

May attack different parts of the system according to threat scenarios.

Example: insider attacker may intercept communication links in LAN.

To evaluate security of specification, simulate jointly with adversary model.

Cryptography

Keys are symbols, crypto-algorithms are abstract operations.

- Can only decrypt with right keys.
- Can only compose with available messages.
- Cannot perform statistical attacks.

Cryptographic Expressions I

Exp: quotient of term algebra generated from sets *Data*, *Keys*, *Var* of symbols using

- *_::_* (concatenation), *head(_)*, *tail(_)*,
- $(\_)^{-1}$ (inverse keys)
- *{ _ }_* (encryption)
- *Dec_( )* (decryption)
- *Sign_( )* (signing)
- *Ext_( )* (extracting from signature)

under equations …

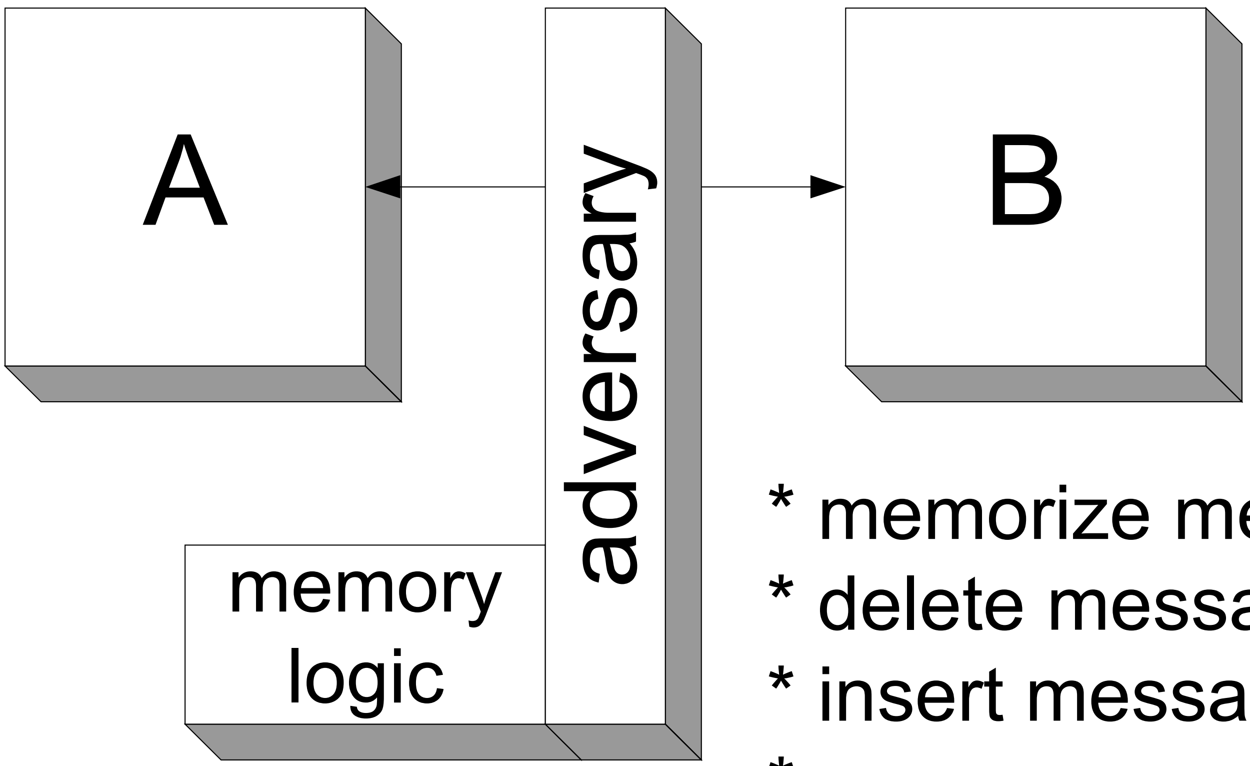
Cryptographic Expressions II

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

Write $E_1::E_2::E_3$ for $E_1::(E_2::E_3)$ and $fst(E_1::E_2)$ for $head(E_1::E_2)$ etc.

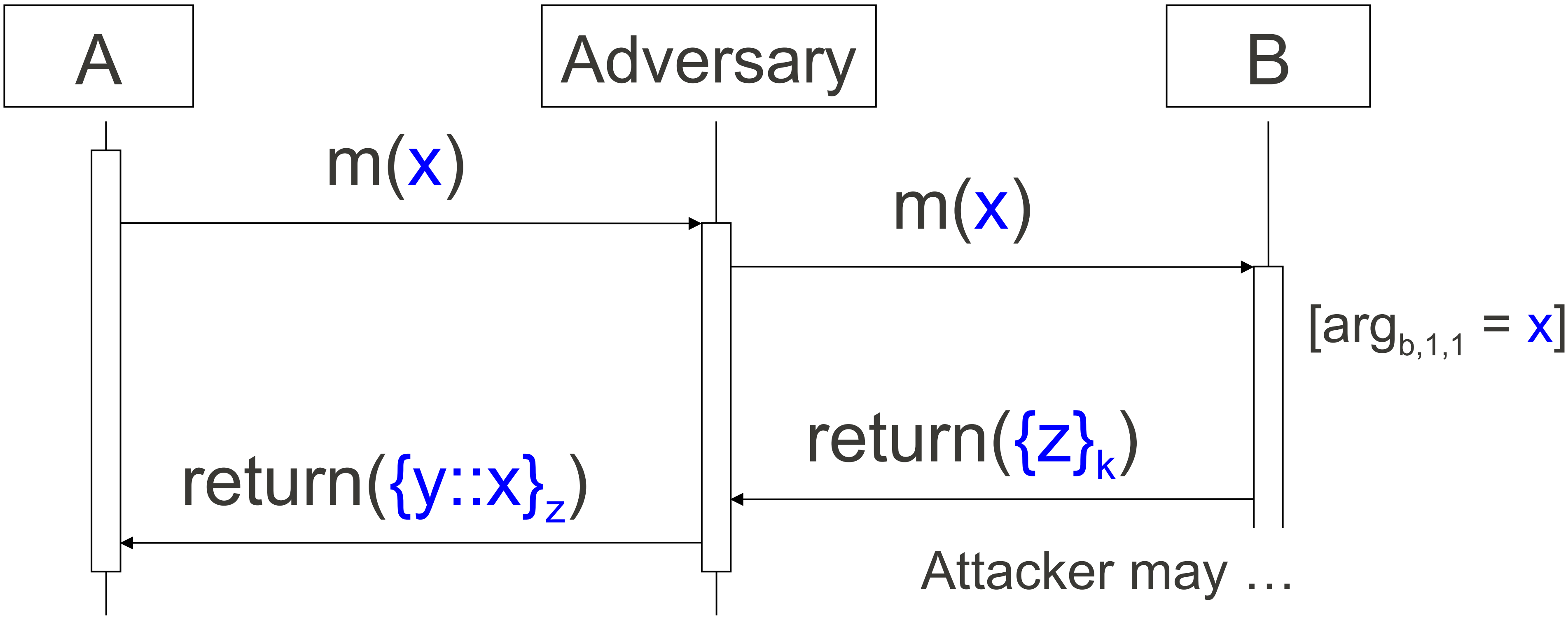
Can include further crypto-specific primitives and laws (XOR, …).

Adversary Model



* memorize message
* delete message
* insert message
* compose own message
* use cryptographic primitives

Crypto-based Software (e.g. Protocols)



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

(cf. [Dolev, Yao 1982])

Attacker may …

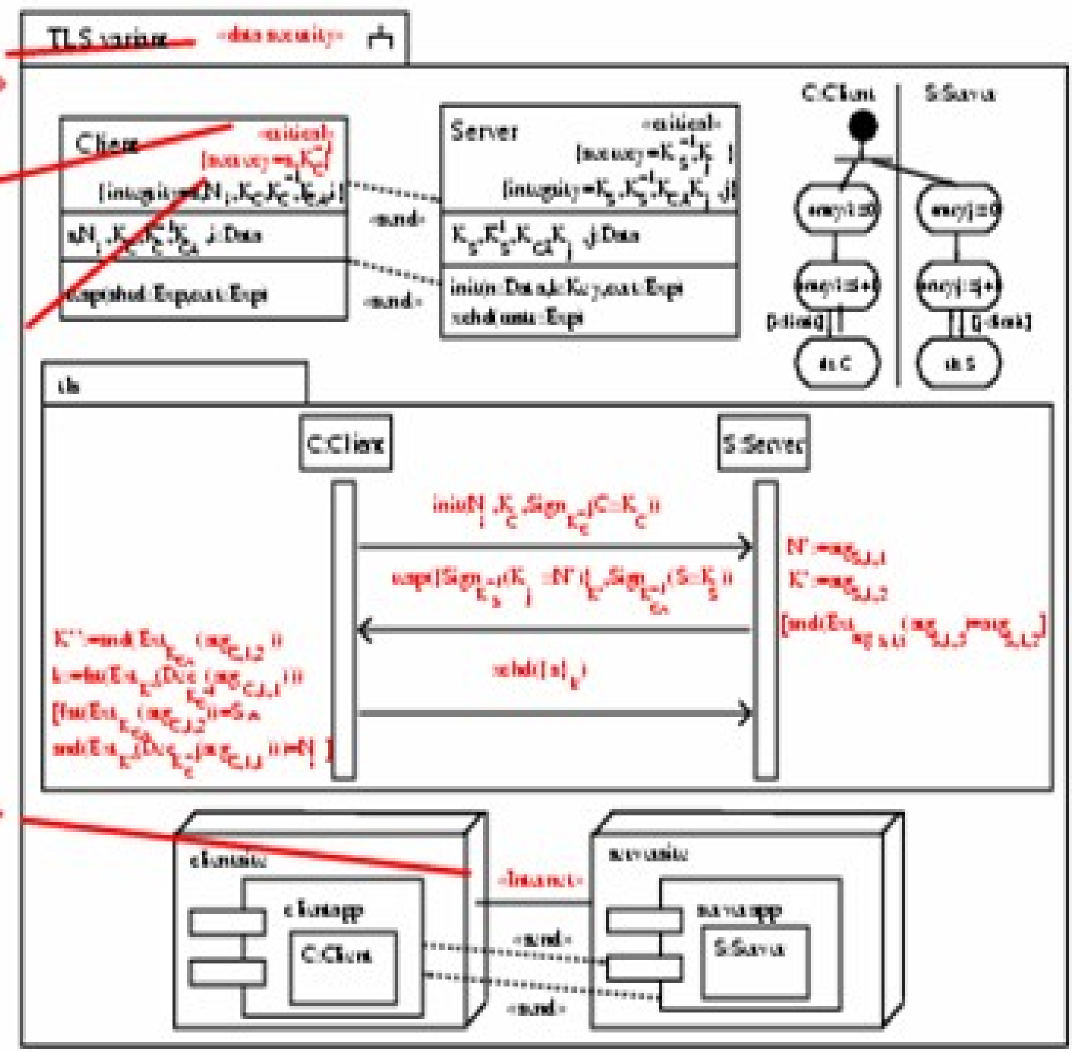
- control system parts,
- know data in advance,
- intercept messages,
- delete messages,
- inject messages.

Beispiel: Variante von TLS (SSL)

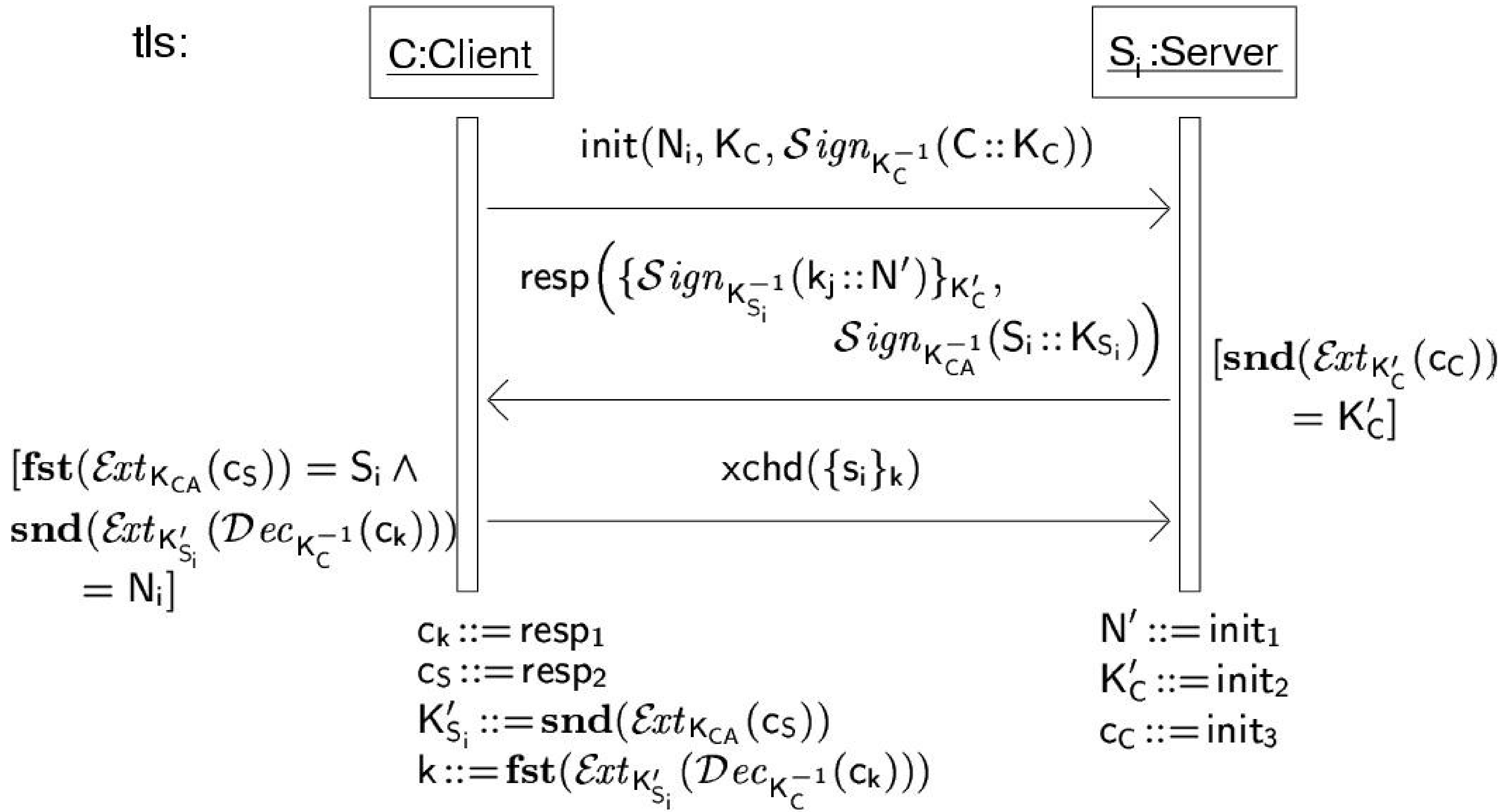
IEEE Infocom 1999.

Ziel: Vertrauliche Daten verschlüsselt unter Sitzungsschlüssel.

Weniger Serverbelastung als bei TLS.



Protokollablauf



Man-in-the-Middle Angriff

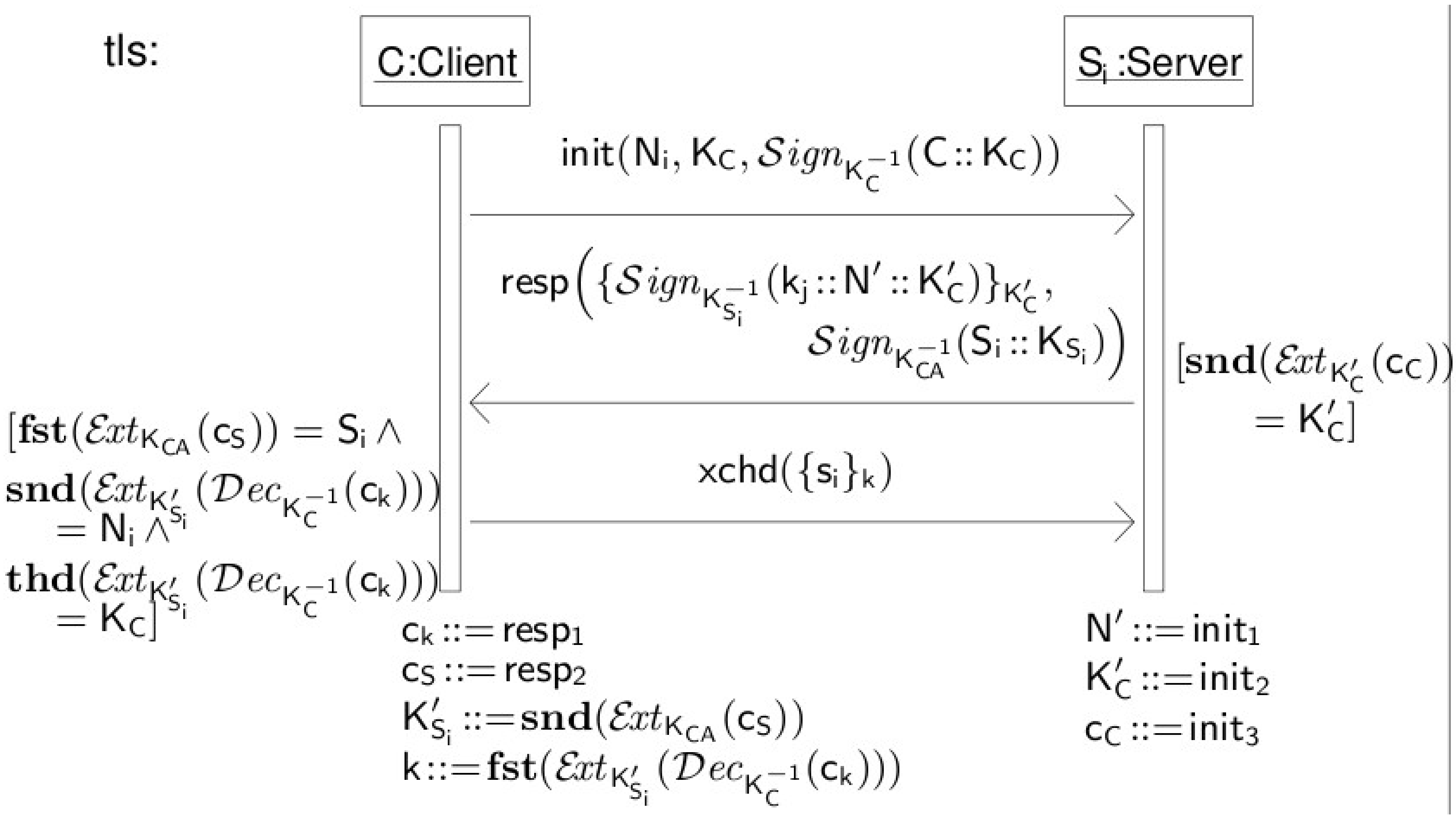
$$C \xrightarrow{N_i::K_C::Sign_{K_C^{-1}}(C::K_C)} A \xrightarrow{N_i::K_A::Sign_{K_A^{-1}}(C::K_A)} S$$

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

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

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

Korrigiertes Protokoll



Sicher gemäß automatischer Sicherheitsanalyse.

Aufgabe 6.1

a) Zeichne den regulären Protokollablauf (d.h. ohne Angriff) des Protokolles auf Folie 7 als Sequenzdiagramm. [3 P.]

b) 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 ? Zeichne den Angriffsablauf. [5 P.]

c) Welche Angriffsaktion kann der default Angreifer auf Folie 6 durchführen ? [2 P.]

d) Wie kann er den Kommunikationsablauf darüberhinaus auf Folie 7 auch ohne den Signaturschlüssel beeinflussen (Hinweis: Nachrichteneingang) ? [3 P.]

Adversary Knowledge

Specify set K_A^0 of initial knowledge of an adversary of type A. Let K_A^{n+1} be the Exp-subalgebra generated by K_A^n and the expressions received after $n+1$st iteration of the protocol.

Definition (Dolev, Yao 1982).

S keeps secrecy of M against attackers of type A if there is no n with $M \in K_A^n$.

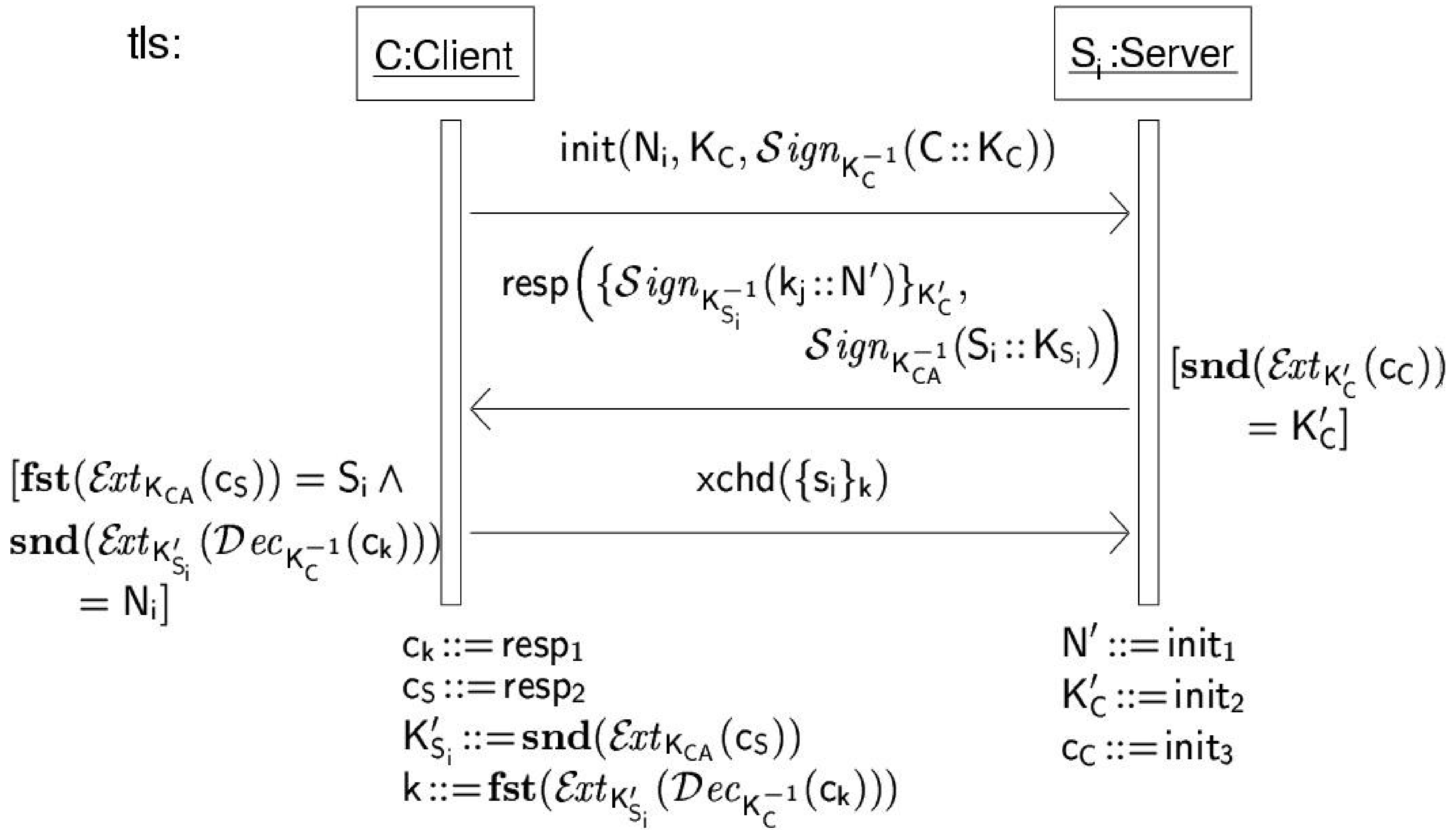
Security Analysis in First-order Logic

Idea: approximate set of possible data values flowing through system from above.

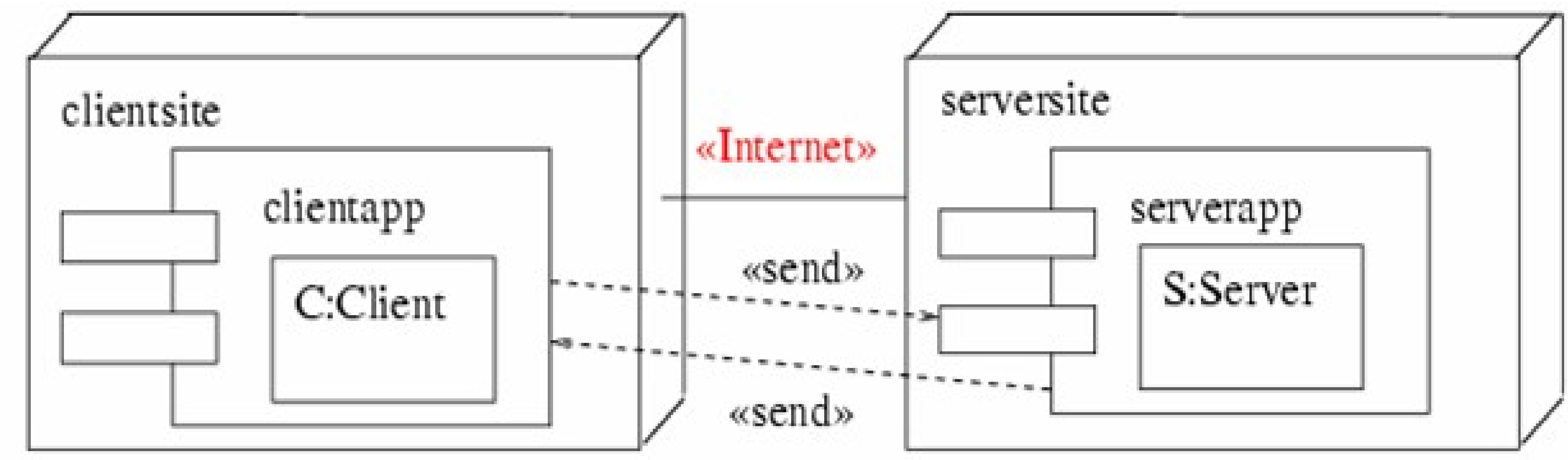
Predicate *knows(E)* meaning that the adversary may get to know *E* during the execution of the protocol.

For any secret *s*, check whether can derive *knows(s)* using automated theorem prover.

Given Sequence Diagram …



… and Physical Layer Model …



Deployment diagram.

Derived adversary model: read, delete, insert data.

… Translate to 1st Order Logic

Connection (or statechart transition)

TR1=(in(msg_in),cond(msg_in),out(msg_out))

followed by *TR2* gives predicate *PRED(TR1)=*

$\forall$ *msg_in. [knows(msg_in)* $\wedge$ *cond(msg_in)*

$\Rightarrow$*knows(msg_out)*

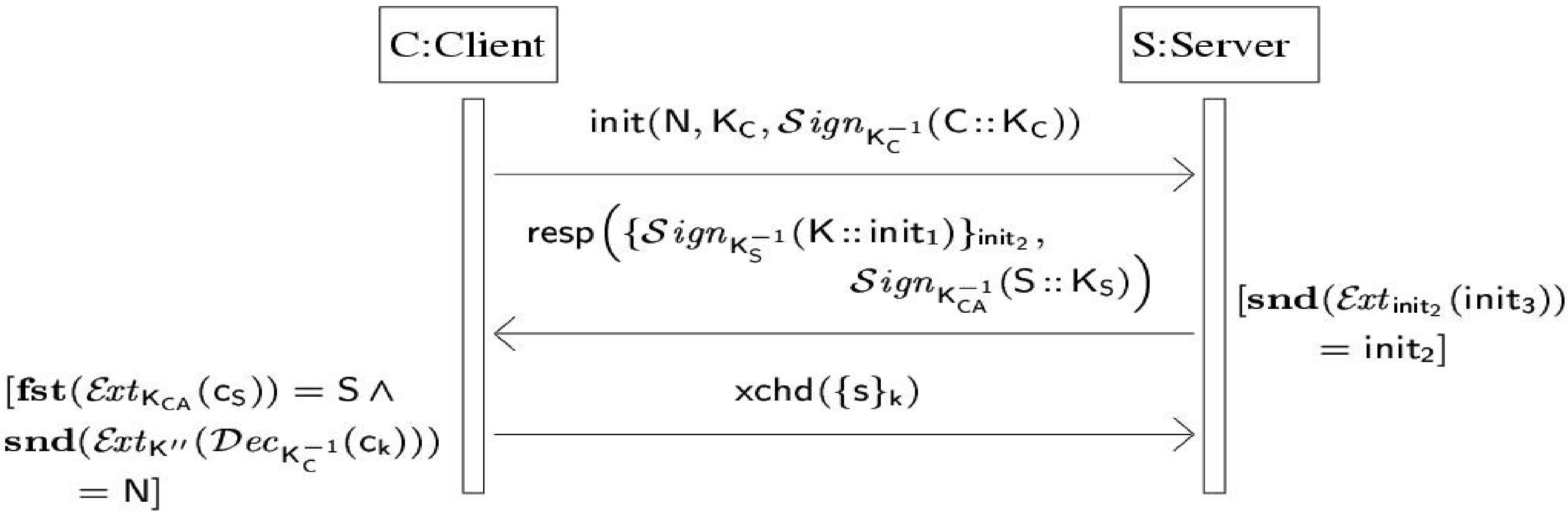
$\wedge$ *PRED(TR2)]*

(Assume: order enforced (!).)

Can include senders, receivers in messages.

Abstraction: find all attacks, may have false positives.

Example TLS Variant



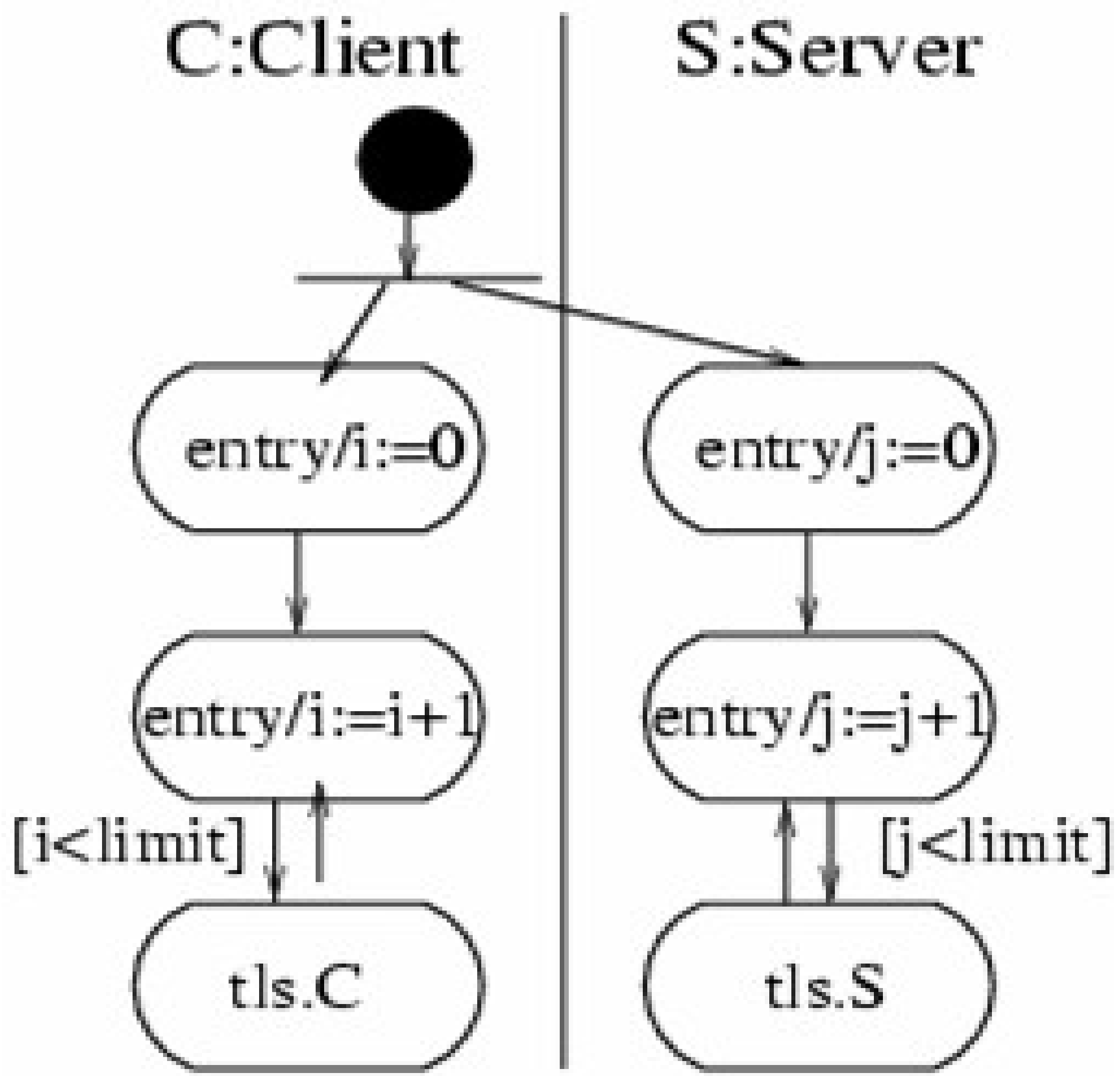
$$knows(N) \wedge knows(K_C) \wedge knows(Sign_{K_C^{-1}}(C::K_C))$$

$$\wedge\ \forall init_1, init_2, init_3.[knows(init_1) \wedge knows(init_2) \wedge knows(init_3) \wedge snd(Ext_{init_2}(init_3)) = init_2$$

$$\Rightarrow knows(\{Sign_{K_S^{-1}}(\ldots)\}_{\ldots}) \wedge [knows(Sign\ldots)] \wedge$$

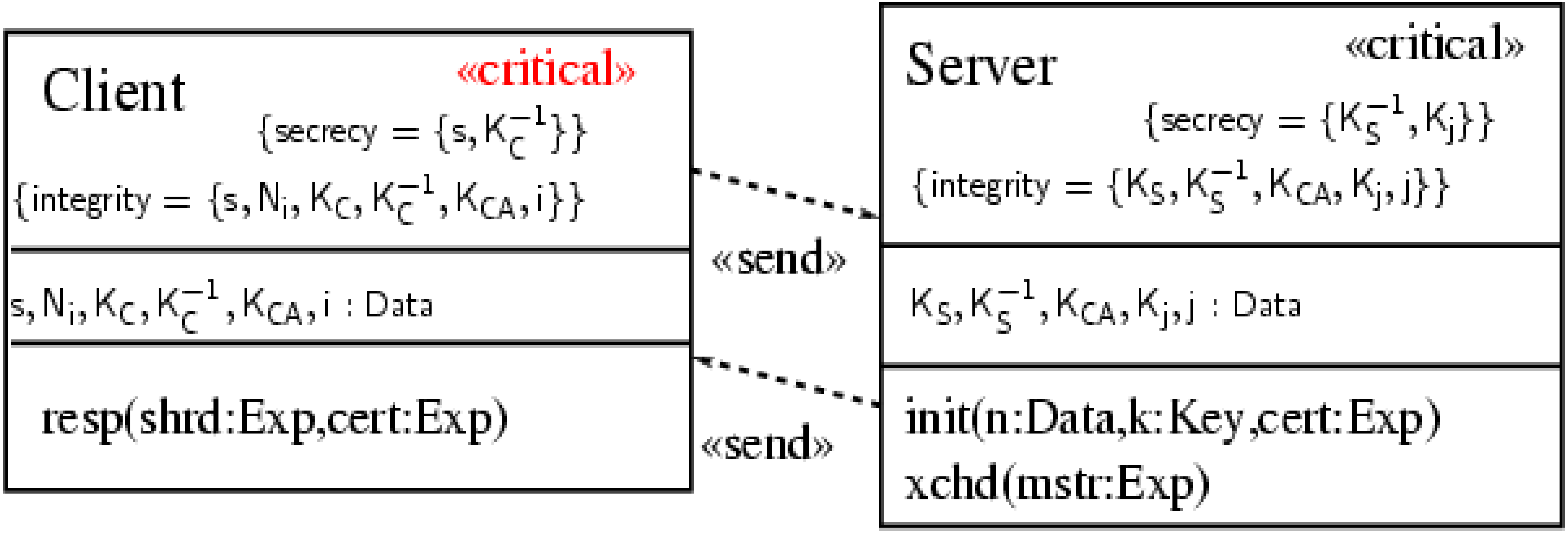
$$\forall\ resp_1, resp_2.[\ldots \Rightarrow \ldots]]$$

Execute in System Context



Activity diagram.

Formulate Data Security Requirements



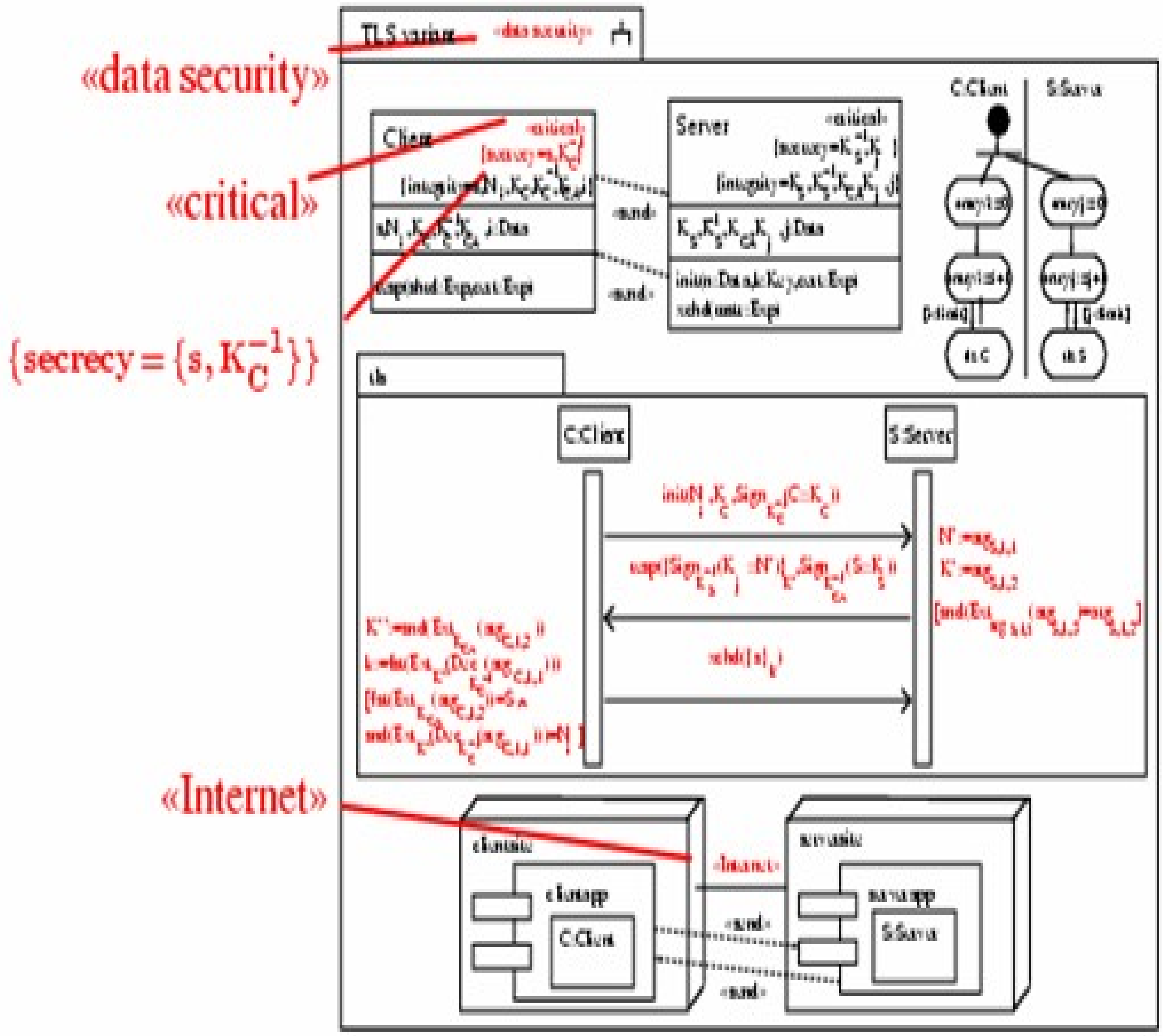
Class diagram.

Gives conjecture: *knows(s)* derivable ?

LEHRSTUHL 14 SOFTWARE ENGINEERING

Proposed Variant of TLS (SSL)

[IEEE Infocom 1999]
Goal:
send secret protected by session key using fewer server resources.



TLS Variant in TPTP Notation I

```
input_formula(tls_abstract_protocol,axiom,(
 ![ArgS_11, ArgS_12, ArgS_13, ArgC_11, ArgC_12] : (
   ![DataC_KK, DataC_k, DataC_n] : (
      % Client -> Attacker (1. message)
      (       knows(n)
            & knows(k_c)
            & knows(sign(conc(c, k_c), inv(k_c) ) )   )
    & % Server -> Attacker (2. message)
      (   (   knows(ArgS_11)
            & knows(ArgS_12)
            & knows(ArgS_13)
            & ( ? [X] : equal( sign(conc(X, ArgS_12), inv(ArgS_12) ),
                               ArgS_13 ) )   )
      => (    knows(enc(sign(conc(kgen(ArgS_12), ArgS_11), inv(k_s) ),
                        ArgS_12 ) )
            & knows(sign(conc(s, k_s), inv(k_ca) ) ) )   )
```

TLS Variant in TPTP Notation II

```
    & % Client -> Attacker (3. message)
      (  (    knows(ArgC_11)
            & knows(ArgC_12)
            & equal(sign(conc(s, DataC_KK), inv(k_ca)), ArgC_12 )
            & equal(enc(sign(conc(DataC_k, DataC_n), inv(DataC_KK) ),
                         k_c), ArgC_11 )
            & ( ? [DataC_ks] : equal(sign(conc(s, DataC_ks), inv(k_ca) ),
                                      ArgC_12 ) )
            & equal(enc(sign(conc(DataC_k, n), inv(DataC_KK) ), k_c),
                    ArgC_11 )
            & equal(enc(sign(conc(DataC_k, DataC_n), inv(DataC_KK) ), k_c),
                    ArgC_11 )
         )
      => ( knows(symenc(secret, DataC_k)) )  )
) )  )).
```

Surprise ...

```
E-SETHEO csp03 single processor running on host ...
(c) 2003 Max-Planck-Institut fuer Informatik and
    Technische Universitaet Muenchen

tlsvariant-freshkey-check.tptp
...
time limit information: 300 total (entering statistics module).
problem analysis ...
testing if first-order ...
first-order problem
...
statistics: 19 0 7 46 3 6 2 0 1 2 14 8 0 2 28 6
...
schedule selection: problem is horn with equality (class he).
schedule:605 3 300 597
...
entering next strategy 605 with resource 3 seconds.
...
analyzing results ...
proof found
time limit information: 298 total / 297 strategy (leaving wrapper).
...
e-SETHEO done. exiting
```

Attack

… Which Means:

Can derive *knows(s)* (!).

That is: Protocol does not preserve secrecy of *s* against adversaries.

➔ Completely insecure wrt stated goals.

But why ?

Could look at proof tree.

Or: use prolog-based attack generator.

Man-in-the-Middle Attack

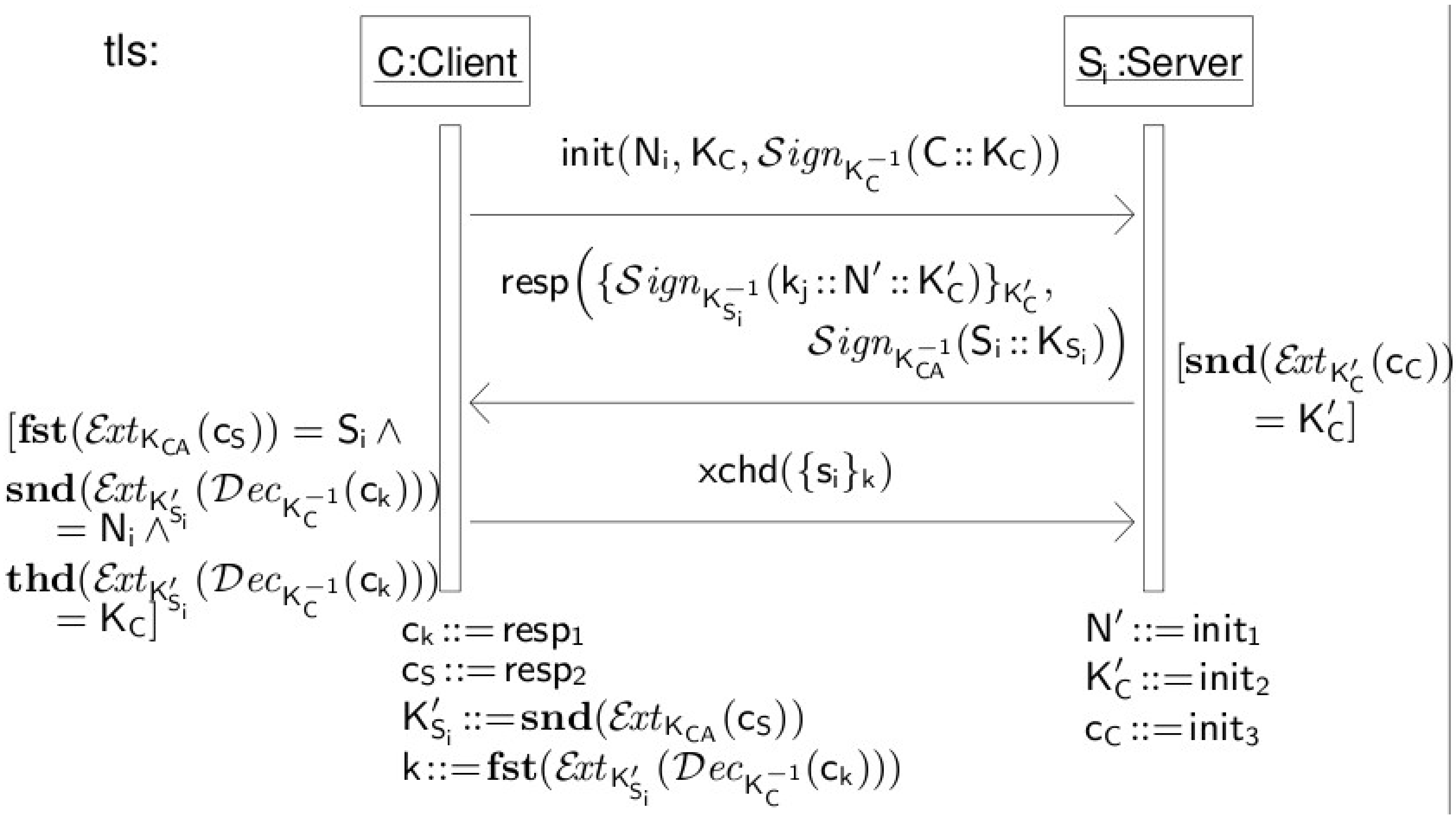
$$C \xrightarrow{N_i::K_C::Sign_{K_C^{-1}}(C::K_C)} A \xrightarrow{N_i::K_A::Sign_{K_A^{-1}}(C::K_A)} S$$

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

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

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

The Fix



e-Setheo: *knows(s)* not derivable. Thus secure.

Aufgabe 6.2

Zeige, wie der Angriff von Folie 34 in Form einer logischen Ableitung der conjecture von den axioms in der TPTP Datei

http://ls14-www.cs.tu-dortmund.de/main2/jj/teaching/ss10/arch/exercises/tlsvariant.tptp

(s. Handout) demonstriert werden kann [8 P.].