Protokolldesign

Alexander Aprelkin

TUM

16.06.2009

Gliederung

1. Einführung
   - Begriffe
   - Notation
   - Motivation
2. Design Prinzipien
   - Basisprinzip 1. Explicitness
   - Prinzip 3. Naming
   - Prinzip 4. Encryption
   - Prinzip 5. Signing encrypted Data
3. Schlusswort

Gliederung

1. Einführung
   - Begriffe
   - Notation
   - Motivation
2. Design Prinzipien
3. Schlusswort

Begriffe

Definition

Protokoll - eine Menge von Regeln, die festlegen, wie der Austausch von Nachrichten zwischen 2 oder mehr Partnern abläuft.

Definition

Kryptographisches Protokoll - Protokoll, bei dem Teile oder ganze Nachrichten verschlüsselt sind.

Begriffe

Definition

Protokoll - eine Menge von Regeln, die festlegen, wie der Austausch von Nachrichten zwischen 2 oder mehr Partnern abläuft.

Definition

Kryptographisches Protokoll - Protokoll, bei dem Teile oder ganze Nachrichten verschlüsselt sind.

Gliederung

1. Einführung
   - Begriffe
   - Notation
   - Motivation
2. Design Prinzipien
3. Schlusswort

Notation

- A - Alice, B - Bob
- S - Trusted Server
- K_a - Öffentlicher Schlüssel von A, ${K^{-1}}_a$ - Privater Schlüssel von A bzw. inverser Schlüssel zu K_a
- T - Zeitstempel
- $\{X\}_K$ - Nachricht X, verschlüsselt mit K
- K_{ab} - (symmetrischer) Schlüssel für die Kommunikation zwischen A und B
- N_a - Nonce (=Number used Once), generiert durch A

Notation

- A - Alice, B - Bob
- S - Trusted Server
- K_a - Öffentlicher Schlüssel von A, ${K^{-1}}_a$ - Privater Schlüssel von A bzw. inverser Schlüssel zu K_a
- T - Zeitstempel
- $\{X\}_K$ - Nachricht X, verschlüsselt mit K
- K_{ab} - (symmetrischer) Schlüssel für die Kommunikation zwischen A und B
- N_a - Nonce (=Number used Once), generiert durch A

Notation

- A - Alice, B - Bob
- S - Trusted Server
- K_a - Öffentlicher Schlüssel von A, ${K^{-1}}_a$ - Privater Schlüssel von A bzw. inverser Schlüssel zu K_a
- T - Zeitstempel
- $\{X\}_K$ - Nachricht X, verschlüsselt mit K
- K_{ab} - (symmetrischer) Schlüssel für die Kommunikation zwischen A und B
- N_a - Nonce (=Number used Once), generiert durch A

Notation

- A - Alice, B - Bob
- S - Trusted Server
- K_a - Öffentlicher Schlüssel von A, ${K^{-1}}_a$ - Privater Schlüssel von A bzw. inverser Schlüssel zu K_a
- T - Zeitstempel
- $\{X\}_K$ - Nachricht X, verschlüsselt mit K
- K_{ab} - (symmetrischer) Schlüssel für die Kommunikation zwischen A und B
- N_a - Nonce (=Number used Once), generiert durch A

Notation

- A - Alice, B - Bob
- S - Trusted Server
- K_a - Öffentlicher Schlüssel von A, ${K^{-1}}_a$ - Privater Schlüssel von A bzw. inverser Schlüssel zu K_a
- T - Zeitstempel
- $\{X\}_K$ - Nachricht X, verschlüsselt mit K
- K_{ab} - (symmetrischer) Schlüssel für die Kommunikation zwischen A und B
- N_a - Nonce (=Number used Once), generiert durch A

Notation

- A - Alice, B - Bob
- S - Trusted Server
- K_a - Öffentlicher Schlüssel von A, ${K^{-1}}_a$ - Privater Schlüssel von A bzw. inverser Schlüssel zu K_a
- T - Zeitstempel
- $\{X\}_K$ - Nachricht X, verschlüsselt mit K
- K_{ab} - (symmetrischer) Schlüssel für die Kommunikation zwischen A und B
- N_a - Nonce (=Number used Once), generiert durch A

Notation

- A - Alice, B - Bob
- S - Trusted Server
- K_a - Öffentlicher Schlüssel von A, ${K^{-1}}_a$ - Privater Schlüssel von A bzw. inverser Schlüssel zu K_a
- T - Zeitstempel
- $\{X\}_K$ - Nachricht X, verschlüsselt mit K
- K_{ab} - (symmetrischer) Schlüssel für die Kommunikation zwischen A und B
- N_a - Nonce (=Number used Once), generiert durch A

Beispiel

Message 4: $B \rightarrow A : \{T_a + 1\}_{K_{ab}}$

Interpretation

- 4. Nachricht im Protokol
- Bob schickt an Alice
- Zeitstempel, den A generiert hat, um 1 inkrementiert
- Verschlüsselt mit dem Schlüssel für die Kommunikation zwischen A und B

Beispiel

Message 4: $B \rightarrow A : \{T_a + 1\}_{K_{ab}}$

Interpretation

- 4. Nachricht im Protokol
- Bob schickt an Alice
- Zeitstempel, den A generiert hat, um 1 inkrementiert
- Verschlüsselt mit dem Schlüssel für die Kommunikation zwischen A und B

Beispiel

Message 4: $B \rightarrow A : \{T_a + 1\}_{K_{ab}}$

Interpretation

- 4. Nachricht im Protokol
- Bob schickt an Alice
- Zeitstempel, den A generiert hat, um 1 inkrementiert
- Verschlüsselt mit dem Schlüssel für die Kommunikation zwischen A und B

Beispiel

Message 4: $B \rightarrow A : \{T_a + 1\}_{K_{ab}}$

Interpretation

- 4. Nachricht im Protokol
- Bob schickt an Alice
- Zeitstempel, den A generiert hat, um 1 inkrementiert
- Verschlüsselt mit dem Schlüssel für die Kommunikation zwischen A und B

Beispiel

Message 4: $B \rightarrow A : \{T_a + 1\}_{K_{ab}}$

Interpretation

- 4. Nachricht im Protokol
- Bob schickt an Alice
- Zeitstempel, den A generiert hat, um 1 inkrementiert
- Verschlüsselt mit dem Schlüssel für die Kommunikation zwischen A und B

Motivation

Ziel eines kryptographischen Protokolls

den erforderlichen Sicherheitsgrad des Systems zu gewährleisten

In manchen Protokollen wird dieses Ziel nicht erreicht

- nicht aufgrund einer schwachen Verschlüsselung
- sondern wegen eines Designfehlers

Motivation

Ziel eines kryptographischen Protokolls

den erforderlichen Sicherheitsgrad des Systems zu gewährleisten

In manchen Protokollen wird dieses Ziel nicht erreicht

- nicht aufgrund einer schwachen Verschlüsselung
- sondern wegen eines Designfehlers

Motivation

Ziel eines kryptographischen Protokolls

den erforderlichen Sicherheitsgrad des Systems zu gewährleisten

In manchen Protokollen wird dieses Ziel nicht erreicht

- nicht aufgrund einer schwachen Verschlüsselung
- sondern wegen eines Designfehlers

Motivation

Ziel eines kryptographischen Protokolls

den erforderlichen Sicherheitsgrad des Systems zu gewährleisten

In manchen Protokollen wird dieses Ziel nicht erreicht

- nicht aufgrund einer schwachen Verschlüsselung
- sondern wegen eines Designfehlers

Motivation

Ziel eines kryptographischen Protokolls

den erforderlichen Sicherheitsgrad des Systems zu gewährleisten

In manchen Protokollen wird dieses Ziel nicht erreicht

- nicht aufgrund einer schwachen Verschlüsselung
- sondern wegen eines Designfehlers

Techniken zum Design von Security-Protokollen:

- Formale Methoden (werden nicht weiter betrachtet)
- *Design Prinzipien* → Faustregeln zur Vermeidung von bekannten Fehlern

Techniken zum Design von Security-Protokollen:

- Formale Methoden (werden nicht weiter betrachtet)
- *Design Prinzipien* → Faustregeln zur Vermeidung von bekannten Fehlern

Techniken zum Design von Security-Protokollen:

- Formale Methoden (werden nicht weiter betrachtet)
- *Design Prinzipien* $\longrightarrow$ Faustregeln zur Vermeidung von bekannten Fehlern

Design-Prinzipien für kryptographische Protokolle:

- sind für die Korrektheit *weder notwendig noch hinreichend*
- ihre strikte Einhaltung *verhindert viele Fehler*

- **1994** - M. Abadi und R. Needham: 11 Prinzipien
- **1995** - R. Anderson und R. Needham: 8 zusätzliche Prinzipien für Public-Key-Protokolle

Design-Prinzipien für kryptographische Protokolle:

- sind für die Korrektheit *weder notwendig noch hinreichend*
- ihre strikte Einhaltung *verhindert viele Fehler*

- **1994** - M. Abadi und R. Needham: 11 Prinzipien
- **1995** - R. Anderson und R. Needham: 8 zusätzliche Prinzipien für Public-Key-Protokolle

Design-Prinzipien für kryptographische Protokolle:

- sind für die Korrektheit *weder notwendig noch hinreichend*
- ihre strikte Einhaltung *verhindert viele Fehler*

- **1994** - M. Abadi und R. Needham: 11 Prinzipien
- **1995** - R. Anderson und R. Needham: 8 zusätzliche Prinzipien für Public-Key-Protokolle

Gliederung

1. Einführung
2. Design Prinzipien
   - Basisprinzip 1. Explicitness
   - Prinzip 3. Naming
   - Prinzip 4. Encryption
   - Prinzip 5. Signing encrypted Data
3. Schlusswort

Gliederung

1 Einführung

2 Design Prinzipien
- Basisprinzip 1. Explicitness
- Prinzip 3. Naming
- Prinzip 4. Encryption
- Prinzip 5. Signing encrypted Data

3 Schlusswort

Explicitness

Prinzip 1

- **Every message should say what it means.**
  The interpretation of the message should depend only on its content.

→ Allumfassendes universales Prinzip
→ Es soll möglich sein, einen deutschen Satz zu formulieren, der die Nachricht beschreibt

Explicitness

Prinzip 1

- **Every message should say what it means.**
  The interpretation of the message should depend only on its content.

→ Allumfassendes universales Prinzip
→ Es soll möglich sein, einen deutschen Satz zu formulieren, der die Nachricht beschreibt

Gliederung

1. Einführung
2. Design Prinzipien
   - Basisprinzip 1. Explicitness
   - Prinzip 3. Naming
   - Prinzip 4. Encryption
   - Prinzip 5. Signing encrypted Data
3. Schlusswort

Naming

Prinzip 3

- **If the identity of a party is essential to the meaning of the message, it is prudent to mention the party's name (identity) explicitly in the message.**

→ Ein Spezialfall des Basisprinzips 1
→ Manchmal können die Namen der Kommunikationspartner aus anderen Daten implizit hervorgehen

Naming

Prinzip 3

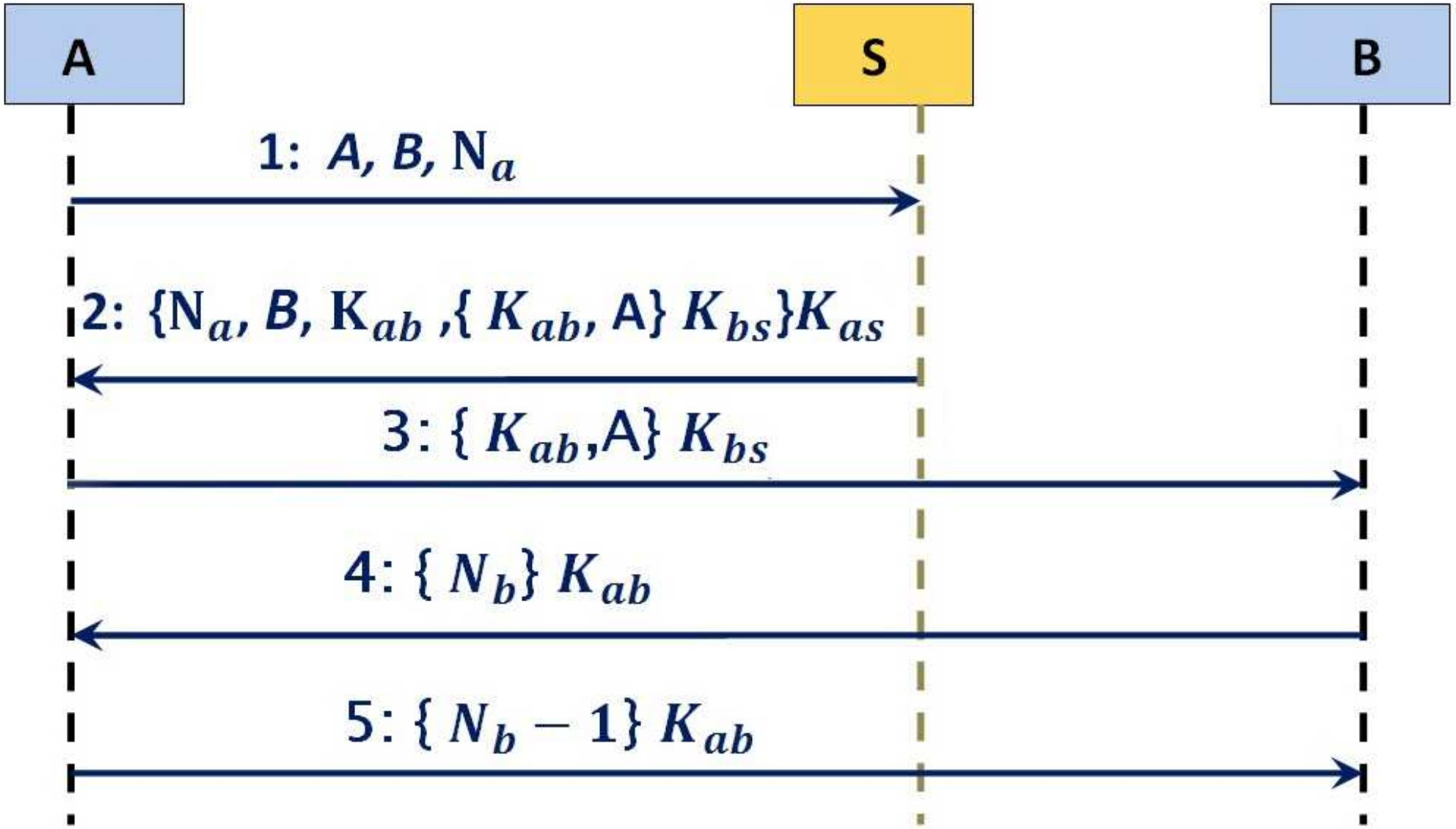
- **If the identity of a party is essential to the meaning of the message, it is prudent to mention the party's name (identity) explicitly in the message.**

→ Ein Spezialfall des Basisprinzips 1
→ Manchmal können die Namen der Kommunikationspartner aus anderen Daten implizit hervorgehen

Beispiel: Needham-Schröder-Protokoll

- Ziel: die Verbindung zwischen A und B mit Schlüssel K_{ab} aufbauen
- Den Schlüssel K_{ab} erzeugt der Server
- Nur A kontaktiert den Server

Beispiel: Needham-Schröder-Protokoll



Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: S generiert den Schlüssel auf Anfrage von A
- Message 2: N_a als Nachweis der Aktualität
- Message 3: A schickt den Schlüssel an B
- Messages 4-5: Handshaking von A und B

Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: S generiert den Schlüssel auf Anfrage von A
- Message 2: N_a als Nachweis der Aktualität
- Message 3: A schickt den Schlüssel an B
- Messages 4-5: Handshaking von A und B

Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: S generiert den Schlüssel auf Anfrage von A
- Message 2: N_a als Nachweis der Aktualität
- Message 3: A schickt den Schlüssel an B
- Messages 4-5: Handshaking von A und B

Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: S generiert den Schlüssel auf Anfrage von A
- Message 2: N_a als Nachweis der Aktualität
- Message 3: A schickt den Schlüssel an B
- Messages 4-5: Handshaking von A und B

Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: S generiert den Schlüssel auf Anfrage von A
- Message 2: N_a als Nachweis der Aktualität
- Message 3: A schickt den Schlüssel an B
- Messages 4-5: Handshaking von A und B

Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: A wird implizit durch K_{as} angegeben.
- Message 2: B muss explizit angegeben werden!
- Sonst: Man-In-The-Middle-Attacke möglich wie folgt:

Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: A wird implizit durch K_{as} angegeben.
- Message 2: B muss explizit angegeben werden!
- Sonst: Man-In-The-Middle-Attacke möglich wie folgt:

Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: A wird implizit durch K_{as} angegeben.
- Message 2: B muss explizit angegeben werden!
- Sonst: Man-In-The-Middle-Attacke möglich wie folgt:

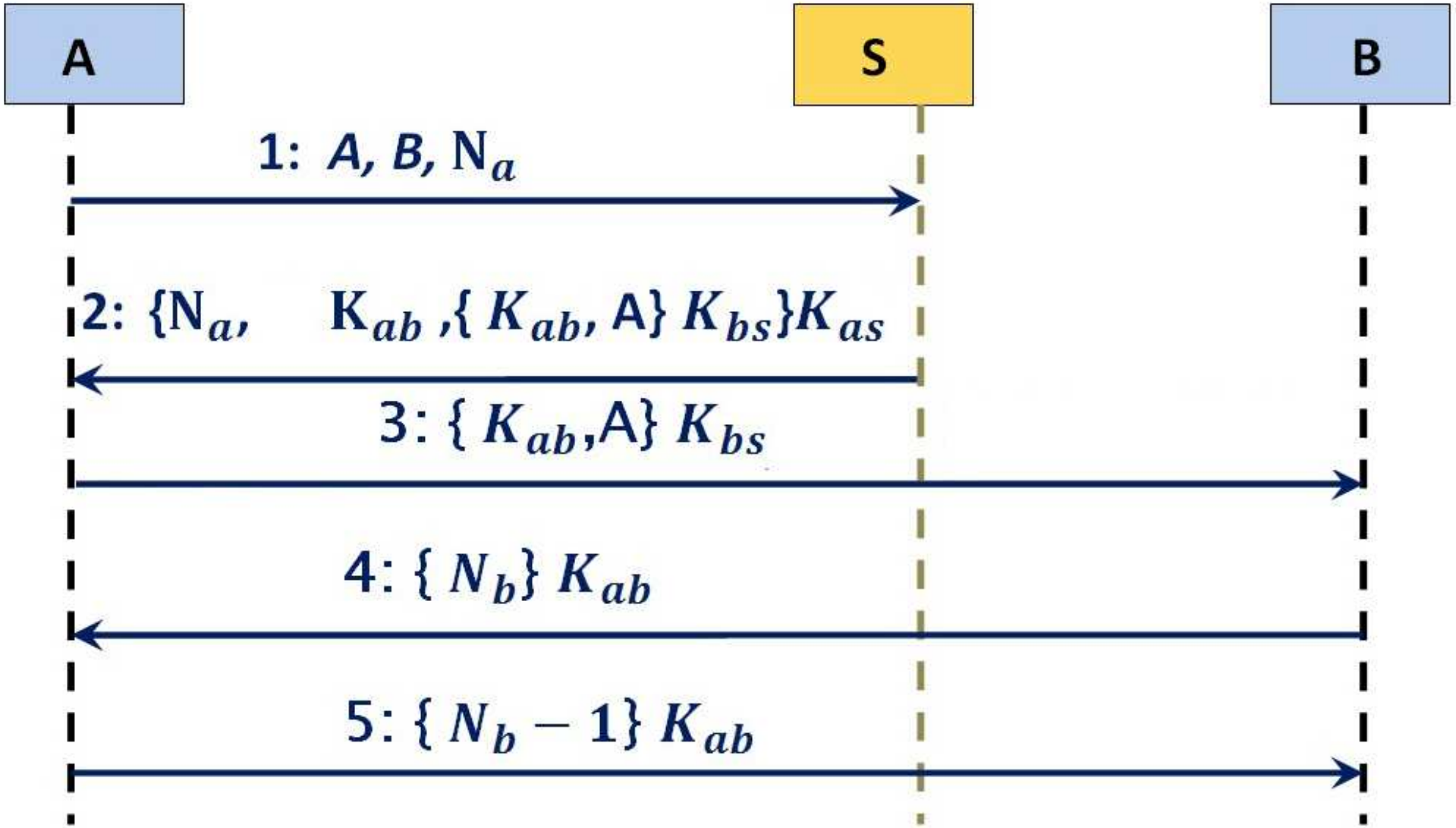
Beispiel: Needham-Schröder-Protokoll

Needham-Schröder

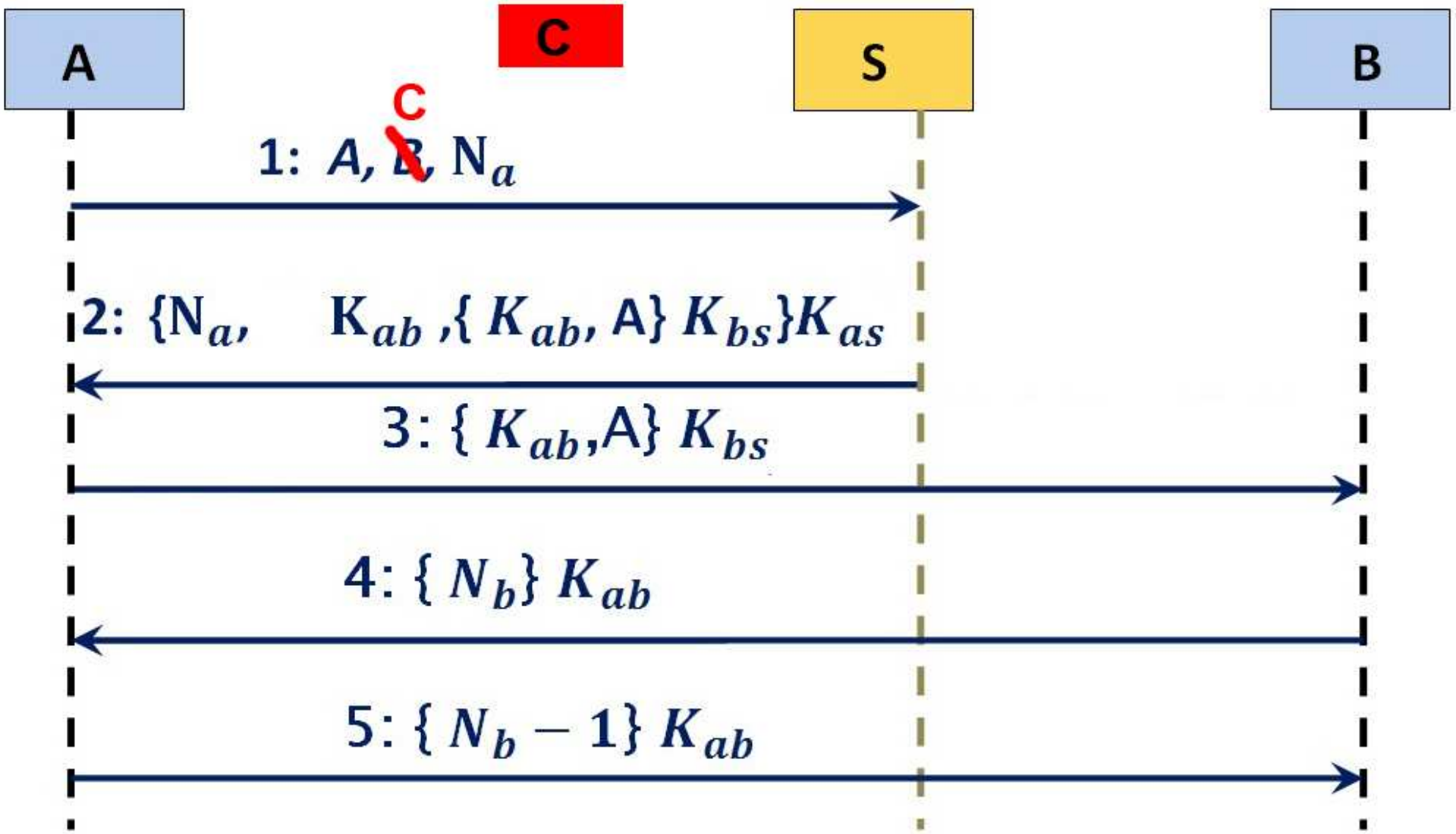
Message 1: $A \rightarrow S : \{A, B, N_a\}$
Message 2: $S \rightarrow A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
Message 3: $A \rightarrow B : \{K_{ab}, A\}_{K_{bs}}$
Message 4: $B \rightarrow A : \{N_b\}_{K_{ab}}$
Message 5: $A \rightarrow B : \{N_b - 1\}_{K_{ab}}$

- Message 2: A wird implizit durch K_{as} angegeben.
- Message 2: B muss explizit angegeben werden!
- Sonst: Man-In-The-Middle-Attacke möglich wie folgt:

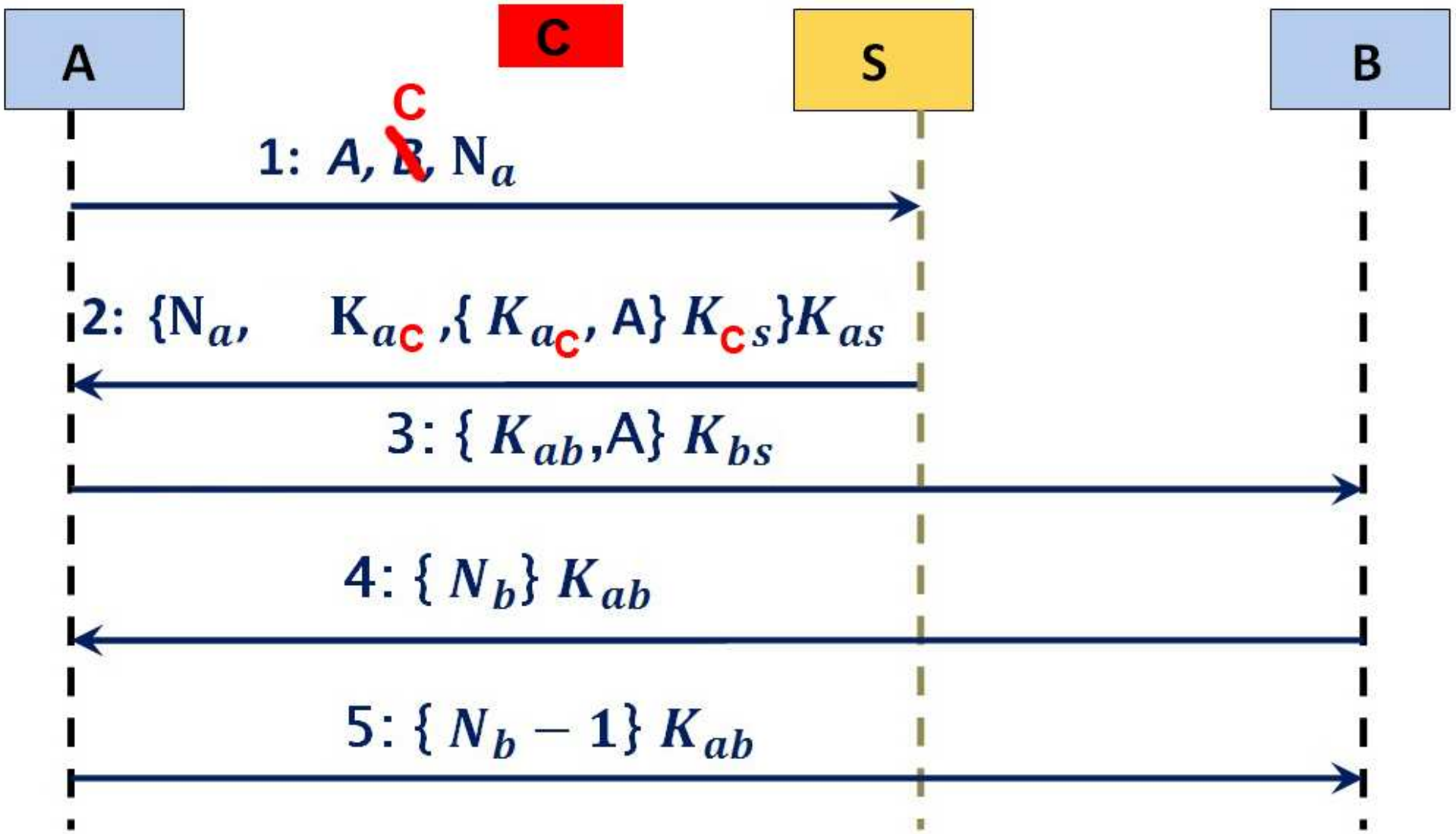
Beispiel: Needham-Schröder-Protokoll



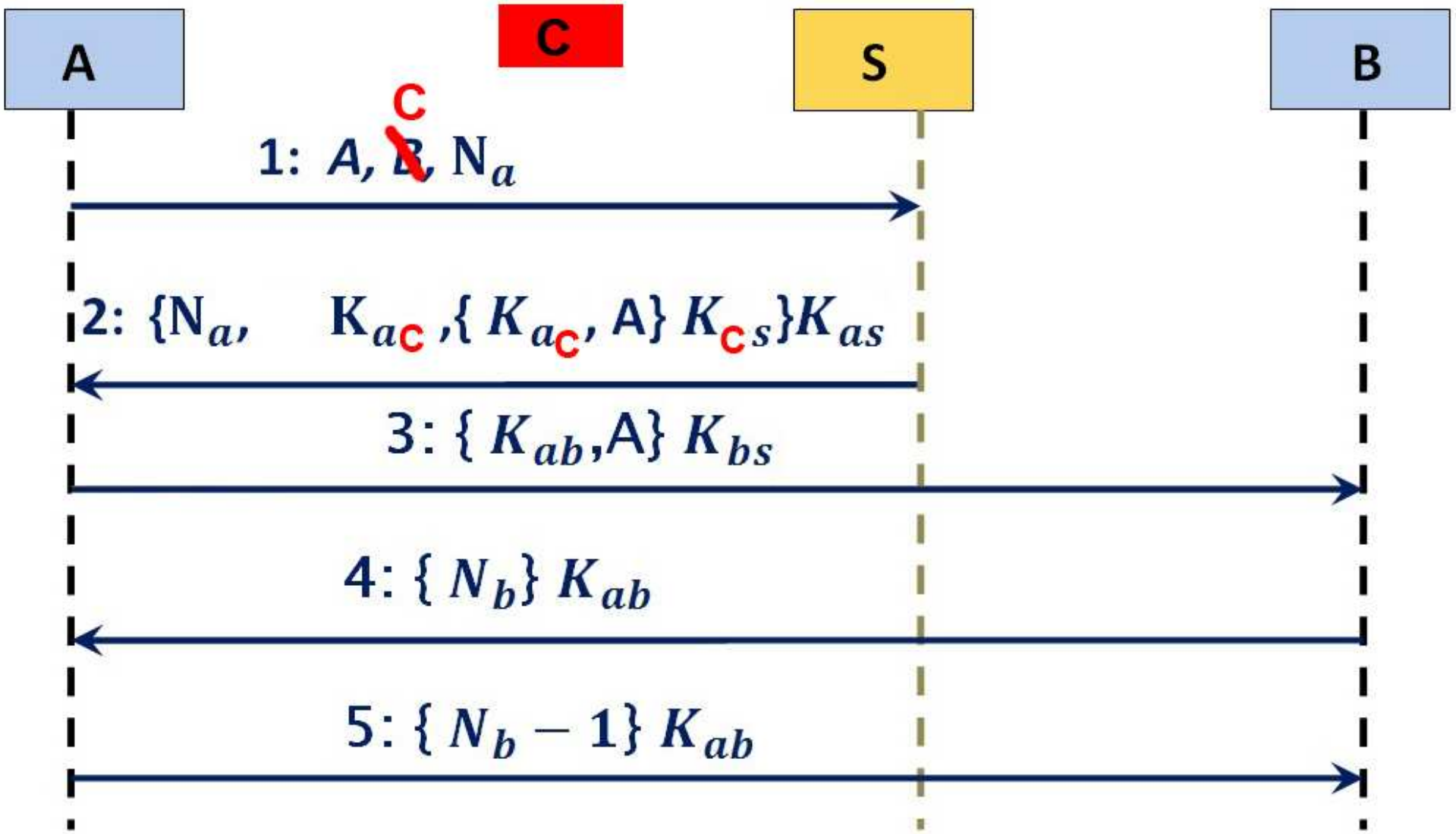
Beispiel: Needham-Schröder-Protokoll



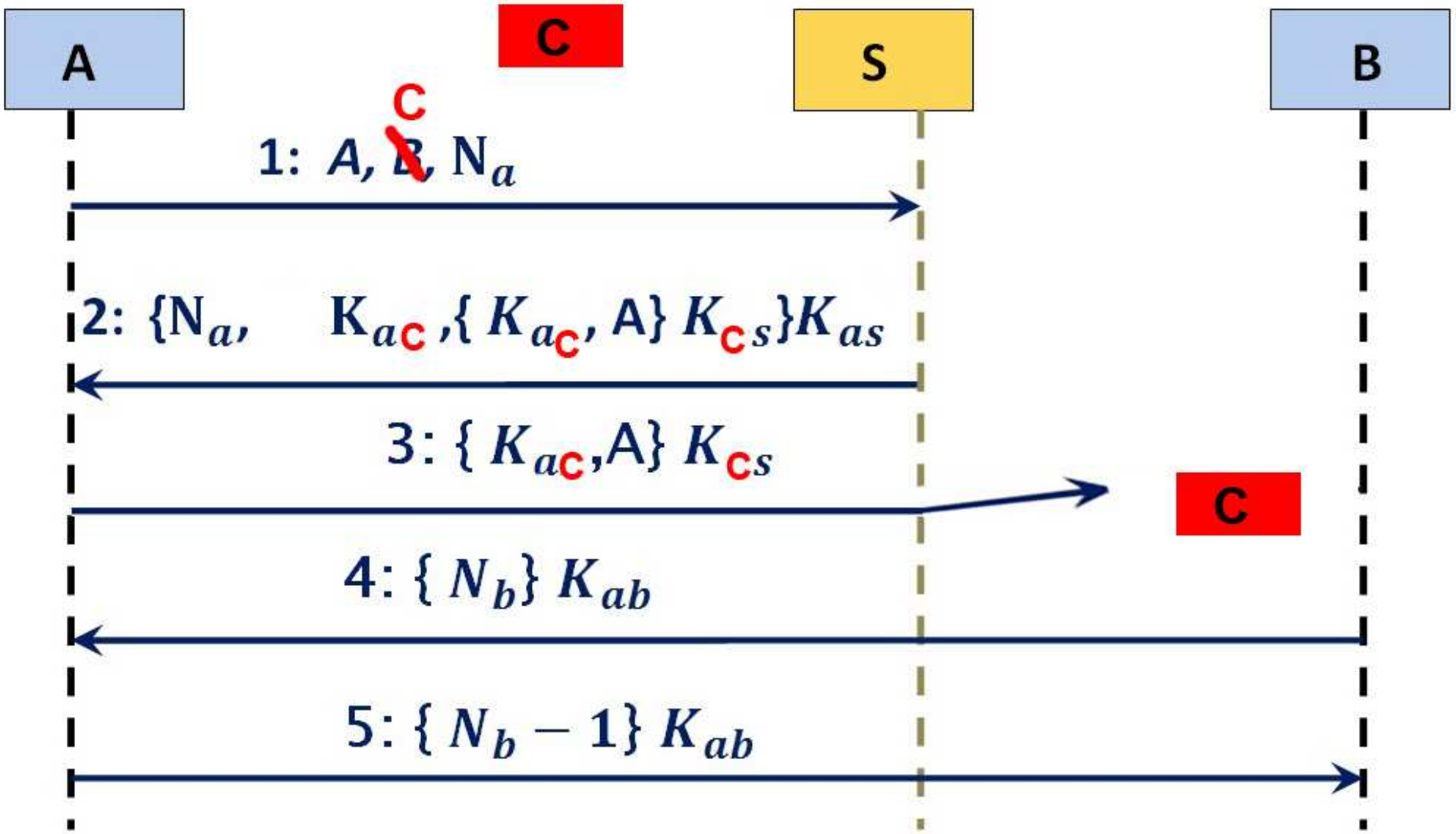
Beispiel: Needham-Schröder-Protokoll



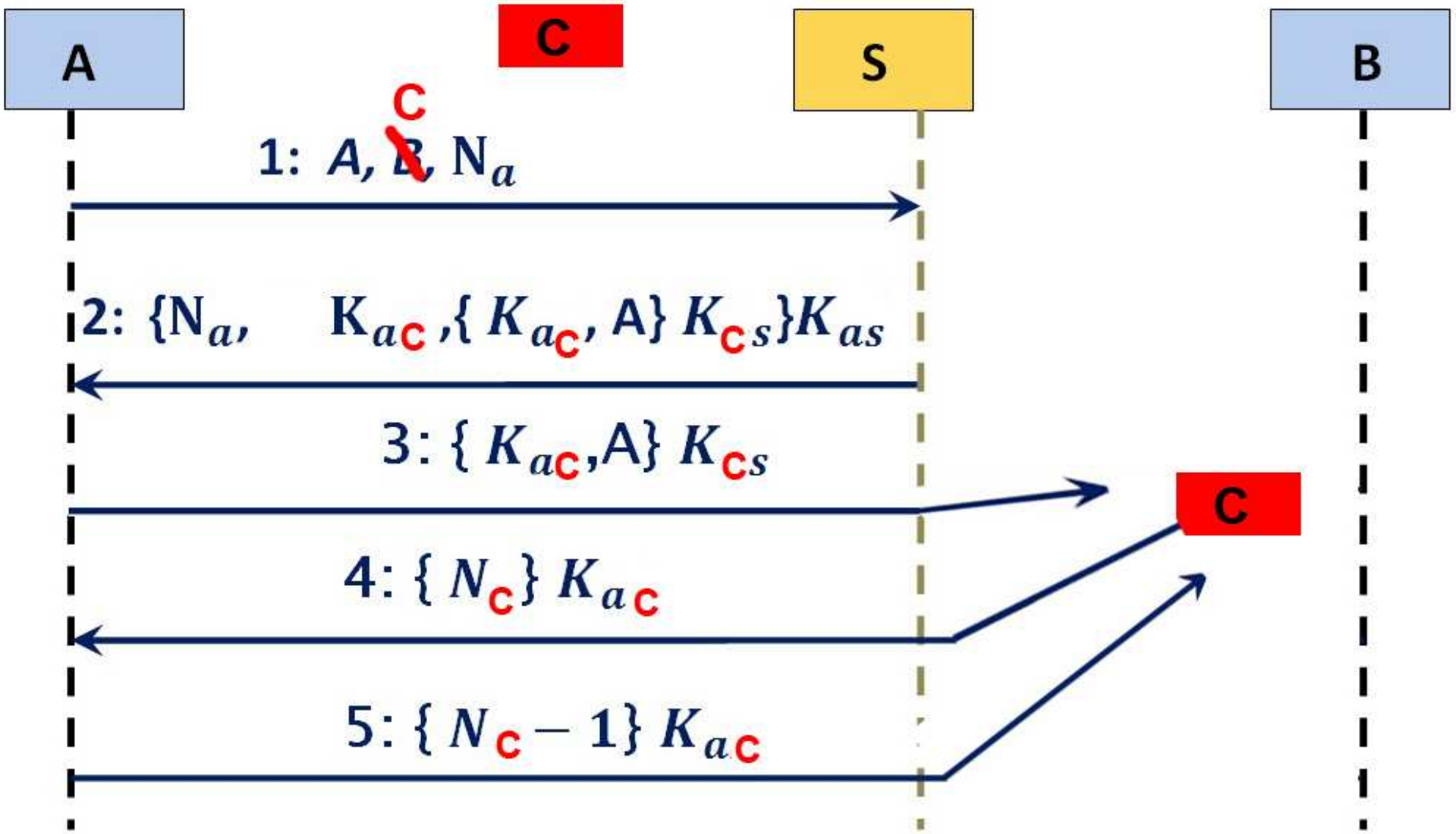
Beispiel: Needham-Schröder-Protokoll



Beispiel: Needham-Schröder-Protokoll



Beispiel: Needham-Schröder-Protokoll



Gliederung

1. Einführung
2. Design Prinzipien
   - Basisprinzip 1. Explicitness
   - Prinzip 3. Naming
   - Prinzip 4. Encryption
   - Prinzip 5. Signing encrypted Data
3. Schlusswort

Encryption

Prinzip 4

- **Be clear about why encryption is applied**
- **Encryption is not cheap, and not asking precisely why it is being done can lead to redundancy**
- **Encryption is not synonymous with security, and its improper use can lead to errors**

Einsatz der Verschlüsselung

→ um Vertraulichkeit zu garantieren
→ um Authentizität zu garantieren
→ um Integrität zu garantieren
→ manchmal nicht notwendig

Encryption

Prinzip 4

- **Be clear about why encryption is applied**
- **Encryption is not cheap, and not asking precisely why it is being done can lead to redundancy**
- **Encryption is not synonymous with security, and its improper use can lead to errors**

Einsatz der Verschlüsselung

→ um Vertraulichkeit zu garantieren
→ um Authentizität zu garantieren
→ um Integrität zu garantieren
→ manchmal nicht notwendig

Beispiel: vereinfachtes Kerberos

TICKET =

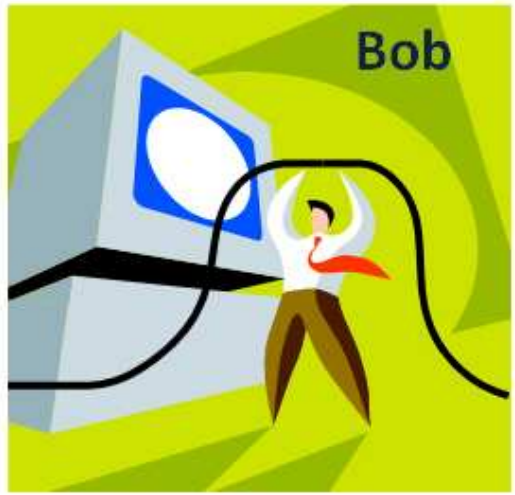
{Ts, Lifetime, K_{ab}, A} K_{bs}

1. Alice,
   Bob

2. {Lifetime, K_{ab}, TICKET, Bob, Ts} K_{as}

3. TICKET, {Alice, Ta} K_{ab}

4. {Ta + 1}K_{ab}



Gliederung

1 Einführung

2 Design Prinzipien
- Basisprinzip 1. Explicitness
- Prinzip 3. Naming
- Prinzip 4. Encryption
- Prinzip 5. Signing encrypted Data

3 Schlusswort

Encryption

Prinzip 5

- **When a party signs material that has already been encrypted, it should not be inferred that the signing party knows the content of the message.**
- **On the other hand, it is proper to infer that the party that signs a message and then encrypts it for privacy knows the content of the message.**

→ Signatur wird manchmal als Beweis angenommen, dass der signierende Kommunikationspartner den Inhalt der Nachricht kennt
→ Dies führt zu Fehlern

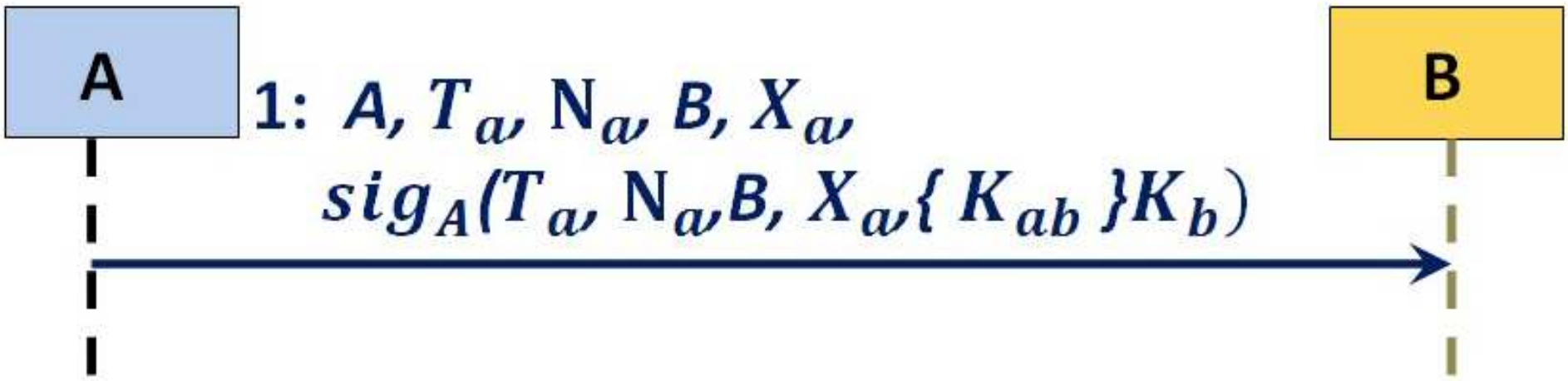
Encryption

Prinzip 5

- **When a party signs material that has already been encrypted, it should not be inferred that the signing party knows the content of the message.**
- **On the other hand, it is proper to infer that the party that signs a message and then encrypts it for privacy knows the content of the message.**

→ Signatur wird manchmal als Beweis angenommen, dass der signierende Kommunikationspartner den Inhalt der Nachricht kennt
→ Dies führt zu Fehlern

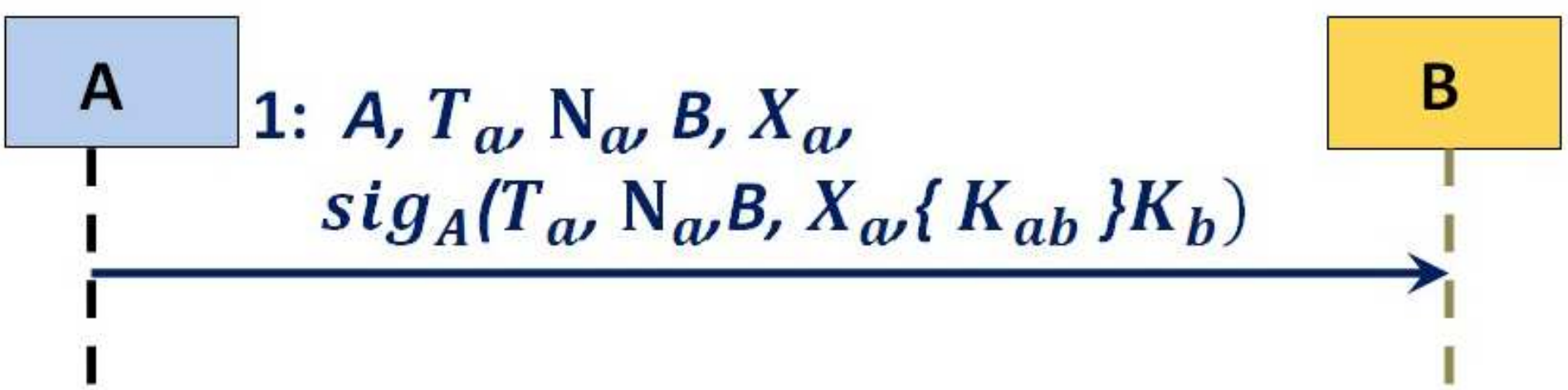
Beispiel: CCITT X.509. One Pass Authentication



Idee:

- K_{ab} geheim übertragen
- Authentikation von A gegenüber B

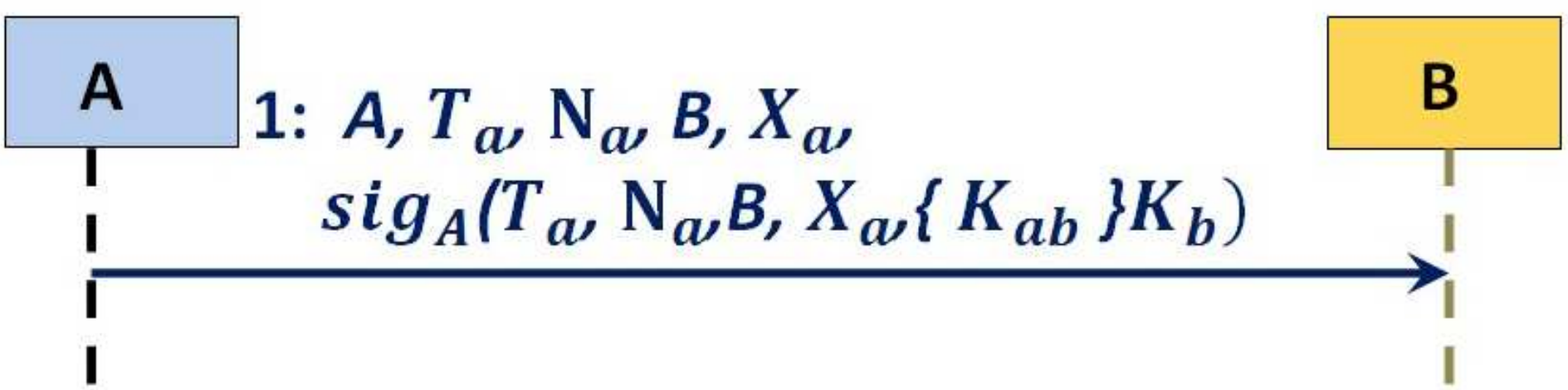
Beispiel: CCITT X.509 One Pass Authentication



Anfälligkeiten:

- Möglich, dass Sender den Inhalt der verschlüsselten Nachricht nicht kennt
- Attacke möglich

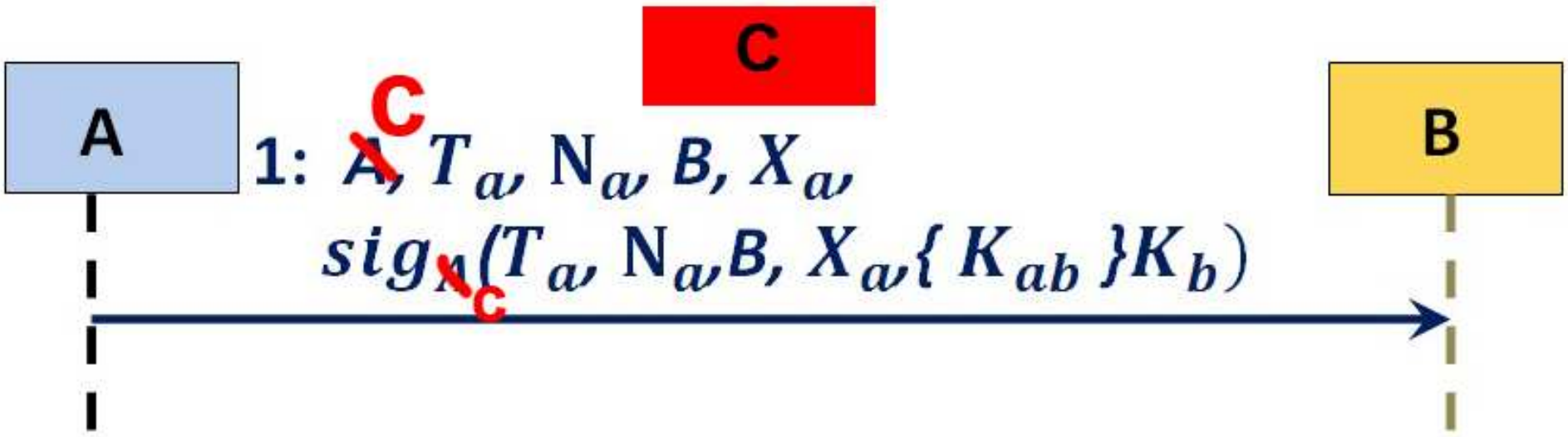
Beispiel: CCITT X.509 One Pass Authentication



Anfälligkeiten:

- Möglich, dass Sender den Inhalt der verschlüsselten Nachricht nicht kennt
- Attacke möglich

Beispiel: CCITT X.509 One Pass Authentication



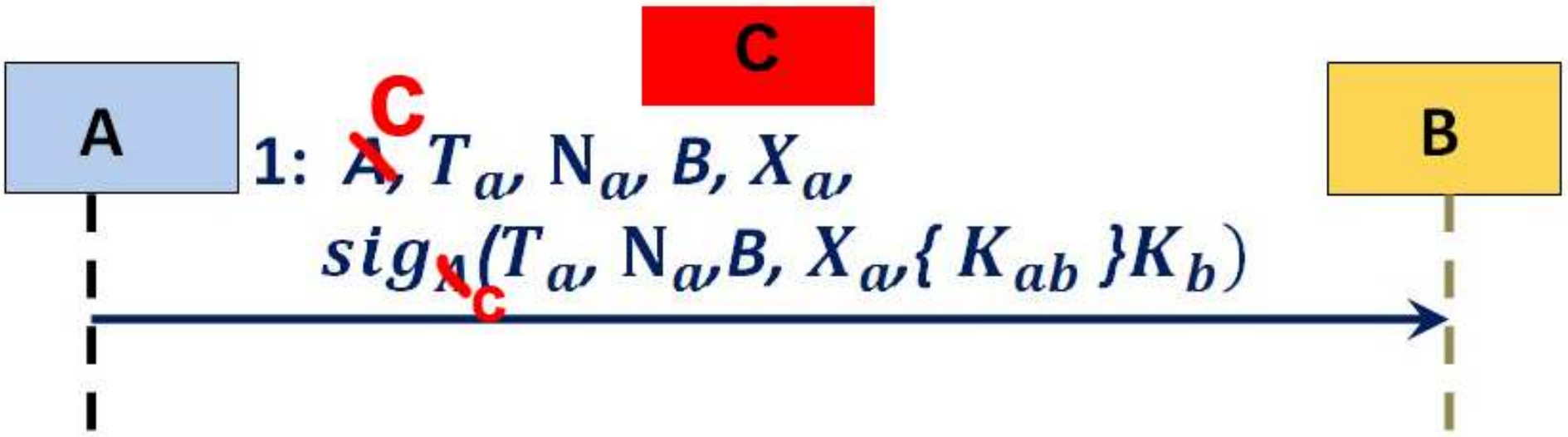
Attacke

- C entfernt die Signatur von A und ersetzt sie durch seine
- Verschlüsselte Nachricht wird blind weitergeleitet

Konsequenz:

- Verletzung der Authentizität von X_a und K_{ab}
- B denkt, er teile den Schlüssel mit C, was falsch ist

Beispiel: CCITT X.509 One Pass Authentication



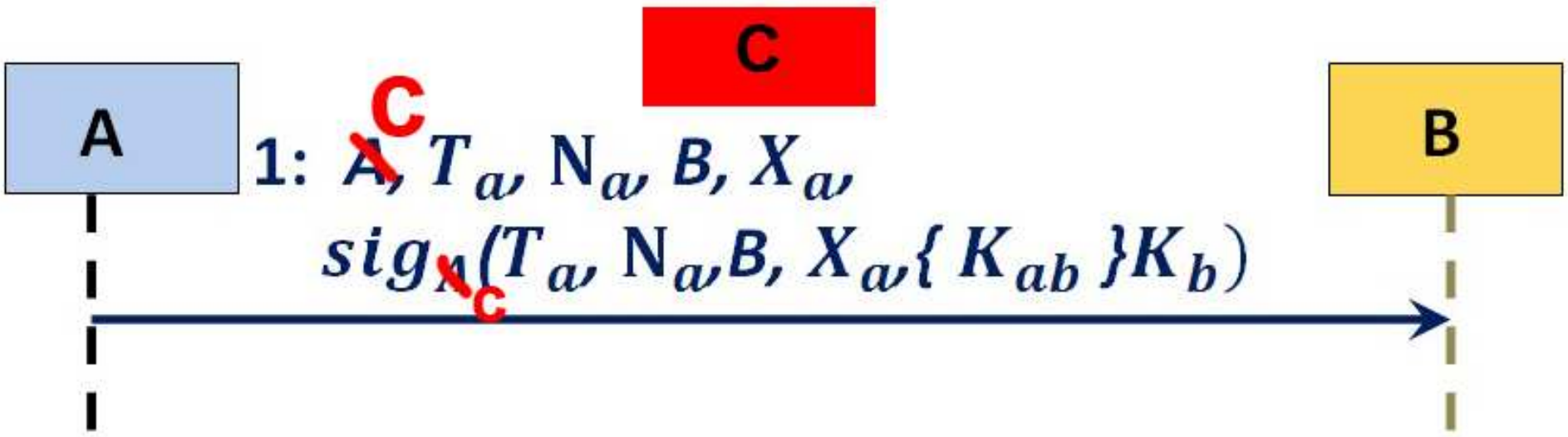
Attacke

- C entfernt die Signatur von A und ersetzt sie durch seine
- Verschlüsselte Nachricht wird blind weitergeleitet

Konsequenz:

- Verletzung der Authentizität von X_a und K_{ab}
- B denkt, er teile den Schlüssel mit C, was falsch ist

Beispiel: CCITT X.509 One Pass Authentication



Verhinderung

- **Erst signieren, dann verschlüsseln**
- **Message 1:** $A \rightarrow B : A, T_a, N_a, B, X_a, (sig)_a(T_a, N_a, B, X_a, \{K_{ab}\})_{K_b}$

Gliederung

1 Einführung

2 Design Prinzipien
- Basisprinzip 1. Explicitness
- Prinzip 3. Naming
- Prinzip 4. Encryption
- Prinzip 5. Signing encrypted Data

3 Schlusswort

Schlusswort

Protokolldesign

- Design von kryptographischen Protokollen sollte bewusst anhand der Prinzipien durchgeführt werden.
- Einhaltung der Regeln reduziert die Kosten, wie auch in anderen Engineering-Bereichen.
- KISS "Keep It Simple Stupid" wird oft ignoriert.

Weiterführende Literatur

- Martin Abadi and Roger Needham "Prudent Engineering Practice for Cryptographic Protocols". 1994
- Martin Abadi "Security Protocols: Principles and Calculi Tutorial Notes". 1994
- Debra S. Herrmann "A Practical Guide to Security Engineering and Information Assurance". Auerbach Publications, 2002