Protokolldesign

Hauptseminar: Kryptographische Protokolle

Autor: Franz Saller

16.06.2009

0. Gliederung

1. Grundlegende Sicherheitseigenschaften kryptographischer Protokolle
2. Analyse von Protokolleigenschaften
3. Das Needham-Schroeder-Protokoll
4. Timestamps
5. Nonces
6. Vertrauen
7. Zusammenfassung wichtiger Gestaltungsprinzipien
8. Schluss und Ausblick

1. Grundlegende Sicherheitseigenschaften

Grundsätzlich sollten kryptographische Protokolle eine oder mehrere dieser Kriterien erfüllen:

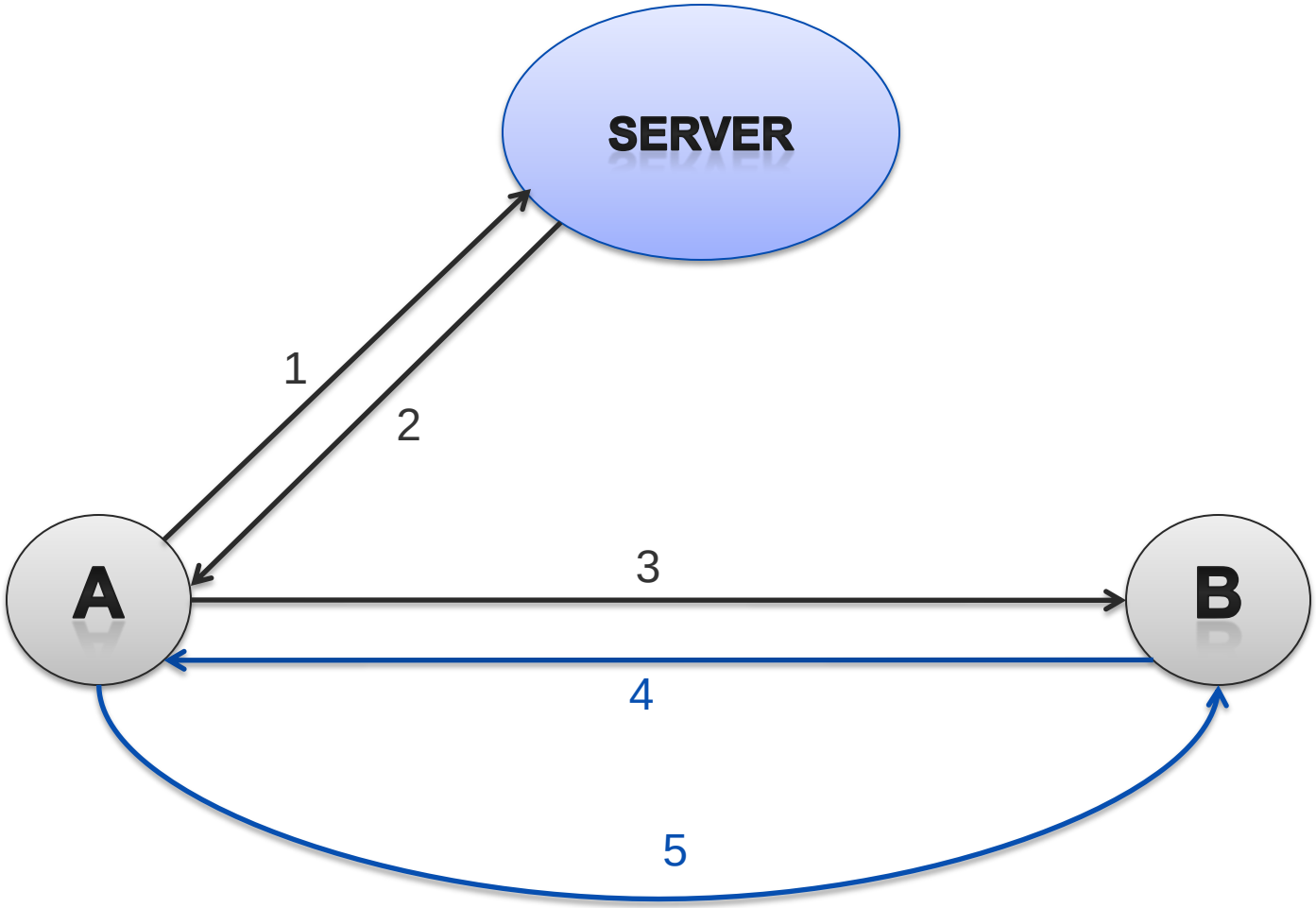
- Secrecy (Kerckhoff-Prinzip)
- Integrity (Änderungsschutz)
- Authenticity (Fälschungsschutz)
- Originality (Ursprünglichkeit)
- Reliability (Nichtabstreitbarkeit)
- Confidentiality (Vertraulichkeit)

2. Analyse von Protokolleigenschaften

- Protokolleigenschaften lassen sich am besten durch Beispiele veranschaulichen
- Beispiel: Das **Needham-Schroeder-Protokoll**
- Ziel des Protokolls: Sicheren Sitzungsschlüssel für die Kommunikation zweier Teilnehmer A und B bereitstellen und für einen sicheren Schlüsselaustausch sorgen

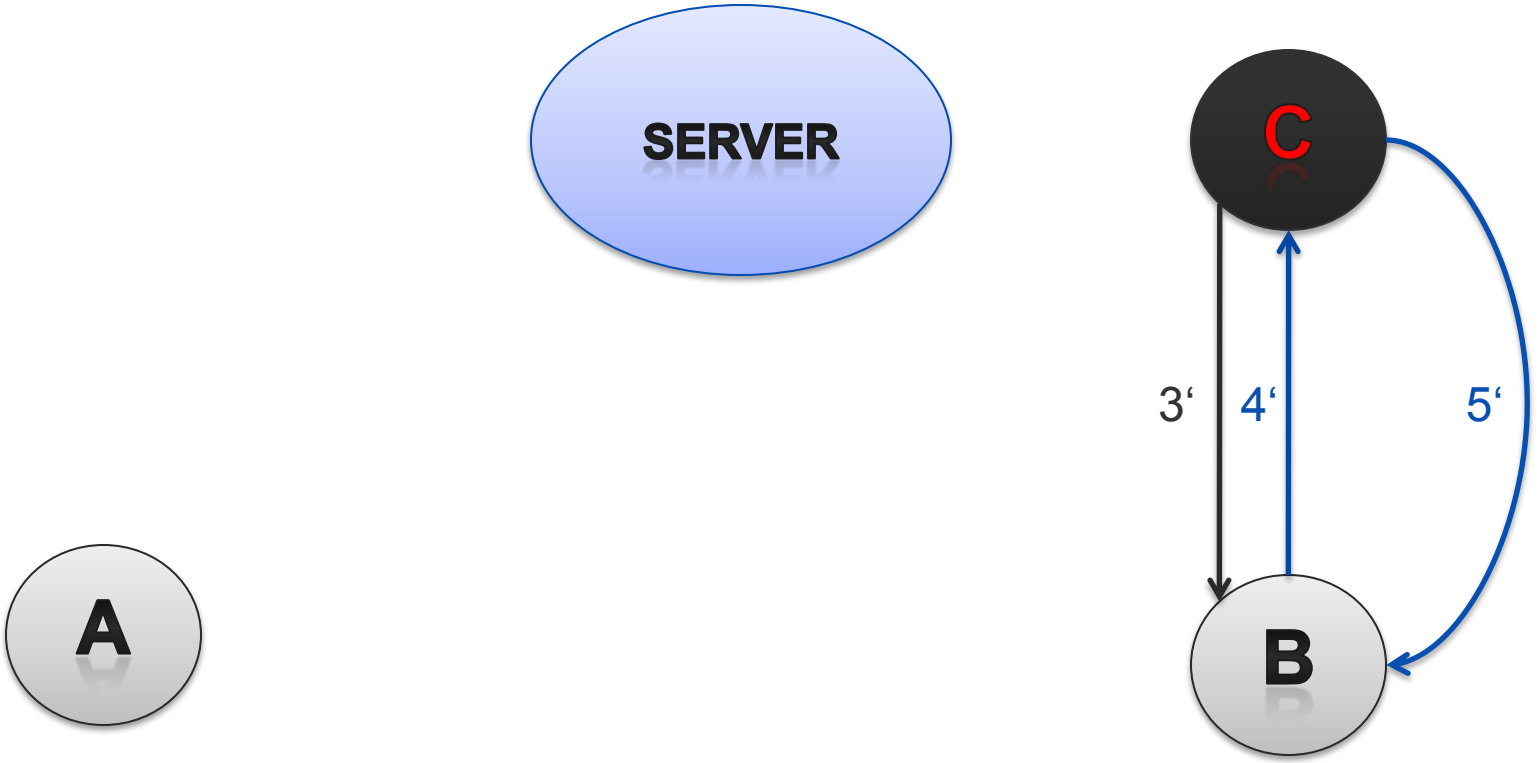
- Die Bedrohung wird durch das *Dolev-Yao Angreifer-Modell* beschrieben. Ein Angreifer kann:
  - Nachrichten abhören
  - Nachrichten abfangen
  - Nachrichten manipulieren
  - Aber er kann **nicht verschlüsselte Nachrichten entschlüsseln, wenn ihm der dazu passende Schlüssel fehlt!**

3. Das Needham-Schroeder-Protokoll



3. Das Needham-Schroeder-Protokoll

Schwachstelle des Protokolls: Was passiert, wenn ein Sitzungsschlüssel zweier Teilnehmer geknackt wurde?



Lösung: Timestamps

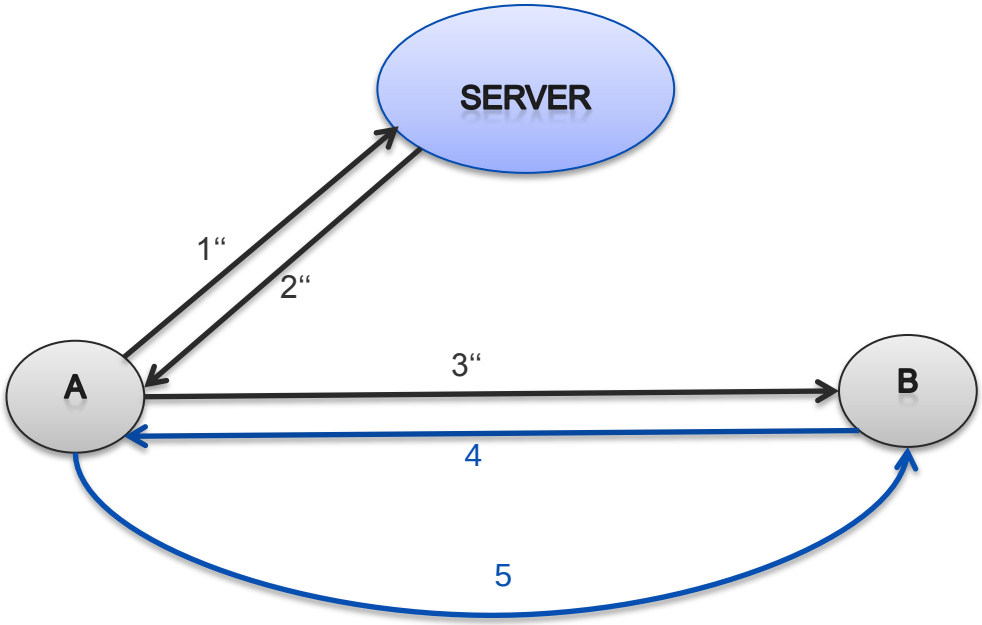
4. Zeitstempel (Timestamps)

- Motivation: Verhinderung von Replay-Attacken

- Gewährleistung der Neuheit (*Freshness*) einer Nachricht mithilfe einer Referenz zur *absoluten* Zeit

4.1 Anwendung von Timestamps

- Motivation: Neuheit (Freshness) der Sitzungsschlüssel garantieren und der Verwendung alter, evtl. kompromittierter Schlüssel vorbeugen
- Lösung durch Modifikation des Needham-Schroeder-Protokolls mit Timestamps



4.2 Probleme bei der Verwendung von Timestamps

- Verwendung von Timestamps impliziert synchronisierte Systemuhren
- Entstehendes Problem: Keine einheitliche Systemzeit bei allen Teilnehmern des Protokolls
- Zeitdifferenzen können verursacht werden durch:
  - **Langsame** oder **schnelle** Uhren
  - Zurückgesetzte Systemuhr durch Batterieausfall

→ Einmaliges Setzen der Uhrzeiten auf allen beteiligten Rechner reicht nicht aus!

→ Uhren müssen fortlaufend synchronisiert werden

5. Nonces: Verwendungszweck und Ausprägungsformen

- Dienen vorwiegend zur Verhinderung von Replay-Angriffen
- Challenge-Response-Verfahren mit Nonces können auch als Alternative zu Timestamps dienen

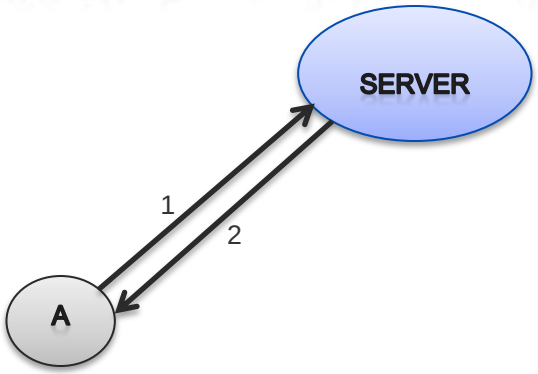
- Nonces müssen nicht unbedingt zufällig gewählt werden – falls sie vorhersehbar sind, müssen dann aber geschützt werden (z.B. in Form eines Zählers)
- Beispiel für einen potentiellen Angriff:
  - Zeitsynchronisierungs-Protokoll

5.1 Gefahr durch vorhersehbare Nonces

- Beispiel: Protokoll zur Zeitsynchronisierung

Message 1: $A \rightarrow S : A, N_a$

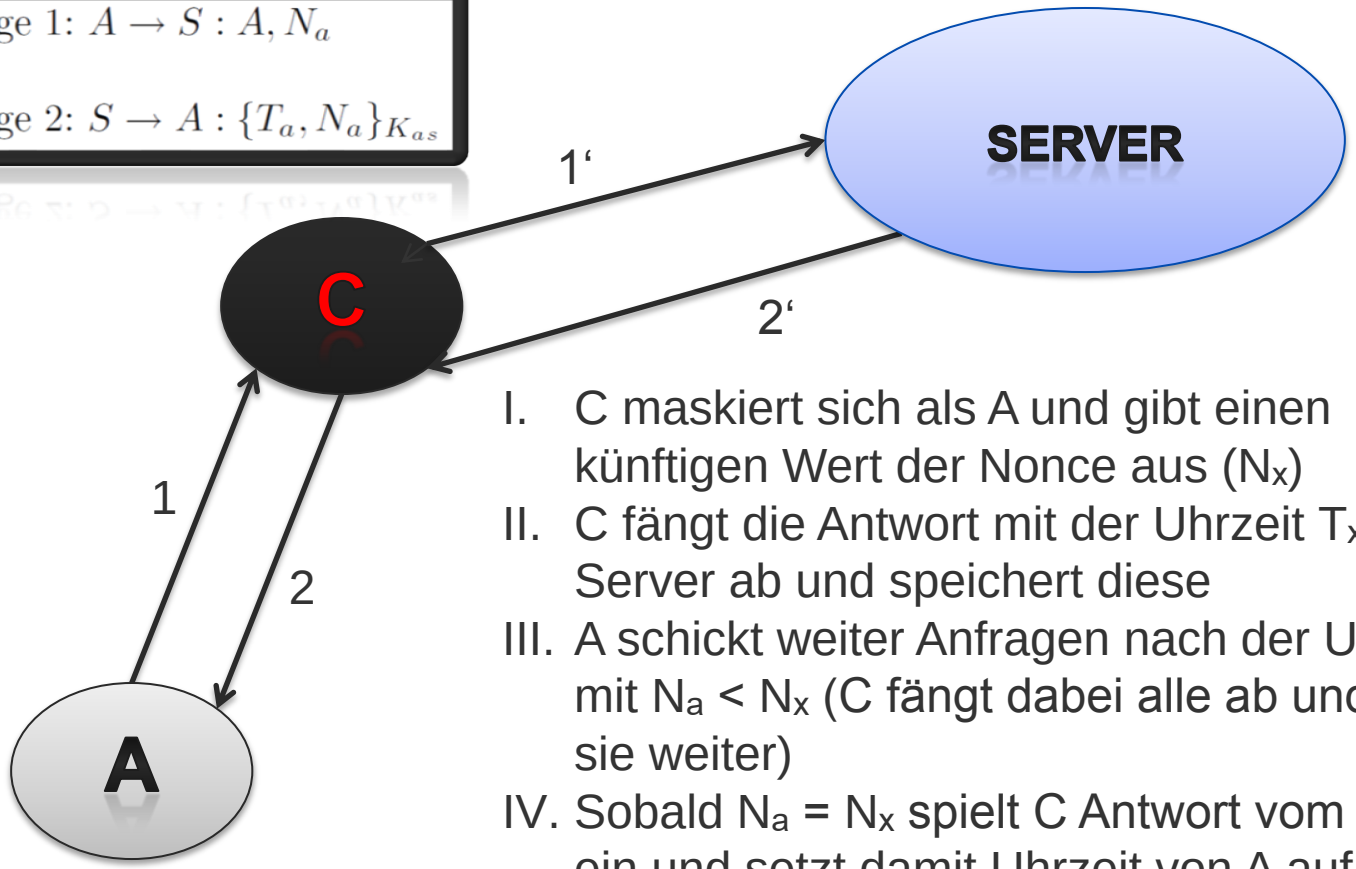
Message 2: $S \rightarrow A : \{T_a, N_a\}_{K_{as}}$



5.2 Zurücksetzen der Systemzeit von A durch Replay-Attacke

Message 1: $A \rightarrow S : A, N_a$

Message 2: $S \rightarrow A : \{T_a, N_a\}_{K_{as}}$



I. C maskiert sich als A und gibt einen künftigen Wert der Nonce aus (N_x)
II. C fängt die Antwort mit der Uhrzeit T_x vom Server ab und speichert diese
III. A schickt weiter Anfragen nach der Uhrzeit mit $N_a < N_x$ (C fängt dabei alle ab und leitet sie weiter)
IV. Sobald $N_a = N_x$ spielt C Antwort vom Server ein und setzt damit Uhrzeit von A auf T_x zurück

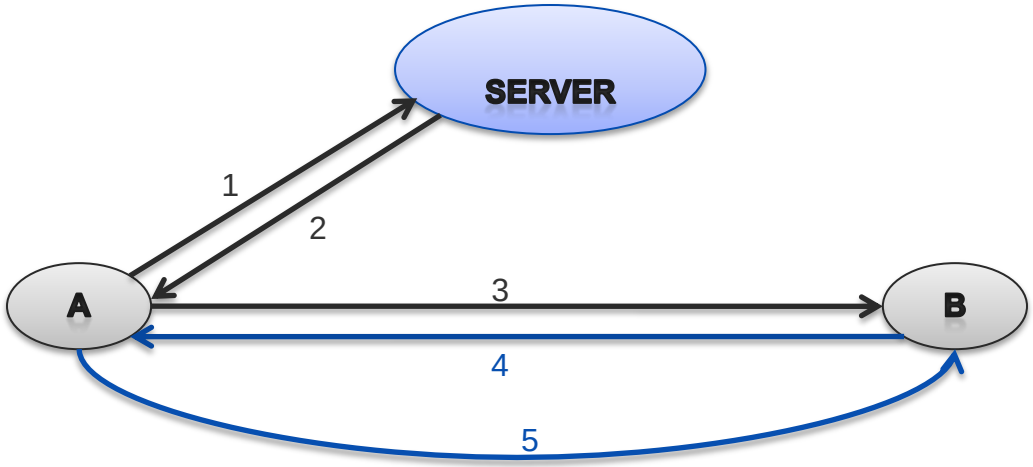
6. Vertrauen

- Entscheidende Fragen beim Protokolldesign
- Wer soll den gemeinsam genutzten Sitzungsschlüssel zweier Teilnehmer erzeugen?
- Wie können die Teilnehmer einander vertrauen?

- Grundsätzliche Alternativen: zentraler oder dezentraler Ansatz

6.1 Zentraler Ansatz

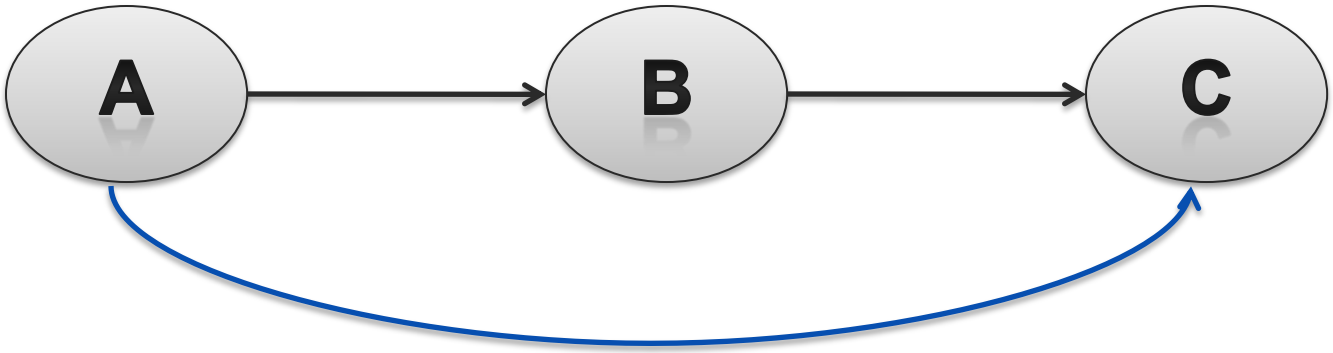
- Lösung beim Needham-Schroeder-Protokoll durch dezentralen Ansatz:
  Authentifizierungs- bzw. Schlüsselverteilungsserver **S** als vertrauenswürdige, sichere Basis des Protokolls



6.1 Dezentraler Ansatz

- Direktes Vertrauen
- A gibt seine direkten Vertrauensbeziehungen explizit an

- Indirektes Vertrauen (Transitivität)
- A vertraut auch Teilnehmern, denen seine direkten Bekannten vertrauen



7. Zusammenfassung wichtiger Protokollgestaltungsprinzipien

Zeitstempel

- Bei Verwendung von Zeitstempeln wird die Gewährleistung einer Zeit-Synchronisation automatisch impliziert
- Bei fehlender Synchronisation kann Freshness nicht garantiert werden

Schlüssel

- Der Verwendungszeitpunkt eines Schlüssels ist kein Anhaltspunkt für dessen Sicherheit bzw. Vertraulichkeit

Vorhersehbare Nonces

- Falls berechenbare Nonces wie Zähler verwendet werden, um die Neuheit von Nachrichten zu garantieren, müssen diese geschützt werden

Vertrauen

- Es muss explizit geklärt werden, wie das Vertrauensverhältnis zweier Teilnehmer zustande kommt und warum diese Beziehung notwendig ist.

8. Schluss und Ausblick

- Durch Beachtung grundlegender Prinzipien bei der Protokollgestaltung können häufige Fehler vermieden werden

- Neben einem sicheren Verschlüsselungsverfahren ist ein sicheres Schlüsselaustauschprotokoll zentral für die Kommunikationssicherheit

Vielen Dank für die Aufmerksamkeit!