

Übungsblatt 6

zur Vorlesung

Verteilte Systeme/Ubiquitous Computing

Wintersemester 2007/2008

Achtung: Die Klausuranmeldung ist unter

<http://www.mobile.ifi.lmu.de/Vorlesungen/ss06/vs/>

freigeschaltet. Diese bleibt bis zum 03.02.2008, 23:59 Uhr (MESZ) geöffnet. Die Klausur findet am Donnerstag, dem 07.02.2008, in den Räumen der Theresienstraße 39 um 18:30 statt (Einlass ab 18:15). Für die Teilnahme ist eine Anmeldung zwingend erforderlich! Nähere Details zur Klausur finden sich auf der oben genannten Web-Site.

Aufgabe 20: (H) Atomare Transaktionen in Verteilten Systemen

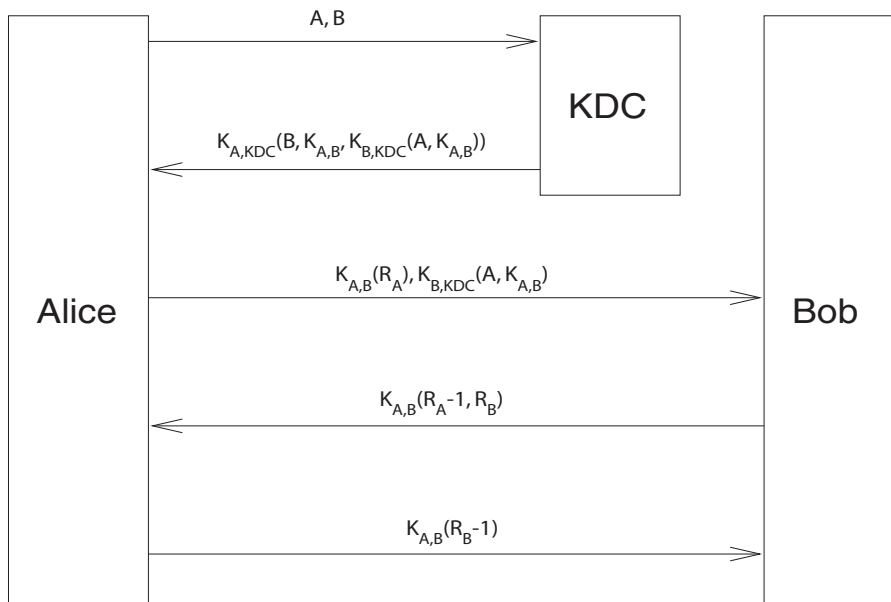
- Welche Eigenschaften besitzen Transaktionen?
- Welche Voraussetzungen müssen hierfür gegeben sein?
- Welche Transaktionsarten existieren in Verteilten Systemen?

Aufgabe 21: (H) Grundlagen der Sicherheit in Verteilten Systemen

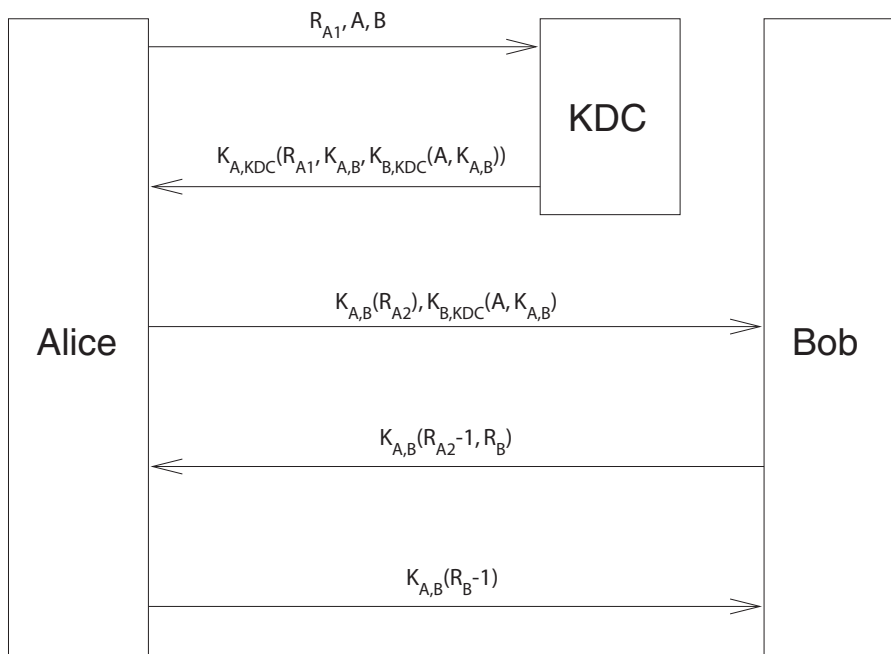
- Was ist der Unterschied zwischen Verfügbarkeit (Availability) und Zuverlässigkeit (Reliability)?
- Was ist der Unterschied zwischen Authentifizierung und Autorisierung?
- Warum müssen Authentifizierung und Nachrichtenintegrität immer gemeinsam gewährleistet werden, um sinnvoll zu sein?

Aufgabe 22: (H) Authentifizierung unter Verwendung eines Key Distribution Centers

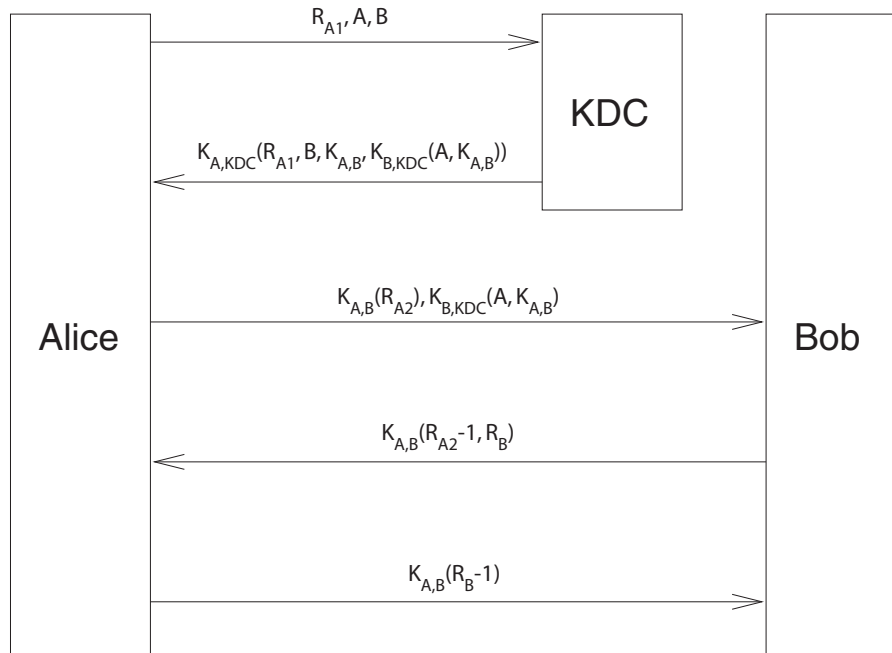
- a. Wie kann jemand, der auf irgendwelchen Wegen $K_{B,KDC}$ herausbekommen hat, in den vermeintlich sicheren Kanal eindringen, der durch das folgende Protokoll hergestellt wurde?



- b. Wenn Bob weiß, dass einem Fremden sein Schlüssel $K_{B,KDC}$ bekannt ist, kann er einen Angriff dadurch verhindern, dass er $K_{B,KDC}$ ändert?
- c. Wie kann bei Verwendung des folgenden Protokolls selbst ohne Kenntnis von $K_{B,KDC}$ ein Angriff erfolgen?



- d. Wo liegt die Schwachstelle des Needham-Schroeder-Protokolls, so wie es unten abgebildet ist? Wie kann diese Schwachstelle beseitigt werden?



Aufgabe 23: (H) RSA

Der RSA-Algorithmus wurde nach seinen Erfindern Rivest, Shamir, Adleman benannt. Der öffentliche und der private Schlüssel werden nach diesem Algorithmus wie folgt bestimmt:

- Es werden zwei sehr große Primzahlen, p und q , bestimmt.
- Aus diesen wird $n = p * q$ und $z = (p - 1) * (q - 1)$ berechnet.
- Es wird eine Zahl e gewählt, so dass diese keinen gemeinsamen Teiler mit z besitzt.
- Darauf wird eine Zahl d ermittelt, die $d * e \equiv 1 \pmod{z}$ erfüllt. Dies ist gleichbedeutend mit $d = e^{-1} \pmod{z}$.

Eins der beiden Zahlenpaare (d, n) und (e, n) kann als öffentlicher Schlüssel verwendet werden, während das andere als privater Schlüssel dient.

- Wählen Sie $p = 47$ und $q = 71$. Berechnen Sie daraus n und z . Nehmen Sie für e die Zahl 79 an. Über den erweiterten Euklidischen Algorithmus kann daraus d mit 1019 bestimmt werden.
- Benutzen Sie e als privaten Schlüssel und kodieren Sie damit die Nachricht "6882326879666683". Dabei müssen Sie die Nachricht in Blöcke gleicher Länge zerteilen, die jeweils kleiner als n sind. Diese Blöcke transformieren Sie, indem Sie diese mit e exponentieren und Modulo n nehmen. Hängen Sie die Ergebnisse einfach aneinander, um die verschlüsselte Nachricht zu generieren. Für diese Rechnungen empfiehlt es sich, den "Basic Calculator" unter Unix zu verwenden, den Sie über den Befehl "bc" aufrufen können. (Modulo wird durch das Zeichen % repräsentiert.) Handelsübliche Taschenrechner sind durch die Größe der Ergebnisse überfordert und liefern falsche Ergebnisse.
- Entschlüsseln Sie die Nachricht, indem Sie die verschlüsselten Blöcke mit d potenzieren und anschließend eine Modulo- n -Operation ausführen.

Aufgabe 24: (T) IDEA

Der RSA-Algorithmus ist zwar sicherer als symmetrische Verfahren, benötigt aber 100 bis 200 mal mehr Rechenzeit als diese, sofern eine Implementierung in Software gewählt wurde. Bei Implementierungen in Hardware liegt dieser Faktor bei etwa 1000. Deswegen wird RSA in der Regel verwendet, um einen so genannten Sitzungsschlüssel auszutauschen, während die eigentliche Kommunikation dann mit Hilfe eines symmetrischen Verfahrens unter Verwendung dieses Sitzungsschlüssels erfolgt.

Das häufig verwendete Freeware-Produkt “Pretty Good Privacy” (PGP) von Philip Zimmermann benutzt beispielsweise das symmetrische IDEA-Verfahren (IDEA = International Data Encryption Algorithm) für die Kommunikation. Dieses soll im Rahmen eines Tutorials in der Übung exemplarisch vorgestellt werden.