Kryptographie II
Asymmetrische Kryptographie

Christopher Wolf

Fakultät für Mathematik
Ruhr-Universität Bochum

Sommersemester 2010

Organisatorisches

- Vorlesung: **Mi 10:15–11:45** in NA 6/99 (2+2 SWS, 6 CP)
- Übung: **Mi 12:15–13:45** in NA 6/99
- Assistent: **Alexander Meurer**, Korrektor: **Florian Giesen**
- Übungsbetrieb: jeweils abwechselnd alle 2 Wochen
  - Präsenzübung, Start 21. April
  - Zentralübung, Start 28. April (Abgabe: 26. April)
- Übungsaufgaben werden korrigiert.
- Gruppenabgaben bis 3 Personen
- Bonussystem:
  1/3-Notenstufe für 50%, 2/3-Notenstufe für 75%
  Gilt nur, wenn man die Klausur besteht!
- Klausur: Mitte-Ende August (vermutlich)

Literatur

Vorlesung richtet sich nach

- Jonathan Katz, Yehuda Lindell, “Introduction to Modern Cryptography”, Taylor & Francis, 2008

Weitere Literatur

- S. Goldwasser, M. Bellare, “Lecture Notes on Cryptography”, MIT, online, 1996–2008
- O. Goldreich, “Foundations of Cryptography – Volume 1 (Basic Tools)”, Cambridge University Press, 2001
- O. Goldreich, “Foundations of Cryptography – Volume 2 (Basic Applications)”, Cambridge University Press, 2004s
- A.J. Menezes, P.C. van Oorschot und S.A.Vanstone, “Handbook of Applied Cryptography”, CRC Press, 1996

Erinnerung an Kryptographie I

Symmetrische Kryptographie

- Parteien besitzen gemeinsamen geheimen Schlüssel.
- Erlaubt Verschlüsselung, Authentifikation, Hashen, Pseudozufallspermutationen.
- **Frage:** Wie tauschen die Parteien einen Schlüssel aus?

Nachteile

1. U Teilernehmer benötigen $\binom{U}{2} = \Theta(U^2)$ viele Schlüssel.
2. Jeder Teilnehmer muss $U - 1$ Schlüssel sicher speichern. Update erforderlich, falls Teilnehmer hinzukommen oder gelöscht werden.
3. Schlüsselaustausch funktioniert nicht in offenen Netzen.

Schlüsselverteilungs-Center (KDC)

Partielle Lösung: Verwenden vertrauenswürdige Instanz

- IT-Manager eröffnet Key Distribution Center (KDC).
- Teilnehmer besitzen gemeinsamen, geheimen Schlüssel mit KDC.
- Alice schickt Nachricht “Kommunikation mit Bob” an KDC.
- Alice authentisiert Nachricht mit ihrem geheimen Schlüssel.
- KDC wählt einen *Session-Key* k, d.h. einen neuen Schlüssel.
- KDC schickt Verschlüsselung $E_{Alice}(k)$ an Alice.
- KDC schickt Verschlüsselung $E_{Bob}(k)$ an Bob.

Alternativ im Needham Schröder Protokoll:
KDC schickt $E_{Bob}(k)$ an Alice und diese leitet an Bob weiter.

Vor- und Nachteile von KDCs

Vorteile

- Jeder Teilnehmer muss nur *einen* Schlüssel speichern.
- Hinzufügen/Entfernen eines Teilnehmers erfordert Update *eines* Schlüssels.

Nachteile

- Kompromittierung von KDC gefährdet das gesamte System.
- Falls KDC ausfällt, ist sichere Kommunikation nicht möglich.

Praktischer Einsatz von KDCs

- Kerberos (ab Windows 2000)

Diffie Hellman Gedankenexperiment

Szenario

- Alice will eine Kiste zu Bob schicken.
- Post ist nicht zu trauen, d.h. die Kiste muss verschlossen werden.
- Sowohl Alice als auch Bob besitzen ein Schloss.

Algorithmus 3-Runden Diffie-Hellman Austausch

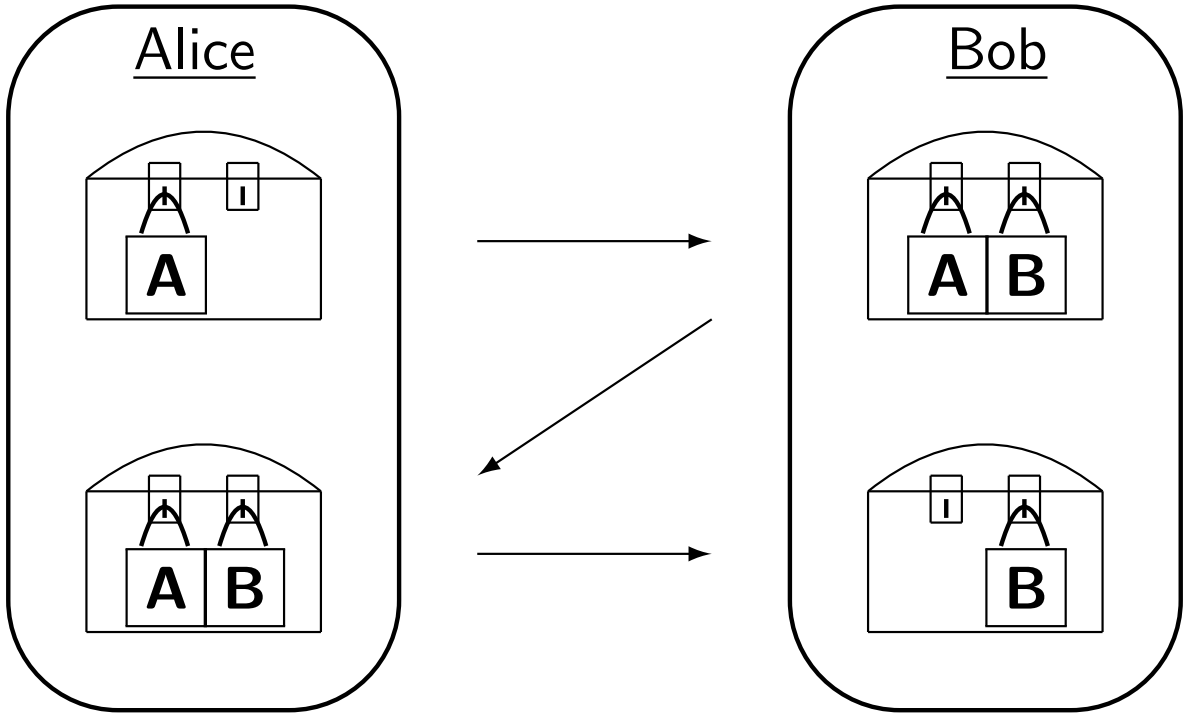
1. Alice sendet die Kiste an Bob, verschlossen mit ihrem Schlüssel.
2. Bob sendet die Kiste zurück, verschlossen mit seinem Schlüssel.
3. Alice entfernt ihr Schloss und sendet die Kiste an Bob.
4. Bob entfernt sein Schloss und öffnet die Kiste.

(Nicht sicher gegen Man-in-the-middle-Angriff—aktiver Angreifer!)

Beobachtung: Viele Funktionen sind inherent asymmetrisch.

- Zudrücken eines Schlosses ist leicht, Öffnen ist schwer.
- Multiplizieren von Zahlen ist leicht, Faktorisieren ist schwer.
- Exponentieren von Zahlen ist leicht, dlog ist (oft) schwer.

Diffie Hellman Gedankenexperiment



Diffie-Hellman Schlüsselaustausch (1976)

Szenario:

- Alice und Bob verwenden öffentlichen Kanal.
- Beide wollen einen zufälligen Bitstring k austauschen.
- **Angreifer ist passiv**, d.h. kann nur lauschen, nicht manipulieren.

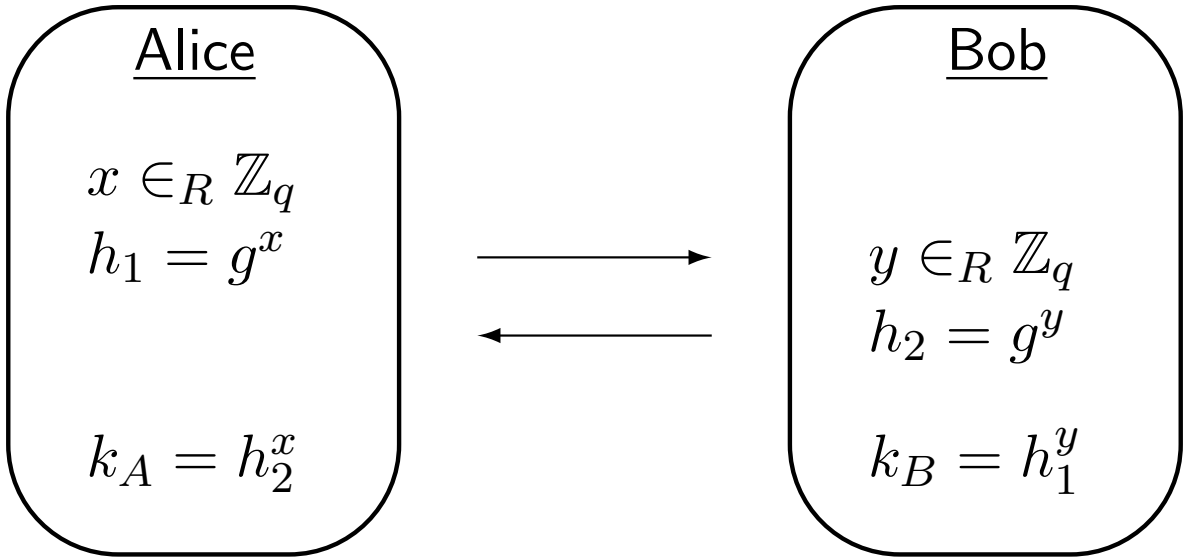
Systemparameter:

- Sicherheitsparameter 1^n
- Schlüsselerzeugung $(G, q, g) \leftarrow \mathcal{G}(1^n)$
  - $\mathcal{G}$ ist probabilistischer polynomial-Zeit (in n) Algorithmus
  - G ist multiplikative Gruppe mit Ordnung q und Generator g.

2-Runden Diffie-Hellman Schlüsselaustausch

Protokoll 2-Runden Diffie-Hellman Schlüsselaustausch

1. Alice: Wähle $x \in_R \mathbb{Z}_q$. Sende $h_1 = g^x$ an Bob.
2. Bob: Wähle $y \in_R \mathbb{Z}_q$. Sende $h_2 = g^y$ an Alice.
3. Alice: Berechne $k_A = h_2^x$.
4. Bob: Berechne $k_B = h_1^y$.



Korrektheit und Schlüsselerzeugung

Korrektheit: $k_A = k_B$

- Alice berechnet Schlüssel $k_A = h_2^x = (g^y)^x = g^{xy}$.
- Bob berechnet Schlüssel $k_B = h_1^y = (g^x)^y = g^{xy}$.

Schlüsselerzeugung:

- Gemeinsamer Schlüssel $k_A \in G$ ist Gruppenelement, kein Zufallsstring $k \in \{0,1\}^m$.
- Konstruktion von Zufallsstring mittels sog. *Zufallsextraktoren*.
- Sei k_A ein zufälliges Gruppenelement aus G.
- Zufallsextraktor liefert bei Eingabe k_A einen Schlüssel $k \in \{0,1\}^m$, ununterscheidbar von einem Zufallsstring derselben Länge.

Übung: Schlüssel k + sichere symmetrische Verschlüsselung liefert zusammen ein beweisbar sicheres Verfahren.

Spiel zur Unterscheidung des Schlüssels

Spiel Schlüsselaustausch $KE_{\mathcal{A},\Pi}(n)$

Sei Π ein Schlüsselaustausch-Protokoll für Gruppenelemente aus G. Sei $\mathcal{A}$ ein Angreifer für Π.

1. $(k, \text{trans}) \leftarrow \Pi(n)$, wobei k der gemeinsame Schlüssel und trans der Protokollablauf ist.
2. Wähle $b \in_R \{0,1\}$. Falls $b = \begin{cases} 1 & k' = k \\ 0 & k' \in_R G \end{cases}$.
3. $b' \leftarrow \mathcal{A}(\text{trans}, k')$. Ausgabe $\begin{cases} 1 & \text{falls } b' = b \\ 0 & \text{sonst} \end{cases}$.

- $\mathcal{A}$ gewinnt, falls $KE_{\mathcal{A},\Pi}(n) = 1$.
- D.h. $\mathcal{A}$ gewinnt, falls er erkennt, welches der korrekte Schlüssel k des Protokolls Π und welches der zufällige Schlüssel $k' \in_R G$ ist.
- $\mathcal{A}$ kann trivialerweise mit Ws $\frac{1}{2}$ gewinnen. (Wie?)

Spiel zur Unterscheidung des Schlüssels