Grundlagen der Verschlüsselung und Authentifizierung (2)

Benjamin Klink

Friedrich-Alexander Universität Erlangen-Nürnberg

Benjamin.Klink@informatik.stud.uni-erlangen.de

Übersicht

- Asymmetrische Verschlüsselung
  - Einwegfunktionen
  - RSA
- Diffie-Hellman
- Hashfunktionen
  - MD5
- Zufallszahlen

Asymmetrische Verschlüsselung

- Was ist anders? Begriffserklärung
  - Statt nur einem Schlüssel existiert folgendes Schlüsselpaar:
  - Ein öffentlicher Schlüssel e zum Verschlüsseln
    - für jeden beliebigen Sender A sichtbar
  - Ein zugehöriger privater Schlüssel d zum Entschlüsseln
    - nur dem Empfänger B bekannt
- Welche Vorteile? Welche Anforderungen?
  - Kein neuer Schlüssel für einen neuen Sender A notwendig
  - Trotzdem sollen mehrere Sender A nicht gegenseitig die verschlüsselten Daten lesen können

Baustein Einwegfunktion

- Wichtiger Bestandteil: Einwegfunktion mit „Falltür“
  - Einwegfunktion bedeutet, dass etwas einfach ist in die eine Richtung zu berechnen z.B. $y = f(x)$, aber die Umkehrung, also $x = f^{-1}(y)$, sehr schwer bzw. unmöglich
  - mit „Falltür“: Unter Kenntnis einer geheimen Zusatzinformation z soll es möglichst leicht sein $x = f^{-1}(y, z)$ zu berechnen
  - Kandidaten:
    - $y = x^e \bmod n$ mit $n = p \cdot q$ — Faktorisierung von n (→ RSA)
    - $y = g^x \bmod p$ — diskreter Logarithmus (→ D.-H.)
- Beispiel Telefonbuch

RSA

- Wer hat's erfunden?
  - R. Rivest, A. Shamir und L. Adleman (1978)
- Wie funktioniert's?
  - Schlüsselpaar-Erzeugung
    - Zwei große Primzahlen p und q (mit $p \cdot q > 10^{200}$), $n = p \cdot q$
    - Eine beliebige Zahl $e > 1$ mit $ggt(e, (p-1) \cdot (q-1)) = 1$
    - Berechne d, sodass $d \cdot e \bmod (p-1) \cdot (q-1) = 1$
    - privater Schlüssel: {d, n}
    - öffentlicher Schlüssel: {e, n}
  - Verschlüsseln: $C = M^e \bmod n$
  - Entschlüsseln: $M' = C^d \bmod n$

RSA – ein Beispiel

- Erzeugung des Schlüsselpaares:

B

Wahl von p = 11, q = 13

=> n = 143, (p-1)·(q-1) = 120

Wahl von e = 23

d·23 mod 120 = 1 => d = 47

veröffentlichen von {23, 143}

- Verschlüsselung des Textes "KvBK" (= 75 118 66 75):

A

holt sich {23, 143}

"K": 75^{23} mod 143 = 69

"v": ->105 "B": ->66 "K": ->69

versandt an B

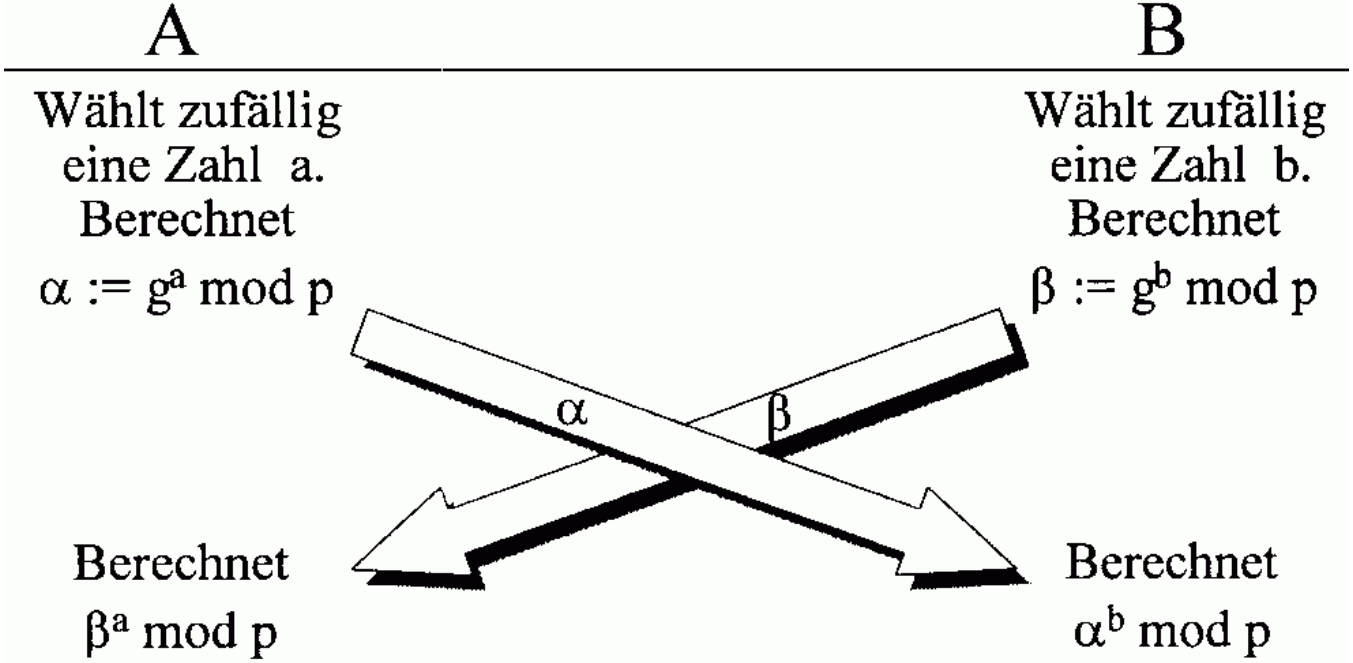
- Entschlüsselung:

69^{47} mod 143 = 75

105 ->118 66 ->66 69 ->75

Diffie-Hellman (1)

- Schlüsselvereinbarung über unsicheren Kanal?
  - Ein Angreifer könnte das gesamte Gespräch belauschen…
- Lösung:
  - Vereinbarung einer Basis g und einer Primzahl p (öffentlich)



aus: [Beu95]

Diffie-Hellman (2)

- Der somit berechnete Schlüssel k ist wegen
  $k = \beta^a \bmod p = (g^b)^a \bmod p = (g^a)^b \bmod p = \alpha^b \bmod p$
  für A und B gleich („gemeinsames Geheimnis“)
- Warum ist das Verfahren selbst dann sicher, wenn ein Angreifer den kompletten Datenfluss zwischen A und B abhört?
  - Es werden lediglich g, p, α und β übertragen, nicht aber die geheimen Zahlen a oder b
  - $\alpha = g^a \bmod p$ ist einfach zu berechen; rückwärts also a = diskreter Logarithmus von α zur Basis g aber sehr schwierig
  - Ohne a bzw. b kann der Schlüssel k nicht berechnet werden

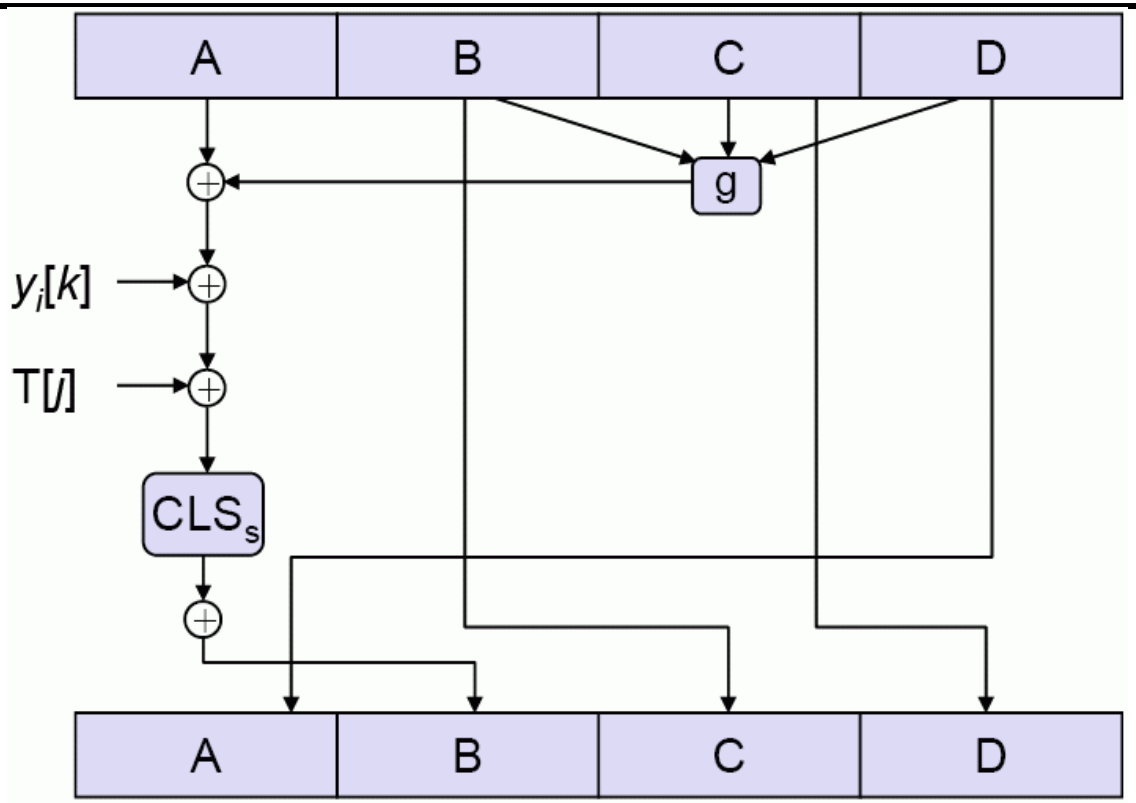
Kryptographische Hashfunktionen

- Ziel: Schutz von Daten vor absichtlicher Änderung
  - Prüfsummen helfen nur bei (unabsichtlichen) Fehlern…
- Anforderungen:
  - Komprimieren einer beliebigen Nachricht auf eine feste Länge
  - „Einwegfunktion“
  - Praktisch kollisionsfrei
    - Es soll praktisch unmöglich sein zwei verschiedene Nachrichten mit gleicher Hashsumme zu finden
  - Leicht zu berechnen
- Beispiele:
  - Blockverschlüsselung im CBC-Mode, MD5, SHA-1, …

MD5 (1)

- Funktionsweise:
  - Die Nachricht wird durch Auffüllen und einer Längeninformation auf ein Vielfaches von 512 Bit gebracht
  - Es gibt eine gemeinsame Tabelle T mit 64 Konstanten
  - Die Ergebnisregister werden vorinitialisiert:
    - `A = 0x 67 45 23 01` `-B = 0x EF CD AB 89`
    - `C = 0x 98 BA DC FE` `-D = 0x 10 32 54 67`
  - Jeder der 512 Bit großen Blöcke der Nachricht wird zu Teilen von 32 Bit in 4 Runden à 16 Schritten verarbeitet
  - Jede der 4 Runden verwendet eine spezifische Funktion g

MD5 (2)



- g ist eine der vier logischen Funktionen
- y_i[k] ist der k-te 32-Bit-Teil eines Blockes
- T[j] ist der j-te Eintrag aus der Tabelle T der 64 Konstanten
- CLS_s ist ein Schieberegister, das s Bits nach links rotiert

aus: [NetSec]

Zufallszahlen

- Erzeugung durch Hardware:
  - Zeitliche Messung radioaktiven Zerfalls
  - „Signalrauschen“ (→ z.B. Mikrophon oder Kamera)
  - usw. …
- Erzeugung durch Software
  - Systemzeit
  - Zeitabstände zwischen Tastendruck oder Mausbewegung
  - Betriebssystem-Parameter wie z.B. Computerauslastung
- Pseudo-Zufallszahlen-Generator
  - Erzeugt aus einem Startwert (Seed) eine Folge „zufälliger“ Zahlen
  - Die Schwierigkeit besteht in der Nicht-Vorhersagbarkeit der Folge

Zusammenfassung

- Asymmetrische Verschlüsselung
  - Privater & öffentlicher Schlüssel + Einwegfunktion mit „Falltür“
  - RSA
    - Geheime Erzeugung von e, d und n; öffentlicher Schlüssel = {e, n}
    - Ver- und Entschlüsseln: $C = M^e \bmod n$ $M' = C^d \bmod n$
- Diffie-Hellman
  - Schlüsselvereinbarung per $k = \beta^a \bmod p = \alpha^b \bmod p$
- Hashfunktionen
  - Manipulationsdetektion
  - Praktisch kollisionsfrei
    - keine zwei Nachrichten mit gleicher Hashsumme auffindbar
- Zufallszahlen
  - Hardware, Software, Pseudo-Zufallszahlen

Vielen Dank für eure Aufmerksamkeit!

Fragen? Diskussionsbedarf?

Quellen

- [NetSec]: Folien zur Vorlesung Systemsicherheit/ Netzwerksicherheit, Kapitel 4-6
  www7.informatik.uni-erlangen.de/~dressler/lectures/netzwerksicherheit-ws0708/
- [Beu95]: A. Beutelspacher: "Moderne Verfahren der Kryptographie. Von RSA zu Zero-Knowledge", 1995, Vieweg Verlag, Braunschweig/Wiesbaden