Vorlesung Kryptographie

Aufzeichnungen vom 24.5.

4 Hashfunktionen

Gesucht: Funktion h, so dass $h(m)$ ein String fester Länge (z.B. 160 bit), sowie

- Geg. m, dann soll $h(m)$ leicht zu berechnen sein;
- Aus $h(m)$ auf m schliessen: nicht machbar
- m ändern, ohne $h(m)$ zu ändern: " "
- (schwach Kollisionsfrei)
- $m \neq m'$ finden mit $h(m)=h(m')$ " "
(stark Kollisionsfrei)

Allgemeines Prinzip (oft) (real: SHA-1, SHA-2, MD5)

Merkle-Damgård-Schema:

- Teile m in Blöcke $m_1, m_2, \dots, m_n$ fester Länge (m_0)
Hänge die Länge als m_{n+1} an (in geeigneter Notation, „Padding").
Startwert (fest) s. Benutze wiederholt $f(x_i, m_i) = x_{i+1}$:

$$s = x_0 \to x_1 \to x_2 \to x_3 \to \dots \to x_{n+2}$$

(m_0 auf x_1, m_1 auf x_2, m_2 auf x_3, m_{n+1} auf x_{n+2})

$x_{n+2} = h(m)$ Hashwert.

Spielzeugbsp: $f(x,y) = x + 7y \bmod 100$

$S = Q \hat{=} 16$

A B C ... L M N ... Z
00 01 02 11 12 13 25

$m =$ „ALICE", $m_5 = 5 =$ „F"

00 11 08 02 04 05

$s = 16 \to 16 \to 93 \to 49 \to 63 \to 91 \to \underline{26}$

$h(\text{„ALICE"}) = \underline{26}$.

Eigsch:
- leicht ✓
- $h(m)$ auf m schließen ✓
- Kollisionsfrei: Nein.

Probieren: $\xrightarrow{-7}$

$+7 \downarrow$

$h(\text{alice}) = 26$ $h(\text{akice}) = 19$ $h(\text{ajice}) = 12$ ...
$h(\text{blice}) = 33$ ...
$h(\text{clice}) = 40$ $h(\text{bkice}) = 26$
$h(\text{dlice}) = 47$ $h(\text{cjice}) = 26$
⋮ ⋱

Spielzeugbsp: Wie oben, nur $f(x,y) = 3^x + 7^y \bmod 100$

Total schlecht: $h(m) \in \{56, 96\}$.

Bsp (ernst) MD5 (1991).

- Zu MD5: siehe Video auf der Vorlesungs-Webseite, sowie: Joshua Holden: A good hash function is hard to find, and vice versa" (link auf Vorlesungswebseite)

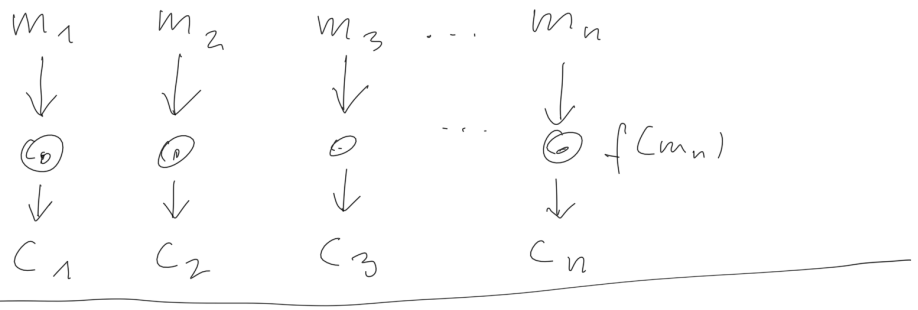
Das $F(B,C,D)$: Kombinationen von $\wedge, \vee, XOR$ bitweise

Pseudocode von MD5: siehe Video

Längliches Verfahren ... Dennoch:

- 1996 theoretische Lücke entdeckt
- 2005 Kollisionen produziert (2 verschiedene ps-files)
- Gilt als unsicher für kryptographische
- Als Prüfsumme für Downloads viel genutzt.

Bemerk.: Gute symmetrische Verschlüsselungsverfahren arbeiten mit ganz ähnlichen Zutaten (AES, DES), nur nicht mit dem „Weiterleiten" bei Merkle-Damgård.



5. Signaturen

RSA-Signatur:

- Alice verschlüsselt m mit Bobs öffentlichem Schlüssel $c = m^{e_B} \bmod p$

- Alice verschlüsselt $h(m)$ mit ihrem geheimen Schlüssel: $s = h(m)^{d_A} \bmod p$. Sendet (c, s)

Bob berechnet $c^{d_B} \equiv m^{e_B d_B} \equiv m \bmod p$;
daraus $h(m)$, und prüft:

$$h(m) \overset{?}{=} s^{e_A} \quad \left(\equiv h(m)^{d_A e_A} \equiv h(m) \bmod p \right)$$

(falls s von Alice erzeugt.)

Wenn "=" (und wenn h kollisionsfrei) dann muss" s; somit m, von Alice kommen.

Lässt sich auf viele andere Public-Key-Verfahren anwenden.

Interessant: El-Gamal liefert Signaturen auch ohne Hashfunktion.

El-Gamal-Signatur: Öffentliche Schlüssel:

- $(p, g^a \bmod p)$ von Alice (a geheim)
- $(p, g^b \bmod p)$ von Bob (b geheim)

BISHER:
Bob wählt r mit $ggT(r, p-1) = 1$ und berechnet $K = g^r \bmod p$.
Bob verschlüsselt m mittels K als c; und schickt (c, K) an Alice.
Alice entschlüsselt c als m

Neu:
- Bob berechnet r^{-1} in $\mathbb{Z}^*_{p-1}$ und $s \equiv (m - bK) r^{-1} \bmod p-1$ und sendet (c, K, s) an Alice.

(c: Geheimtext; K: Teilschlüssel; s: Signatur)

- Alice prüft ob $g^m \equiv (g^b)^K \cdot K^s \bmod p$
  Falls „$\equiv$", dann ist (es)wirklich von Bob.

Bemerk.
- machbar: ✓ (alle Zutaten ✓)
- Korrekt: Übung 19 auf Blatt 6
- sicher: So sicher El-Gamal.

Insbesondere muss r wirklich zufällig sein (und r geheim).
Und jedesmal verschieden!

* * *