

# Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

12.05.2014

## 1 Hashfunktionen

- Erinnerung
- Angriffe auf Hashfunktionen
- Zusammenfassung Hashfunktionen

## 2 Asymmetrische Verschlüsselung

- Idee
- Beispiel: RSA
- Sicherheit von RSA

## 1 Hashfunktionen

- Erinnerung
- Angriffe auf Hashfunktionen
- Zusammenfassung Hashfunktionen

## 2 Asymmetrische Verschlüsselung

- Idee
- Beispiel: RSA
- Sicherheit von RSA

- Hashfunktion:  $H : \{0, 1\}^* \rightarrow \{0, 1\}^k$   
( $k$  Sicherheitsparameter)
- Haupteigenschaft: Kollisionsresistenz  
schwer,  $X \neq X'$  mit  $H(X) = H(X')$  zu finden

## 1 Hashfunktionen

- Erinnerung
- **Angriffe auf Hashfunktionen**
- Zusammenfassung Hashfunktionen

## 2 Asymmetrische Verschlüsselung

- Idee
- Beispiel: RSA
- Sicherheit von RSA

# Birthday Attack

- Idee: betrachte viele  $Y_i := H(X_i)$  für zufällige  $X_i$
- Vereinfachend:  $Y_i \in \{0, 1\}^k$  unabhängig gleichverteilt

## Theorem (Birthday Bound, ohne Beweis)

Sei  $n \leq 2^{k/2}$  und  $Y_1, \dots, Y_n \in \{0, 1\}^k$  unabhängig gleichverteilt.  
Dann gibt es  $i \neq j$  mit  $Y_i = Y_j$  mit Wkt.  $p > (1/11) \cdot (n^2/2^k)$ .

- Konsequenz: für  $n = 2^{k/2}$  zufällige (verschiedene)  $X_i$   
Kollisionen unter den  $Y_i$  mit Wahrscheinlichkeit  $p > 1/11$
- Vorgehen (Aufwand  $\Theta(k \cdot 2^{k/2})$  Schritte,  $\Theta(k \cdot 2^{k/2})$  Bits):
  - 1 Schreibe  $(X_i, Y_i)$  in Liste ( $X_i \in \{0, 1\}^{2k}$  glv.,  $Y_i = H(X_i)$ )
  - 2 Sortiere Liste nach  $Y_i$
  - 3 Untersuche Liste auf  $Y_i$ -Kollisionen

- Auch Meet-in-the-Middle-Angriff manchmal möglich
  - Setzt spezielle Struktur von Hashfunktionen voraus:  
Hashwert sollte sich „rückwärts“ berechnen lassen
  - Aufwand: asymptotisch wie Birthday Attack
- **Lehre:** Hash-Ausgabe  $\geq k$  Bits für  $k/2$  Bits „Sicherheit“
  - Ausgabelängen: MD5 128, SHA-1 160, SHA-3 variabel

## 1 Hashfunktionen

- Erinnerung
- Angriffe auf Hashfunktionen
- Zusammenfassung Hashfunktionen

## 2 Asymmetrische Verschlüsselung

- Idee
- Beispiel: RSA
- Sicherheit von RSA



# Zusammenfassung Hashfunktionen

- Hashwert „Fingerabdruck“ der Eingabe
- Kollisionsresistenz  $\Rightarrow$  Einwegeigenschaft
- Populäre Strategie: Merkle-Damgård
- Ausgabelänge  $\geq k$  Bits für  $k/2$  Bits Sicherheit
- Populäre Verfahren: MD5, SHA-1, SHA-3 (Keccak)
- **Aber:** MD5, SHA-1 **gebrochen**

- Hashfunktionen, deren Sicherheit auf gut untersuchten Problemen beruht (z.B. Berechnungsprobleme in Gittern)
- Für Passwortabfragen: „universelle“ Einwegfunktionen
- Untersuchung von generischen Strategien (wie Merkle-Damgård) in idealisierten Modellen
- Kryptoanalyse: Weitere „Anwendungen“ und Erweiterungen von bekannten Angriffen auf z.B. MD5, SHA-1, Beispiele:
  - gefälschte Webseiten-Zertifikate (Flame)
  - gefälschte Signaturen für (Postscript-)Dokumente
  - Grundidee: gegeben (Signatur von)  $H(M)$ , finde „sinnvolles“  $M'$  mit  $H(M') = H(M)$  (für das die Signatur dann gilt)

## 1 Hashfunktionen

- Erinnerung
- Angriffe auf Hashfunktionen
- Zusammenfassung Hashfunktionen

## 2 Asymmetrische Verschlüsselung

- Idee
- Beispiel: RSA
- Sicherheit von RSA

- 1 Hashfunktionen
  - Erinnerung
  - Angriffe auf Hashfunktionen
  - Zusammenfassung Hashfunktionen
  
- 2 Asymmetrische Verschlüsselung
  - **Idee**
  - Beispiel: RSA
  - Sicherheit von RSA

# Motivation

- Symmetrische Verschlüsselung: gemeinsames Geheimnis  $K$

$$\text{Alice}_K \quad \xleftarrow{C := \text{Enc}(K, M)} \quad \text{Bob}_K$$

- Bei  $n$  Benutzern  $\binom{n}{2} = n \cdot (n - 1) / 2$  Schlüsselpaare
- Zudem kann Schlüsselverteilung problematisch sein
- Auftritt Merkle (1974) und Diffie und Hellman (1976):



Quelle: Wikipedia

- Asymmetrische (oder Public-Key-)Verschlüsselung:

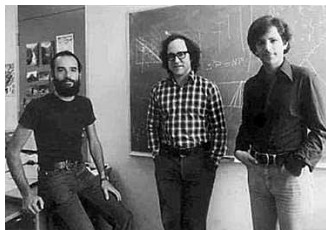
$$\text{Alice}_{sk} \quad \xleftarrow{C := \text{Enc}(pk, M)} \quad \text{Bob}_{pk}$$

- Verschlüsselung mit  $pk$ :  $C \leftarrow \text{Enc}(pk, M)$
- Entschlüsselung mit  $sk$ :  $M \leftarrow \text{Dec}(sk, C)$
- $pk$  und  $sk$  gemeinsam generiert:  $(pk, sk) \leftarrow \text{Gen}(1^k)$
- $pk$  darf veröffentlicht werden,  $sk$  muss geheim bleiben

# Erste Eigenschaften

$$\text{Alice}_{sk} \longleftarrow \text{Enc}(pk, M) \longrightarrow \text{Bob}_{pk}$$

- Keine (geheime) Schlüsselverteilung,  $pk$  öffentlich
- Bei  $n$  Benutzern  $n$  öffentliche (und  $n$  geheime) Schlüssel
- **Problem:** Diffie und Hellman hatten kein Verfahren
- Auftritt Rivest, Shamir und Adleman (1977): (Quelle: [ams.org](http://ams.org))



- 1 Hashfunktionen
  - Erinnerung
  - Angriffe auf Hashfunktionen
  - Zusammenfassung Hashfunktionen
  
- 2 Asymmetrische Verschlüsselung
  - Idee
  - Beispiel: RSA
  - Sicherheit von RSA



# Überblick RSA

$$pk = (N, e) \quad sk = (N, d)$$

- $N = PQ$  für (hinreichend große) Primzahlen  $P \neq Q$
- Rechnung in  $\mathbb{Z}_N := \mathbb{Z}/N\mathbb{Z}$  (d.h. Rechnung modulo  $N$ )
- $e$  und  $d$  sind „zueinander inverse“ Exponenten
  - Genauer:  $e \cdot d = 1 \pmod{\varphi(N)}$  (mit  $\varphi(N) = (P - 1)(Q - 1)$ )
- Nachrichtenraum ist  $\mathcal{M} := \mathbb{Z}_N$

$$\text{Enc}(pk, M) = M^e \pmod{N} \quad \text{Dec}(sk, C) = C^d \pmod{N}$$

# RSA-Schlüsselgenerierung

- Gesucht:

$$pk = (N, e) \quad sk = (N, d)$$

- Gen wählt  $P$  und  $Q$  zufällig von vorgegebener Bitlänge  $k$ :
  - Gängig: wähle gleichverteiltes *ungerades*  $P \in \{2^k, \dots, 2^{k+1}\}$
  - ... bis  $P$  prim (Primalitätstest z.B. Miller-Rabin), analog für  $Q$
- Um  $e$  und  $d$  zu erhalten:
  - Wähle zufällig gleichverteilt  $e \in \{3, \dots, \varphi(N) - 1\}$
  - ... bis  $\gcd(e, \varphi(N)) = 1$
  - Berechne  $d = e^{-1} \bmod \varphi(N)$  mit erweitertem Euklid:
    - $\text{EE}(e, \varphi(N)) = (\alpha, \beta)$  mit  $\alpha \cdot e + \beta \cdot \varphi(N) = \gcd(e, \varphi(N)) = 1$
    - Damit ist  $\alpha \cdot e = 1 \bmod \varphi(N)$ , setze also  $d := \alpha \bmod \varphi(N)$

# Korrektheit von RSA

- Für Korrektheit sollte gelten:  $(M^e)^d = M^{ed} = M \pmod N$
- Zutaten (ohne Beweis):

## Theorem (Kleiner Satz von Fermat, ohne Beweis)

Für primes  $P$  und  $M \in \{1, \dots, P-1\}$  gilt  $M^{P-1} = 1 \pmod P$ .

- Konsequenz:  $\forall M \in \mathbb{Z}_P, \alpha \in \mathbb{Z} : (M^{P-1})^\alpha \cdot M = M \pmod P$

## Theorem (Chinesischer Restsatz, ohne Beweis)

Sei  $N = PQ$  mit  $P, Q$  teilerfremd. Dann ist die Abbildung  $\mu : \mathbb{Z}_N \rightarrow \mathbb{Z}_P \times \mathbb{Z}_Q$  mit  $\mu(M) = (M \pmod P, M \pmod Q)$  bijektiv.

- Also:  $(X = Y \pmod P) \wedge (X = Y \pmod Q) \implies X = Y \pmod N$

# Korrektheit von RSA

## Theorem (Korrektheit von RSA)

Sei  $N, e, d$  wie oben. Dann ist  $M^{ed} = M \pmod N$  für alle  $M \in \mathbb{Z}_N$ .

## Beweis.

- Es gilt  $ed = 1 \pmod{(P-1)(Q-1)}$  nach Definition, deshalb:

$$\begin{aligned}(P-1)(Q-1) \mid ed - 1 &\Rightarrow P-1 \mid ed - 1 \\ &\Rightarrow ed = \alpha(P-1) + 1 \quad \text{für } \alpha \in \mathbb{Z} \\ &\Rightarrow M^{ed} = (M^{P-1})^\alpha \cdot M \stackrel{\text{Fermat}}{=} M \pmod P\end{aligned}$$

- Analog:  $M^{ed} = M \pmod Q$
- Chinesischer Restsatz  $\Rightarrow M^{ed} = M \pmod N$



- 1 Hashfunktionen
  - Erinnerung
  - Angriffe auf Hashfunktionen
  - Zusammenfassung Hashfunktionen
  
- 2 Asymmetrische Verschlüsselung
  - Idee
  - Beispiel: RSA
  - Sicherheit von RSA

- **Diskussion:** Was wollen wir eigentlich?  
Wieder Einschränkung auf passive Sicherheit/Angriffe

## Definition (Semantische Sicherheit für Public-Key-Verfahren)

Ein Public-Key-Verschlüsselungsschema ist semantisch sicher, wenn es für jede  $M$ -Verteilung von Nachrichten gleicher Länge, jede Funktion  $f$  und jeden PPT-Algorithmus  $\mathcal{A}$  einen PPT-Algorithmus  $\mathcal{B}$  gibt, so dass

$$\Pr \left[ \mathcal{A}(1^k, pk, \text{Enc}(pk, M)) = f(M) \right] - \Pr \left[ \mathcal{B}(1^k) = f(M) \right]$$

vernachlässigbar (als Funktion im Sicherheitsparameter) ist.

- Unhandlich, aber (wie symmetrisch) äquivalent zu IND-CPA (... wobei IND-CPA für Public-Key-Verschlüsselung analog)

- RSA nicht semantisch sicher
  - $f(M) = M^e \bmod N$  kann mit Chiffprat berechnet werden
  - Aber ohne Chiffprat keine Information über  $M$
  - „Angriff“ nutzt aus, dass RSA deterministisch
- Intuitiv überzeugender: beispielsweise

$\text{Enc}(pk, \text{annehmen})$  und  $\text{Enc}(pk, \text{ablehnen})$

bei RSA effizient unterscheidbar (keine IND-CPA-Sicherheit)



# Weitere Angriffe auf RSA

- Was, wenn  $e = 3$  (aus Effizienzgründen) für alle Benutzer?
  - Problem, wenn  $M$  an  $\geq 3$  Benutzer gesendet wird
  - Angreifer kennt Chiffre  $M^3 \bmod N_i$  für  $1 \leq i \leq 3$
  - Chinesischer Restsatz  $\rightsquigarrow M^3 \bmod N_1 N_2 N_3$
  - Wegen  $0 \leq M \leq N_1, N_2, N_3$  ist  $M^3 \bmod N_1 N_2 N_3 = M^3 \in \mathbb{Z}$
  - „Wurzelziehen“ über  $\mathbb{Z}$  liefert  $M$
- Könnte mit probabilistischem  $\text{Enc}$  behoben werden
- Weitere schlechte Idee: gleiches  $N$  für mehrere Benutzer

# Homomorphie von RSA

- Es gilt (Rechnung modulo  $N$ ):

$$\begin{aligned}\text{Enc}(pk, M) \cdot \text{Enc}(pk, M') &= M^e \cdot M'^e \\ &= (M \cdot M')^e = \text{Enc}(pk, M \cdot M')\end{aligned}$$

- Problem z.B. bei Auktionen:
  - Auktionator  $A$  veröffentlicht  $pk$ , behält  $sk$
  - Bieter  $B_1, B_2$  senden verschlüsselte Gebote  $C_i := \text{Enc}(pk, M_i)$  an Auktionator
  - $A$  entschlüsselt  $C_i$ , Bieter mit höchstem Gebot erhält Ware
  - Angriff:  $B_2$  wartet  $B_1$ 's Gebot ab, setzt

$$C_2 := C_1 \cdot \text{Enc}(pk, 2) \bmod N$$