Symm. Verschl.

Stromchiffre
- → Simuliere OTP
  - ↳ Generiere langen Schlüssel aus kurzem Schl.
  - ⇒ $G(K) = K'$
    - ↳ $C := M \oplus K'$

One-Time-Pad
- → Vigenere, aber Schlüssel K so lang wie Nachricht M

Korrektheit
$D(K, E(K, M)) \stackrel{!}{=} M$

Blockchiffre

ECB Mode

1. M in L-Bit Blöcke teilen
2. $C := C_1 \cdots C_n \quad C_i := E(K, M_i)$

CBC Mode

1. M in L-Bit Blöcke teilen
2. $C_0 = IV$
3. $C_i := E(K, M_i \oplus C_{i-1})$

DES Feistelstruktur

$M = L_0 R_0$

↳ in jedem Schritt (K_i Rundenschlüssel)

$L_{i+1} = R_i$
$R_{i+1} = L_i \oplus F(R_i, K_i)$

2DES:

Zwei Schlüssel:
K_1, K_2
↳ erst DES mit K_1 en-, dann DES mit K_2

3DES genauso

Semantische Sicherheit:

Ein Symm. Verschl. verf. ist semantisch sicher, wenn

$$Pr[A^{Enc(K,\cdot)}(Enc(K, M)) = f(M)] - Pr[B(\varepsilon) = f(m)]$$

Hashfunktionen "Fingerabdrücke" $H: \{0,1\}^* \to \{0,1\}^k$

↳ Kollisionsresistent: $X \neq X'$ mit $H(X) = H(X')$ schwer zu finden

Einwegeigenschaft: Gegeben $Y = H(X)$, ist es schwer X' mit $H(X') = Y$ zu finden

Target Collision Resistance: Für ein gegebenes X soll es schwer sein, ein X' mit $X \neq X'$ und $H(X) = H(X')$ zu finden

Über k parametrisierte Fkt ist kollisionsresistent

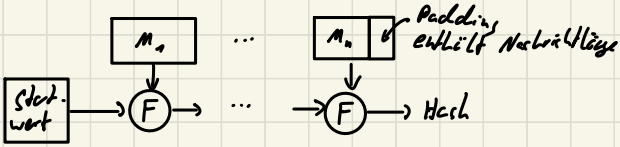
$\Leftrightarrow$

jeder poly (PPT) Algorithmus nur mit vernachlässigbarer W'keit eine Kollision findet

Merkle - Damgård - Konstruktion

→ Hashfkt. H mit beliebiger Länge

Sei $F: \{0,1\}^{2k} \to \{0,1\}^k$ Kompression



Kleiner Satz von Fermat

Sei P Primzahl, $M \in \{1, \dots, P-1\}$

$\Rightarrow M^{P-1} = 1 \bmod P$

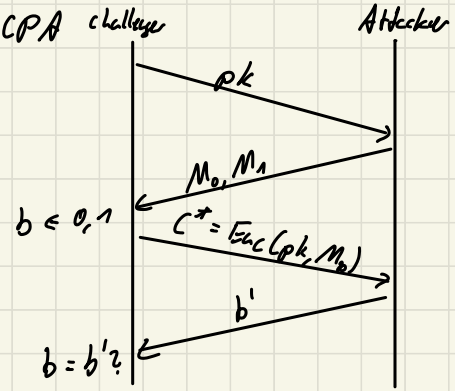
Chinesischer Restsatz

Sei $N = P \cdot Q$, P, Q Teilerfremd

$\Rightarrow$ Wenn $X = Y \bmod P$
$\wedge\ X = Y \bmod Q$
$\Rightarrow X = Y \bmod N$

$(N = P \cdot Q)$
$\Rightarrow \varphi(N) = (P-1)(Q-1)$

IND-CPA



RSA-Gen: Gesucht: $pk = (N, e)$, $sk = (N, d)$

1. Wähle zufällig P und Q der Länge lc, bis beide Primzahlen sind
2. Wähle zufällig $e \in \{3, \dots, \varphi(N)-1\}$, bis $gcd(e, \varphi(N)) = 1$ (ggT, teilerfremd)
3. Berechne $d = e^{-1} \bmod \varphi(N)$
   Mittels erweitertem Euklid gilt:
   $(\alpha \cdot e = 1 \bmod \varphi(N))$
   $\Rightarrow d := \alpha \bmod \varphi(N)$
   um $\alpha e + \beta \cdot \varphi(N) = 1$ zu erfüllen

Euklid in RSA:

Input e und $\varphi(N)$
mit $gcd(e, \varphi(N)) = 1$

1. Start $\varphi(N) = \lambda e + c$
   ↳ $e = \lambda c + c'$
   $c = \lambda c' + c''$
   ...
   $c' = \lambda c'' + c'''$
   $c'' = \lambda c''' + 0$
   λ in jeder Zeile anders
   Der letzte „Rest" $\neq 0 = c'''$ ist gcd
2. Gehe Rückwärts durch gcd-Rechnung
   ↳ Starte mit vorletzte Zeile
   $gcd = c' - \lambda c''$
   ↳ substituiere c'' durch Rest der Zeile darüber
   ↳ umformen für nächsten
   ...
   $gcd = -\beta \cdot \varphi(N) + \alpha \cdot e \Rightarrow d := \alpha \bmod \varphi(N)$

Beispielrechnung für $d: e = 29$, $\varphi(N) = 192$

1. $192 = 29 \cdot 6 + 18$
   $29 = 18 \cdot 1 + 11$
   $18 = 11 \cdot 1 + 7$
   $11 = 7 \cdot 1 + 4$
   $7 = 4 \cdot 1 + 3$
   $4 = 3 \cdot 1 + 1$
   $3 = 1 \cdot 3 + 0$

Für Korrektheit von RSA sollte gelten:

$(M^e)^d = M \bmod N$

2. $1 = 4 - 1 \cdot 3$

a) $1 = 4 - 1 \cdot (7 - 1 \cdot 4) = 2 \cdot 4 - 1 \cdot 7$

b) $1 = 2 \cdot (11 - 1 \cdot 7) - 1 \cdot 7 = 2 \cdot 11 - 3 \cdot 7$

c) $1 = 2 \cdot 11 - 3 \cdot (18 - 1 \cdot 11) = 5 \cdot 11 - 3 \cdot 18$

d) $1 = 5 \cdot (29 - 1 \cdot 18) - 3 \cdot 18 = 5 \cdot 29 - 8 \cdot 18$

e) $1 = 5 \cdot 29 - 8 \cdot (192 - 6 \cdot 29) =$

$= 53 \cdot 29 - 8 \cdot 192$

↳ $d = 53 \bmod 192$

El Gamal → zyklische Gruppe $G = \langle g \rangle$

$pk = (G, g, g^x \bmod N)$,
$sk = (G, g, x)$ mit x zufällig

$Enc(pk, M) = (g^y \bmod N, g^{xy} \cdot M \bmod N)$

mit y zufällig

$Dec(sk, (Y, Z)) = \frac{Z}{Y^x} = \frac{g^{xy} \cdot M}{(g^y)^x} = M$

Bei einer Gruppe gilt z.B. $\mathbb{Z}_{55}$

$3^{60} \bmod 55 = 3^{60 \bmod \varphi(55)} \bmod 55$

durch Satz von Euler
$\Rightarrow \varphi(N) = N - 1$ ist Ordnung, wenn N Primzahl

---

Annahmen für asymm. Verfahren

RSA: ziehen der e-ten Wurzel mod N ist schwer

El Gamal: Decisional Diffie-Hellman $(g^x, g^y, g^z) \Rightarrow$ ist $z = x \cdot y$? schwer

---

El Gamal & RSA sind homomorph:

RSA:

$Enc(pk, M_1) \cdot Enc(pk, M_2) = M_1^e \cdot M_2^e \bmod N$

$= (M_1 \cdot M_2)^e \bmod N$

$= Enc(pk, M_1 \cdot M_2)$

El Gamal

$Enc(pk, M) \cdot Enc(pk, M') = (g^y, g^{xy} \cdot M) \cdot (g^{y'}, g^{xy'} \cdot M')$

$= (g^{y+y'}, g^{x(y+y')} \cdot M \cdot M')$

$= Enc(pk, M \cdot M')$

Zero-Knowledge Eigenschaft

Ein Public-Key-Identifikationsprotokoll ist Zero-Knowledge

für jeden poly Angreifer A ein poly-Algorithmus S existiert, sodass die Verteilungen

$$(pk, \langle P(sk), A(1^k, pk)\rangle) \text{ und } (pk, S(1^k, pk))$$

ununterscheidbar sind

Beispiel ZK-Protokoll (zw. P und V)

1. P wählt Bijektion $\pi: \{1,2,3\} \to \{1,2,3\}$
2. P committed sich mit $com_i = Com(\pi(\varphi(i)); R_i)$ auf dreigefärbten Graphen mit für jedes i „getauschten" Farben
3. P sendet alle com_i an V
4. V wählt zufällige Kante (i,j) und sendet diese an P
5. P öffnet com_i, com_j
   ↳ sendet also $(\pi(\varphi(i)), R_i), (\pi(\varphi(j)), R_j)$ an V
6. V akzeptiert Openings
   $\Leftrightarrow$
   $\pi(\varphi(i)) \neq \pi(\varphi(j))$

Bernstein: Commitment

Binding

Für jeden poly-Angreifer A ist
$Pr[Com(M;R) = Com(M';R') \wedge M \neq M']$

Hiding

Für beliebige $M, M' \in \{0,1\}^*$ sind die Verteilungen
$Com(M;R)$ und $Com(M';R)$
ununterscheidbar

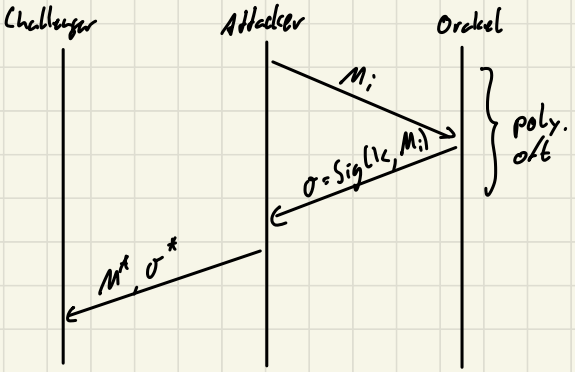
Syntax

$Com(M;R)$ mit Eingabe M, Zufall R
↳ Commitment auf R

Intuition:

Com legt auf M fest
↳ kann nur für ein einziges M aufgedeckt werden

Chosen Message Attacks:



Existential Forgery (EUF)

Mindestens eine Nachricht M^*, die noch nie signiert wurde, funktioniert

Selective Forgery (SUF)

Nachricht M^* wurde vor dem Angriff gewählt

Universal Forgery (UUF)

Der Angreifer kann für jedes beliebige M^* eine valide Signatur σ finden

Total Break

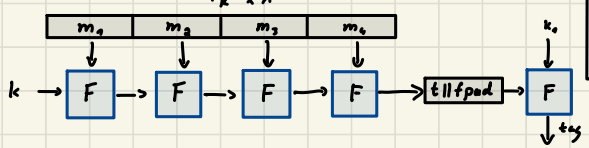
Der Angreifer kann privaten Schlüssel erhalten und somit beliebig signieren

---

MAC Konstruktionen

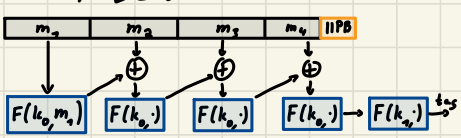
Sei $PRF: K \times X \to X$ pseudo-randomisierte Fkt., z.B. $PRF(k, x) = H(k \| x)$

NMAC $F = PRF|_{K^2 \times X^{SL}} : K^2 \times X^{SL} \to X$

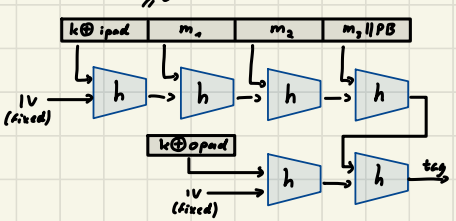


$H(k \| m)$ unsicher → Nachricht mit keys ergeben

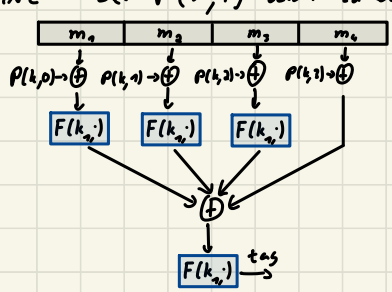
CBC-MAC / ECBC



HMAC - „double NMAC"

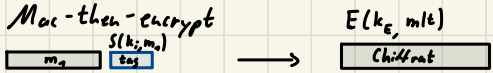


PMAC Sei $P(k, i)$ leicht zu berechnen

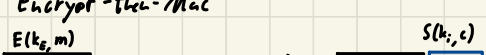


MAC mit Verschlüsselung

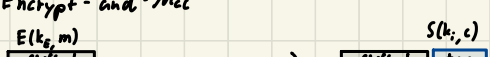
SSL; Mac-then-encrypt



IPSec; Encrypt-then-Mac



SSH; Encrypt-and-Mac



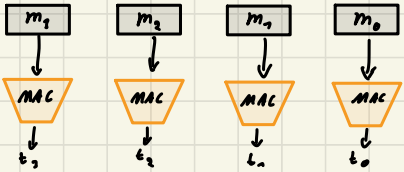
Progressive MACs

Es soll erfüllen:

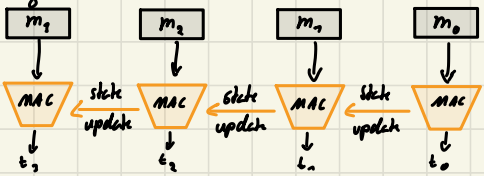
1. Direkt Authentication of each message possible
2. Save Communication Costs
3. Security similar to traditional MACs
4. Resynchronization possible

	trad. MAC	truncated MAC	Sponge	sponge Wrap	XOR MAC	Whips
1.	✓	✓	✗	✓	✓	✓
2.	✗	✓	✓	✓	✓	✓
3.	✓	✗	✓	✓	✗	✓
4.	✓	✓	✗	✗	✓	✓

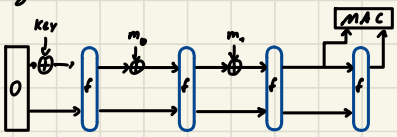
trunc. MAC:



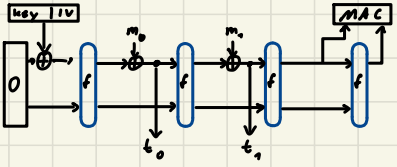
progressive MAC



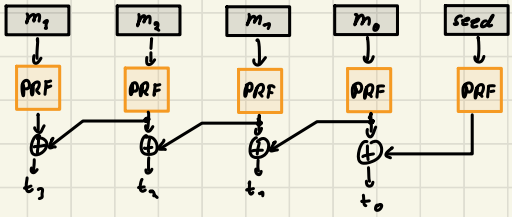
Sponge



Sponge Wrap



XOR-MAC



Whips