Message Authentication Codes

Fabian Eltz / Matthias Schubert

Seminar Kryptographie und Datensicherheit

WS 06/07

Gliederung

1. Message Authentication Code (MAC)
2. CBC-MAC
3. Nested MAC
4. HMAC
5. Hash Families
6. Unconditionally Secure MACs
7. Strongly Universal Hash Families
8. Zusammenfassung
9. Quellen

Grundlagen

„keyed hash function“

Ziel: Authentizität der Daten

Verwendung bei Chipkarten, Online-Banking,...

Vorgehen: Nachricht x wird mit Funktion h und Schlüssel K verschlüsselt

$\mathbf{h_K(x) = y}$ (y = „authentication tag“)

Einsatz

1. Alice berechnet MAC des Dokuments mit Hilfe des geheimen Schlüssels, den sie vorher mit Bob vereinbart hat.
2. Alice sendet Dokument mit MAC an Bob.
3. Bob berechnet ebenfalls MAC mit Hilfe des geheimen Schlüssels. Sind beide MAC-Werte gleich, ist das Dokument authentisch.

Angriffsidee

Angreifer muss geheimen Schlüssel nicht kennen!

Voraussetzung: Orakel kann $y = h_K(x)$ erzeugen

Angreifer kann bis zu q Anfragen $(x_1,x_2,...,x_q)$ an Orakel stellen

Orakel berechnet Paare $(x_1y_1),(x_2y_2),...,(x_qy_q)$

Angriffsidee

Ziel: neues Paar (x,y) erzeugen, wobei x nicht in $\{x_1,...,x_q\}$ liegt

Wenn (x,y) gültiges Paar, dann Fälschung

(x,y) ist mit Wahrscheinlichkeit ε gültig → (ε,q)-Fälschung

Grundlagen

CBC = Cipher Block Chaining

Verwendung von Blockchiffren wie DES/AES

weit verbreitet

standardisiert z.B. in FIPS-113 (ICST 1985) oder ISO 9797

Grundlagen

Nachricht wird in n Blöcke der Länge b unterteilt
$\rightarrow x = (x_1,...,x_n)$, $x_i = \{0,1\}^b$

x_i wird $\oplus$-verknüpft mit Ergebnis aus Blockchiffre von x_{i-1}

Start mit Initialisierungsvektor $IV = \{0\}^b$

Ausgabe ist Ergebnis des letzten Blockchiffres

Algorithmus

$$x = x_1 \,||\, \ldots \,||\, x_n$$

$$IV \leftarrow 00\ldots0$$

$$y_0 \leftarrow IV$$

for i ← 1 to n do

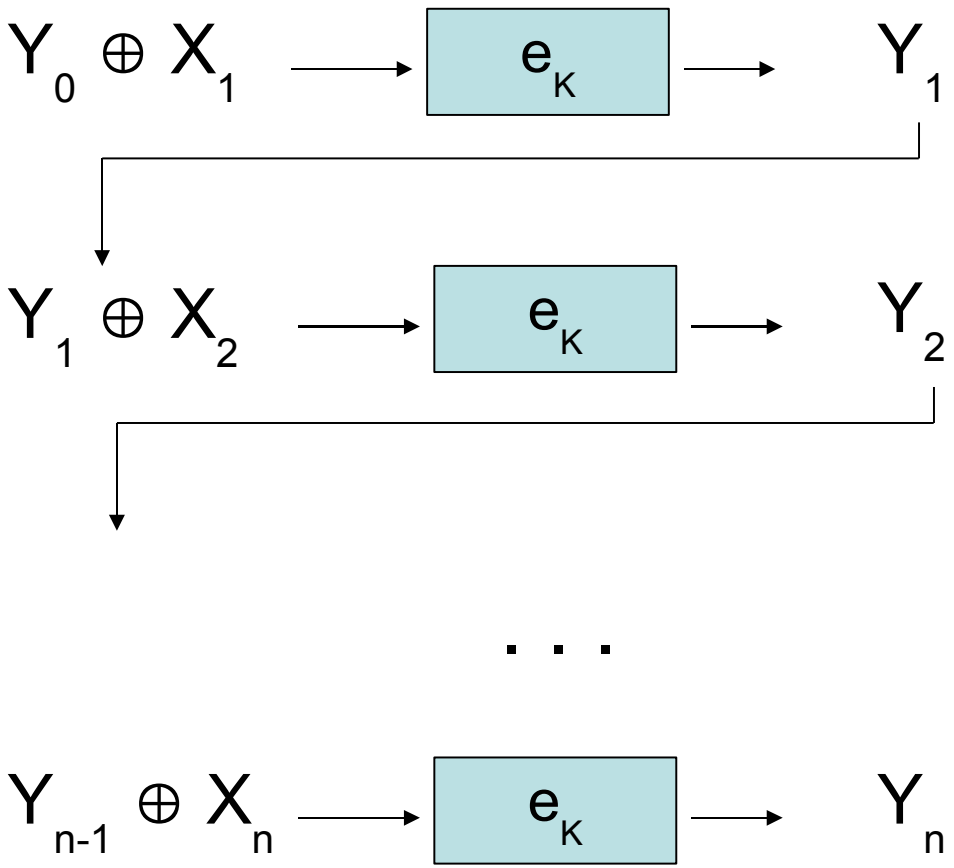
$$y_i \leftarrow e_K(y_{i-1} \oplus x_i)$$

return (y_n)

IV = Initialisierungsvektor

e_K = Verschlüsselung mit Schlüssel K

Algorithmus



Sicherheit

gilt als sicher wenn:

Blockchiffre sicher ist

Nachrichtenlänge konstant ist

Sicherheit

unsicher, wenn Nachrichtenlänge nicht konstant:

erfrage y_1 für $h_K(x_1)$

erfrage y_2 für $h_K(y_1||x_2)$

y_2 ist dann eine Fälschung!

Sicherheit

beste Attacke bei konstanter Länge: Geburtstag-Attacke

Ziel: Kollision $h_K(x_i) = h_K(x_j)$ für $i \neq j$

wegen Geburtstags-Paradoxon liegt die Wahrscheinlichkeit für einen Erfolg bei 1/2 für $2^{q/2}$ Versuchen

$(1/2,2^{q/2})$-Fälschung

Grundlagen

Idee: Verknüpfung zweier Hash-Funktionen (mit Schlüseln)

$$NMAC_{(K,L)}(x) = h_L(g_K(x))$$

h_L heißt dann „little MAC“

Verkettung heißt „big MAC“

Bedingungen

sicher wenn:

g_K muss kollisionsresistent ist

h_L eine sichere MAC-Funktion ist

Angriffe

„big MAC attack“

Angreifer versucht Fälschung für $h_L(g_K(x))$ durch probieren zu erzeugen

„little MAC attack“

Angreifer versucht Fälschung für $g_K(x)$ durch probieren zu erzeugen

Angriffe

„unknown-key collision attack“

Angreifer versucht durch probieren eine Kollision zu erzeugen

Die Sicherheit eines Nested MACs hängt also von der Sicherheit der verwendeten Funktionen g und h ab!

Grundlagen

Spezialfall von Nested MAC

Zentrale Idee: Schlüssel wird dazu verwendet, den Initialwert der Anfangswerte zu beeinflussen.

Standard definiert z.B. in RFC 2104

Algorithmus

$$\mathbf{HMAC_K(x) = H(K \oplus opad || H(K \oplus ipad || x))}$$

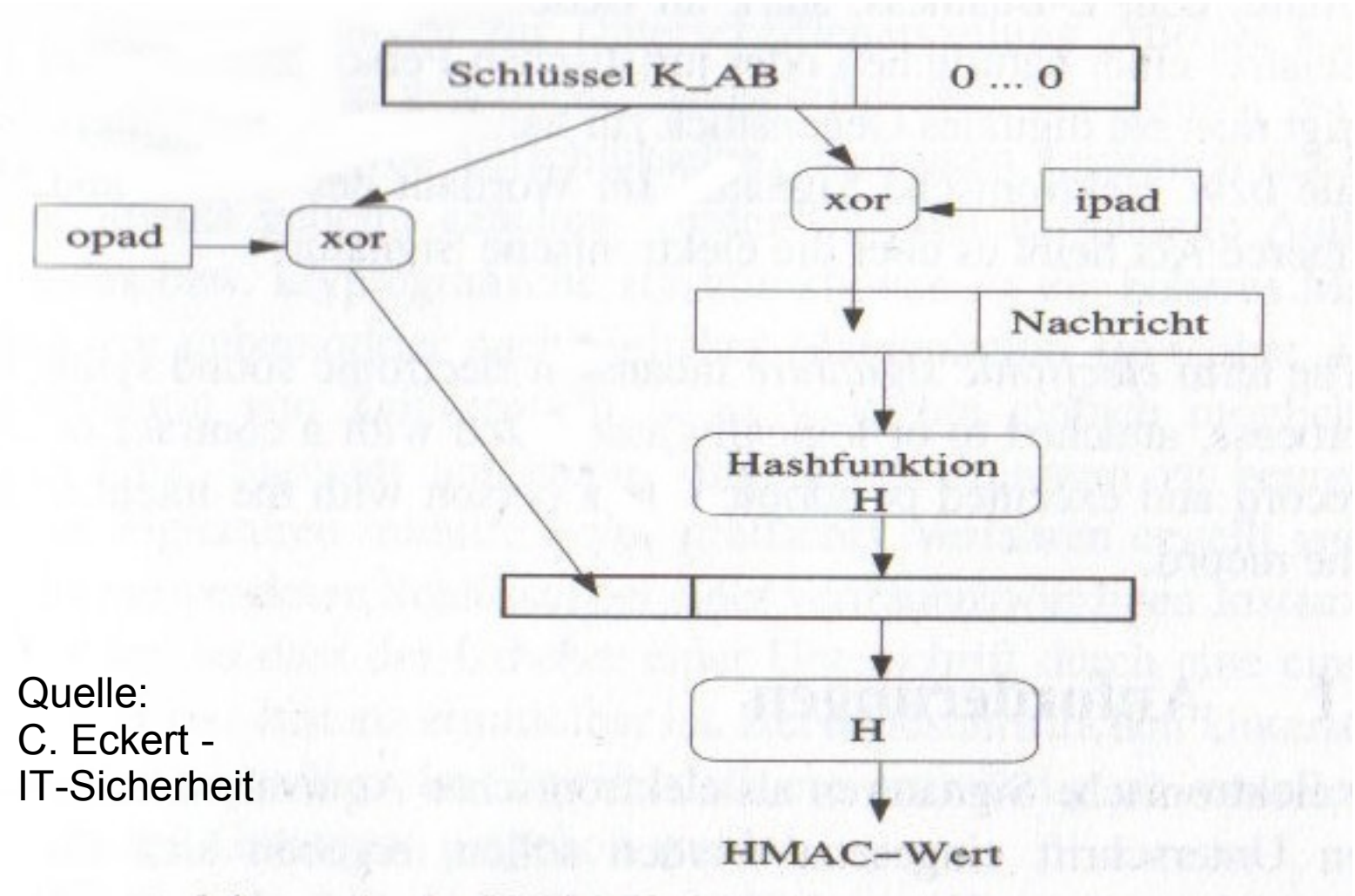
K: Schlüssel
H: Hashfunktion (z.B. SHA-1)
x: Nachricht

$ipad = (363636...36)_{16}$
$opad = (5C5C5C...5C)_{16}$

Algorihmus

1. $\oplus$ von K mit ipad
2. Anfügen von x an Ergebnis von 1.
3. Hash-Funktion auf Ergebnis von 2.
4. $\oplus$ von K mit opad
5. Anfügen von Ergebnis 3. an Ergebnis von 4.
6. Hash-Funktion auf Ergebnis von 5.
7. Ergebnis von 6. ausgeben

Algorithmus



Quelle:
C. Eckert -
IT-Sicherheit

Beispiel

http://postmortem.csd.auth.gr/~gpall/ftp_dir/software/

→ hmac.exe

Beispiel-Code auch im RFC 2104 zu finden!

Wiederholung

Tupel (X, Y, K, H)

X : alle Nachrichten

Y : alle Authentication Tags

K : alle Schlüssel

H : alle Hash-Funktionen

Wiederholung

$$|X| = N$$

$$|Y| = M$$

$$\rightarrow (N,M)\text{-Hash Family}$$

Grundlagen

Verbesserung: Schlüssel wird nur einmal verwendet

Angreifer hat nur einen Versuch für Fälschung

nur noch (ε,0)- oder (ε,1)-Fälschungen möglich

Eigenschaften

kleinstmögliche Wahrscheinlichkeiten ε für (ε,0)- und (ε,1)-Fälschungen

unabhängig von Rechenresourcen des Angreifers

Definitionen

Pd_q für q = {0,1} : Deception Probability

→ max. Wert für ε für eine (ε,q)-Fälschung

payoff(x,y) : Wahrscheinlichkeit, dass (x,y) gültiges Paar ist

payoff(x',y' ; x,y) : bedingte Wahrscheinlichkeit, dass (x',y') gültiges Paar ist, wenn (x,y) gültig ist

Deception Probability

$$\text{payoff}(x,y) = \Pr[y = h_K(x)]$$

$$= \frac{|\{K \in \mathcal{K} : h_K(x) = y\}|}{|\mathcal{K}|}$$

$$Pd_0 = \max\{\text{payoff}(x,y) : x \in \mathcal{X},\ y \in \mathcal{Y}\}$$

Deception Probability

$$\text{payoff}(x',y' ; x,y) = \frac{\Pr[y' = h_K(x') \text{ and } y = h_K(x)]}{\Pr[y = h_K(x)]}$$

$$= \frac{|\{K \in \mathcal{K} : h_K(x') = y', h_K(x) = y\}|}{|\{K \in \mathcal{K} : h_K(x) = y\}|}$$

$$Pd_1 = \max\{\text{payoff}(x',y' ; x,y) : x,x' \in X, y,y' \in Y, x \neq x'\}$$

Authentication Matrix

$X = Y = \{0,1,2\}$

$K = \{0,1,2\} \times \{0,1,2\}$

für alle K = (a,b)

$h_{(a,b)} = ax + b \bmod 3$

(3,3)-Hash Family

key / x	0	1	2
(0,0)	0	0	0
(0,1)	1	1	1
(0,2)	2	2	2
(1,0)	0	1	2
(1,1)	1	2	0
(1,2)	2	0	1
(2,0)	0	2	1
(2,1)	1	0	2
(2,2)	2	1	0

Beispiel (ε,0)-Fälschung

Oscar fängt x ab und rät ein gültiges Paar (x,y)

ein gültiges y kommt in jeder Spalte 3 mal vor

$Pd_0 = 3/9 = 1/3$

key / x	0	1	2
(0,0)	0	0	0
(0,1)	1	1	1
(0,2)	2	2	2
(1,0)	0	1	2
(1,1)	1	2	0
(1,2)	2	0	1
(2,0)	0	2	1
(2,1)	1	0	2
(2,2)	2	1	0

Beispiel (ε,1)-Fälschung

Oscar hat ein gültiges Paar (0,0)

Oscar weiß dadurch, dass (0,0), (1,0) oder (2,0) mögliche Schlüssel sind

(1,1) ist mögliche Fälschung

$Pd_1 = 1/3$

key / x	0	1	2
(0,0)	0	0	0
(0,1)	1	1	1
(0,2)	2	2	2
(1,0)	0	1	2
(1,1)	1	2	0
(1,2)	2	0	1
(2,0)	0	2	1
(2,1)	1	0	2
(2,2)	2	1	0

Definitionen

(X, Y, K, H) ist (N,M)-Hash Family

Strongly Universal, wenn für jedes $x, x' \in X$ $(x \neq x')$ und jedes $y, y' \in Y$ gilt:

$$|\{K \in K : h_K(x) = y,\ h_K(x') = y'\}| = \frac{|K|}{M^2}$$

Beispiel

$$|\{(1,0) \in K : h_{(1,0)}(0) = 0, \; h_{(1,0)}(1) = 1\}| = 1$$

$$\frac{|K|}{M^2} = \frac{9}{3^2} = 1$$

key / x	0	1	2
(0,0)	0	0	0
(0,1)	1	1	1
(0,2)	2	2	2
(1,0)	0	1	2
(1,1)	1	2	0
(1,2)	2	0	1
(2,0)	0	2	1
(2,1)	1	0	2
(2,2)	2	1	0

Zusammenfassung

MACs dienen der Authentifizierung von Daten

Hash-Funktionen kombiniert mit Schlüsseln

Authentifizierung auch über völlig offenen Kanal

Unconditionally Secure MACs bieten größtmögliche Sicherheit

Quellen

Douglas R. Stinson - Cryptography: Theory and Practice. 2nd Edition, Chapman & Hall/CRC 2002

Claudia Eckert – IT-Sicherheit, 4. Auflage, Oldenburg 2006

Günter Schäfer – Netzsicherheit , dpunkt 2003

http://www.cs.uni-potsdam.de/ti/lehre/05-Kryptographie/slides/MAC_vortrag.pdf

http://www.cs.uni-potsdam.de/ti/lehre/04-Kryptographie/slides/MessageAuthenticationCodes.pdf

Quellen

http://www.math.tu-berlin.de/~hess/krypto/vl-8-4.pdf

http://weisskugel.informatik.uni-mannheim.de/people/lucks/vorl0102/v9.ps

http://postmortem.csd.auth.gr/~gpall/ftp_dir/software/