

Message Authentication Codes

Martin Schütte

30. Nov. 2004

Gliederung

Definitionen

Grundlegende Begriffe

Konstruktion von MACs

häufig benutzte MACs

Einschätzung der Sicherheit

Bedingungslos sichere MACs

zusätzliche Definitionen

Universelle Hash-Familien

Schluss

Was ist ein Message Authentication Code (MAC)?

- “keyed hash function“
- Funktion $h(K, x)$ bzw. Funktionenfamilie $h_K(x)$, bildet Nachrichten x und Schlüssel K auf MACs oder Tags ab
- Nur die Kommunikationspartner sollen den MAC berechnen können
- Authentizität: Nachricht stammt vom Kommunikationspartner und wurde nicht manipuliert
- Verschlüsselung allein garantiert die Authentizität nicht!



Angriffsmodell

1. Ein Orakel berechnet $y = MAC_K(x)$
2. Der Angreifer stellt q Anfragen x_1, x_2, \dots, x_q an das Orakel und bekommt so gültige Paare $(x_1, y_1), (x_2, y_2), \dots, (x_q, y_q)$
3. Ziel des Angreifers: neues Paar (x, y) erzeugen ($x \notin \{x_1, \dots, x_q\}$)
 - Wenn (x, y) ein gültiges Paar ist, so ist es eine Fälschung
 - Wenn (x, y) mit Wahrscheinlichkeit ϵ gültig ist, so ist es eine (ϵ, q) -Fälschung

CBC MAC

- $\text{CBC-MAC}_K(x) = y_n$, mit $y_i = e_K(y_{i-1} \oplus x_i)$ ($1 \leq i \leq n$)
- Grundbaustein: symmetrische Blockverschlüsselung
- üblicherweise DES, AES

- Vorteil: verbreitete Verschlüsselungsfunktionen gut analysiert
- Nachteil: normalerweise sehr langsam

Vorsicht bei der Konstruktion: $e_K(x_1) \oplus e_K(x_2) \oplus \dots \oplus e_K(x_n)$
ergibt **keine** gute MAC-Funktion

Einfache Erweiterungen von Hashfunktionen

Sei $h(x)$ eine iterierte Hash-Funktion.

- Secret IV: $\text{MAC}_K(x) = h(x)$, mit dem Initialwert K
- Secret Prefix: $\text{MAC}_K(x) = h(K||x)$
- Secret Suffix: $\text{MAC}_K(x) = h(x||K)$
- Envelope: $\text{MAC}_K(x) = h(K||x||K)$

Nested MAC (NMAC)

- Grundidee: Verknüpfung zweier Hash-Funktionen (mit Schlüsseln)
- $\text{NMAC}_{(K_1, K_2)}(x) = h_{K_1}(g_{K_2}(x))$
- g_{K_2} muss kollisionsresistent sein
- h_{K_1} ist der “little MAC“ und muss eine sichere MAC-Funktion sein
- die Verkettung ist der “big MAC“

Hash MAC (HMAC)

wie nach RFC 2104

- $HMAC_K(x) = H(K \oplus \text{opad} || H(K \oplus \text{ipad} || x))$
- Spezialfall von NMAC mit
 - $h_K(x) = g_K(x) = H(K || x)$
 - $K_1 = K \oplus \text{opad}$
 - $K_2 = K \oplus \text{ipad}$
 - ($\text{opad} = 0x3636\dots36$, $\text{ipad} = 0x5c5c\dots5c$)
- Vorteil: Hash-Funktion H wird als 'black box' benutzt und ist beliebig auswechselbar

Sicherheit von NMACs

- “little MAC“-Angriff: Angreifer erzeugt $(y', h_{K_1}(y'))$
Angenommen hierfür gibt es keinen (ϵ_1, q) -Angriff.
- Kollisionsangriff: Angreifer erzeugt $x' \neq x''$ und
 $g_{K_2}(x') = g_{K_2}(x'')$
Angenommen hierfür gibt es keinen $(\epsilon_2, q + 1)$ -Angriff.
- “big MAC“-Angriff: Angreifer erzeugt $(x', h_{K_1}(g_{K_2}(x')))$

Falls die Annahmen stimmen und es für den “big MAC“ einen (ϵ, q) -Angriff gibt, so ist (beweisbar) $\epsilon < \epsilon_1 + \epsilon_2$

Das heißt: Eine Schwachstelle der Funktionen g, h ist
Vorbedingung für jeden Angriff auf NMAC

Allgemeine Angriffe auf MACs

- bester bekannter Angriff: Kollisions-/Geburtstagsangriff
- Wenn MAC_K die Länge von n bits hat, müssten im Mittel $2^{n/2}$ MACs berechnet werden für einen $(1/2, O(2^{n/2}))$ -Angriff
- In der Praxis unmöglich, weil die Berechnung der MAC-Werte vom Opfer erfolgen muss.
- zum Vergleich:
 - $1 \text{ MAC/sec} \implies 2^{25} \text{ MACs/Jahr}$
 - Über 1 GBit/sec-Leitung können 2^{46} MACs angefordert werden.

Geht es noch sicherer?

Erste Verbesserung

Jeden Schlüssel K nur einmal benutzen.

⇒ nur noch $(\epsilon, 0)$ und $(\epsilon, 1)$ -Fälschungen möglich.

Bedingungslos sichere MACs

Eigenschaften eines bedingungslos sicheren MACs:

- Wahrscheinlichkeiten ϵ_0 und ϵ_1 der möglichen $(\epsilon_0, 0)$ und $(\epsilon_1, 1)$ -Angriffe sind möglichst klein
- Sicherheit ist unabhängig von den Ressourcen des Angreifers

Die Funktion payoff

- Sei $\text{payoff}(x, y)$ die Wahrscheinlichkeit, dass (x, y) ein gültiges Paar ist:

$$\text{payoff}(x, y) = \Pr[y = h_K(x)] = \frac{|\{K \in \mathcal{K} : h_K(x) = y\}|}{|\mathcal{K}|}$$

- Sei $\text{payoff}(x, y; x', y')$ die bedingte Wahrscheinlichkeit dafür, dass (x', y') ein gültiges Paar ist, wenn (x, y) gültig ist:

$$\begin{aligned} \text{payoff}(x, y; x', y') &= \Pr[y' = h_K(x') | y = h_K(x)] \\ &= \frac{\Pr[y' = h_K(x') \wedge y = h_K(x)]}{\Pr[y = h_K(x)]} = \frac{|\{K \in \mathcal{K} : h_K(x') = y', h_K(x) = y\}|}{|\{K \in \mathcal{K} : h_K(x) = y\}|} \end{aligned}$$

Deception Probability

- Pd_q heißt deception probability bzw. Betrugswahrscheinlichkeit von q
- Pd_q ist der größte Wert ϵ , so dass ein (ϵ, q) -Fälscher existiert
- Maximum über alle möglichen Schlüssel K
- $\Rightarrow Pd_0 = \max\{\text{payoff}(x, y)\}$
- $\Rightarrow Pd_1 = \max\{\text{payoff}(x, y; x', y')\} (\forall x \neq x', y, y')$
- Raten ist immer möglich, deshalb:
Bei M möglichen MAC-Werten: $Pd_q \geq 1/M$ (für alle q)

Universelle Hash-Familien

- Eine Hash-Familie, die N Nachrichten auf M Hash-Werte abbildet, heißt (N, M) -Hash-Familie
- Universellen Hash-Familien müssen bestimmte Anforderungen an Kollisionswahrscheinlichkeit, Schlüssel- und Wertebereich erfüllen
- Eine (N, M) -Hash-Familie heißt stark-universell, wenn $\forall x, x', y, y', x \neq x'$ gilt:

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

Zwei stark-universelle Hash-Familien

1. Für eine Primzahl p und $a, b \in \mathbb{Z}_p$ sei

$$f_{(a,b)}(x) := ax + b \pmod{p}$$

$f_{(a,b)} : \mathbb{Z}_p \Rightarrow \mathbb{Z}_p$ ist dann eine stark-universelle (p, p) -Hash-Familie.

2. Seien $j \in \mathbb{N}$, p eine Primzahl, $\mathcal{X} = \{0, 1\}^j \setminus \{(0, \dots, 0)\}$,
 $\vec{r} \in (\mathbb{Z}_p)^j$

$$f_{\vec{r}}(\vec{x}) := \vec{r} \cdot \vec{x} \pmod{p} = \sum_{i=1}^j r_i x_i$$

Dann ist $f_{\vec{r}} : \mathcal{X} \rightarrow \mathbb{Z}_p$ eine stark-universelle $(2^{j-1}, p)$ -Hash-Familie.

Berechnung von Pd_0

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

$$= \sum_{y'} \frac{|\mathcal{K}|}{M^2} = \frac{|\mathcal{K}|}{M}$$

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

$$= \frac{|\mathcal{K}|/M}{|\mathcal{K}|} = 1/M$$

$$\Rightarrow Pd_0 = 1/M$$

Berechnung von Pd_1

$$\begin{aligned} \text{payoff}(x', y'; x, y) &= \frac{|\{K \in \mathcal{K} : h_K(x) = y, h_K(x') = y'\}|}{|\{K \in \mathcal{K} : h_K(x) = y\}|} \\ &= \frac{|\mathcal{K}|/M^2}{|\mathcal{K}|/M} = 1/M \end{aligned}$$

$$\Rightarrow Pd_1 = 1/M$$





Also ist $Pd_0 = Pd_1 = 1/M$.

Die Möglichkeit eines Angriffs ist minimal.

Fazit

1. Auf einem völlig offenen Kanal (Angreifer hat Lese- und Schreibzugriff) lässt sich eine authentifizierte Verbindung aufbauen.
2. Wenn jede Nachricht mit zufälligem Schlüssel authentifiziert wird, ist die Authentifizierung bedingungslos sicher, d.h. mit unbegrenzten Ressourcen nicht zu fälschen.
⇒ In der Realität hängt alles von der Schlüsselquelle ab.

Quellen

-  Douglas Stinson,
Cryptography – Theory and Practice, pp. 136-149
Chapman & Hall/CRC, Boca Raton, Florida, 2nd ed., 2002
-  Mihir Bellare, Ran Canetti, Hugo Krawczyk,
Keying Hash Functions for Message Authentication,
1996, <http://www.cse.ucsd.edu/users/mihir/papers/hmac.html>
-  Shafi Goldwasser, Mihir Bellare,
Lecture Notes on Cryptography,
Chapter 8, 2001, Cambridge, Massachusetts,
<http://www.cse.ucsd.edu/users/mihir/papers/gb.html>
-  Bart Preneell, Paul van Oorschot,
MDx-MAC and Building Fast MACs from Hash Functions,
in: *Advances in Cryptology – CRYPTO '95*, LNCS 963,
pp. 1-14, Springer-Verlag, Berlin, 1995