

Message Authentication Code (MAC)

Szenario: Integrität und Authentizität mittels MACs.

- Alice und Bob besitzen gemeinsamen Schlüssel k .
- Alice berechnet für m einen MAC-Tag t als Funktion von m und k .
- Alice sendet das Tupel (m, t) an Bob.
- Bob verifiziert, dass t ein gültiger Tag für m ist.

Definition Message Authentication Code (MAC)

Ein *Message Authentication Code (MAC)* besteht aus den ppt Alg.

- 1 **Gen:** $k \leftarrow \text{Gen}(1^n)$
- 2 **Mac:** Bei Eingabe von k und $m \in \{0, 1\}^*$ berechne $t \leftarrow \text{Mac}_k(m)$.
- 3 **Vrfy:** Bei Eingabe (m, t) und Schlüssel k berechne

$$\text{Vrfy}_k(m, t) = \begin{cases} 1 & \text{falls } t \text{ ein gültiger MAC für } m \text{ ist} \\ 0 & \text{sonst} \end{cases} .$$

Es gilt $\text{Vrfy}(m, \text{Mac}_k(m)) = 1$ für alle m und k .

Sicherheitsspiel Mac-forge

Spiel Sicherheit von MACs Mac-forge

Sei Π ein MAC mit Sicherheitsparameter n und Angreifer \mathcal{A} .

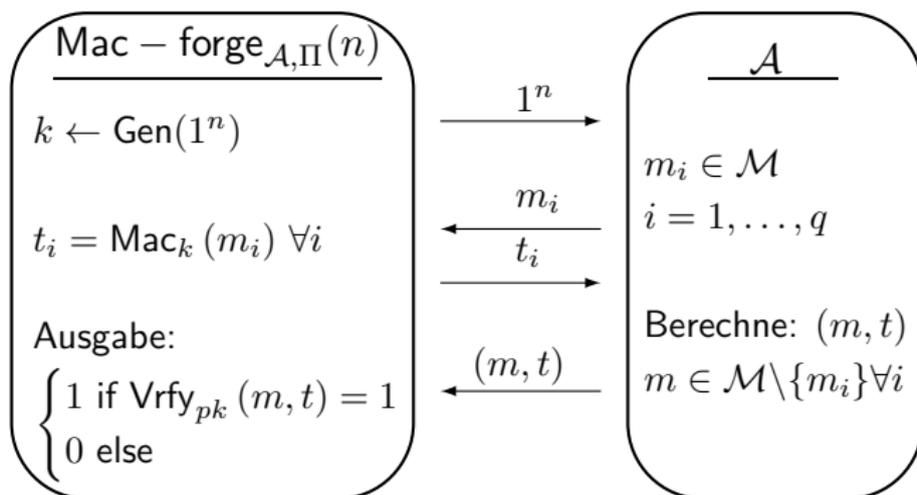
- 1 $k \leftarrow \text{Gen}(1^n)$
- 2 $(m, t) \leftarrow \mathcal{A}^{\text{Mac}_k(\cdot)}(1^n)$. Sei Q die Menge aller $\text{Mac}_k(\cdot)$ -Anfragen von \mathcal{A} an sein Orakel.
- 3 $\text{Mac-forge}_{\mathcal{A}, \Pi}(n) = \begin{cases} 1 & \text{falls } \text{Vrfy}(m, t) = 1 \text{ und } m \notin Q \\ 0 & \text{sonst} \end{cases}$.

Definition Sicherheit eines MACs

Ein MAC Π heißt *existentiell unfälschbar gegenüber adaptiv gewählten Angriffen* bzw. kurz *sicher* falls für alle ppt Angreifer \mathcal{A} gilt

$$\text{Ws}[\text{Mac-forge}_{\mathcal{A}, \Pi}(n) = 1] \leq \text{negl}(n).$$

Sicherheitsspiel Mac-forge



Replay Angriffe

Szenario: Replay Angriff

- Alice schickt an ihre Bank eine authentifizierte Zahlungsanweisung (m, t) über 100 Euro zugunsten Bob's Konto.
- Aufgrund der MAC-Sicherheit kann Bob den Betrag nicht ändern.
- Die MAC-Sicherheit verhindert nicht, dass Bob (m, t) abfängt und dieselbe Nachricht (m, t) weitere Male an die Bank versenden.
- Abhilfe: Verwenden Nummerierung oder Zeitstempel.

Seriennummer:

- Berechnen MAC von $i||m$ für eindeutige i .
- MAC-Sicherheit: \mathcal{A} kann nicht MAC für $i' || m$ berechnen.

Zeitstempel:

- Sender berechnet MAC von $Systemzeit||m$.
- Empfänger verifiziert, dass die $Systemzeit$ aktuell ist.

Konstruktion eines sicheren MACs fester Länge

Algorithmus Π_{MAC} fester Länge

Sei F eine Pseudozufallsfunktion mit Blocklänge n . Wir konstruieren einen MAC für Nachrichten $m \in \{0, 1\}^n$.

- 1 **Gen:** Wähle $k \in_R \{0, 1\}^n$.
- 2 **Mac:** Für $m, k \in \{0, 1\}^n$ berechne $t := F_k(m)$.
- 3 **Vrfy:** Für $(m, t) \in \{0, 1\}^n \times \{0, 1\}^n$ und $k \in \{0, 1\}^n$

$$\text{Ausgabe} = \begin{cases} 1 & \text{falls } t = F_k(m) \\ 0 & \text{sonst} \end{cases} .$$

Sicherheit von Π_{MAC}

Satz Sicherheit von Π_{MAC}

Sei F eine Pseudozufallsfunktion. Dann ist Π_{MAC} sicher.

Beweis:

- Sei \mathcal{A} ein Angreifer für Π_{MAC} mit Erfolgsws $\epsilon(n)$.
- Wir konstruieren Unterscheider U für Pseudozufallsfunktionen.

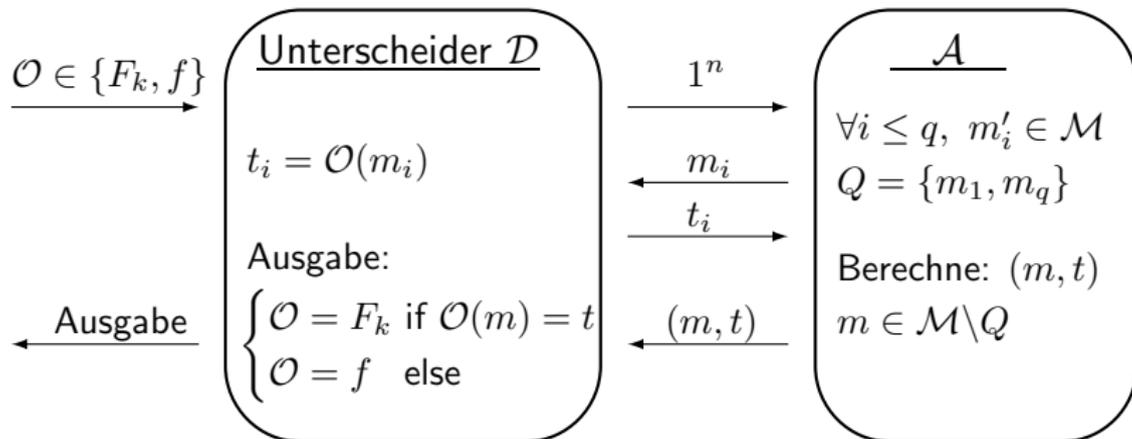
Algorithmus Unterscheider U

EINGABE: 1^n , $\mathcal{O} : \{0, 1\}^n \rightarrow \{0, 1\}^n$ mit $\mathcal{O} = F_k(\cdot)$ oder $\mathcal{O} = f(\cdot)$.

- 1 $(m, t) \leftarrow \mathcal{A}^{Mac(\cdot)}$, beantworte $Mac(m')$ -Anfragen mit $t' := \mathcal{O}(m')$.
- 2 Sei Q die Menge aller von \mathcal{A} gestellten $Mac(\cdot)$ -Anfragen.

AUSGABE =
$$\begin{cases} 1 & \text{falls } t' = \mathcal{O}(m'), m' \notin Q, \text{ Interpretation: } \mathcal{O} = F_k(\cdot) \\ 0 & \text{sonst, Interpretation: } \mathcal{O} = f(\cdot) \end{cases}$$

Sicherheit von Π_{MAC}



Sicherheit von Π_{MAC}

Fall 1: $\mathcal{O} = F_k(\cdot)$, d.h. das Orakel ist eine Pseudozufallsfunktion.

- Dann ist die Verteilung für \mathcal{A} identisch zum Protokoll Π_{MAC} .
- Damit gilt

$$\text{Ws}[U^{F_k(\cdot)}(n) = 1] = \text{Ws}[\text{Mac-forge}_{\mathcal{A}, \Pi_{MAC}}(n) = 1] = \epsilon(n).$$

Fall 2: $\mathcal{O} = f(\cdot)$, d.h. das Orakel ist eine echte Zufallsfunktion.

- Sei Π' das Protokoll Π_{MAC} mit $f(\cdot)$ statt $F_k(\cdot)$.
- Für alle $m \notin Q$ ist $t = f(m)$ uniform verteilt in $\{0, 1\}^n$.
- Damit gilt

$$\text{Ws}[U^{f(\cdot)}(n) = 1] = \text{Ws}[\text{Mac-forge}_{\mathcal{A}, \Pi'}(n) = 1] = \frac{1}{2^n}.$$

Aus der Pseudozufälligkeit von F_k folgt für alle ppt U

$$\text{negl}(n) \geq |\text{Ws}[U^{F_k(\cdot)}(n) = 1] - \text{Ws}[U^{f(\cdot)}(n) = 1]| = |\epsilon - \frac{1}{2^n}|.$$

Damit folgt $\epsilon \leq \text{negl}(n) + \frac{1}{2^n} = \text{negl}(n)$ für alle ppt Angreifer \mathcal{A} .

Von fester zu variabler Länge

Ziel: Konstruiere MAC für $m = m_1 \dots m_\ell$ für variable Blockzahl ℓ .

Überlegungen zu einer sicheren MAC-Konstruktion:

- **MAC des XOR der Blocks**, d.h. $t := \text{Mac}_k(\bigoplus_{i=1}^{\ell} m_i)$.
- Problem: Tag t ist z.B. gültig für $\overline{m_1 m_2} m_3 \dots m_\ell$.
- **MAC jeden Blocks**, d.h. $t = t_1 \dots t_\ell$ für $t_i := \text{Mac}_k(m_i)$.
- Problem: $t' = t_2 t_1 t_3 \dots t_\ell$ ist gültig für $m' = m_2 m_1 m_3 \dots m_\ell$.
- **MAC mit Block-Seriennummer**, d.h. $t_i := \text{Mac}_k(i || m_i)$.
- Problem: $t' = t_1 \dots t_{\ell-1}$ ist gültig für $m' = m_1 \dots m_{\ell-1}$.
- **MAC mit Nachrichtenlänge**, d.h. $t_i := \text{Mac}_k(\ell || i || m_i)$.
- Problem: Seien $t = t_1 \dots t_\ell$, $t' = t'_1 \dots t'_\ell$ gültig für m, m' . Dann ist $t'_1 t_2 \dots t_\ell$ gültig für $m'_1 m_2 \dots m_\ell$, d.h. wir können Tags kombinieren.
- **MAC mit Nachrichten-Seriennummer**: $t_i := \text{Mac}_k(r || \ell || i || m_i)$.
- Wir benötigen pro Nachricht m einen eindeutigen Identifikator r .

Sicherer MAC für Nachrichten variabler Länge

Algorithmus MAC Π_{MAC2} variabler Länge

Sei $\Pi' = (Gen', Mac', Vrfy')$ ein MAC für Nachrichten der Länge n .

1 **Gen:** $k \leftarrow Gen(1^n)$

2 **Mac:** Sei $k \in \{0, 1\}^n$ und $m = m_1 \dots m_\ell \in (\{0, 1\}^{\frac{n}{4}})^\ell$.
Wähle $r \in_R \{0, 1\}^{\frac{n}{4}}$ und berechne

$$t_i \leftarrow Mac'_k(r || \ell || i || m_i) \text{ für } i = 1, \dots, \ell,$$

mit Kodierungen $\ell, i \in \{0, 1\}^{\frac{n}{4}}$. Ausgabe des Tags $t = (r, t_1 \dots t_\ell)$.

3 **Vrfy:** Für $(m, t) = (m_1 \dots m_\ell, r, t_1, \dots, t_\ell)$

$$\text{Ausgabe} = \begin{cases} 1 & \text{falls } Vrfy'_k(r || \ell || i || m_i, t_i) = 1 \text{ für } i = 1, \dots, \ell \\ 0 & \text{sonst} \end{cases}.$$

Anmerkung:

- Benötigen $\ell < 2^{\frac{n}{4}}$, sonst kann ℓ nicht mit $\frac{n}{4}$ Bits kodiert werden.

Sicherheit von Π_{MAC2}

Satz Sicherheit von Π_{MAC2}

Sei Π' sicher. Dann ist Π_{MAC2} ebenfalls sicher.

Beweis:

- Sei \mathcal{A} ein Angreifer für Π_{MAC2} mit Erfolgsws $\epsilon(n)$.
- Wir konstruieren einen Angreifer \mathcal{A}' für Π' .

Algorithmus Angreifer \mathcal{A}'

EINGABE: 1^n , Orakel $Mac'_k(\cdot)$.

- 1 Beantworte $Mac_k(m'_1 \dots m'_{\ell'})$ -Anfragen von \mathcal{A} wie folgt: Wähle $r' \in_R \{0, 1\}^{\frac{n}{4}}$ und berechne $t'_i = Mac'_k(r' || \ell' || i || m'_i)$ für $i = 1, \dots, \ell'$.
- 2 $(m, t) = (m_1 \dots m_\ell, r, t_1 \dots t_\ell) \leftarrow \mathcal{A}^{Mac_k(\cdot)}(1^n)$.

AUSGABE: Nicht-angefragtes $r || \ell || i || m_i$ mit gültigem Tag t_i , falls ein solches in (m, t) existiert.

Sicherheit von Π_{MAC2}

