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)* bzgl. des Nachrichtenraumen $\mathcal{M}$ besteht aus den ppt Alg.

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

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

Korrektheit: Es gilt $Vrfy_k(m, Mac_k(m)) = 1$ für alle m und k.

Sicherheitsspiel $\mathrm{Mac\text{-}forge}$

Spiel Sicherheit von MACs $\mathrm{Mac\text{-}forge}$

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

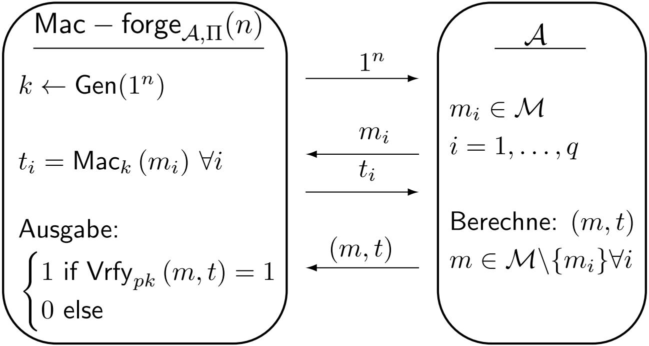
1. $k \leftarrow Gen(1^n)$
2. $(m, t) \leftarrow \mathcal{A}^{Mac_k(\cdot)}(1^n)$. Sei Q die Menge aller $Mac_k(\cdot)$-Anfragen von $\mathcal{A}$ an sein Orakel.
3. $\mathrm{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \text{falls } Vrfy_k(m,t) = 1 \text{ und } m \notin Q \\ 0 & \text{sonst}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

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

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

Sicherheitsspiel Mac-forge



Replay Angriffe

Szenario: Replay Angriff

- Alice schickt an ihre Bank eine authentisierte 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 MAC 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 $\mathcal{D}$ für Pseudozufallsfunktionen.

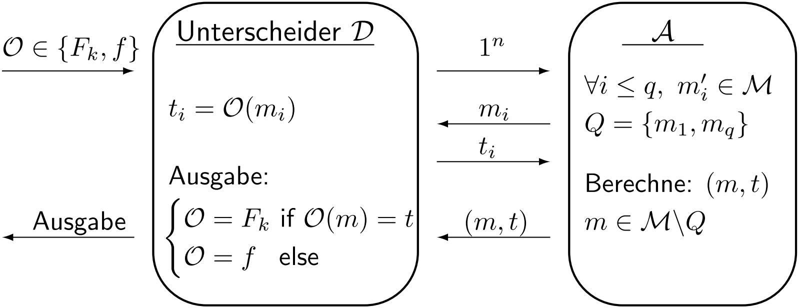
Algorithmus Unterscheider $\mathcal{D}$

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

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

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

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

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

Aus der Pseudozufälligkeit von F_k folgt für alle ppt $\mathcal{D}$

$$\mathrm{negl}(n) \geq \left|\mathrm{Ws}[\mathcal{D}^{F_k(\cdot)}(n) = 1] - \mathrm{Ws}[\mathcal{D}^{f(\cdot)}(n) = 1]\right| = \left|\epsilon - \frac{1}{2^n}\right|.$$

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

Von fester zu variabler Länge

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

Überlegungen zu einer sicheren MAC-Konstruktion:

- **MAC des XOR der Blocks**, d.h. $t := Mac_k(\bigoplus_{i=1}^{\ell} m_i)$.
- Problem: Tag t ist z.B. gültig für $\overline{m_1 m_2} m_3 \ldots m_\ell$.

- **MAC jeden Blocks**, d.h. $t = t_1 \ldots t_\ell$ für $t_i := Mac_k(m_i)$.
- Problem: $t' = t_2 t_1 t_3 \ldots t_\ell$ ist gültig für $m' = m_2 m_1 m_3 \ldots m_\ell$.

- **MAC mit Block-Seriennummer**, d.h. $t_i := Mac_k(i||m_i)$.
- Problem: $t' = t_1 \ldots t_{\ell-1}$ ist gültig für $m' = m_1 \ldots m_{\ell-1}$.

- **MAC mit Nachrichtenlänge**, d.h. $t_i := Mac_k(\ell||i||m_i)$.
- Problem: Seien $t = t_1 \ldots t_\ell$, $t' = t'_1 \ldots t'_\ell$ gültig für m, m'. Dann ist $t'_1 t_2 \ldots t_\ell$ gültig für $m'_1 m_2 \ldots m_\ell$, d.h. wir können Tags kombinieren.

- **MAC mit Nachrichten-Seriennummer**: $t_i := Mac_k(r||\ell||i||m_i)$.
- 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 $m = m_1 \ldots m_\ell \in (\{0,1\}^{\frac{n}{4}})^\ell$.
   Wähle $r \in_R \{0,1\}^{\frac{n}{4}}$ und berechne
   $$t_i \leftarrow Mac'_k(r||\ell||i||m_i) \text{ für } i = 1, \ldots, \ell,$$
   mit Kodierungen $\ell, i \in \{0,1\}^{\frac{n}{4}}$. Ausgabe des Tags $t = (r, t_1 \ldots t_\ell)$.
3. **Vrfy:** Für $(m,t) = (m_1 \ldots m_\ell, r, t_1, \ldots, t_\ell)$
   $$\text{Ausgabe} = \begin{cases} 1 & \text{falls } Vrfy'_k(r||\ell||i||m_i, t_i) = 1 \text{ für } i = 1, \ldots, \ell \\ 0 & \text{sonst} \end{cases}.$$

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 $Mac_k(m_j^1 \ldots m_j^{\ell'})$-Anfragen von $\mathcal{A}$ wie folgt: Wähle $r_j \in_R \{0,1\}^{\frac{n}{4}}$ und berechne $\overline{t}_j^i = Mac'_k(r_j||\ell||i||m_j^i)$ für $i = 1, \ldots, \ell$.
2. $(m,t) = (m^1 \ldots m^\ell, r, t^1 \ldots t^\ell) \leftarrow \mathcal{A}^{Mac_k(\cdot)}(1^n)$.

AUSGABE: Nicht-angefragtes $\bar{m}^i = r||\ell||i||m^i$ mit gültigem Tag t^i, falls ein
solches in $(\bar{m}^i, t^i)$ existiert.

Sicherheit von Π_{MAC2}

$\mathsf{Mac-forge}_{\mathcal{A}',\Pi'}(n)$		$\mathcal{A}'$		$\mathcal{A}$
$k \leftarrow \mathsf{Gen}'(1^n)$	$\xrightarrow{1^n}$		$\xrightarrow{1^n}$	Wähle $Q \subset \mathcal{M}$ mit $Q = \{m_1, \ldots, m_q\}$, $m_j = m_j^1, \ldots, m_j^\ell$ und $m_j^i \in \{0,1\}^{\frac{n}{4}}$.
			$\xleftarrow{m_j}$	
	$\xleftarrow{\bar{m}_j^i}$	$r_j \in_R \{0,1\}^{n/4}$ $\bar{m}_j^i = r_j\|\|\ell\|\|i\|\|m_j^i$ für $i = 1, \ldots, \ell$		
$\bar{t}_j^i = \mathsf{Mac}'_k\,(\bar{m}_j^i)$	$\xrightarrow{\bar{t}_j^i}$	$t_j = (r_j, \bar{t}_j^1, \ldots, \bar{t}_j^\ell)$	$\xrightarrow{t_j}$	Berechne $t = (r, t^1, \ldots, t^\ell)$ für ein $m \notin Q$
$Q' = \bigcup_{i,j} \bar{m}_j^i$		Suche ein nicht angefragtes $\bar{m}^i := r\|\|\ell\|\|i\|\|m^i$ mit gültigem Tag t^i in (m,t).	$\xleftarrow{(m,t)}$	
Ausgabe: 1, falls $\mathsf{Vrfy}'_k(\bar{m}^i, t^i) = 1$ und $m^i \notin Q'$	$\xleftarrow{(\bar{m}^i, t^i)}$			

Sicherheit von Π_{MAC2}

Wir definieren die folgenden Ereignisse

- *Forge*: Ein Block $\bar{m}^i = r||\ell||i||m^i$ in (m, t) wurde nicht an $\text{Mac}'_k(\cdot)$ angefragt, aber $\text{Vrfy}'_k(\bar{m}^i, t^i) = 1$.
- *Repeat*: Bei 2 MAC-Anfragen wird dasselbe $r_i = r_j$ verwendet.

Aufgrund der Sicherheit von Π' gilt

$$\begin{aligned}
\mathrm{negl}(n) &\geq \mathrm{Ws}[\text{Mac-forge}_{\mathcal{A}',\Pi'}(n) = 1] \\
&= \mathrm{Ws}[\text{Mac-forge}_{\mathcal{A},\Pi_{MAC2}}(n) = 1 \wedge \mathit{Forge}] \\
&= \epsilon(n) - \mathrm{Ws}[\text{Mac-forge}_{\mathcal{A},\Pi_{MAC2}}(n) = 1 \wedge \overline{\mathit{Forge}}] \\
&= \epsilon(n) - \underbrace{\mathrm{Ws}[\text{Mac-forge}_{\mathcal{A},\Pi_{MAC2}}(n) = 1 \wedge \overline{\mathit{Forge}} \wedge \overline{\mathit{Repeat}}]}_{\epsilon_1} \\
&\quad - \underbrace{\mathrm{Ws}[\text{Mac-forge}_{\mathcal{A},\Pi_{MAC2}}(n) = 1 \wedge \overline{\mathit{Forge}} \wedge \mathit{Repeat}]}_{\leq \mathrm{Ws}[\mathit{Repeat}]}
\end{aligned}$$

Zeigen nun $\epsilon_1 = 0$ und $\mathrm{Ws}[\mathit{Repeat}] \leq \mathrm{negl}(n)$. Damit $\epsilon(n) \leq \mathrm{negl}(n)$.

Sicherheit von Π_{MAC2}

zu zeigen: $\mathrm{Ws}[Repeat] \leq \mathrm{negl}(n)$

- Sei $q(n)$ die Anzahl der MAC-Anfragen von $\mathcal{A}$ an $\mathcal{A}'$.
- Bei der i-ten MAC-Anfrage wähle $\mathcal{A}'$ den Identifikator r_i.
- *Repeat* tritt ein, falls $r_i = r_j$ für ein $i \neq j$. Sei dies Ereignis $E_{i,j}$.
- Nach Geburtstagsparadoxon gilt:

$$\begin{aligned}\mathrm{Ws}[Repeat] = \mathrm{Ws}\Big[\bigcup_{1\leq i<j\leq q(n)} E_{i,j}\Big] &\leq \sum_{1\leq i<j\leq q(n)} \mathrm{Ws}[E_{i,j}] \\ &\leq \frac{q(n)^2}{2^{\frac{n}{4}}} = \mathrm{negl}(n).\end{aligned}$$

Sicherheit von Π_{MAC2}

zu zeigen: $\mathrm{Ws}[\text{Mac-forge}_{\mathcal{A},\Pi_{MAC2}}(n) = 1 \wedge \overline{Forge} \wedge \overline{Repeat}] = 0$

- **Idee:** $\text{Mac-forge}_{\mathcal{A},\Pi_{MAC2}}(n) = 1$ und $\overline{Repeat}$ impliziert *Forge*.
- Sei $(m, t) = (m^1 \dots m^\ell, r, t^1 \dots t^\ell)$ die Ausgabe von $\mathcal{A}$.

Fall 1: Identifikator r unterscheidet sich von allen r_i.

- Dann ist (z.B.) $r||\ell||1||m^1$ nicht-angefragt mit gültigem Tag t^1.

Fall 2: $r = r_i$ für genau ein $i \in [q(n)]$.

- Sei $m_i = m_i^1 \dots m_i^{\ell_i}$ die von $\mathcal{A}$ angefragte Nachricht.
- Fall $\ell \neq \ell_i$: Dann ist $r||\ell||1||m^1$ nicht-angefragt mit gültigem Tag t^1.
- Fall $\ell = \ell_i$: Wegen $m \notin Q$ gilt $m \neq m_i$.
- D.h. es existiert ein j, so dass $m^j \neq m_i^j$.
- Damit wurde $r||\ell||j||m^j$ nicht angefragt. Tag t^j ist dafür gültig.

Fall 3: $r = r_i$ für mehrere $i \in [q(n)]$.

- Fall ist ausgeschlossen, da wegen $\overline{Repeat}$ gilt $r_i \neq r_j$ für alle $i \neq j$.