Ruhr-Universität Bochum
Lehrstuhl für Kryptologie und IT-Sicherheit
Dr. Christopher Wolf
Alexander Meurer, Florian Giesen

RUHR UNIVERSITÄT BOCHUM RUB

Präsenzübungen zur Vorlesung

Kryptographie 2

SS 2010

Blatt 5 / 16. Juni 2010

Erinnerung: Ein *Message Authentication Code (MAC)* ist ein Tupel $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ mit folgenden Eigenschaften:

- $\mathsf{Gen}(1^n)$ erzeugt einen geheimen Schlüssel k.
- $\mathsf{Mac}_k(m)$ erzeugt zu einer Nachricht $m \in \{0,1\}^*$ einen String tag, genannt Tag für m.
- $\mathsf{Vrfy}_k(m, \mathsf{tag})$ gibt ein Bit $b \in \{0,1\}$ aus, je nachdem ob tag ein gültiges Tag für die Nachricht m ist oder nicht.

Ein MAC ist korrekt, wenn für jedes $n \in \mathbb{N}$, $k \leftarrow \mathsf{Gen}(1^n)$ und $m \in \{0,1\}^*$ gilt

$$\mathsf{Vrfy}_k(m, \mathsf{Mac}_k(m)) = 1 \ .$$

Die Sicherheit für einen MAC Π definiert man analog zur Sicherheit für digitale Signaturen durch das folgende Spiel $\mathsf{MACforge}_{\mathcal{A},\Pi}(n)$:

1. Wähle $k \leftarrow \mathsf{Gen}(1^n)$ und sende 1^n an $\mathcal{A}$.
2. $\mathcal{A}$ erhält Orakel-Zugriff auf $\mathsf{Mac}_k(\cdot)$. Sei $\mathcal{Q}$ die Menge aller Orakel-Anfragen von $\mathcal{A}$.
3. $\mathcal{A}$ gibt (m, tag) aus.

Das Experiment gibt folgenden Ausgang aus.

$$\mathsf{MACforge}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \text{falls } \mathsf{Vrfy}_k(m, \mathsf{tag}) = 1 \text{ und } m \notin \mathcal{Q} \\ 0 & \text{sonst} \end{cases}$$

Ein MAC Π heißt ist existentiell unfälschbar gegen adaptive chosen-message Angriffe, kurz sicher, wenn für jeden ppt-Angreifer $\mathcal{A}$ gilt

$$\mathbf{Pr}\left[\mathsf{MACforge}_{\mathcal{A},\Pi}(n) = 1\right] \leq \mathsf{negl}(n) \ .$$

AUFGABE 1. MAC's und Digitale Signaturen.

MACs kann man als symmetrisches Gegenstück zu digitalen Signaturen auffassen. Vergleichen Sie Message Authentication Codes mit Digitalen Signaturen!

AUFGABE 2. Offline/Online-Signaturen.
Digitale Signaturen sind oft relativ "teuer" (in Bezug auf die Rechenzeit). Die Idee von Offline/Online Signaturen besteht darin, den Signierprozeß in zwei Komponenten aufzuteilen. Während einer *Offline*-Phase wird eine Teilsignatur σ_1 vorberechnet, ohne die zu signierende Nachricht zu kennen. Anschliessend wird in der *Online*-Phase aus den Informationen σ_1 und m eine zweite Teilsignatur σ_2 berechnet und zur gesamten Signatur σ zusammengefasst.
Sei nun $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ ein sicherers Signaturschema und sei $\Pi' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vrfy}')$ ein sicheres One-Time Signaturschema. Wir konstruieren daraus ein Offline/Online Schema $(\mathsf{Gen}^{\mathsf{off/on}}, \mathsf{Sign}^{\mathsf{off/on}}, \mathsf{Vrfy}^{\mathsf{off/on}})$ wie folgt:

- $\mathsf{Gen}^{\mathsf{off/on}}(1^n)$ gibt ein Schlüsselpaar gemäß Π aus, d.h. $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^n)$.
- $\mathsf{Sign}^{\mathsf{off/on}}_{\mathsf{sk}}(m)$
  - Offline-Phase: Wähle $(\mathsf{sk}', \mathsf{pk}') \leftarrow \mathsf{Gen}'(1^n)$ und berechne $\sigma_1 = \mathsf{Sign}_{\mathsf{sk}}(\mathsf{pk}')$.
  - Online-Phase: Berechne $\sigma_2 = \mathsf{Sign}'_{\mathsf{sk}'}(m)$.

  Gib die Signatur $\sigma = (\sigma_1, \sigma_2, \mathsf{pk}')$ aus.

a) Geben Sie eine sinnvolle Verifikation $\mathsf{Vrfy}^{\mathsf{off/on}}$ an und begründen Sie die Korrektheit.

b) Beweisen Sie die Sicherheit des Verfahrens *oder* geben Sie einen effizienten Angreifer an.

AUFGABE 3. Random Oracle.
Sei $H : \{0,1\}^{2n} \to \{0,1\}^n$ eine Funktion. Definiere

$$F_k(x) := H(k||x)$$

wobei $|k| = |x| = n$. Zeigen Sie, dass F_k eine Pseudozufallsfunktion ist, wenn H als Random Oracle modelliert ist.