

Voll homomorphe Verschlüsselung

Definition Voll homomorphe Verschlüsselung

Sei Π ein Verschlüsselungsverfahren mit $Enc : R \rightarrow R'$ für Ringe R, R' . Π heißt *voll homomorph*, falls

- 1 $Enc(m_1) + Enc(m_2)$ eine gültige Verschlüsselung von $m_1 + m_2$
 - 2 $Enc(m_1) \cdot Enc(m_2)$ eine gültige Verschlüsselung von $m_1 \cdot m_2$
- für alle $m_1, m_2 \in R$ ist.

Anwendung: Cloud Computing

- Sende verschlüsselt Algorithmus $\mathcal{A}$, Eingabe x an einen Server S .
- S berechnet daraus die verschlüsselte Ausgabe $Enc(\mathcal{A}(x))$.
- Erlaubt Auslagern von Berechnungen an S .
- S lernt nichts über das Programm $\mathcal{A}$ oder die Eingabe x .

Erste voll homomorphe Verschlüsselung:

Gentry Verfahren (2009), basierend auf Problemen der Gittertheorie.

Digitale Signaturen

Funktionsweise von digitalen Signaturen:

- Schlüsselgenerierung erzeugt pk , sk von Alice.
- Signieren ist Funktion von sk .
- Verifikation ist Funktion von pk .

Idee: Es soll unmöglich sein, ein gültiges Paar von Nachricht m mit zugehöriger Signatur σ zu erzeugen, ohne sk zu kennen.

Eigenschaften digitaler Signaturen: Sei σ eine gültige Signatur für m .

- **Integrität:** m kann nicht verändert werden, da man keine gültige Signatur zu einem $m' \neq m$ erstellen kann.
- **Authentizität:** Falls σ eine gültige Signatur zu m ist, so kommt die Signatur von Alice, der Besitzerin von sk .
- **Transferierbarkeit:** Jeder kann die Gültigkeit von (m, σ) überprüfen. Insbesondere kann (m, σ) weitergereicht werden.
- **Nicht-Abstreitbarkeit:** Alice kann nicht behaupten, dass eine andere Person eine gültige Signatur erzeugt hat.

Definition Signaturverfahren

Definition Signaturverfahren

Ein *Signaturverfahren* ist ein 3-Tupel $(Gen, Sign, Vrfy)$ von ppt Alg mit

- 1 **Gen:** $(pk, sk) \leftarrow Gen(1^n)$.
- 2 **Sign:** $\sigma \leftarrow Sign_{sk}(m)$ für $m \in \{0, 1\}^*$.
- 3 **Vrfy:** $Vrfy_{pk}(m, \sigma) = \begin{cases} 1 & \text{falls } \sigma \text{ gültig für } m \text{ ist.} \\ 0 & \text{sonst} \end{cases}$.

Es gilt $Vrfy_{pk}(m, Sign_{sk}(m)) = 1$ für alle $m \in \{0, 1\}^*$.

Unfälschbarkeit von Signaturen

Spiel CMA-Spiel $\text{Forge}_{\mathcal{A},\Pi}(n)$

Sei Π ein Signaturverfahren mit Angreifer $\mathcal{A}$.

- 1 $(pk, sk) \leftarrow \text{Gen}(1^n)$
- 2 $(m, \sigma) \leftarrow \mathcal{A}^{\text{Sign}_{sk}(\cdot)}(pk)$, wobei $\text{Sign}_{sk}(\cdot)$ ein Signierorakel für beliebige Nachrichten $m' \neq m$ ist.
- 3 $\text{Forge}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \text{falls } \text{Vrfy}_{pk}(m, \sigma) = 1, \text{Sign}_{sk}(m) \text{ nicht angefragt} \\ 0 & \text{sonst} \end{cases}$.

Definition CMA-Sicherheit

Sei Π ein Signaturverfahren. Π heißt *existentiell unfälschbar* unter *Chosen Message Angriffen (CMA)*, falls für alle ppt Angreifer $\mathcal{A}$ gilt

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

Wir bezeichnen Π auch abkürzend als *CMA-sicher*.

Unsicherheit von Textbook RSA Signaturen

Algorithmus Textbook RSA Signaturen

- 1 **Gen:** $(N, e, d) \leftarrow \text{GenRSA}(1^n)$. Setze $pk = (N, e)$, $sk = (N, d)$.
- 2 **Sign:** Für $m \in \mathbb{Z}_N$ berechne $\sigma = m^d \bmod N$.
- 3 **Vrfy:** Für $(m, \sigma) \in \mathbb{Z}_N^2$ Ausgabe 1 gdw $\sigma^e \stackrel{?}{=} m \bmod N$.

Unsicherheit: gegenüber CMA-Angriffen

- Wähle beliebiges $\sigma \in \mathbb{Z}_N$. Berechne $m \leftarrow \sigma^e \bmod N$.
- Offenbar ist σ eine gültige Signatur für m .
- Angreifer besitzt keine Kontrolle über m (existentielle Fälschung).

Fälschen einer Signatur für ein gewähltes $m \in \mathbb{Z}_N$:

- Wähle $m_1 \in_R \mathbb{Z}_N^*$ mit $m_1 \neq m$. Berechne $m_2 = \frac{m}{m_1} \bmod N$.
- Lasse m_1, m_2 vom Orakel $\text{Sign}_{sk}(\cdot)$ unterschreiben.
- Seien σ_1, σ_2 die Signaturen. Dann ist
$$\sigma := \sigma_1 \cdot \sigma_2 = m_1^d \cdot m_2^d = (m_1 m_2)^d = m^d \bmod N$$
 gültig für m .

Hashfunktionen und Kollisionen

Definition Hashfunktion

Eine *Hashfunktion* ist ein Paar (Gen, H) von pt Algorithmen mit

- 1 **Gen:** $s \leftarrow Gen(1^n)$. Gen ist probabilistisch.
- 2 **H:** $\{0, 1\}^n \leftarrow H_s(x)$ für alle $x \in \{0, 1\}^*$. H ist deterministisch.

Spiel $HashColl_{\mathcal{A}, \Pi}(n)$

- 1 $s \leftarrow Gen(1^n)$
- 2 $(x, x') \leftarrow \mathcal{A}(s)$
- 3 $HashColl_{\mathcal{A}, \Pi} = \begin{cases} 1 & \text{falls } H_s(x) = H_s(x') \text{ und } x \neq x' \\ 0 & \text{sonst} \end{cases}$.

Definition Kollisionsresistenz

Eine Hashfunktion Π heißt *kollisionsresistent*, falls für alle ppt $\mathcal{A}$ gilt $\mathbb{W}_s[HashColl_{\mathcal{A}, \Pi}(n) = 1] \leq \text{negl}(n)$.

Hashed RSA

Algorithmus Hashed RSA

- ➊ **Gen:** $(N, e, d, H) \leftarrow \text{GenHashRSA}(1^n)$ mit $H : \{0, 1\}^* \rightarrow \mathbb{Z}_N$.
Ausgabe $pk = (N, e, H)$, $sk = (N, d, H)$.
- ➋ **Sign:** Für $m \in \{0, 1\}^*$ berechne $\sigma = H(m)^d \bmod N$.
- ➌ **Vrfy:** Für $(m, \sigma) \in \mathbb{Z}_N^2$ Ausgabe 1 gdw $\sigma^e \stackrel{?}{=} H(m) \bmod N$.

Einfacher Angriff:

- Sei $m_1 \neq m_2$ eine Kollision für H ist, d.h. $H(m_1) = H(m_2)$.
- Frage (m_1, σ) an. Dann ist (m_2, σ) eine gültige Fälschung.
- D.h. wir benötigen für H Kollisionsresistenz.

Anmerkung: Sicherheit gegen unsere Angriffe für Textbook RSA

- ➊ Wähle $\sigma \in \mathbb{Z}_N$, $m' \leftarrow \sigma^e$. Müssen $m \in H^{-1}(m')$ bestimmen.
Übung: Urbildbestimmung ist schwer für kollisionsresistentes H .
- ➋ Für ein $m \in \mathbb{Z}_N^*$ benötigen wir m_1, m_2 mit $H(m) = H(m_1) \cdot H(m_2)$ in $\mathbb{Z}_N$. Scheint Invertierbarkeit von H zu erfordern.

Später: Zeigen CMA-Sicherheit einer Hashed RSA Variante. (im ROM)

Hash-and-Sign Paradigma

Ziel: Signaturen für Nachrichten beliebiger Länge

- Starten mit Signaturverfahren Π für $m \in \{0, 1\}^n$.
- Verwenden Hashfunktion $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$.
- Unterschreiben Hashwerte statt der Nachrichten.

Definition Hash-and-Sign Paradigma

Sei $\Pi = (\text{Gen}, \text{Sign}, \text{Vrfy})$ und $\Pi_H = (\text{Gen}_H, H)$ eine Hashfunktion.

- 1 **Gen'**: $(pk, sk) \leftarrow \text{Gen}(1^n)$, $s \leftarrow \text{Gen}_H(1^n)$.
Ausgabe $pk' = (pk, s)$ und $sk' = (sk, s)$.
- 2 **Sign'**: Für eine Nachricht $m \in \{0, 1\}^*$ berechne
$$\sigma \leftarrow \text{Sign}_{sk}(H_s(m)).$$
- 3 **Vrfy'**: Für eine Nachricht $m \in \{0, 1\}^*$ mit Signatur σ prüfe
$$\text{Vrfy}_{pk}(H_s(m), \sigma) \stackrel{?}{=} 1.$$

Intuition: Fälschung impliziert Fälschung in Π oder Kollision in H .

Sicherheit von Hash-and-Sign

Satz Sicherheit des Hash-and-Sign Paradigmas

Sei Π CMA-sicher und Π_H kollisionsresistent. Dann ist das Hash-and-Sign Signaturverfahren Π' CMA-sicher.

Beweis:

- Sei $\mathcal{A}'$ ein Angreifer für Hash-and-Sign Π' mit Ausgabe (m, σ) .
- Sei Q die Menge der von $\mathcal{A}$ an das Signierorakel $\text{Sign}_{sk}(\cdot)$ gestellten Anfragen. Es gilt $m \notin Q$.
- Sei coll das Ereignis, dass $m_i \in Q$ mit $H_s(m_i) = H_s(m)$.
- Dann gilt $\text{Ws}[\text{Forge}_{\mathcal{A}', \Pi'}(n) = 1]$

$$\begin{aligned} &= \text{Ws}[\text{Forge}_{\mathcal{A}', \Pi'}(n) = 1 \wedge \text{coll}] + \text{Ws}[\text{Forge}_{\mathcal{A}', \Pi'}(n) = 1 \wedge \overline{\text{coll}}] \\ &\leq \text{Ws}[\text{coll}] + \text{Ws}[\text{Forge}_{\mathcal{A}', \Pi'}(n) = 1 \wedge \overline{\text{coll}}] \end{aligned}$$

- Wir zeigen nun, dass beide Summanden vernachlässigbar sind.