

CPA-Sicherheit ist ungenügend

Definition CCA

CCA (=Chosen Ciphertext Attack) ist ein Angriff, bei dem der Angreifer sich Chiffretext seiner Wahl entschlüsseln lassen kann.

Beispiele in denen CPA nicht genügt:

- Eve fängt verschlüsselte Email $c = Enc(m)$ an Bob ab.
- Eve verschickt c selbst an Bob.
- Bob antwortet Eve und hängt dabei m an die Antwort an.
- D.h. Bob fungiert als Entschlüsselungssorakel.
- **Frage:** Lernt Eve Information, um andere c' zu entschlüsseln?

- Alice und Eve nehmen als Bieter an einer Auktion von Bob teil.
- Alice sendet ihr Gebot $c = Enc(m)$ verschlüsselt an Bob.
- Enc soll CPA-sicher sein, d.h. Eve erhält keine Information über m .
- **Frage:** Ist es Eve möglich, $c' = Enc(2m)$ aus c zu berechnen, ohne m zu kennen, und damit Alice zu überbieten? (Malleability)
- Man kann zeigen: CCA-sichere Verschlüsselung ist non-malleable.

CCA Ununterscheidbarkeit

Spiel CCA Ununterscheidbarkeit von Chiffretexten $PubK_{\mathcal{A}, \Pi}^{cca}(n)$

Sei Π ein PK-Verschlüsselungsverfahren mit Angreifer \mathcal{A} .

- 1 $(pk, sk) \leftarrow Gen(1^n)$
- 2 $(m_0, m_1) \leftarrow \mathcal{A}^{Dec_{sk}(\cdot)}(pk)$, wobei $Dec_{sk}(\cdot)$ ein Entschlüsselungs-orakel für \mathcal{A} für beliebige Chiffretexte ist.
- 3 Wähle $b \in_R \{0, 1\}$. Verschlüssele $c \leftarrow Enc_{pk}(m_b)$.
- 4 $b' \leftarrow \mathcal{A}^{Dec_{sk}(\cdot)}(c)$, wobei \mathcal{A} beliebige Chiffretexte $c' \neq c$ durch das Orakel $Dec_{sk}(\cdot)$ entschlüsseln lassen darf.
- 5 $PubK_{\mathcal{A}, \Pi}^{cca}(n) = \begin{cases} 1 & \text{für } b = b' \\ 0 & \text{sonst} \end{cases}$

Anmerkungen:

- Zusätzlich zum Verschlüsselungs-Orakel bei CPA besitzt \mathcal{A} bei CCA ein weiteres Entschlüsselungs-Orakel $Dec_{sk}(\cdot)$.
- Falls \mathcal{A} in Schritt 4 auch c entschlüsseln darf, ist das Spiel trivial.

Definition CCA-Sicherheit

Ein Verschlüsselungsverfahren Π heißt *CCA-sicher* bzw. besitzt *ununterscheidbare Chiffretexte unter CCA*, falls für alle ppt Angreifer \mathcal{A} gilt $\mathbb{W}_s[\text{PubK}_{\mathcal{A},\Pi}^{\text{cca}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n)$.

Anmerkungen:

- Erstes effizientes CCA-sicheres Verfahren Cramer-Shoup (1998).
- Cramer-Shoup verwendet die DDH-Annahme.
- Chiffretexte sind doppelt so lang wie bei ElGamal.
- Sicherheitsbeweis von Cramer-Shoup ist nicht-trivial.
- Später in der Vorlesung: CCA-sichere Verfahren im sogenannten Random Oracle Modell.

CCA Angriff und Malleability von Textbuch RSA

CCA Angriff auf Textbuch RSA

- Wollen $c = m^e \bmod N$ entschlüsseln.
- Man beachte: c darf nicht direkt angefragt werden.
- Berechne $c' = c \cdot r^e = (mr)^e \bmod N$ für $r \in \mathbb{Z}_N^* \setminus \{1\}$.
- Berechne $mr \leftarrow Dec_{sk}(c')$ mittels Entschlüsselungs-Orakel.
- Berechne $mr \cdot r^{-1} = m \bmod N$.

Malleability von Textbuch RSA

- Voriger Angriff zeigt: Für $c = m^e \bmod N$ kann die Verschlüsselung von mr berechnet werden, ohne m selbst zu kennen.
- D.h. Textbuch RSA ist malleable.

CCA Angriff und Malleability von ElGamal

Praktischer CCA-Angriff auf Padded RSA Variante PKCS #1 v1.5

- Bleichenbacher Angriff: Sende adaptiv Chiffretexte an Server.
- Falls die Entschlüsselung nicht das korrekte Format besitzt, sendet der Server eine Fehlermeldung zurück.
- Genügt, um einen beliebigen Chiffretext c zu entschlüsseln.

CCA Angriff auf ElGamal

- Ziel: Entschlüssele $c = (g^y, g^{xy} \cdot m)$.
- Lasse $c' = (g^y, g^{xy} \cdot m \cdot r)$ für $r \in G \setminus \{1\}$ entschlüsseln.
- Berechne $mr \cdot r^{-1} = m$.
- ElGamal ist malleable, da c' korrekte Verschlüsselung von mr .

Einwegfunktionen

Ziel: CPA-sichere Verschlüsselung aus Trapdoor-Einwegpermutation

Später: CCA-sichere Verschlüsselung aus Trapdoor-Einwegperm.

Spiel Invertieren $Invert_{\mathcal{A},f}(n)$

Sei $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ effizient berechenbar, \mathcal{A} ein Invertierer für f .

1 Wähle $x \in_R \{0, 1\}^n$. Berechne $y \leftarrow f(x)$.

2 $x' \leftarrow \mathcal{A}(1^n, y)$

3 $Invert_{\mathcal{A},f}(n) = \begin{cases} 1 & \text{falls } f(x') = y \\ 0 & \text{sonst} \end{cases}$.

Definition Einwegfunktion

Eine Funktion $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ heißt *Einwegfunktion*, falls

1 Es existiert ein deterministischer pt Alg \mathcal{B} mit $f(x) \leftarrow \mathcal{B}(x)$.

2 Für alle ppt Algorithmen \mathcal{A} gilt $\text{Ws}[Invert_{\mathcal{A},f}(n) = 1] \leq \text{negl}(n)$.

Die Faktorisierungsannahme

- **Problem:** Existenz von Einwegfunktionen ist ein offenes Problem.
- Konstruktion unter Komplexitätsannahme (z.B. Faktorisierung)
- Verwenden dazu $(N, p, q) \leftarrow \text{GenModulus}(1^n)$ von RSA.

Spiel Faktorisierungsspiel $\text{Factor}_{\mathcal{A}, \text{GenModulus}}(n)$

1 $(N, p, q) \leftarrow \text{GenModulus}(1^n)$

2 $(p', q') \leftarrow \mathcal{A}(N)$ mit $p', q' > 1$.

3 $\text{Factor}_{\mathcal{A}, \text{GenModulus}}(n) = \begin{cases} 1 & \text{falls } p'q' = N \\ 0 & \text{sonst} \end{cases}$.

Definition Faktorisierungsannahme

Faktorisieren ist hart bezüglich GenModulus falls für alle ppt Algorithmen \mathcal{A} gilt $\text{Ws}[\text{Factor}_{\mathcal{A}, \text{GenModulus}}(n) = 1] \leq \text{negl}(n)$.

Faktorisierungsannahme: Faktorisieren ist hart bezüglich GenModulus .

Konstruktion aus Faktorisierungsannahme

- Sei $p(n)$ ein Polynom, so dass $GenModulus(1^n)$ höchstens $p(n)$ Zufallsbits verwendet.
- OBdA sei $p(n) : \mathbb{N} \rightarrow \mathbb{N}$ monoton wachsend.

Algorithmus FACTOR-ONEWAY f_{FO}

Eingabe: $x \in \{0, 1\}^*$

- 1 Berechne n mit $p(n) \leq |x| < p(n+1)$.
- 2 $(N, p, q) \leftarrow GenModulus(1^n, x)$, wobei $GenModulus$ die Eingabe x als internen Zufallsstring verwendet.

Ausgabe: N

Bemerkung:

- $GenModulus(1^n, x)$ ist deterministisch. (Derandomisierung)

Existenz von Einwegfunktionen

Satz Einweg-Eigenschaft von f_{FO}

Unter der Faktorisierungsannahme ist f_{FO} eine Einwegfunktion.

Beweis:

- f_{FO} ist mittels FACTOR-ONEWAY effizient berechenbar.
- z.z.: Invertierer \mathcal{A} von f_{FO} impliziert Faktorisierer \mathcal{A}' .
- Sei \mathcal{A} ein Invertierer für f_{FO} mit Erfolgsws $\text{Ws}[Invert_{\mathcal{A},f_{FO}}(N) = 1]$.
- Sei $x' \leftarrow \mathcal{A}(N)$ mit $f(x') = N$.
- Berechne die Faktorisierung $(N, p, q) \leftarrow GenModulus(1^n, x')$.
- Unter der Faktorisierungsannahme gilt

$$\text{negl} \geq \text{Ws}[Factor_{\mathcal{A}', GenModulus}(n) = 1] = \text{Ws}[Invert_{\mathcal{A},f_{FO}}(n) = 1].$$