Ruhr-Universität Bochum
Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Alexander Meurer, Felix Heuer

RUHR UNIVERSITÄT BOCHUM RUB

Hausübungen zur Vorlesung

Kryptographie 2

SS 2011

Blatt 4 / 18. Mai 2011 / Abgabe 1. Juni, 08:30 Uhr, Kasten NA 02

Ziel der nächsten drei Aufgaben ist der Beweis des folgenden Theorems.

Theorem 1. Sei $f : \{0,1\}^n \to \{0,1\}^n$ eine Einwegpermutation. Dann ist auch $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$ definiert durch $g(x,r) := (f(x), r)$ eine Einwegpermutation. Ferner ist die Funktion $\mathsf{gl}(x,r) :=< x, r >:= \sum_{i=1}^{n} x_i r_i \bmod 2$ ein Hardcore-Prädikat für g, d.h. für alle ppt-Angreifer $\mathcal{A}$ gilt

$$\mathbf{Pr}_{x,r \in_R \{0,1\}^n} [\mathcal{A}(g(x,r)) = \mathsf{gl}(x,r)] \leq \frac{1}{2} + \mathsf{negl}(n) \ .$$

Die erste Aufgabe dient als Warm-Up und beschäftigt sich mit der ersten Aussage des Theorems.

AUFGABE 1. Warm-Up. (5 Punkte)
Sei $f : \{0,1\}^n \to \{0,1\}^n$ eine Einwegpermutation. Dann ist auch $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$ definiert durch $g(x,r) := (f(x), r)$ eine Einwegpermutation.

Wir untersuchen nun einen Algorithmus Recover, welcher zur Eingabe $f(x)$ und mit Orakelzugriff auf $\mathcal{O}_x$ unter Benutzung des Algorithmus Predict (siehe Präsenzübung) x berechnet.

Algorithmus $\textsc{Recover}^{\mathcal{O}_x}$

Input: $1^n, f(x)$
Output: $y \in \{0,1\}^n$
Parameter: m

01 Wähle $r_i \in_R \{0,1\}^n$ und setze $r = (r_1, \ldots, r_m)$
02 **For all** $\sigma \in \{0,1\}^m$ **do**
03 **For** $k = 1$ **to** n **do**
04 $y_k \leftarrow \textsc{Predict}^{\mathcal{O}_x}(e_k, r, \sigma)$ // e_k ist der k-te Einheitsvektor
05 **End For**
06 $y \leftarrow (y_1, \ldots, y_n)$
07 **If** $f(y) = f(x)$ **then** output y und halte an
08 **End For**

AUFGABE 2. Algorithmenanalyse. (5 Punkte)
Sei $M = 2^m$. Zeigen Sie:

a) Die Laufzeit von RECOVER ist $t_{\text{RECOVER}}(n) = \mathcal{O}(n \cdot M^2 \cdot t_{\mathcal{O}_x}(n))$ wobei $t_{\mathcal{O}_x}(n)$ die Zeit für eine Orakelantwort ist.

b) $\mathbf{Pr}_{r\in\{0,1\}^{n\times m},\sigma\in\{0,1\}^m}\left[\text{RECOVER}^{\mathcal{O}_x}(1^n, f(x)) \neq x\right] \leq \frac{n}{M\varepsilon^2}$.

Hinweis: Untersuchen Sie zunächst, welchen Wert y_k annimmt, falls die Ausgabe von PREDICT korrekt ist, d.h. wenn $\text{PREDICT}(e_k, r, \sigma) = \mathsf{gl}(x, e_k)$ gilt. Überlegen Sie dann, wie sie Aufgabe PÜ 4.3 anwenden können.

c) Geben Sie eine Wahl von m in Abhängigkeit von n an, sodass RECOVER mit Wahrscheinlichkeit $\geq \frac{1}{2}$ das richtige x findet. Begründen Sie, dass für diese Wahl von m die Laufzeit von RECOVER durch

$$t_{\text{RECOVER}}(n) = \mathcal{O}(\mathsf{poly}(n, \varepsilon^{-1}) \cdot t_{\mathcal{O}_x}(n))$$

gegeben ist.

Wir sind nun in der Lage, alle Puzzlestücke zum Beweis von Theorem 1 zusammenzusetzen. Die Grundidee ist, aus einem Prädiktor $\mathcal{A}$ für $\mathsf{gl}(x, r)$ einen Invertierer $\mathcal{A}'$ für $g(x, r) = (f(x), r)$ zu konstruieren. Hierbei verwenden wir $\mathcal{A}$ für festes x als Orakel $\mathcal{O}_x$ in den obigen Konstruktionen, d.h. $\mathcal{O}_x(r) := \mathcal{A}(g(x, r))$ für beliebiges $r \in \{0, 1\}^n$. In der Reduktion haben wir keine Kontrolle über x. Wir müssen somit sicherstellen, dass wir für „genügend viele“ x ein Orakel $\mathcal{O}_x$ erhalten, welches Gleichung (1) vom Präsenzzettel erfüllt. Dieser letzte Schritt ist Inhalt der folgenden Aufgabe.

AUFGABE 3. Puzzlestücke. (5 Punkte)
Nehmen Sie an, es gibt einem Algorithmus $\mathcal{A}$ mit Laufzeit $t_{\mathcal{A}}(n) = \mathsf{poly}(n)$ und ein Polynom $p(n)$ mit

$$\mathbf{Pr}_{x,r\in_R\{0,1\}^n}\left[\mathcal{A}(g(x, r)) = \mathsf{gl}(x, r)\right] \geq \frac{1}{2} + \frac{1}{p(n)} \ .$$

Zeigen Sie:

a) Sei $\mathsf{Good}(n) := \{x \in \{0, 1\}^n \ : \ \mathbf{Pr}_{r\in_R\{0,1\}^n}\left[\mathcal{A}(g(x, r)) = \mathsf{gl}(x, r)\right] \geq \frac{1}{2} + \frac{1}{2p(n)}\}$. Dann gilt

$$|\mathsf{Good}(n)| \geq \frac{2^n}{2p(n)} \ .$$

b) Beweisen Sie Theorem 1. Betrachten Sie hierfür den Invertierer

$$\mathcal{A}'(g(x, r')) := (\text{RECOVER}^{\mathcal{O}_x}(1^n, f(x)), r')$$

mit Orakel $\mathcal{O}_x(r) := \mathcal{A}(g(x, r))$ und $\varepsilon(n) := \frac{1}{p(n)}$. Zeigen Sie, dass

$$\mathbf{Pr}_{x,r'\in_R\{0,1\}^n}\left[\mathcal{A}'(g(x, r')) = (x, r')\right] \geq \frac{1}{4p(n)} > \mathsf{negl}(n)$$

mittels Teil a) und zeigen Sie, dass $\mathcal{A}'$ Laufzeit

$$t_{\mathcal{A}'}(n) = \mathcal{O}(\mathsf{poly}(n) \cdot t_A(n)) = \mathsf{poly}(n)$$

hat. Verwenden Sie in in beiden Fällen die Wahl von m aus Aufgabe 2c).

Wir wollen die Konstruktion eines CPA-sicheren Public-Key Verfahrens im Random Oracle Modell gemäß Folie 74 verallgemeinern, indem wir den durch das Random Oracle H gegebenen Zufallsstring $H(r)$ nicht direkt als One-Time Pad für die zu verschlüsselnde Nachricht benutzen, sondern $H(r)$ als Schlüssel eines symmetrischen Verschlüsselungsverfahrens $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ verwenden. Wir betrachten also folgende modifizierte Konstruktion:

Sei $H : \mathbb{Z}_N^* \to \{0,1\}^{\ell(n)}$ ein Random Oracle und erzeuge GenRSA wie gewohnt ein RSA-Schlüsselpaar (e, d) und Modulus N. Sei Π' wie oben. Konstruiere ein Public-Key Verschlüsselungsverfahren $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ durch

- $\mathsf{Gen}(1^n)$: Berechne $(N, e, d) \leftarrow \mathsf{GenRSA}(1^n)$ und gib $\mathsf{pk} \leftarrow (N, e)$ sowie $\mathsf{sk} \leftarrow (N, d)$ aus.
- $\mathsf{Enc}_{\mathsf{pk}}(m)$: Zur Eingabe $\mathsf{pk} = (N, e)$ und $m \in \{0,1\}^{\ell(n)}$ wähle $r \in_R \mathbb{Z}_N^*$ und berechne $k \leftarrow H(r)$. Gib den Chiffretext

  $$(c_1, c_2) = (r^e \bmod N, \mathsf{Enc}'_k(m))$$

  aus.
- $\mathsf{Dec}_{\mathsf{sk}}((c_1, c_2))$: Zur Eingabe $\mathsf{sk} = (N, d)$ berechne $r \leftarrow c_1^d \bmod N$ und $k \leftarrow H(r)$. Gib dann $m \leftarrow \mathsf{Dec}'_k(c_2)$ aus.

AUFGABE 4. Random Oracle. (5 Punkte)
Es gelte die RSA-Annahme bzgl. GenRSA und es sei Π' ein symmetrisches Verschlüsselungsverfahren welches ununterscheidbare Chiffretexte gegenüber Lauschern hat (siehe Krypto I, Folie 20 für das Sicherheitsspiel und Folie 31 für die Definition). Zeigen Sie, dass Π dann ein CPA-sicheres Public-Key Verfahren ist.