Gregor Leander
30.06.2008

Fakultät für Mathematik
Ruhr-Universität Bochum

Kryptographie II – Lösungsvorschläge zum Übungsblatt 11

Aufgabe 1 *Secret Sharing* 10 Punkte

Zeigen Sie, dass bei Shamir's (t, n)-Secret Sharing Protokoll weniger als t Parteien keinerlei Informationen über das gemeinsame Geheimnis haben.

Lösungsvorschlag: Seien weniger als t Parteien gegeben. Dann gibt es Stützstellen (i, y_i) mit $i \in I$, wobei I eine Teilmenge von $\mathbb{Z}_p$ mit $|I| < t$ ist. Wir behaupten, dass es für jedes mögliche Geheimnis $s' \in \mathbb{Z}_p$ die gleiche Anzahl von Polynomen f gibt mit den Eigenschaften

$$f(i) = y_i \text{ für alle } i \in I$$

und

$$f(0) = s'.$$

Jedes Polynom vom Grad kleiner t ist mit t (oder mehr) Stützstellen eindeutig bestimmt. Wir wählen daher weitere Stützstellen $i \in J$ mit $J \cap I = \emptyset$ und $|J| = t - |I| - 1$ aus. Für jede Wahl von Tupeln $(y_j)_{j \in J}$ gibt es daher ein eindeutig bestimmtes Polynom mit

$$f(i) = y_i \text{ für alle } i \in I$$

und

$$f(j) = y_j \text{ für alle } j \in J$$

und

$$f(0) = s'.$$

Die Anzahl dieser Polynome ist $p^{|J|}$ und unabhängig von s'. Damit haben weniger als t Personen keine Information über das Geheimnis, aus ihrer Sicht sind alle Geheimnisse gleich wahrscheinlich.

Aufgabe 2 *RSA und CCA* 5 Punkte

Zeigen Sie, dass das standard RSA Verfahren nicht sicher gegen Chosen Ciphertext Attacken sicher ist. Sei $PK = (n, e)$ ein öffentlicher RSA Schlüssel und d der entsprechende geheime Exponent. Finden Sie also eine Attacke auf RSA im folgenden Model.

1. Der Angreifer bekommt einen öffentlichen RSA Schlüssel (n, e) und eine Ciphertext $c = m^e \bmod n$.

2. Der Angreifer hat Zugriff auf ein Entschlüsselungs-Orakel und kann sich alle Ciphertexte $c' \neq c$ entschlüsseln lassen. Das Orakel berechnet also $m' = (c')^d \bmod n$

3. Am Ende muss der Angreifer die Nachricht m zu dem Ciphertext c ausgeben.

Lösungsvorschlag: Die Idee ist die Homomorphie von RSA auszunutzen. Wir dürfen im zweiten Schritt alle Ciphertexte außer dem gegebenen c entschlüsseln lassen. Wir wählen nun ein $\alpha \in \mathbb{Z}_n^*$, $\alpha \neq 1$ zufällig und berechnen $c' = \alpha^e c$. Es gilt $c' \neq c$ denn ansonsten wäre $\alpha^e = 1$ und wegen $\gcd(e, \phi(n)) = 1$ auch $\alpha = 1$. Damit liefert das Entschlüsselungs-Orakel bei Eingabe von c' den Wert

$$m' = c'^d = (\alpha^e c)^d = \alpha m.$$

Da wir α kennen, ist es nun leicht m zu berechnen.

Aufgabe 3 *Deterministische Verfahren* 5 Punkte

Zeigen Sie, dass ein deterministisches Public Key Verschlüsselungs-Verfahren niemals semantisch sicher sein kann. Finden Sie also für jedes solche Verfahren einen Angriff im folgenden Model.

1. Der Angreifer wählt zwei (gleich lange) Nachricht m_0 und m_1 und schickt m_0 und m_1 an das Orakel.

2. Das Orakel wählt zufällig ein Bit b und schickt dem Angreifer die Verschlüsselung von m_b.

3. Der Angreifer gibt ein Bit b' aus und gewinnt, wenn $b = b'$ gilt.

Lösungsvorschlag: Das ist ganz einfach: Hier kann man einfach die Nachrichten m_0 und m_1 selbst verschlüsseln (das Verfahren ist ja Public Key!). Da das Verfahren deterministisch ist, reicht nun ein einfacher Vergleich mit dem vom Orakel zurückgegebenen Wert um b immer richtig zu bestimmen. Fazit: Sichere Verfahren müssen randomisiert sein.