



Lösungsblatt 13

Aufgabe 1 (Endliche Körper)

Wir betrachten den Endlichen Körper

$$\mathbb{K} := \mathbb{F}_2[X] / (X^2 + X + 1)\mathbb{F}_2[X]$$

- (a) Da g_1 vom Grad 3 ist und keine Nullstelle in \mathbb{K} hat, ist es auch irreduzibel.
- (b) Ein Beispiel für ein solches Polynom ist $g_2 = (1)Y^2 + (1)Y + (X) \in \mathbb{K}[Y]$.
- (c) Es gilt $\mathbb{K}_1 \approx \mathbb{F}_{2^6}$ und $\mathbb{K}_2 \approx \mathbb{F}_{2^4}$.
- (d) Wir berechnen mit dem erweiterten Euklidischen Algorithmus:

a, b	$(1)Y^3 + (1)Y + (1)$	$(1)Y^2 + (X)$	$(X + 1)Y + (1)$	$(X) + (X)Y$	(X)
q	$(1)Y$	$(1)Y$	$(X)Y$	(X)	(X)
x_i	(1)	0	(1)	$(X)Y$	$(X + 1)Y + (1)$
y_i	0	(1)	$(1)Y$	$(X)Y^2 + (1)$	$(X + 1)Y^2 + (X) + (1)Y$

Da $(X+1)X = 1 \in \mathbb{K}_1$ gilt: $p^{-1} = (X+1) ((X+1)Y^2 + (X) + (1)Y) = (X)Y^2 + (X+1)Y + (1)$.

Aufgabe 2 (Identifikations- und Singnaturverfahren)

Als Identifikationsverfahren haben wir das Fiat-Shamir-Verfahren kennengelernt. Wir wollen die Anzahl der Runden verringern, nach denen wir glauben, daß unser Gegenüber tatsächlich derjenige ist, der er behauptet zu sein. Dazu bilden wir den geheimen Schlüssel aus zwei Quadratwurzeln s_1, s_2 und den öffentlichen Schlüssel aus $(S_1 = s_1^2 \pmod n, S_2 = s_2^2 \pmod n, n)$. Die Faktorisierung von n bleibt allen Parteien unbekannt.

- (a) Wie könnte nun ein Zero-Knowledge-Beweis für die Kenntnis von s_1 und s_2 aussehen? Der Prover wählt $r \in \mathbb{Z}_n$ zufällig und sendet $x = r^2 \pmod n$ an den Verifier. Der Verifier wählt dann $(e_1, e_2) \in \{0, 1\}^2$ und sendet diese Challenge an den Prover. Dieser sendet $y = rs_1^{e_1} s_2^{e_2}$ an den Verifier. Dieser kann nun prüfen, ob

$$y^2 = xS_1^{e_1} S_2^{e_2}$$

Wenn der Verifier aus diesem Protokoll etwas lernen könnte, dann könnte er auch bei dem Fiat-Shamir-Verfahren etwas lernen.

- (b) Ein falscher Beweiser müßte (e_1, e_2) richtig raten, um y richtig zu berechnen, was er nur mit der Wahrscheinlichkeit $1/4$ tun kann.
- (c) Der geheime Schlüssel wird aus $n = 160$ Elementen $s_i, i = 1, \dots, 160$ gebildet. Der öffentliche Schlüssel ist dann $(n, S_1, S_2, \dots, S_{160})$ gebildet, wobei $S_i = s_i^2 \pmod n$. Das Commitment bleibt gleich, die Challenge ist ein 160-Bit-Vector $(e_1, e_2, \dots, e_{160})$ und die Response $y = r \prod_{i=1}^{160} s_i^{e_i}$. Der Verifier prüft dann, ob $y^2 = x \prod_{i=1}^{160} S_i^{e_i}$.
- (d) Zunächst wählt der Signierer eine zufällige Zahl $r \in \mathbb{Z}_n$ und berechnet das Commitment $x = r^2 \pmod n$. Dann berechnet man den 160-Bit Hashwert h von $x||m$, der die Challenge bildet. Ein Betrüger hat nur die Chance 2^{-160} , die Challenge richtig raten zu können. Die Signatur ist dann (x, y) , wobei y die korrekte Response ist.
- (e) Lösung wird nachgereicht.

Aufgabe 3 (Secret-Sharing)

- (a) Wir können das Geheimnis durch eine Geradengleichung darstellen. Wenn wir eine ganzzahlige Steigung wählen, dann kann man das Geheimnis wie gewünscht darstellen. Das Geheimnis ist dann $b = 13$ und die Shares sind $(x, ax + b)$ mit $a \in \mathbb{Z}$.

(b) Wir stellen p dar als $X^2p_2 + Xp_1 + p_0$. Damit gilt für ein x die Gleichung

$$\begin{pmatrix} 1 & x & x^2 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} = p(x).$$

Wir erhalten drei solcher Gleichungen aus den Shares:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 1 \end{pmatrix}$$

Nach dem Hinweis gilt daher:

$$\begin{pmatrix} 3 & -3 & 1 \\ -5/2 & 4 & -3/2 \\ 7 & -1 & 7 \end{pmatrix} \begin{pmatrix} 5 \\ 7 \\ 1 \end{pmatrix} \equiv \begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} \equiv \begin{pmatrix} 8 \\ 1 \\ 9 \end{pmatrix} \pmod{13}$$

Aufgabe 4 (AES – SubBytes)

$$X^{-2} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad X^{-3} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (X^2 + X^3)^{-1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

$$\text{SubBytes}(X^2) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad \text{SubBytes}(X^3) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{SubBytes}(X^3 + X^2) = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Wäre SubBytes eine lineare Funktion müßte unter anderem gelten

$$\text{SubBytes}(04) + \text{SubBytes}(08) = \text{SubBytes}(0C).$$

Wie uns die berechneten Werte zeigen, gilt diese Beziehung nicht.