Übungsblatt 7

Aufgabe 52 ***mündlich***

(a) Betrachten Sie folgende Angriffsmöglichkeit auf DSA: Für ein gegebenes x sei $z \equiv_q (\text{SHA-1}(x))^{-1}$ und $\epsilon \equiv_p \beta^z$. Nehmen Sie an, dass $\gamma, \lambda \in \mathbb{Z}_q^*$ gefunden werden können, sodass

$$\left((\alpha\epsilon^\gamma)^{\lambda^{-1} \bmod q} \bmod p\right) \bmod q = \gamma.$$

Sei $\delta \equiv_q \lambda\text{SHA-1}(x)$. Zeigen Sie, dass (γ, δ) eine gültige Signatur für x ist.

(b) Beschreiben sie eine ähnliche Angriffsmöglichkeit auf ECDSA.

Aufgabe 53 ***mündlich***

Sei E die elliptische Kurve $y^2 \equiv_{127} x^3 + x + 26$ mit $\|E\| = 131$ Elementen. Betrachten Sie ECDSA in E mit $A = (2, 6)$ und $m = 54$.

(a) Berechnen Sie den öffentlichen Schlüssel $B = mA$.

(b) Berechnen Sie die Signatur einer Nachricht x mit SHA-1$(x) = 10$; verwenden Sie die Zufallszahl $z = 75$.

(c) Prüfen Sie die Verifikationsbedingung für die in (b) berechnete Signatur.

Aufgabe 54 ***mündlich***

Was wären die Folgen, wenn man beim ECDSA-Signaturverfahren $\gamma = 0$ oder $\delta = 0$ zulassen würde?

Aufgabe 55 ***mündlich***

Bei der Lamport-Signatur wird ein Dokument $x = x_1 \dots x_n \in \{0,1\}^n$ durch die Folge $(u_{(i,x_i)})_{i=1,\dots,n}$ signiert, d. h. durch x wird die Teilmenge $A_x = \{(i, x_i) \mid i = 1, \dots, n\}$ aus der Indexmenge $A = \{1, \dots, n\} \times \{0, 1\}$ ausgewählt. Eine Familie $\{A_i \subseteq A \mid i \in I\}$ heißt *Spernersystem* über A, falls für alle $i, j \in I$ gilt: $i \neq j \Rightarrow A_i \not\subseteq A_j$.

(a) Zeigen Sie, dass die Spernereigenschaft notwendig für die Sicherheit der Lamport-Signatur ist.

(b) Bestimmen Sie für $B = \{1, \dots, 2m\}$ ein Spernersystem der Größe $\|I\| = \binom{2m}{m}$.

(c) Benutzen Sie das Spernersystem aus Teilaufgabe (b) für die Konstruktion einer Signatur, deren Signaturlänge gegenüber der Lamport-Signatur um ca. 50% verkürzt ist. Beschreiben Sie hierzu den Signieralgorithmus und die Verifikationsbedingung.

Hinweis: Verwenden Sie $\binom{n+1}{k+1} = \binom{n}{k} + \binom{n}{k+1}$, um eine injektive Funktion $f\colon \{0,1\}^n \to I$ anzugeben.

(d) Zeigen Sie, dass kein Spernersystem der Größe $\|I\| > \binom{2m}{m}$ über der Grundmenge $B = \{1, \dots, 2m\}$ existiert.

Aufgabe 56 ***mündlich***

Geben Sie eine Variante der Lamport-Signatur an, bei der mehrere Nachrichten signiert werden können. Die Größe der öffentlichen Schlüssel soll nicht linear in der Anzahl der Nachrichten wachsen, sondern nur vom Sicherheitsparameter abhängen. Beweisen Sie die Fälschungssicherheit Ihrer Konstruktion.

Hinweis: Konstruieren Sie einen Baum, dessen Blätter öffentliche Lamport-Schlüssel und dessen innere Knoten einen Hashwert über ihre Kinder enthalten.

Aufgabe 57 ***mündlich***

Ein wesentlicher Nachteil des Lamport-Signaturverfahrens ist die Größe der Schlüssel. In Aufgabe 56 wurde gezeigt, wie die Größe der öffentlichen Schlüssel durch Einsatz einer Hashfunktion reduziert werden kann. Zeigen Sie, wie auch die privaten Schlüssel verkleinert werden können. Verwenden Sie hierfür einen Pseudozufallsgenerator.

Aufgabe 58 ***10 Punkte***

Bei der Verifikation einer Signatur im ElGamal-Signaturverfahren (oder einer seiner Varianten) ist es nötig, einen Wert der Form $\alpha^c\beta^d$ zu berechnen. Wenn c und d zufällige ℓ-Bit-Exponenten sind, würde die naheliegende Implementierung durch wiederholtes Quadrieren und Multiplizieren (im Durchschnitt) jeweils $\ell/2$ Multiplikationen und ℓ Quadrierungen benötigen. Das Ziel dieser Aufgabe ist es, $\alpha^c\beta^d$ effizienter zu berechnen.

(a) Beschreiben Sie eine Variante des wiederholten Quadrierens und Multiplizierens, bei der in jeder Iteration höchstens eine Multiplikation nötig ist, wenn $\alpha\beta$ schon im Voraus berechnet wurde.

(b) Sei $c = 26$ und $d = 17$. Zeigen Sie, wie Ihr Algorithmus $\alpha^c\beta^d$ berechnet, indem Sie für jede Runde die Exponenten i und j des Zwischenergebnisses $z = \alpha^i\beta^j$ angeben.

(c) Weisen Sie nach, warum Ihr Algorithmus im Durchschnitt ℓ Quadrierungen und $3\ell/4$ Multiplikationen benötigt, wenn c und d zufällige ℓ-Bit-Zahlen sind.

(d) Schätzen Sie den Geschwindigkeitsgewinn im Vergleich zum ursprünglichen Algorithmus ab, bei dem α^c und β^d unabhängig voneinander berechnet und am Schluss multipliziert werden. Nehmen Sie an, dass Quadrieren und Multiplizieren ungefähr gleich viel Zeit braucht.