Kryptologie 2 Sommersemester 2010
Johannes Köbler, Sebastian Kuhnert 23. Juni 2010

Übungsblatt 9

Aufgabe 67 ***mündlich***

Sei E die elliptische Kurve $y^2 \equiv_{127} x^3 + x + 26$. Da $\|E\| = 131$ prim ist, sind alle nichtneutralen Elemente Erzeuger der durch E definierten additiven Gruppe. 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 $k = 75$.

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

Aufgabe 68 ***mündlich***

Angenommen, Alice verwendet das ElGamal-Signaturverfahren und möchte bei der Berechnung der beim Signieren verwendeten Zufallszahlen Zeit sparen, indem sie ein k_0 wählt und die i-te Nachricht unter Verwendung von $k_i \equiv_{p-1} k_0 + 2i$ signiert. (Es gilt also $k_i \equiv_{p-1} k_{i-1} + 2$.)

(a) Zeigen Sie, wie Bob bei Kenntnis von zwei aufeinander folgenden signierten Nachrichten $(x_i, sig(x_i, k_i))$ und $(x_{i+1}, sig(x_{i+1}, k_{i+1}))$ den privaten Schlüssel a berechnen kann, ohne einen diskreten Logarithmus zu berechnen.
Bemerkung: Für diesen Angriff muss der Wert von i nicht bekannt sein.

(b) Führen sie den Angriff durch, wenn Bob die Werte $p = 28703$, $\alpha = 5$, $\beta = 11339$, $x_i = 12000$, $sig(x_i, k_i) = (26530, 19862)$, $x_{i+1} = 24567$ und $sig(x_{i+1}, k_{i+1}) = (3081, 7604)$ kennt.

Aufgabe 69 ***mündlich***

Wir haben gesehen, dass das ElGamal-Signaturverfahren gebrochen werden kann, wenn die gleiche Zufallszahl k mehrmals verwendet wird. Zeigen Sie, wie ähnliche Angriffe auf das Schnorr-Signaturverfahren, DSA und ECDSA möglich sind.

Aufgabe 70 ***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

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

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

(b) Beschreiben sie eine ähnliche Angriffsmöglichkeit auf ECDSA.

Aufgabe 71 ***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 72 ***mündlich***

Benutzen Sie das Chaum-van-Antwerpen-Verfahren mit den Parametern $p = 467$, $\alpha = 4$, $a = 101$ und $\beta = 449$, um eine verbindliche digitale Signatur für das Dokument $x = 64$ zu erzeugen. Zeigen Sie, wie Alice mit Hilfe des Abstreitungsprotokolls Bob davon überzeugen kann, dass eine ihr vorgelegte Signatur $y = 25$ für das Dokument $x = 157$ gefälscht ist (unter der Annahme, dass Bob die Zufallszahlen $e_1 = 46$, $f_1 = 123$, $e_2 = 198$ und $f_2 = 11$ benutzt).

Aufgabe 73 ***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.