Vorlesung am 7.5.2015

Digital Signature Algorithm (DSA)

Sicherheit beruht auf der Schwierigkeit des Diskreten Logarithmusproblems

Schlüsselgenerierung

1. Wähle zwei Primzahlen p und q mit q teilt $p-1$
2. Wähle x in $\mathbb{Z}_p^*$ und berechne $g := x^{(p-1)/q} \bmod p$.
3. Falls $g = 1$, gehe zu 2.
4. Wähle eine Zahl $a \in \{1, \ldots, q-1\}$ und setze $A := g^a$.

(p, q, g, A) : öffentlicher Schlüssel, a : geheimer Schlüssel.

Bemerkung. (i) Wahl von g: $\mathbb{Z}_p^*(g) := \{g^1, g^2, ...\}$ hat genau q Elemente
Übung: Zeigen Sie diese Aussage

(ii) Aus (p, q, g, A) lässt sich a mittels Logarithmus bestimmen ($a = \log_g A$)
Zwei Möglichkeiten, Logarithmus zu berechnen:
In $\mathbb{Z}_p^*$: Laufzeit $\mathcal{O}(e^{\sqrt{\ln p \ln \ln p}})$
In $\mathbb{Z}_p^*(g)$: Laufzeit $\mathcal{O}(\sqrt{q})$
Übung Wie groß müssen p, q sein, um Sicherheitsniveau 100 Bit zu erhalten?

Signaturerzeugung Signatur (s_1, s_2) von m (genauer $h = H(m)$)

1. Wähle eine zufällige Zahl $1 < s < q$
2. Berechne $s_1 = (g^s \bmod p) \bmod q$ und $s_2 = s^{-1}(h + s_1 \cdot a) \bmod q$

Signaturverifikation

1. Prüfe, ob $0 < s_1, s_2 < q$ gilt.
2. Berechne $w = s_2^{-1} \bmod q$, $u_1 = h \cdot w \bmod q$, $u_2 = s_1 \cdot w \bmod q$
3. Berechne $v = (g^{u_1} \cdot A^{u_2} \bmod p) \bmod q$

4. Ist $v = s_1$, so akzeptiere die Signatur.

Satz 6.1. *Es gilt:* (s_1, s_2) *ist korrekte Signatur genau dann, wenn* $v = s_1$.

Beweis.

- Ist (s_1, s_2) korrekt, dann gilt $s_2 = s^{-1}(h + s_1 \cdot a) \bmod q$.
- Multiplikation mit $sw \bmod q$ ergibt $s_2 sw = (hw + s_1 aw) \bmod q$.
- Mit $w = s_2^{-1} \bmod q$ also $s = (hw + s_1 aw) \bmod q$.
- Wegen $u_1 = h \cdot w \bmod q$, $u_2 = s_1 \cdot w \bmod q$ also $s = u_1 + u_2 a \bmod q$.
- Es existiert also $n \in \mathbb{N}$ mit $s + nq = u_1 + u_2 a$.
- Weiter gilt $g^q = (x^{(p-1)/q})^q = x^{p-1} = 1 \bmod p$ (Satz von Euler).
- Es folgt $g^s = g^{s+nq} = g^{u_1+u_2 a} = g^{u_1}(g^a)^{u_2} = g^{u_1} A^{u_2} \bmod p$.
- Daraus folgt nun $s_1 = (g^s \bmod p) \bmod q = (g^{u_1} \cdot A^{u_2} \bmod p) \bmod q = v$.

□

Bemerkung. DSA ist ein probabilistischer Algorithmus
Für jede Signatur wird ein Zufall $s < q$ genutzt

Nebenbedingungen für s:

(i) s muss geheim gehalten werden, sonst lässt sich a berechnen:

$$a = (\underbrace{(s_2 \cdot s)}_{= h(m) + s_1 \cdot a \bmod q} - h(m)) \cdot s_1^{-1} \bmod q$$

(ii) Entropie (Unvorhersagbarkeit) von s muss 100 Bit groß sein
sonst lässt sich a mit Wkeit $< 1/2^{100}$ bestimmen (durchprobieren von s)

(iii) Für jede Signatur muss ein anderer Wert s genutzt werden:

- Seien zwei Nachrichten m, m' unter Nutzung von s signiert mit zugehörigen Signaturen (s_1, s_2) und (s_1', s_2')
- Dann gilt $s_1 = s_1'$, $s_2 = s^{-1}(m + s_1 a) \bmod q$ und $s_2' = s^{-1}(m' + s_1' a) \bmod q = s^{-1}(m' + s_1 a) \bmod q$.
- Also $s_2 - s_2' = s^{-1}(m - m') \bmod q$. d.h. $s = (m - m')(s_2 - s_2')^{-1}$.
- s lässt sich also ausrechnen und damit auch a (siehe (ii))

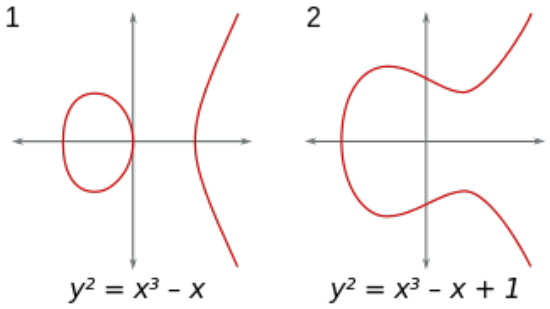
Elliptische Kurven Kryptographie (ECC)

Elgamal und DSA: Kryptographisch starke Gruppen (Logarithmus ist schwer)
Bisher kennen wir nur die (multiplikativen) Gruppen $\mathbb{Z}_p^*$.

Elliptische Kurve: Lösungsmenge der Gleichung $y^2 = x^3 + ax + b$ über Körper
Beispiel: $E = \{(x, y) \in \mathbb{Z}_p; y^2 = x^3 + ax + b\}$ (über Körper $(\mathbb{Z}_p, +, \cdot)$).

Wir betrachten im Folgenden elliptische Kurven über $\mathbb{R}$:

- Form wird durch Diskriminante des Polynoms $x^3 + ax + b$ festgelegt ($\Delta_E = -4a^3 - 27b^2$)
- Für $\Delta_E < 0$ besteht die Kurve aus 2 Komponenten
- Für $\Delta_E > 0$ aus einer Komponente (siehe Abbildung)



- Kurven mit Diskriminante 0 können nicht genutzt werden (siehe unten)

Rechnen in E:

- Wir benötigen zusätzlichen Punkt $O \notin E$, $\bar{E} := E \cup \{O\}$
  O ist neutrales Element in $\bar{E}$

- Für $P, Q \in \bar{E}$ mit $P = (x_P, y_P)$ und $Q = (x_Q, y_Q)$ sei

$$P{+}Q := \begin{cases} P, & \text{falls } Q = O \\ Q, & \text{falls } P = O \\ O, & \text{falls } x_P = x_Q \text{ und } y_P = -y_Q \text{ (d.h. } Q = -P) \\ O, & \text{falls } P = Q \text{ und } y_P = 0 \\ & \\ x_R = s^2 - 2x_P, & \text{falls } P = Q \text{ und } y_P \neq 0, \text{ mit} \\ y_R = -y_P + s(x_P - x_R) & s = (3x_P^2 + a)/2y_P \\ & \\ x_R = s^2 - x_P - x_Q, & \text{falls } x_P \neq x_Q, \text{ mit} \\ y_R = -y_P + s(x_P - x_R) & s = (y_P - y_Q)/(x_P - x_Q) \end{cases}$$

- Für eine geomtrische Interpretation siehe Tafelbild
- $(\bar{E}, +)$ ist eine abelsche Gruppe

Für Anwendungen in der Kryptographie: Kurven über $\mathbb{Z}_p$
Diese sind unter betimmten Voraussetzungen kr. stark, d.h.

Diskretes Logarithmusproblem (in $\bar{E}$):
Gegeben: Gegeben G und $n \cdot G = \underbrace{G + \cdots + G}_{n-mal}$.
Lösung: Finde n.

Dieses Problem ist in elliptischen Gruppen schwerer als DL in $\mathbb{Z}_p$
Bester derzeit bekannter Algorithmus hat Laufzeit $\mathcal{O}(\sqrt{p})$.

Also: Für Sicherheitsniveau 100 Bit muss $p \approx 2^{200}$ gelten.
Auf ell. Kurven basierende Kryptographie benötigt deutlich kürzere Schlüssellängen.

Übung: Studieren Sie den Signaturalgorithmus ECDSA. Stellen Sie insb. Schlüsselerzeugung, Signaturerzeugung und -verifikation dar. An welchen Stellen geht für die Sicherheit die Schwere des DL-Problems ein? Welche Bedingung muss für G (in der obigen Formulierung des Problems) gelten, damit das Problem tatsächlich schwer ist?