- Alice berechnet den Hashwert zu der Nachricht x Sig 6
- Alice berechnet $S = h(x)^d \bmod n$
- Sie schickt das signierte Dokument (x, S) an Bob.
- Bob überprüft: $S^e = h(x) \bmod n$. Falls das stimmt, dann weiß er, die Nachricht kommt von Eve.
- Jeder kann die Signatur von Alice überprüfen.

10.4. Die ElGamal-Signatur

Sei p eine große Primzahl, $\langle \alpha \rangle = \mathbb{Z}_p^*$ und $\beta = \alpha^a \bmod p$ für ein $a \in \{2, \dots, p-2\}$. a ist der geheime Schlüssel und (p, α, β) der öffentliche. Sie will die Nachricht x ~~ein~~ $\mathbb{Z}_p = \mathcal{P}$ unterschreiben.

Dafür wählt sie ein zufälliges $1 \leq k \leq p-2$ mit $ggT(k, p-1) = 1$. und signiert durch

$$u_k : \mathcal{P} = \mathbb{Z}_p \to \mathcal{U} = \mathbb{Z}_p \times \mathbb{Z}_{p-1}$$

$$x \mapsto (u_1, u_2) \quad \text{mit}$$

$u_1 \equiv \alpha^k \bmod p$ ~~und~~, $0 \leq u_1 < p$, und

$u_2 \equiv (x - a u_1) k^{-1} \bmod (p-1)$.

~~Die öffentliche~~

Sie schickt (x, u_1, u_2).

Die öffentliche Verifikation ist

$$v(x, u_1, u_2) = \begin{cases} \text{wahr} & \beta^{u_1} u_1^{u_2} \equiv \alpha^{x} \bmod p \\ \text{falsch} & \text{sonst.} \end{cases}$$

Verifikation

Ist $(u_1, u_2) = u_k(x)$, dann folgt

$$\beta^{u_1} u_1^{u_2} \equiv \alpha^{a u_1} \alpha^{k(x - a u_1) k^{-1}} \equiv \alpha^{x} \bmod p$$

Also gilt $v(x, u_k(x))$ ~~ist~~ $\equiv$ wahr.

Ist $v(x, u_1, u_2) =$ falsch, dann folgt also

$u_k(x) \neq (u_1, u_2)$.

Beispiel ← Sig 7b

Sicherheit

- k muß geheim bleiben (wie bei ElGamal).
  Ansonsten kann Oskar aus x und (u_1, u_2) den geheimen Schlüssel a von Alice bestimmen:

  $$a \equiv (x - k u_2)\, u_1^{-1} \bmod (p-1).$$

- Alice muß k jedesmal neu wählen. Benutzt sie dasselbe k 2x, dann kann Oskar k herausfinden. (siehe Lösungszettel)

Beispiel

Sei $p = 19$, Dann ist $\langle 2 \rangle = \mathbb{Z}_{19}^{*}$

Wähle $a = 13$. Dann ist $\beta = \alpha^{a} = 2^{13} \equiv 3 \bmod 19$.

~~und~~ Also ist der private Schlüssel 13

und der öffentliche $(19, 2, 3)$.

Die Nachricht sei $x = 5 \in \mathbb{Z}_{19}$ und $k = 7$.

Dann ist $u_1 = \alpha^{k} = 2^{7} \equiv 14 \bmod 19$

und $u_2 = (x - a u_1) k^{-1} = (5 - 13 \cdot 14) 7^{-1} \equiv 3 \cdot 7^{-1}$

$\equiv 3 \cdot 13 \equiv 3 \bmod 18$

Also sendet Alice $(5, 14, 3)$

- Falls x in $\mathbb{Z}_{p-1}^{*}$ invertierbar ist, d.h. $ggT(x, p-1) = 1$, dann kann der Angreifer Oskar aus der Signatur $s = u_a(x)$ eine gültige Signatur einer Nachricht $\tilde{x}$ erstellen:

  - Oskar berechnet $U \equiv \tilde{x}\, x^{-1} \mod (p-1)$
  - Oskar berechnet $\tilde{u}_2 :\equiv u_2\, U \mod (p-1)$

  $u_a(x) = (u_1, u_2)$.

  - Oskar berechnet mit dem chinesischen Restsatz (Beachte $ggT(p, p-1) = 1$) eine Lösung $\tilde{u}_1$ der Kongruenzen

$$X \equiv u_1 U \mod (p-1)$$
$$X \equiv u_1 \mod p.$$

  - Dann ist $(\tilde{u}_1, \tilde{u}_2)$ eine gültige Signatur von $\tilde{x}$.

Verifikation:

$$\beta^{\tilde{u}_1}\, \tilde{u}_1^{\tilde{u}_2} = \alpha^{a u_1 U}\, \tilde{u}_1^{u_2 U} \equiv \alpha^{a u_1 U}\, u_1^{u_2 U}$$

$$\equiv \alpha^{a u_1 U}\, \alpha^{k u_2 U} = \alpha^{(a u_1 + k u_2) U} \underset{a u_1 + k u_2 \equiv x \mod p-1}{\equiv} \alpha^{x U} = \alpha^{\tilde{x}} \mod p.$$

Beispiel: Wähle das obige Beispiel und $\tilde{x} = 14$. Dann ist $u = \tilde{x} x^{-1} = 14 \cdot 5^{-1} \equiv 14 \cdot 11 \equiv 10$ mod 18. Also $u = 10$

$\tilde{u}_2 \equiv u_2 u \mod p-1$, d.h. $\tilde{u}_2 \equiv 3 \cdot 10 \equiv 12 \mod 18$

$\tilde{u}_2 = 12$

$\tilde{u}_1 \equiv 14 \cdot 10 \mod 18$ und

$\tilde{u}_1 \equiv 14 \mod 19$

$\Rightarrow \tilde{u}_1 = 14$

Die Signatur von $\tilde{x} = 14$ ist $(\tilde{u}_1, \tilde{u}_2) = (14, 12)$.

RSA beruht darauf, dass es sehr schwer ist natürliche Zahlen zu faktorisieren, d.h. gegeben $n = p \cdot q$, p, q zwei verschiedene Primzahlen, p und q herauszufinden.

ElGamal beruht darauf, dass es sehr schwer ist für $g \in G$, G Gruppe, $o(g)$ groß, aus g^a auf den Exponenten a zu schließen. Diese Fragestellung wollen wir nun genauer anschauen.

19. Der diskrete Logarithmus

In diesem Paragraphen wollen wir Algorithmen vorstellen zur Lösung des diskreten Logarithmus.

Sei G eine Gruppe, $g \in G$, $o(g) = n$ und $h = g^x$ mit $x \in \{0, \dots, n-1\}$ ($\Rightarrow$ x ist durch h eindeutig bestimmt!)

Das <u>diskrete Logarithmus Problem</u> ist das Problem

$$x = \log_g h$$ zu bestimmen.

19.1 Enumeration

Ist n klein, dann können wir $g, g^2, g^3 \dots$ bestimmen und abgleichen, wann $g^j = h$ gilt. Dann ist $j = x$. Aber das Verfahren benötigt $x-1$ Multiplikationen und x Vergleiche. In kryptographischen Verfahren ist $x > 2^{160}$. Daher ist dieses Verfahren Enumeration nicht praktikabel.

10.2 Baby-Step-Giant-Step von Shanks

Sei $m = \lceil \sqrt{n} \rceil$. Wegen des euklidischen Algorithmus

~~Baby-Step Be~~ (Division mit Rest) gibt es

$j, r \in \mathbb{N}$ mit $x = jm + r$ und $0 \leq r \leq m-1$.

Der Algorithmus berechnet j und r

Zuerst wird r bestimmt:

Baby-Step Bestimme die Menge

$$B = \{ (hg^{-r}, r) \mid 0 \leq r \leq m-1 \}$$

und speichere die Elemente aus B ab.

Ist $(\cancel{hg}, r) \in B \Rightarrow h = g^r$ und $x = r$.

Sonst fahre fort mit

Giant-Step Für $j = 1, .., m-1$ suche in B nach einem

Paar (hg^{-r}, r) so, dass $hg^{-r} = g^{mj}$.

Wenn dies gefunden wurde, dann ~~ist~~ setze $x = mj + r$.

Der Algorithmus terminiert:

Es ist $x = jm + r \leq n-1 \Rightarrow j \leq m-1$

$$\Rightarrow hg^{-r} = g^x g^{-r} = g^{mj+r-r} = g^{mj}$$