Week 12

Digital Signature Schemes

Def^n: A signature scheme is a five tuple $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$ where the following conditions are satisfied:

(i) $\mathcal{P}$ is a finite set of possible messages

(ii) $\mathcal{A}$ is a finite set of possible signatures

(iii) $\mathcal{K}$, the keyspace, is a finite set of possible keys

(iv) For each $K \in \mathcal{K}$, there is a signing algm. $sig_K \in \mathcal{S}$ and a corresponding verification algm. $ver_K \in \mathcal{V}$.

Each $sig_K : \mathcal{P} \to \mathcal{A}$ and $ver_K : \mathcal{P} \times \mathcal{A} \to \{true, false\}$ are functions such that the following equation is satisfied for every message $x \in \mathcal{P}$ and for every signature $y \in \mathcal{A}$:

$$ver_K(x, y) = \begin{cases} true & \text{if } y = sig_K(x) \\ false & \text{if } y \neq sig_K(x). \end{cases}$$

- For every $K \in \mathcal{K}$, the functions sig_K and ver_K should be polynomial-time functions

$x \to$ message

$y \to$ signature

- ver_K will be a public fu^n.
- sig_K will be secret.

A signature scheme is not unconditional. ← Oscar can go for brute force attack by computing all possible signature on x using public fu^n ver & find the right signature

- Goal: design computationally secure signature schemes.

RSA Signature Scheme

- Let $n = pq$, where p & q are primes.
- Let $\mathcal{P} = \mathcal{A} = \mathbb{Z}_n$, and define

$$\mathcal{K} = \{ (n, p, q, a, b) : n = pq,\ p, q \text{ prime},\ ab \equiv 1 \pmod{\phi(n)} \}.$$

- n, b are public, p, q, a are secret.

For $K = (n, p, q, a, b)$, define

$$\mathrm{Sig}_K(x) = x^a \bmod n$$

and

$$\mathrm{ver}_K(x, y) = \text{true} \iff x = y^b \pmod{n}.$$

$(x, y \in \mathbb{Z}_n)$

Oscar

Bob.

Alice

x

z, y

$z = \mathcal{E}_{Bob}(x)$.

$y = \mathrm{Sig}_{Alice}(z)$

z, y'

$y' = \mathrm{Sig}_{Oscar}(z)$

Bob may infer that the plaintext x is originated with Oscar

- Signing before encrypting.

(ii) Oscar first chooses δ & then tries to find γ. Then he has to solve the eqn.

$$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} \pmod{p}.$$

for the unknown γ.

- The random value k used in computing a signature should not be revealed.

$$sig_K(x) = (\gamma, \delta) \qquad \gamma = \alpha^{k}$$
$$\delta = (x - a\gamma)k^{-1}$$
$$\Rightarrow a = (x - k\delta)\gamma^{-1}.$$

k known $\Rightarrow$ a known $\Rightarrow$ system is broken.
(Oscar can forge signatures at will)

- Same value of k in signing two different messages ~~should not be used~~ makes ~~the system~~ it easy for Oscar to compute a & hence break the system.

$x_1, x_2 \rightarrow$ two different messages

$$sig_K(x_1) = (\gamma, \delta_1), \quad \gamma = \alpha^{k}, \quad \delta_1 = (x_1 - a\gamma)k^{-1}$$
$$sig_K(x_2) = (\gamma, \delta_2), \quad \gamma = \alpha^{k}, \quad \delta_2 = (x_2 - a\gamma)k^{-1}.$$

$$\left.\begin{aligned} \beta^{\gamma}\gamma^{\delta_1} &\equiv \alpha^{x_1} \pmod{p} \\ \beta^{\gamma}\gamma^{\delta_2} &\equiv \alpha^{x_2} \pmod{p} \end{aligned}\right\} \Rightarrow \alpha^{x_1 - x_2} \equiv \gamma^{\delta_1 - \delta_2} \pmod{p}$$
$$\Rightarrow x_1 - x_2 \equiv k(\delta_1 - \delta_2) \pmod{p-1}$$

ElGamal Signature Scheme

- p be a prime s.t. DLP in Z_p is intractable,
- $\alpha \in Z_p^*$ be a primitive element.
- $P = Z_p^*$, $A = Z_p^* \times Z_{p-1}$, & define

$$K = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod p\}$$

- p, α, β are public, a is secret.
- for $K = (p, \alpha, a, \beta)$, & for a (secret) random $k \in Z_{p-1}^*$,

define $\quad Sig_K(x, k) = (\gamma, \delta)$

where
$$\gamma = \alpha^k$$
$$\delta = (x - a\gamma)k^{-1} \bmod (p-1)$$

- for $x, \gamma \in Z_p^*$ and $\delta \in Z_{p-1}$, define

$$Ver_K(x, \gamma, \delta) = \text{true} \iff \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p.$$

$$\beta^\gamma \gamma^\delta = \alpha^{a\gamma}(\alpha^k)^{(x-a\gamma)k^{-1}} = \alpha^{a\gamma} \cdot \alpha^{x-a\gamma} = \alpha^x \pmod p$$

Security: Oscar tries to forge a signature (γ, δ) for a given message x, without knowing a.

(i) Oscar chooses γ & then tries to find the corresponding δ.

$$\beta^\gamma \gamma^\delta \equiv \alpha^x \bmod p \Rightarrow \gamma^\delta \equiv \alpha^x \beta^{-\gamma} \bmod p$$

i.e. $\delta = \log_\gamma \alpha^x \beta^{-\gamma}$

∴ Oscar must compute discrete logarithm $\log_\gamma \alpha^x \beta^{-\gamma}$

$x_1 - x_2 \equiv k(\delta_1 - \delta_2) \pmod{p-1}$

~~$b \equiv ax \pmod{m}$, $d = \gcd(a, m)$, $d \mid b$.~~

Let $d = \gcd(\delta_1 - \delta_2, p-1)$.

$d \mid x_1 - x_2 \rightarrow$ otherwise no soln. for k.

$\frac{x_1 - x_2}{d} \equiv k \frac{\delta_1 - \delta_2}{d} \pmod{\left(\frac{p-1}{d}\right)}$.

$\Downarrow$

Solve this linear congruence.

$x' \equiv k\delta' \pmod{p'}$.

$\gcd(\delta', p') = 1$

$\therefore \varepsilon = (\delta')^{-1} \bmod p'$ exists.

$x' = \frac{x_1 - x_2}{d}$

$\delta' = \frac{\delta_1 - \delta_2}{d}$

$p' = \frac{p-1}{d}$.

$\therefore k$ is determined modulo p' as

$k = x'\varepsilon \bmod p'$

we get d candidate values for k:

$k = x'\varepsilon + ip' \pmod{p-1}$

for some i, $0 \le i \le d-1$.

$\downarrow$

Find the correct k by testing the condition

$\gamma \equiv \alpha^k \pmod{p}$

k known $\Rightarrow$ a known $\Rightarrow$ system is broken.

$ax \equiv b \pmod{c}$

$d = \gcd(a, c)$

$d \mid b$. soln.

$\frac{a}{d} x \equiv \frac{b}{d} \pmod{\frac{c}{d}}$

$\downarrow$ soln.

x_0

soln. $x_0 + i\frac{c}{d} \bmod c$

$0 \le i \le d$

The Digital Signature Standard (6)

↓ Shorter signature to implement in smart card

$P = Z_p^*$ $\quad p \to 512$ bits

$A = Z_p^* \times Z_{p-1}$.

sig $\to 1024$ bits

- Let p be a 512-bit prime such that DLP in Z_p is intractable,
- q be a 160-bit prime that divides $p-1$.
- Let $\alpha \in Z_p^*$ be a q-th root of 1 modulo p.
- $P = Z_q^*$, $\quad A = Z_q \times Z_q$, and define

$$K = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod p\}.$$

- p, q, α, β are public, a is secret $\alpha < q$

for $K = (p, q, \alpha, a, \beta)$ and for a (secret) random number k, $1 \le k \le q-1$, define

$\alpha = \sqrt[q]{1} \bmod p$

$\alpha^q = 1 \bmod p$

$$\mathrm{Sig}_K(x, k) = (\gamma, \delta)$$

when $\gamma = (\alpha^k \bmod p) \bmod q$.

& $\delta = (x + a\gamma)k^{-1} \bmod q$.

for $x \in Z_q^*$ and $\gamma, \delta \in Z_q$, verification is done by performing the following computations:

$e_1 = x\delta^{-1} \bmod q$

$e_2 = \gamma\delta^{-1} \bmod q$

$\mathrm{Ver}_K(x, \gamma, \delta) = \text{true} \iff (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma$.

$\alpha^{x\delta^{-1}} \alpha^{a\gamma\delta^{-1}} = \alpha^{(x+a\gamma)\delta^{-1}} = \alpha^k$