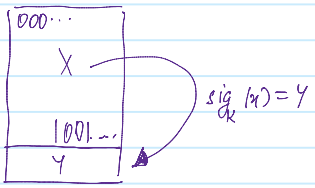
Digital Signatures

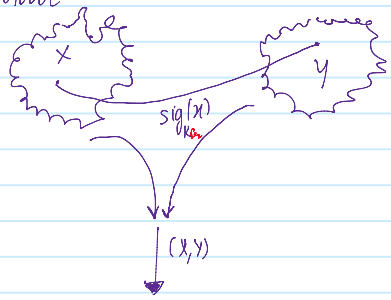
Sunday, 8 December 2019 12:22 AM

INTRODUCTION TO DIGITAL SIGNATURES:

Signature like function for the electronic world.
↳ proof of authenticity of the sender.



protocol:



SECURITY SERVICES

The objectives of a security system are called SECURITY SERVICES.

(1) CONFIDENTIALITY: Info is kept secret from all but the authorised parties.

(2) MESSAGE AUTHENTICATION: Sender of a message is authentic

(3) (MESSAGE) INTEGRITY: message has not been modified during transmission

(4) NON-REPUDIATION: Sender of the message cannot deny the creation of the message.

Basic protocol with digital signature:

ALICE BOB

K_{prA}, K_{pubA} —K_{pubA}→

$S = sig_{K_{pr}}(X)$ —(X,S)→ $ver_{K_{pub}}(X,S)$

RSA Digital Signature:

SETUP:

$K_{prA} = (d)$

$K_{pubA} = (n,e)$

Alice Bob

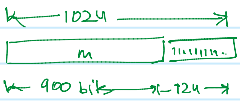
$K_{prA} = d$

$K_{pubA} = (n,e)$ —(n,e)→

∴ we cannot use this "SCHOOLBOOK RSA".

In practice, Impose formatting rules on x which can be checked by ALICE.

Eg:



PADDING

now OSCAR computes

$x \equiv s^e \bmod n$

∴ for getting the adding right, OSCAR has probability of doing so $\left(\frac{1}{2}\right)^{124}$

ELGAMAL DIGITAL SIGNATURE.

ALICE BOB

key generation:

choose p, primitive element α

random $k_{pr} = d$

$\in \{2, 3, \ldots, p-2\}$

$\beta = k_{pub} \equiv \alpha^d \bmod p$

$K_{pub} = (p, \alpha, \beta)$ ←

ephimeral key:

$k_E \in \{2, 3, \ldots, p-2\}$

s.t. $\gcd(k_E, p-1) = 1$

$r \equiv \alpha^{k_E} \bmod p$

SIGNATURE:

$s \equiv (x - dr) k_E^{-1} \bmod p-1$

$(x, (r,s))$ →

signature

ALICE VERIFIES:

$t \equiv \beta^r \cdot r^s \bmod p$

$t = \begin{cases} \equiv \alpha^x \bmod p & \checkmark \\ \not\equiv \alpha^x \bmod p & \times \end{cases}$

proof of correctness:

Alice:

$\beta^r r^s \equiv (\alpha^d)^r \alpha^{k_E})^s \bmod p$

$\equiv \alpha^{d \cdot r + k_E \cdot s}$

by fermat's little theorem:

$a^m = a^{q(p-1)+r}$

$= (a^q)^{p-1} \cdot a^r$

$\equiv 1 \cdot a^r \bmod p$

$\Rightarrow a^m \equiv a^{m \bmod p-1} \bmod p$

$K_{prA} = d$

$K_{pubA} = (n, e)$ $\xrightarrow{(n,e)}$

$S = Sig_{K_{pr}}(x)$

$\equiv x^d \bmod n$

$\xrightarrow{(x,S)}$ $ver_{K_{pubA}}(x, S)$

$S^e \equiv x' \bmod n$

$x' \begin{cases} \equiv x \Rightarrow \text{valid sign} \\ \neq x \Rightarrow \text{invalid sign} \end{cases}$

proof of correctness:

Bob computes:

$s^e = (x^d)^e$

$= x^{de}$

$= x \bmod n$ ✓

COMPUTATIONAL ASPECTS:

(1) signing: S-a-M algorithm (COSTLY) (SLOW)

(2) Verification:

$s^e \bmod n$

in practice often

$e = 3, 2^{16}+1$

(FAST)

EXISTENTIAL FORGERY ATTACK AGAINST RSA DIGITAL SIGNATURE

Alice Bob

$\leftarrow K_{pub} = (n, e)$

$K_{pr} = d$

OSCAR

(1) chooses $s \in Z_n$

(2) computes $x \equiv s^e \bmod n$

$\xleftarrow{(x,S)}$

Does the verification checkout?

$x' \equiv s^e \bmod n$ (ALICE)

$x \equiv s^e \bmod n$ (OSCAR)

$\therefore$ VALID

BROKEN.

only problem: x is fixed here. Its very hard for OSCAR to get x s.t. x says "TRANSFER me 4 bakh rupees".

"OSCAR cannot directly control the SEMANTICS of x".

$= (a^{p-1})^q \cdot a^s$

$\equiv 1 \cdot a^s \bmod p$

$\Rightarrow a^m = a^{m \bmod p-1} \bmod p$

$d \cdot r + k_E \cdot s = x \bmod p-1$

$k_E \cdot s = x - dr \bmod p-1$

$s = (x - dr) k_E^{-1} \bmod p-1$

$\Downarrow$ exactly the construction of s by BOB

REMARK: (1) The signature (r, s) has twice the bit length of x (bb)

(2) Elgamal DS is the base for DSA → DIGITAL SIGNATURE ALGORITHM

WEAKNESSES OF ELGAMAL DIGITAL SIGNATURE:

(1) Reuse of the ephemeral key:

assume Bob uses K_E for 2 messages x_1, x_2.

OSCAR: $x_1, (r, s_1)$

$x_2, (r, s_2)$

his goal is to find d?

$s_1 \equiv (x_1 - dr) k_E^{-1} \bmod p-1$

$s_2 \equiv (x_2 - dr) k_E^{-1} \bmod p-1$

....

$k_E = \dfrac{x_1 - x_2}{s_1 - s_2} \bmod p-1$

only thing unknown is $d = K_{prB}$

$\equiv \dfrac{x_1 - s_1 k_E}{r} \bmod p-1$

$\therefore$ DO NOT REUSE THE EPHEMERAL KEY

(2) ELGAMAL EXISTENTIAL FORGERY ATTACK:

ALICE BOB

OSCAR $K_{pub} = (p, \alpha, \beta)$,

$\leftarrow K_{pr} = d$

(1) select integer i, d with $gcd(i, p-1) = 1$

(2) compute signature

$r = \alpha^i \beta^j \bmod p$

$s = -r \cdot j^{-1} \bmod p-1$

(3) compute

$x \equiv si \bmod p-1$

$\xleftarrow{x, (r,s)}$

ALICE verifies:

$t = \beta^r \cdot r^s \bmod p$

Alice verifies:

$t = \beta^{r} \cdot r^{s} \bmod p$

$\Rightarrow t = \alpha^{x} \bmod p$

$\therefore$ VERIFIED

Why does the attack work?

$t = \beta^{r} r^{s} \bmod p$

$\equiv (\alpha^{d})^{r} \cdot r^{s} \bmod p$

$\equiv \alpha^{dr} \cdot (\alpha^{i} \beta^{q})^{s} \bmod p$

$\equiv \alpha^{dr} \cdot (\alpha^{i} \alpha^{dq})^{s} \bmod p$

$\equiv \alpha^{ds} \alpha^{(i+dq)(-s)^{-1}} \bmod p$

$\equiv \alpha^{dr} \cdot \alpha^{-ir s^{-1}} \alpha^{-drs^{-1}} \bmod p$

$\equiv \alpha^{-rsi^{-1}} \bmod p$

$\equiv \alpha^{si} \bmod p$

But Alice checks

$t = \alpha^{x} \bmod p$

$= \alpha^{si} \bmod p$

DRAWBACK

$\Rightarrow r$ is hard to mould into the required format/ form.