Signatures Schemes

Guevara Noubir
http://www.ccs.neu.edu/home/noubir/Courses/CSG252/F04

Textbook: "Cryptography: Theory and Applications",

Douglas Stinson, Chapman & Hall/CRC Press, 2002

Reading: Chapter 7.1-7.3

Outline

- Introduction to Signatures Schemes
- RSA digital signature
- Security characteristics
- El Gamal digital signature

Digital Signatures

- Goal:
  - Specify the entity (e.g., person) responsible for a message
- Differences with conventional signatures
  - Not physically attached to the physical document
    - ⇒ Need a way to bind it
  - Verification by comparison cannot be used
    - Digital signatures can be verified using a publicly known verification algorithm
  - Copies of conventional signatures can be physically detected
    - ⇒ Need a way to detect replay and limit use (e.g., date)
- Signature Scheme:
  - Signing Algorithm + Verification Algorithm

Formal Definition

- Signature Scheme is a 5-tuple (P, A, K, S, V):
  1. P is finite set of possible messages
  2. A is a finite set of possible signatures
  3. K the keyspace is a finite set of possible keys
  4. For each $k \in K$ there is a signing algorithm $sig_k \in S$ and a corresponding verification algorithm $ver_K \in S$.
     - sig_k: $P \rightarrow A$ **[Private]**
     - ver_K: $P \times A \rightarrow \{true, false\}$ **[Public]**
     - $ver_K(x, y) = \{$true if $y = sig_k(x)$, false if $y \neq sig_k(x)\}$
     - sig_k, ver_k: polynomial time functions

RSA Signature Scheme

- Let $n = pq$, where p and q are primes
- $P = C = Z_n$
- $K = \{(n, p, q, a, b) : ab \equiv 1 \pmod{\phi(n)}\}$

- Signature:
  - $sig_k(x) = x^a \bmod n$
- Verification:
  - $ver_k(x, y) = \text{true} \Leftrightarrow x = y^b \bmod n$

- Public key: n and b
- Private key: p, q, a

Simple Example of Using Signatures

- Two possibilities:
  - Send $e_{Bob}(x, y)$, where $y = sig_A(x)$, or
  - Send $z = e_{Bob}(x)$, and $sig_A(z)$
    - Problem authenticating the origin

Security Requirements for Signatures Schemes

- Attack model, goal of adversary, type of security
- Attack Models:
  - Key-only attack:
    - Only the public key is available to the adversary
  - Known message attack:
    - Attacker possesses a list of messages previously signed by Alice: (x_1, y_1), …
  - Chosen message attack:
    - Attacker can request Alice's signatures on a list of messages
- Goals:
  - Total break: determine private key
  - Selective forgery: with non-negligible probability the adversary is capable of creating a valid signature on a message chosen by someone else
  - Existential forgery: the adversary should be able to create a signature for at least one message [not previously known]
- Notes:
  - Unconditional security cannot be provided
  - Existential forgery against RSA? Two ways?

Signatures and Hash Functions

- Signatures are almost always used in conjunction with hash functions
- Scheme:
- Required properties:
  - To prevent existential forgery the hash function should be second pre-image resistant, collision resistant, and pre-image resistant

El Gamal Signature Scheme

- Let p be a prime s.t. discrete log in Z_p is intractable
- Let $\alpha \in Z_p^*$ be a primitive element
- $P = Z_p^*$, $A = Z_p^* \times Z_{p-1}$
- $K = \{(p, \alpha, a, \beta) : \alpha^a \equiv \beta \pmod{p}\}$
  - p, α, β : public ; a: private
- For a secret random number $k \in Z^*_{p-1}$
  - $sig(x, k) = (\gamma, \delta)$
  - $\gamma = \alpha^k \bmod p$
  - $\delta = (x - a\gamma)k^{-1} \bmod (p-1)$
- For $x, \gamma, \in Z_p^*$ and $\delta \in Z_{p-1}$:
  - $ver(x, (\gamma, \delta)) = true \Leftrightarrow \gamma^\delta \beta^\gamma = \alpha^x \pmod{p}$

Example

- Parameters:
  - $p = 467$, $\alpha = 2$, $a = 127$
  - $\beta = 2^{127} \bmod 467 = 132$
- Signing $x = 100$
  - Choose random $k = 213$ s.t. $\gcd(213, 466) = 1$
  - $k^{-1} \bmod 466 = 431$
  - $\gamma = 2^{213} \bmod 467 = 29$
  - $\delta = (100-127 \times 29) 431 \bmod 466 = 51$
- Public Verification:
  - $132^{29} 29^{51} \equiv 189 \pmod{467}$, and
  - $2^{100} \equiv 189 \pmod{467}$

Security of El Gamal Scheme

- Forging a signatures (without knowing a): alternatives for attacker
  1. Choose γ and tries to find a corresponding δ
     $\Rightarrow$ Need to solve a discrete log problem: $\delta = \log_{\gamma} \alpha^x \beta^{-\gamma}$
  - Chooses δ and tries to find a corresponding γ
     $\Rightarrow$ Another problem for which no solution is known
  1. Choose γ, and δ, and try to solve for x
     $\Rightarrow$ Discrete log problem
  2. Existential forgery [key-only attack assuming no hash function is used]:
     - Generate $\gamma = \alpha^i \beta^j$, δ, and x s.t. $\alpha^x = \gamma^\delta \beta^\gamma \pmod p$
     - Can be satisfied if: $x - i\delta \equiv 0 \bmod p-1$ and $\gamma + j\delta \equiv 0 \bmod p-1$
     - Given i, and j we can solve these two equations for x and δ
     - Example: $p = 467$, $\alpha = 2$, $\beta = 132$;
       - $i = 99$, $j = 179$, $j^{-1} \bmod 466 = 151$;
       - $\gamma = 2^{99} 132^{179} \bmod 467 = 117$;
       - $\delta = -117 * 151 \bmod 466 = 41$;
       - $x = 99 * 41 \bmod 466 = 331$