Signatures Schemes

Guevara Nouibi

http://www.cs.rutgers.edu/~nouibi/Courses/CS251/04

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 \(\text{sig}_k \in S\) and a corresponding verification algorithm \(\text{ver}_k \in S\).
   - \(\text{sig}_k: P \rightarrow A\) [Private]
   - \(\text{ver}_k: P \times A \rightarrow \{\text{true, false}\}\) [Public]
   - \(\text{ver}_k(x, y) = \text{true} \iff y = \text{sig}_k(x)\)
   - \(\text{sig}_k, \text{ver}_k\) polynomial time functions

**RSA Signature Scheme**

- Let \(n = pq\), where \(p\) and \(q\) are primes
- \(P = C = \mathbb{Z}_n^*\)
- \(K = \{(n, p, q, a, b) : ab = 1 \pmod{\varphi(n)}\}\)

- Signature:
  - \(\text{sig}_k(x) = y^x \mod n\)

- Verification:
  - \(\text{ver}_k(x, y) = \text{true} \iff x = y^x \mod n\)

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

**Simple Example of Using Signatures**

- Two possibilities:
  - Send \(e_{pub}(x, y)\), where \(y = \text{sig}_k(x)\), or
  - Send \(z = e_{pub}(x)\), and \(\text{sig}_k(z)\)
    - 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:
      - Adversary possesses a list of messages previously signed by Alice: \( \langle x, y \rangle, \ldots \)
    - Chosen message attack:
      - Adversary can request Alice’s signature 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 \( \mathbb{Z}_p^* \) is intractable
- Let \( \alpha \in \mathbb{Z}_p^* \) be a primitive element
- \( \mathbb{G} = \mathbb{Z}_p^* \times \mathbb{Z}_p^* \)
- \( K = \{ (p, a, (\alpha, \beta)) : \alpha^a = \beta \ (\text{mod} \ p) \} \)
  - \( p, a, \beta : \text{public} \) ; \( \alpha : \text{private} \)
- For a secret random number \( k \in \mathbb{Z}_p^* \):
  - \( \text{sig}(x, k) = (y, \delta) \)
  - \( y = (x \oplus \alpha k) \ (\text{mod} \ p) \)
  - \( \delta = (x \oplus \alpha k \delta) \ (\text{mod} \ p) \)
- For \( x, y, \delta \in \mathbb{Z}_p^* \) and \( \delta \in \mathbb{Z}_p^* \):
  - \( \text{ver}(x, (y, \delta)) = \text{true} \iff y^\delta = \alpha^x \ (\text{mod} \ p) \)
Example

Parameters:
- \( p = 467 \), \( a = 2 \), \( a = 127 \)
- \( \beta = 2^{127} \mod 467 = 132 \)

Signing \( x = 100 \)
- Choose random \( k = 213 \) s.t. \( \gcd(213, 466) = 1 \)
- \( k^2 \mod 466 = 431 \)
- \( \gamma = 2^{127} \mod 467 = 29 \)
- \( \delta = (100-2\times 29)431 \mod 466 = 51 \)

Public Verification:
- \( 132^{2981} \equiv 189 \mod 467 \), and
- \( 2^{132} \equiv 189 \mod 467 \)

Security of El Gamal Scheme

Forging a signature (without knowing \( a \)): alternatives for attacker

1. Choose \( \gamma \) and tries to find a corresponding \( \delta \)
   - \( \Rightarrow \) Need to solve a discrete log problem: \( \delta = \log_{a^n}\gamma \)
2. Chooses \( \delta \) and tries to find a corresponding \( \gamma \)
   - \( \Rightarrow \) Another problem for which no solution is known

1. Choose \( \gamma \), and \( \delta \), and try to solve for \( x \)
   - \( \Rightarrow \) Discrete log problem

2. Existential forgery (key-only attack assuming no hash function is used):
   - Generate \( \gamma = \gamma_{\phi} \), \( \delta \), and \( x \) s.t. \( x = \gamma_{\phi} \mod p \) and \( x = \delta + \gamma \mod p \)
   - Given \( \gamma \) and \( \delta \), we can solve these two equations for \( x \) and \( \delta \)

Example:
- \( p = 467 \), \( a = 2 \), \( \beta = 132 \);
- \( x = 58 \mod 467 = 58 \);
- \( \gamma = 2^{179} \mod 466 = 151 \);
- \( \delta = 137 \mod 466 = 41 \);
- \( x = 3\times 41 \mod 466 = 331 \)