A Biometric Identity Based Signature Scheme

Seminar Biometry & Security

18.01.2010
Bonn-Aachen International Center for Information Technology
Nina Keller
Outline

- Motivation
  - Why Identity Based Signatures
  - Identity Based Signatures
- Preliminaries
  - Bilinear pairings
  - Fuzzy Extractor
  - Error Correction Codes
- The Extraction Process
- Signature Scheme
- Security Issues
Why Identity Based Signatures

- Identity Based cryptography is a type of public-key cryptography.
- Public key of a user is some unique information about the identity of the user (e.g. a user's email address).
- No exchanging of private or public keys.
- No key directories.
- No third party, no certificates.
- No key revocation, once issued, keys are always valid.
- If there are only a finite number of users, after all users have been issued with keys master key can be destroyed.
Identity Based Signatures

- An identity-based scheme can be described as a collection of the following four algorithms:

1. **Private Key Generator (PKG)**
2. **Setup**: Given parameters and $K_m$, computes $k$.
3. **Extract**: Given parameters, $K_m$, computes $d$.
4. **Sign**: Given parameters, $m$, computes $\sigma$.
5. **Verify**: Given parameters, $m$, accepts/rejects $\sigma$.

[Diagram showing the flow of data through the algorithms]

- $ID$ is the input for each algorithm, and the output is passed to the next algorithm.
- $k$, $d$, $\sigma$, and accept/reject are the outputs for each step.
- $m$ is the message input for the **Sign** and **Verify** steps.
- $param$, $K_m$ are parameters used in the **Setup** and **Extract** steps.

---

Identity Based Signatures

- An identity-based scheme can be described as a collection of the following four algorithms:

1. **Private Key Generator (PKG)**
2. **Setup**: Given parameters and $K_m$, computes $k$.
3. **Extract**: Given parameters, $K_m$, computes $d$.
4. **Sign**: Given parameters, $m$, computes $\sigma$.
5. **Verify**: Given parameters, $m$, accepts/rejects $\sigma$.

[Diagram showing the flow of data through the algorithms]

- $ID$ is the input for each algorithm, and the output is passed to the next algorithm.
- $k$, $d$, $\sigma$, and accept/reject are the outputs for each step.
- $m$ is the message input for the **Sign** and **Verify** steps.
- $param$, $K_m$ are parameters used in the **Setup** and **Extract** steps.
Preliminaries

Bilinear pairing

\[ e : G_1 \times G_1 \rightarrow G_2 \]

Properties:

• Bilinearity: \( \forall P, Q \in G_1, a, b \in \mathbb{Z}_p^*: e(aP, bQ) = e(P, Q)^{ab} \)

• Non-Degeneracy: \( \forall P \in G_1, P \neq \infty: e(P, P) \neq 1 \)

• \( e \) has to be computable in an efficient manner
Preliminaries

**Tate pairing** $\tau'_q$

$$\tau'_q : E(F_p)[q] \times E(F_p^2)/qE(F_p^2) \rightarrow \mu(q)$$

$$\tau'_q(P, Q) = a + bi \in \mu(q)$$

$F_p = \{0, 1, 2, 3, \ldots, p - 2, p - 1\}$

$F_p^2 = \{a + ib\}, \quad a, b \in F_p, \quad i = \sqrt{-1}$

$E(F_p): y^2 = x^3 + x$

$E(F_p^2)[q]$ - subgroup of $E(F_p)$ consisting of points of order $q$

$\mu(q) = \{a \in F_p^2 | a^q = 1\}$
Preliminaries

**Fuzzy Extractor**

\[ M = \{0,1\}^v \] - a finite dimensional metric space consisting of biometric data points

\[ dis : M \times M \rightarrow Z^+ \] - distance function

\( l \) - number of bits of the extracted output string \( ID \)

\( t \) - error threshold

\( (M,l,t) \) - fuzzy extractor is generated using two functions:

- **Gen**: probabilistic generation procedure, on input \( b \in M \) outputs an “extracted” string \( ID \in \{0,1\}^l \) and public string \( PAR \)

- **Rep**: deterministic reproduction procedure allowing recovery of \( ID \) from the corresponding public string \( PAR \) and any \( b' \) sufficiently close to \( b \)
Preliminaries

**Fuzzy Extractor**

Our fuzzy extractor uses:

- *Hamming Distance* is defined to be the number of bit positions that differ between $b'$ and $b$

- Error Correcting Codes
Preliminaries

**Error Correcting Codes**

- \( C \) - subset of \( n \)-bit words (i.e. \( C \subseteq \{0,1\}^n \)) with \( n > n \) and \( k \) having at least \( 2^k \) elements for some positive integer.

- \( C_e : M \to C \) – one-to-one encoding function.

- \( C_d : \{0,1\}^n \to C \) - decoding function that has an error threshold of \( t \).

- The decoding function \( C_d \) will take an arbitrary \( n \)-bit string and “correct” it to the nearest codeword in \( C \).
Preliminaries

Error Correcting Codes

- Example:
  - Let $n=3$, $v=1$ and $C=\{000,111\}$
  - Let $C_d$ compute majority
    - Then $C_d$ would map a bitstring $x \in \{0,1\}^3$ to 000 if at least two bits of $x$ are 0s and to 111 if at least two bits are 1s.
  - Threshold $t=1$
The Extraction Process

**Step 1:** obtaining $b$ from the biometric reader
**Step 2:** use an error correcting codes to fuzzy extract some data from the biometric input.

- *Gen* function:
  
  \[
  ID = H(b) \\
  RAP = b \oplus C_e(ID)
  \]
**The Extraction Process**

**Step 2:** use an error correcting codes to fuzzy extract some data from the biometric input.

- *Rep function:*

\[
ID' = C_d(b' \oplus PAR) = C_d(b' \oplus b \oplus C_e(ID))
\]
Signature Scheme

Private Key Generator (PKG)

**Setup**
- param, $K_m$

**Extract**
- $d$

**Sign**
- $\sigma$

**Verify**
- accept / reject

1. $ID \rightarrow Private Key Generator (PKG)$
2. $ID \rightarrow Extract\ d$
3. $k \rightarrow Setup\ param,\ K_m$
4. $m \rightarrow Sign\ \sigma$
5. $param\ Extract\ d$
6. $param\ Verify\ \sigma$
7. $m \rightarrow Verify\ accept / reject$
Signature Scheme

Setup:

Given a security parameter $k$ PKG selects:

- Groups $G_1, G_2$ of prime order $q > 2^k$
- A generator $P$ of $G_1$
- Randomly chosen master key $s \in \mathbb{Z}_q^+$ and the associated public key $P_{pub} = sP$
- Cryptographic hash functions of the same domain range $H_1, H_2 : \{0, 1\}^+ \rightarrow G_1^+$, $H_3 : b \rightarrow \{0, 1\}^+$
- Encoding function $C_e$ and a decoding function $C_d$
- It also selects a method for extracting the features of a biometric, $F_e$
Signature Scheme

Key Generation

- After obtaining $b$ using feature extractor $F_e$ the identity string is calculated $ID = H_3(b)$

- The PKG computes the public key $Q_{ID} = H_1(ID) \in G_1$ and the associated private key $d_{ID} = sQ_{ID} \in G_1$
Signature Scheme

Sign

To sign a message $M$:

1) Pick a random integer $r \in \mathbb{Z}_q$ and compute $U = rP \in G_1$
   Then $H = H_2(ID, M, U) \in G_1$.

2) Compute $V = d_{ID} + rH \in G_1$

3) The value $PAR = b \oplus C_e(ID)$ is included as part of the signature

The signature on $M$ is the triple $\sigma = \langle U, V, PAR \rangle$
Signature Scheme

Verify

To verify a signature \( \sigma = \langle U, V, PAR \rangle \) on a message \( M \) for an identity \( ID \), the verifier performs the following steps:

1) Obtain a biometric reading \( b' \) and calculate
   \[
   ID' = \text{Rep}(b', PAR)
   \]

2) Calculate
   \[
   Q'_{ID} = H_1(ID') \in G_1 \quad \text{and} \quad H = H_2(ID', M, U) \in G_1.
   \]

3) Signature is verified if
   \[
   e(P, V) = e(P_{pub}, Q'_{ID}) e(U, H)
   \]
   and rejected otherwise.
Security Issues

• Malicious signing
  - Biometric data can be obtained relatively easily
  - Attacker can't sign a document (he still doesn't have private key)
  - Attacker can attempt to acquire the private key

• Solution: imposing proper authentication procedures for users applying to the PKG (certificates)
Security Issues

- **Disavowal**
  - If fuzzy extractor is not robust, attacker can tamper with ID
  - Attacker can try to alter his biometrics (wear a thin film with another print on his finger)
    - **Solution**: inspect the biometrics before measurement.