III - Pairing-based Cryptography

Gap Groups

Definition (Pairing Setting)
- Let $G_1$ and $G_2$ be two cyclic groups of prime order $p$.
- Let $g_1$ and $g_2$ be generators of $G_1$ and $G_2$ respectively.
- Let $e : G_1 \times G_2 \rightarrow G_T^*$ be a bilinear map.

Definition (Various Cases)
- The symmetric case: $G_1 = G_2$.
- There exists an isomorphism $\psi$, from $G_2$ onto $G_1$:
  1. $\psi$ is efficiently computable; as well as $\psi^{-1}$
  2. $\psi$ is efficiently computable; but no efficient isomorphism from $G_1$ onto $G_2$
  3. no efficiently computable isomorphism in any direction
Gap Groups

Definition (co-Diffie-Hellman Problems)

Let $(p, G_1, g_1, G_2, g_2, G^T, e)$ be a pairing setting

- **co-CDH** in $(G_1, G_2)$: Given $g, g^a \in G_2$ and $h \in G_1$, compute $h^a$
- **co-DDH** in $(G_1, G_2)$: Given $g, g^a \in G_2$ and $h, h^b \in G_1$, decide whether $a = b$ or not

Note: when $G_1 = G_2 = G$, **co-CDH** in $(G_1, G_2)$ is **CDH** in $G$, and **co-DDH** in $(G_1, G_2)$ is **DDH** in $G$

Definition (Gap Groups)

We say that a group $G$ is a gap group if **CDH** in $G$ is hard, whereas **DDH** in $G$ is simple.

Admissible Bilinear Map

Definition (Admissible Bilinear Map)

Let $(p, G_1, g_1, G_2, g_2, G^T, e)$ be a pairing setting, with $e : G_1 \times G_2 \rightarrow G^T$ a non-degenerated bilinear map

- Bilinear: for any $g \in G_1$, $h \in G_2$ and $u, v \in \mathbb{Z}$,
  
  $e(g^u, h^v) = e(g, h)^{uv}$
- Non-degenerated: $e(g_1, g_2) \neq 1$

**co-DDH** in $(G_1, G_2)$ easy

Given $g, g^a \in G_2$ and $h, h^b \in G_1$

$a = b \mod p \iff e(h, g^a) = e(h^b, g)$

Bilinear Diffie-Hellman Problems

We now focus on the symmetric case: $G_1 = G_2 = G$.

Diffie-Hellman Problems

- **CDH** in $G$: Given $g, g^a, g^b \in G$, compute $g^{ab}$
- **DDH** in $G$: Given $g, g^a, g^b, g^c \in G$, decide whether $c = ab$ or not

**CDH** can be hard to solve, but **DDH** is easy in gap-groups.

Bilinear Diffie-Hellman Problems

- **CBDH** in $G$: Given $g, g^a, g^b, g^c \in G$, compute $e(g, g)^{abc}$
- **DBDH** in $G$: Given $g, g^a, g^b, g^c \in G$ and $h \in G^T$, decide whether $h^? = e(g, g)^{abc}$
Signature in Gap Groups

Let $G$ be a gap-group of prime order $p$, with a generator $g$.

**Signature Scheme**

- Key generation: choose $x \in \mathbb{Z}_p$, and set $y = g^x$;
- Signature of $M \in G$: $\sigma = M^x$;
- Verification of $(M, \sigma)$: check $\text{DDH}(g, y, M, \sigma)$.

**Full-Domain Hash**

$H : \{0, 1\}^* \rightarrow G$

- In order to sign $m$, one first computes $M = H(m) \in G$
- then $\sigma = M^x = \text{CDH}(g, y, H(m))$

The signature of a message $m$ is thus an element $\sigma \in G$.

Identity-Based Cryptography

- **Public-Key Cryptography**
  Each user $ID$ owns
  - a public key $pk$
  - a certificate that guarantees the link between $ID$ and $pk$
  - a private key $sk$, related to $pk$

  One has to access a dictionary in order to get $pk$, the public key of $ID$, together with the certificate, in order to encrypt a message to $ID$.

- **Identity-Based Cryptography**
  Each user $ID$ owns
  - a private key $sk$, related to $ID$
  - the public key $pk$ is indeed $ID$ itself

Key Computation

- **Public-Key Cryptography**
  - User $ID$ chooses his private key $sk$
  - derives his public key $pk$
  - asks a TTP for the certification of $pk$ w.r.t. $ID$

- **Identity-Based Cryptography**
  - Each user $ID$ asks a TTP for the computation of the private key $sk$, related to $ID$
    $\Rightarrow$ extraction

Note

For signature, the two scenarios are quite similar.
Identity-Based Encryption

Setup

The authority generates a master secret key msk, and publishes the public parameters, PK

Extraction

Given an identity ID, the authority computes the private key sk granted the master secret key msk

Encryption

Any one can encrypt a message m to a user ID using only m, ID and the public parameters PK

Decryption

Given a ciphertext, user ID can recover the plaintext, with his secret key sk

Outline

1 Introduction
2 Identity-Based Encryption
   ■ Security
3 Without Random Oracles

Security Model: IND – ID – CCA

Definition (IND – ID – CCA Security)

The adversary
- receives the global parameters
- asks any extraction-query, and any decryption-query
- outputs a target identity ID* and two messages (m0, m1)

The challenger flips a bit b, and encrypts m_b for ID* into c*, then the adversary
- asks any extraction-query, and any decryption-query
- outputs its guess b' for b

\[ \text{Adv}^{\text{ind-id-cca}} = 2 \times \Pr[b' = b] - 1 \]

Restrictions

- IND – ID – CCA: semantic security, full-identity, chosen-ciphertext attacks
  The adversary is just restricted not to ask:
  - the target identity ID* to the extraction-oracle,
  - nor the challenge ciphertext c* to the decryption-oracle with ID*
- sID: selective-identity
  The adversary provides the target identity ID* before receiving the global parameters
- CPA: chosen-plaintext attacks
  The adversary does not have access to the decryption-oracle
Identity-Based Encryption

[Boneh-Franklin – Crypto ’01]

Setup

The authority sets up a gap-group framework:
- a group \( G \) of prime order \( p \), with a generator \( g \),
- with an admissible bilinear map \( e : G \times G \rightarrow G^T \)

It selects a master secret key \( msk = s \in \mathbb{Z}_p \)

It publishes the public parameters: \( PK = (p, G, e, g, P = g^s) \)

Extraction

Given an identity \( ID \), the authority compute
the private key \( sk = H(ID)^s \)

Note that \( sk \) is a BLS signature of \( ID \),
which can be checked by the user: \( e(sk, g) = e(H(ID), P) \)

Encryption

In order to encrypt a message \( m \) to a user \( ID \)

- one chooses a random \( r \in \mathbb{Z}_p \)
- computes \( A = g^r \) and \( K = e(P, H(ID)^r) \)
- sends \( (A, B = K \times m) \)

\[
K = e(P, H(ID)^r) = e(g^s, H(ID)^r) = e(g^r, H(ID)^s) = e(A, sk)
\]

Decryption

Upon reception of \( (A, B) \), user \( ID \)

- computes \( K = e(A, sk) \)
- gets \( m = B/K \)

BF IBE Security Analysis

Theorem

The BF IBE is IND – ID – CPA secure
under the DBDH problem,
in the random oracle model.

By masking \( m \) with \( H(K) \):
\( B = m \oplus H(K) \),
the BF IBE is IND – ID – CPA secure
under the CBDH problem,
in the random oracle model

CCA Security

[Fujisaki-Okamoto – Crypto ’01]

Usual tricks in the random oracle model to achieve IND – ID – CCA.

- How to avoid the random oracle model?
- How to avoid a full-domain hash function onto \( G \)?

Outline

1 Introduction
2 Identity-Based Encryption
3 Without Random Oracles
   - BB Signature/IBE
   - Extension
Boneh-Boyen’s Signature

Let $G$ be a cyclic group of prime order $p$, with two independent generators $g, h$, equipped with an admissible bilinear map

\[ e : G \times G \to G^T \]

For any message $m \in \mathbb{Z}_p$ (output by a hash function), we define $F(m) = u v^m$, where $u$ and $v$ are independent public elements in $G$.

Boneh-Boyen’s Signature (Cont’d)

Signature Scheme

Key generation: choose $x \in \mathbb{Z}_p$, and set $G = g^x$ as well as $H = h^x$; the public key is $G$, whereas $H$ is kept private.

Signature of $m \in \mathbb{Z}_p$: $\sigma = (H \times F(m)^r, g^r)$, for a random $r \in \mathbb{Z}_p$.

Verification of $(m, (\sigma_1, \sigma_2))$: check whether

\[
\frac{\sigma_1}{\sigma_2} = e(G, h) \times e(F(m), F(m))
\]

Boneh-Boyen’s Signature: Security Analysis

Simulation: CMA

For any query $m \neq m^*$, we simulate a signature:

$\sigma_1 = h^{-\beta/(m-m^*)} F(m)^r$ and $\sigma_2 = g^r h^{1/(m^*-m)}$.

Let us set $\rho = r - b/(m - m^*)$:

\[
\sigma_1 = h^{-\beta/(m-m^*)} F(m)^r = h^{-\beta/(m-m^*)} \times (G^{m-m^*} g^{\beta \rho + b/(m-m^*)}) = h^{-\beta/(m-m^*)} \times G^{\beta (m-m^*)} \times g^{\beta \rho} \times h^{\beta/(m-m^*)} = h^\rho \times G^{\beta (m-m^*)} \times g^{\beta \rho} = h^\rho \times F(m)^\rho
\]

\[
\sigma_2 = g^r \times h^{1/(m^*-m)} = g^{r-b/(m - m^*)} = g^\rho
\]
**Identity-Based Encryption**

**Setup**
- The authority sets up a gap-group framework:
  - A group $G$ of prime order $p$
  - With three independent generators $g$, $h$, and $u$
  - With an admissible bilinear map $e : G \times G \rightarrow G^T$
- It selects a master secret key $s \in \mathbb{Z}_p$
- It keeps $H = h^s$
- It publishes the parameters: $(p, G, e, g, h, G = g^s)$

**Extraction**
Given an identity $ID$, the authority computes the key $sk = (sk_1 = H \times F(ID)^t, sk_2 = g^t)$, where $F(x) = uG^x$

**Encryption**
In order to encrypt a message $m \in G^T$ to a user $ID$
- One chooses a random $t \in \mathbb{Z}_p$
- Computes $A = F(ID)^t$, $B = g^t$ and $K = e(G, h)^t$
- Sends $(A, B, C = K \times m)$

**Decryption**
Upon reception of $(A, B, C)$, user $ID$ computes $K = e(B, sk_1)/e(A, sk_2)$ and gets $m = C/K$

**Outline**

1. **Introduction**
2. **Identity-Based Encryption**
3. **Without Random Oracles**
   - BB Signature/IBE
   - Extension

**BB IBE Security Analysis**
The BB IBE is IND − sID − CPA secure under the DBDH problem
Let \( G \) be a cyclic group of prime order \( p \), with two independent generators \( g, h \), equipped with an admissible bilinear map
\[
e : G \times G \to G^T
\]
For any message \( m \in \{0, 1\}^k \) (output by a hash function), we define
\[
F(m) = u'(\prod u_i^{m_i}), \quad m = m_1 \ldots m_k,
\]
where \( u' \) and \( u_1, \ldots, u_k \) are independent public elements in \( G \)

**Signature Scheme**

- **Key generation:** choose \( x \in \mathbb{Z}_p \), and set \( G = g^x \) as well as \( H = h^x \);
The public key is \( G \), whereas \( H \) is kept private.
- **Signature of** \( m \in \{0, 1\}^k \): \( \sigma = (H \times F(m)^{r}, g^r) \), for a random \( r \in \mathbb{Z}_p \);
- **Verification of** \( (m, (\sigma_1, \sigma_2)) \): check whether
\[
e(g, \sigma_1) = e(g, h^x \times F(m)^{r})
\]
\[
= e(g, h^x) \times e(g, F(m)^{r}) = e(g^x, h) \times e(g^r, F(m))
\]
\[
= e(G, h) \times e(\sigma_2, F(m))
\]

**Theorem**

The Water’s IBE is IND – ID – CPA secure under the DBDH problem.