

III – Pairing-based Cryptography

David Pointcheval

Ecole normale supérieure, CNRS & INRIA



ENS – Paris – 2016/2017

- 1 Introduction**
 - Gap Groups
 - Pairings
 - Short Signatures
- 2 Identity-Based Encryption**
 - Security
- 3 Without Random Oracles**
 - BB Signature/IBE
 - Extension

Outline

Gap Groups

- 1 Introduction**
 - Gap Groups
 - Pairings
 - Short Signatures

2 Identity-Based Encryption

3 Without Random Oracles

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)

- 1** The symmetric case: $G_1 = G_2$.
- 2** There exists an isomorphism ψ , from G_2 onto G_1 :
 - 1** ψ is efficiently computable; as well as ψ^{-1}
 - 2** ψ is efficiently computable; but no efficient isomorphism from G_1 onto G_2
 - 3** no efficiently computable isomorphism in any direction

Definition (co-Diffie-Hellman Problems)

Let $(p, \mathbb{G}_1, g_1, \mathbb{G}_2, g_2, \mathbb{G}^T, e)$ be a pairing setting

- **co-CDH** in $(\mathbb{G}_1, \mathbb{G}_2)$: Given $g, g^a \in \mathbb{G}_2$ and $h \in \mathbb{G}_1$, compute h^a
- **co-DDH** in $(\mathbb{G}_1, \mathbb{G}_2)$: Given $g, g^a \in \mathbb{G}_2$ and $h, h^b \in \mathbb{G}_1$, decide whether $a = b$ or not

Note: when $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$, **co-CDH** in $(\mathbb{G}_1, \mathbb{G}_2)$ is **CDH** in \mathbb{G} , and **co-DDH** in $(\mathbb{G}_1, \mathbb{G}_2)$ is **DDH** in \mathbb{G}

Definition (Gap Groups)

We say that a group \mathbb{G} is a **gap group** if **CDH** in \mathbb{G} is hard, whereas **DDH** in \mathbb{G} is simple.

1 Introduction

- Gap Groups
- Pairings
- Short Signatures

2 Identity-Based Encryption

3 Without Random Oracles

Admissible Bilinear Map

[Joux – ANTS '00]

Bilinear Diffie-Hellman Problems

Definition (Admissible Bilinear Map)

Let $(p, \mathbb{G}_1, g_1, \mathbb{G}_2, g_2, \mathbb{G}^T, e)$ be a pairing setting, with $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}^T$ a non-degenerated bilinear map

- Bilinear: for any $g \in \mathbb{G}_1, h \in \mathbb{G}_2$ 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 $(\mathbb{G}_1, \mathbb{G}_2)$ easy

Given $g, g^a \in \mathbb{G}_2$ and $h, h^b \in \mathbb{G}_1$

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

We now focus on the symmetric case: $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$.

Diffie-Hellman Problems

- **CDH** in \mathbb{G} : Given $g, g^a, g^b \in \mathbb{G}$, compute g^{ab}
- **DDH** in \mathbb{G} : Given $g, g^a, g^b, g^c \in \mathbb{G}$, 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 \mathbb{G} : Given $g, g^a, g^b, g^c \in \mathbb{G}$, compute $e(g, g)^{abc}$
- **DBDH** in \mathbb{G} : Given $g, g^a, g^b, g^c \in \mathbb{G}$ and $h \in \mathbb{G}^T$, decide whether $h \stackrel{?}{=} e(g, g)^{abc}$

1 Introduction

- Gap Groups
- Pairings
- Short Signatures

2 Identity-Based Encryption

3 Without Random Oracles

Let \mathbb{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 $M \in \mathbb{G}$: $\sigma = M^x$;
- Verification of (M, σ) : check **DDH**(g, y, M, σ).

Full-Domain Hash

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

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

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

Identity-Based Cryptography

[Shamir – Crypto '84]

Key Computation

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

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
⇒ extraction

Note

For signature, the two scenarios are quite similar.

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 sk
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

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

Security Model: IND – ID – CCA

Restrictions

Definition (IND – ID – CCA Security)
The adversary <ul style="list-style-type: none"> ■ receives the global parameters ■ asks any extraction-query, and any decryption-query ■ outputs a target identity ID^* and two messages (m_0, m_1) The challenger flips a bit b , and encrypts m_b for ID^* into c^* , then the adversary <ul style="list-style-type: none"> ■ asks any extraction-query, and any decryption-query ■ outputs its guess b' for b

- **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

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

Setup

- The authority sets up a gap-group framework: a group \mathbb{G} of prime order p , with a generator g , with an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}^T$
- It selects a master secret key $\text{msk} = s \in \mathbb{Z}_p$
- It publishes the public parameters: $\text{PK} = (p, \mathbb{G}, e, g, P = g^s)$

Extraction

Given an identity ID , the authority compute the private key $sk = \mathcal{H}(\text{ID})^s$

Note that sk is a BLS signature of ID , which can be checked by the user: $e(sk, g) \stackrel{?}{=} e(\mathcal{H}(\text{ID}), P)$

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 $K = e(P, \mathcal{H}(\text{ID})^r)$
 - sends $(A, B = K \times m)$

$$K = e(P, \mathcal{H}(\text{ID})^r) = e(g^s, \mathcal{H}(\text{ID})^r) = e(g^r, \mathcal{H}(\text{ID})^s) = e(A, sk)$$

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 \mathbb{G} ?

Outline

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

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

$$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}^T$$

For any message $m \in \mathbb{Z}_p$ (output by a hash function),
we define $F(m) = uv^m$, where u and v are independent public
elements in \mathbb{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 $m \in \mathbb{Z}_p$: $\sigma = (H \times F(m))^r, g^r$,
for a random $r \in \mathbb{Z}_p$;
Here, $F(m) = G^m \times u$
- Verification of $(m, (\sigma_1, \sigma_2))$: check whether

$$\begin{aligned} e(g, \sigma_1) &= e(g, h^x \times F(m)^r) \\ &= e(g, h^x) \times e(g, F(m)^r) = e(g^x, h) \times e(g^r, F(m)) \\ &\stackrel{?}{=} e(G, h) \times e(\sigma_2, F(m)) \end{aligned}$$

Boneh-Boyen's Signature: Security Analysis

Theorem (Selected-Message CMA)

For a message m^* chosen ahead, before having seen the parameters
and the public key, signing m^* under a chosen-message attack is
intractable under the **CDH** problem in \mathbb{G} .

Simulation: Selected-Message Forgery

Let us be given $g, G = g^a$ and $h = g^b$,
we want to extract $H = h^a = g^{ab}$.

We set $u = G^{-m^*} g^\beta$ for a random β :

$$F(m) = G^m u = G^{m-m^*} g^\beta \quad F(m^*) = g^\beta$$

A forgery for m^* : (σ_1, σ_2) , such that

$$e(g, \sigma_1) = e(G, h) e(\sigma_2, g^\beta) \implies e(G, h) = e(g, \sigma_1 / \sigma_2^\beta)$$

$$\mathbf{CDH}(g, h, G) = \sigma_1 / \sigma_2^\beta$$

Boneh-Boyen's Signature: Security Analysis

Simulation: CMA

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

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

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

$$\begin{aligned} \sigma_1 &= h^{-\beta/(m-m^*)} \times F(m)^r \\ &= h^{-\beta/(m-m^*)} \times (G^{m-m^*} g^\beta)^{\rho+b/(m-m^*)} \\ &= h^{-\beta/(m-m^*)} \times G^{\rho(m-m^*)} \times G^b \times g^{\beta\rho} \times h^{\beta/(m-m^*)} \\ &= h^a \times G^{\rho(m-m^*)} \times g^{\beta\rho} \\ &= h^a \times F(m)^\rho \\ \sigma_2 &= g^r \times h^{1/(m^*-m)} = g^{r-b/(m-m^*)} = g^\rho \end{aligned}$$

Setup

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

Extraction

Given an identity \mathcal{ID} , the authority computes the key $sk = (sk_1 = H \times F(\mathcal{ID})^r, sk_2 = g^r)$, where $F(x) = uG^x$

Note that sk is a BB signature of \mathcal{ID} : $e(g, sk_1) \stackrel{?}{=} e(G, h) \times e(sk_2, F(\mathcal{ID}))$

Encryption

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

$$\begin{aligned}
 K &= e(G, h)^t = e(g^s, h)^t = e(g^t, h^s) = e(g^t, H) \\
 &= e(g^t, sk_1 / F(\mathcal{ID})^r) = e(g^t, sk_1) / e(g^t, F(\mathcal{ID})^r) \\
 &= e(B, sk_1) / e(g^r, F(\mathcal{ID})^t) = e(B, sk_1) / e(sk_2, A)
 \end{aligned}$$

Decryption

Upon reception of (A, B, C) , user \mathcal{ID} computes $K = e(B, sk_1) / e(A, sk_2)$ and gets $m = C / K$

BB IBE Security Analysis

The BB IBE is **IND – sID – CPA** secure under the **DBDH** problem

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

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

$$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}^T$$

For any message $m \in \{0, 1\}^k$ (output by a hash function), we define

$$F(m) = u'(\prod u_i^{m_i}), \quad m = m_1 \dots m_k,$$

where u' and u_1, \dots, u_k are independent public elements in \mathbb{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

$$\begin{aligned} e(g, \sigma_1) &= e(g, h^x \times F(m)^r) \\ &= e(g, h^x) \times e(g, F(m)^r) = e(g^x, h) \times e(g^r, F(m)) \\ &\stackrel{?}{=} e(G, h) \times e(\sigma_2, F(m)) \end{aligned}$$

Waters' Signature (Cont'd)

Theorem

The Water's IBE is **IND – ID – CPA** secure
under the **DBDH** problem