Identity based cryptography

The case of encryption schemes

David Galindo

d.galindo@cs.ru.nl

Security of Systems

Department of Computer Science

Radboud Universiteit Nijmegen

Outline

- Motivation

Outline

- Motivation
- Definitions
  - Identity Based Encryption (IBE)
  - Secure IBEs

Outline

- Motivation
- Definitions
  - Identity Based Encryption (IBE)
  - Secure IBEs
  - Bilinear maps and problems

Outline

- Motivation
- Definitions
  - Identity Based Encryption (IBE)
  - Secure IBEs
  - Bilinear maps and problems
- Schemes

2001 Boneh&Franklin scheme (ROM)

2004 Waters scheme (standard model)

Outline

- Motivation
- Definitions
  - Identity Based Encryption (IBE)
  - Secure IBEs
  - Bilinear maps and problems
- Schemes

2001 Boneh&Franklin scheme (ROM)

2004 Waters scheme (standard model)

- Future research

Motivation: PKI

To use Public Key Cryptography we need to bind identities and keys.

Public Key Infrastructures

Motivation: PKI

To use Public Key Cryptography we need to bind identities and keys.

Public Key Infrastructures

A Certification Authority (CA) issues certificates:

- U user's identity
- PK public key
- D_1 issue date
- D_2 expiration date

Motivation: PKI

To use Public Key Cryptography we need to bind identities and keys.

Public Key Infrastructures

A Certification Authority (CA) issues certificates:

- U user's identity
- PK public key
- D_1 issue date
- D_2 expiration date

} Certificate(U, PK)
$Sig_{\mathsf{CA}}(U, PK, D_1, D_2)$

Motivation: PKI

To use Public Key Cryptography we need to bind identities and keys.

Public Key Infrastructures

A Certification Authority (CA) issues certificates:

- U user's identity
- PK public key
- D_1 issue date
- D_2 expiration date

} Certificate(U, PK)
$Sig_{\mathsf{CA}}(U, PK, D_1, D_2)$

Certificate Revocation Problem

Motivation: PKI (ii)

Before performing the cryptographic operation involving the public key, we must validate Certificate(U, PK).

Motivation: PKI (ii)

Before performing the cryptographic operation involving the public key, we must validate Certificate(U, PK).

Easy for signature schemes. User U sends the certificate along with its signature on a message m

$$(\text{Certificate}(U, PK), \text{Sig}_{PK}(m), m)$$

Motivation: PKI (ii)

Before performing the cryptographic operation involving the public key, we must validate Certificate(U, PK).

Easy for signature schemes. User U sends the certificate along with its signature on a message m

$$(\text{Certificate}(U, PK), \text{Sig}_{PK}(m), m)$$

Difficult for encryption schemes. Before sending a message m to user U, we should know if it is in possession of a valid certificate.

Motivation: PKI (ii)

Before performing the cryptographic operation involving the public key, we must validate Certificate(U, PK).

Easy for signature schemes. User U sends the certificate along with its signature on a message m

$$(\text{Certificate}(U, PK), \text{Sig}_{PK}(m), m)$$

Difficult for encryption schemes. Before sending a message m to user U, we should know if it is in possession of a valid certificate.

We would like to perform the public operation without extra communication.

Identity Based Encryption (IBE)

Identity Based Encryption (IBE)

Main idea The public key is an identity ID $\in \{0,1\}^*$
A Key Generation Center KGC issues private keys for ID

Identity Based Encryption (IBE)

Main idea The public key is an identity ID $\in \{0,1\}^*$
A Key Generation Center KGC issues private keys for ID

An IBE scheme consists of 4 algorithms:

Setup Takes a security parameter ℓ and outputs system paramaters params and master-key.

Identity Based Encryption (IBE)

Main idea The public key is an identity ID $\in \{0,1\}^*$
A Key Generation Center KGC issues private keys for ID

An IBE scheme consists of 4 algorithms:

Setup Takes a security parameter ℓ and outputs system paramaters params and master-key.

Encrypt Takes as inputs params, ID $\in \{0,1\}^*$ and message M and outputs a ciphertext C.

Identity Based Encryption (IBE)

Main idea The public key is an identity ID $\in \{0,1\}^*$
A Key Generation Center KGC issues private keys for ID

An IBE scheme consists of 4 algorithms:

Setup Takes a security parameter ℓ and outputs system paramaters params and master-key.

Encrypt Takes as inputs params, ID $\in \{0,1\}^*$ and message M and outputs a ciphertext C.

ExtractPrivateKey Takes as inputs params, master-key and ID $\in \{0,1\}^*$ and outputs a private decryption key d_{ID}.

Identity Based Encryption (IBE)

Main idea The public key is an identity ID $\in \{0,1\}^*$
A Key Generation Center KGC issues private keys for ID

An IBE scheme consists of 4 algorithms:

Setup Takes a security parameter ℓ and outputs system paramaters params and master-key.

Encrypt Takes as inputs params, ID $\in \{0,1\}^*$ and message M and outputs a ciphertext C.

ExtractPrivateKey Takes as inputs params, master-key and ID $\in \{0,1\}^*$ and outputs a private decryption key d_{ID}.

Decrypt Takes as inputs params, private key d_{ID} and message C and outputs a message M.

Identity Based Encryption (IBE)

Main idea The public key is an identity ID $\in \{0,1\}^*$
A Key Generation Center KGC issues private keys for ID

An IBE scheme consists of 4 algorithms:

Setup Takes a security parameter ℓ and outputs system paramaters params and master-key.

Encrypt Takes as inputs params, ID $\in \{0,1\}^*$ and message M and outputs a ciphertext C.

Certificate revocation problem can be "avoided" using
ID $= \texttt{bob@company.com}||\texttt{year}||\texttt{month}||\texttt{day}$

Security notions for IBE schemes

IND-ID-CPA security for an IBE scheme $\mathcal{E}$

Security notions for IBE schemes

IND-ID-CPA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Security notions for IBE schemes

IND-ID-CPA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$

Security notions for IBE schemes

IND-ID-CPA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$

Challenge $\mathcal{A}$ outputs two equal length M_0, M_1 and an ID_{ch} on which it wishes to be challenged. The challenger $b \leftarrow \{0,1\}$ and sets $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}_{ch}, M_b)$

Security notions for IBE schemes

IND-ID-CPA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$

Challenge $\mathcal{A}$ outputs two equal length M_0, M_1 and an ID_{ch} on which it wishes to be challenged. The challenger $b \leftarrow \{0,1\}$ and sets $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}_{ch}, M_b)$

Phase 2 As in Phase 1, except submitting ID_{ch}.

Security notions for IBE schemes

IND-ID-CPA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$

Challenge $\mathcal{A}$ outputs two equal length M_0, M_1 and an ID_{ch} on which it wishes to be challenged. The challenger $b \leftarrow \{0,1\}$ and sets $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}_{ch}, M_b)$

Phase 2 As in Phase 1, except submitting ID_{ch}.

Guess $\mathcal{A}$ outputs a bit b' and wins if $b' = b$.

Security notions for IBE schemes

IND-ID-CCA security for an IBE scheme $\mathcal{E}$

Security notions for IBE schemes

IND-ID-CCA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Security notions for IBE schemes

IND-ID-CCA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$

Security notions for IBE schemes

IND-ID-CCA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$

Challenge $\mathcal{A}$ outputs two equal length M_0, M_1 and an ID_{ch} on which it wishes to be challenged. The challenger $b \leftarrow \{0,1\}$ and sets $C = \text{Encrypt(params,}\mathsf{ID}_{ch}, M_b)$

Security notions for IBE schemes

IND-ID-CCA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$

Challenge $\mathcal{A}$ outputs two equal length M_0, M_1 and an ID_{ch} on which it wishes to be challenged. The challenger $b \leftarrow \{0,1\}$ and sets $C = \mathsf{Encrypt}(\mathsf{params}, \mathsf{ID}_{ch}, M_b)$

Phase 2 As in Phase 1, except submitting ID_{ch}.

Security notions for IBE schemes

IND-ID-CCA security for an IBE scheme $\mathcal{E}$

Initialization The challenger runs **setup**, gives the adversary $\mathcal{A}$ the description of $\mathcal{E}$, params and keeps d_{ID} secret.

Phase 1 $\mathcal{A}$ issues adaptive queries of the type

- Extraction query $\langle \mathsf{ID}_i \rangle$
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$

Challenge $\mathcal{A}$ outputs two equal length M_0, M_1 and an ID_{ch} on which it wishes to be challenged. The challenger $b \leftarrow \{0,1\}$ and sets $C = \text{Encrypt(params,}\mathsf{ID}_{ch}, M_b)$

Phase 2 As in Phase 1, except submitting ID_{ch}.

Guess $\mathcal{A}$ outputs a bit b' and wins if $b' = b$.

Bilinear maps and bilinear groups

Let $\mathbb{G}, \mathbb{G}_T$ be prime order p abelian groups in which the discrete logarithm is believed to be hard.

Bilinear maps and bilinear groups

Let $\mathbb{G}, \mathbb{G}_T$ be prime order p abelian groups in which the discrete logarithm is believed to be hard.
By a **bilinear map** we will refer to a non-degenerate bilinear function $\widehat{t} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Bilinear maps and bilinear groups

Let $\mathbb{G}, \mathbb{G}_T$ be prime order p abelian groups in which the discrete logarithm is believed to be hard.
By a **bilinear map** we will refer to a non-degenerate bilinear function $\widehat{t} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Computational Diffie-Hellman problem on $\mathbb{G}$ Given $P, aP, bP \leftarrow \mathbb{G}$ as input, compute $abP \in \mathbb{G}$, where $a \leftarrow \mathbb{Z}_p^*$.

Bilinear maps and bilinear groups

Let $\mathbb{G}, \mathbb{G}_T$ be prime order p abelian groups in which the discrete logarithm is believed to be hard.
By a **bilinear map** we will refer to a non-degenerate bilinear function $\widehat{t} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Computational Diffie-Hellman problem on $\mathbb{G}$ Given $P, aP, bP \leftarrow \mathbb{G}$ as input, compute $abP \in \mathbb{G},$ where $a \leftarrow \mathbb{Z}_p^*$.

Decisional Diffie-Hellman problem on $\mathbb{G}$ Given $P, aP, bP, cP \leftarrow \mathbb{G}$ as input, output yes if $c = ab$ and no otherwise, where $a, b \leftarrow \mathbb{Z}_p^*$.

Bilinear maps and bilinear groups

Let $\mathbb{G}, \mathbb{G}_T$ be prime order p abelian groups in which the discrete logarithm is believed to be hard.
By a **bilinear map** we will refer to a non-degenerate bilinear function $\widehat{t} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.

Computational Diffie-Hellman problem on $\mathbb{G}$ Given $P, aP, bP \leftarrow \mathbb{G}$ as input, compute $abP \in \mathbb{G}$, where $a \leftarrow \mathbb{Z}_p^*$.

Decisional Diffie-Hellman problem on $\mathbb{G}$ Given $P, aP, bP, cP \leftarrow \mathbb{G}$ as input, output yes if $c = ab$ and no otherwise, where $a, b \leftarrow \mathbb{Z}_p^*$.

(P, aP, bP, cP) is a DH tuple iff $\widehat{t}(aP, bP) = \widehat{t}(P, abP)$.

BDH problems

BDH problems

Bilinear Diffie-Hellman (BDH) Problem on $\mathbb{G}$. Given $P, aP, bP, cP \leftarrow \mathbb{G}$ as input, compute $W = \widehat{t}(P,P)^{abc} \in \mathbb{G}_T$.

BDH problems

Bilinear Diffie-Hellman (BDH) Problem on $\mathbb{G}$. Given $P, aP, bP, cP \leftarrow \mathbb{G}$ as input, compute $W = \widehat{t}(P,P)^{abc} \in \mathbb{G}_T$.

Decision Bilinear Diffie-Hellman (DBDH) Problem on $\mathbb{G}$. Given $P, aP, bP, cP \leftarrow \mathbb{G}$ as input, and $T \leftarrow \mathbb{G}_T$,; output `yes` if $T = \widehat{t}(P,P)^{abc}$ and `no` otherwise.

Boneh-Franklin identity based encryption scheme

Basic scheme

An IND-ID-CPA is defined first.

BasicIdent

Basic scheme

An IND-ID-CPA is defined first.

BasicIdent

Setup.

- Choose $P \leftarrow \mathbb{G}$, $s \leftarrow \mathbb{Z}_p^*$ and set $P_{pub} = sP \in \mathbb{G}^*$.

Basic scheme

An IND-ID-CPA is defined first.

BasicIdent

Setup.

- Choose $P \leftarrow \mathbb{G}$, $s \leftarrow \mathbb{Z}_p^*$ and set $P_{pub} = sP \in \mathbb{G}^*$.
- Choose $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n$.

Basic scheme

An IND-ID-CPA is defined first.

BasicIdent

Setup.

- Choose $P \leftarrow \mathbb{G}$, $s \leftarrow \mathbb{Z}_p^*$ and set $P_{pub} = sP \in \mathbb{G}^*$.
- Choose $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n$.
- Set $\mathcal{M} = \{0,1\}^n$ and $\mathcal{C} = \mathbb{G}^* \times \{0,1\}^n$.

Basic scheme

An IND-ID-CPA is defined first.

BasicIdent

Setup.

- Choose $P \leftarrow \mathbb{G}$, $s \leftarrow \mathbb{Z}_p^*$ and set $P_{pub} = sP \in \mathbb{G}^*$.
- Choose $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n$.
- Set $\mathcal{M} = \{0,1\}^n$ and $\mathcal{C} = \mathbb{G}^* \times \{0,1\}^n$.
- params $= \langle p, \mathbb{G}, \mathbb{G}_T, \widehat{t}, P, P_{pub}, H_1, H_2 \rangle$.
- The master-key is $s \in \mathbb{Z}_p^*$.

Basic scheme

Extract.

- Given $\mathsf{ID} \in \{0,1\}^*$, compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}^*$.
- Set $d_{\mathsf{ID}} = sQ_{\mathsf{ID}} \in \mathbb{G}^*$.

Basic scheme

Extract.

- Given $\mathsf{ID} \in \{0,1\}^*$, compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}^*$.
- Set $d_{\mathsf{ID}} = sQ_{\mathsf{ID}} \in \mathbb{G}^*$.

Encrypt. To encrypt $M \in \{0,1\}^n$ under the public key ID

- Compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_2^*$.
- Choose $r \leftarrow \mathbb{Z}_p^*$
- Set $C = \langle rP, M \oplus H_2(g_{\mathsf{ID}}^r) \rangle$ where $g_{\mathsf{ID}} = \widehat{t}(P_{pub}, Q_{\mathsf{ID}}) \in \mathbb{G}_T$.

Basic scheme

Extract.

- Given $\mathsf{ID} \in \{0,1\}^*$, compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}^*$.
- Set $d_{\mathsf{ID}} = sQ_{\mathsf{ID}} \in \mathbb{G}^*$.

Encrypt. To encrypt $M \in \{0,1\}^n$ under the public key ID

- Compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_2^*$.
- Choose $r \leftarrow \mathbb{Z}_p^*$
- Set $C = \langle rP, M \oplus H_2(g_{\mathsf{ID}}^r) \rangle$ where $g_{\mathsf{ID}} = \widehat{t}(P_{pub}, Q_{\mathsf{ID}}) \in \mathbb{G}_T$.

Decrypt.

- $C = \langle U, V \rangle \in \mathcal{C}$
- Compute $V \oplus H_2(\widehat{t}(U, d_{\mathsf{ID}})) = M$.

Basic scheme

Extract.

- Given $\mathsf{ID} \in \{0,1\}^*$, compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}^*$.
- Set $d_{\mathsf{ID}} = sQ_{\mathsf{ID}} \in \mathbb{G}^*$.

Encrypt. To encrypt $M \in \{0,1\}^n$ under the public key ID

- Compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_2^*$.
- Choose $r \leftarrow \mathbb{Z}_p^*$
- Set $C = \langle rP, M \oplus H_2(g_{\mathsf{ID}}^r) \rangle$ where $g_{\mathsf{ID}} = \widehat{t}(P_{pub}, Q_{\mathsf{ID}}) \in \mathbb{G}_T$.

Decrypt.

- $C = \langle U, V \rangle \in \mathcal{C}$
- Compute $V \oplus H_2(\widehat{t}(U, d_{\mathsf{ID}})) = M$.

$$\widehat{t}(U, d_{\mathsf{ID}}) = \widehat{t}(rP, sQ_{\mathsf{ID}}) = \widehat{t}(P, Q_{\mathsf{ID}})^{sr} = \widehat{t}(P_{pub}, Q_{\mathsf{ID}})^r = g_{\mathsf{ID}}^r$$

Full scheme

FullIdent is obtained by applying Fujisaki-Okamoto conversion from Crypto'99 to BasicIdent

Full scheme

FullIdent is obtained by applying Fujisaki-Okamoto conversion from Crypto'99 to BasicIdent

FO conversion If we denote by $E_{\mathsf{pk}}(M, r)$ the encryption of M using randomness r under public key pk

Full scheme

FullIdent is obtained by applying Fujisaki-Okamoto conversion from Crypto'99 to BasicIdent

FO conversion If we denote by $E_{\mathsf{pk}}(M, r)$ the encryption of M using randomness r under public key pk

$$E^{hy}_{\mathsf{pk}}(M) = \langle E_{\mathsf{pk}}(\sigma, H_3(\sigma, M)), H_4(\sigma) \oplus M \rangle$$

where $\sigma \leftarrow \{0,1\}^n$.

Full scheme

FullIdent is obtained by applying Fujisaki-Okamoto conversion from Crypto'99 to BasicIdent

FO conversion If we denote by $E_{\mathsf{pk}}(M, r)$ the encryption of M using randomness r under public key pk

$$E^{hy}_{\mathsf{pk}}(M) = \langle E_{\mathsf{pk}}(\sigma, H_3(\sigma, M)), H_4(\sigma) \oplus M \rangle$$

where $\sigma \leftarrow \{0,1\}^n$.

This adds n bits to the resulting ciphertext

Full scheme (ii)

Setup.

- Choose $P \leftarrow \mathbb{G}$, $s \leftarrow \mathbb{Z}_p^*$ and set $P_{pub} = sP \in \mathbb{G}^*$.
- Choose $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*, \quad H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n,$
  $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_p^*, \; H_4 : \{0,1\}^n \rightarrow \{0,1\}^n.$
- Set $\mathcal{M} = \{0,1\}^n$ and $\mathcal{C} = \mathbb{G}^* \times \{0,1\}^n \times \{0,1\}^n$.
- params $= \langle p, \mathbb{G}, \mathbb{G}_T, \widehat{t}, P, P_{pub}, H_1, H_2, H_3, H_4 \rangle$.
- The master-key is $s \in \mathbb{Z}_p^*$.

Full scheme (iii)

Extract.

- Just as before, $d_{\mathsf{ID}} = sH_1(\mathsf{ID}) \in \mathbb{G}^*$.

Full scheme (iii)

Extract.

- Just as before, $d_{\mathsf{ID}} = sH_1(\mathsf{ID}) \in \mathbb{G}^*$.

Encrypt. To encrypt $M \in \{0,1\}^n$ under the public key ID

- Compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_*$.
- Choose $\sigma \leftarrow \{0,1\}^n$
- Set $C = \langle rP, \sigma \oplus H_2(g_{\mathsf{ID}}^r, M \oplus H_4(\sigma)) \rangle$ where $g_{\mathsf{ID}} = \widehat{t}(P_{pub}, Q_{\mathsf{ID}}) \in \mathbb{G}_T$, and $r = H_3(\sigma, M)$.

Full scheme (iii)

Extract.

- Just as before, $d_{\mathsf{ID}} = sH_1(\mathsf{ID}) \in \mathbb{G}^*$.

Encrypt. To encrypt $M \in \{0,1\}^n$ under the public key ID

- Compute $Q_{\mathsf{ID}} = H_1(\mathsf{ID}) \in \mathbb{G}_*$.
- Choose $\sigma \leftarrow \{0,1\}^n$
- Set $C = \langle rP, \sigma \oplus H_2(g_{\mathsf{ID}}^r, M \oplus H_4(\sigma)) \rangle$ where $g_{\mathsf{ID}} = \widehat{t}(P_{pub}, Q_{\mathsf{ID}}) \in \mathbb{G}_T$, and $r = H_3(\sigma, M)$.

Decrypt.

- $C = \langle U, V, W \rangle \in \mathcal{C}$
- Compute $V \oplus H_2(\widehat{t}(U, d_{\mathsf{ID}})) = M$ and $W \oplus H_4(\sigma) = M$.
- Set $r = H_3(\sigma, M)$. Check that $U = rP$. If not reject.

Security result

Theorem Let $\mathcal{A}$ an IND-ID-CCA adversary running in time t and with advantage ε against FullIdent making at most q_E private key extraction queries, q_D decryption queries and q_H hash queries. Then there is an algorithm $\mathcal{B}$ running in time roughly t that has advantage at least $\frac{\varepsilon}{q_H^2 q_D}$ against BDH problem in $\mathbb{G}$.

Security result

Theorem Let $\mathcal{A}$ an IND-ID-CCA adversary running in time t and with advantage ε against FullIdent making at most q_E private key extraction queries, q_D decryption queries and q_H hash queries. Then there is an algorithm $\mathcal{B}$ running in time roughly t that has advantage at least $\frac{\varepsilon}{q_H^2 q_D}$ against BDH problem in $\mathbb{G}$.

Bilinear Diffie-Hellman (BDH) Problem on $\mathbb{G}$. Given $P, aP, bP, cP \leftarrow \mathbb{G}$ as input, compute $W = \widehat{t}(P, P)^{abc} \in \mathbb{G}_T$.

Waters IBE scheme in the standard model

Waters scheme

Setup.

- Choose $s \leftarrow \mathbb{Z}_p^*$.
- Choose $P_2 \leftarrow \mathbb{G}$, and set $P_1 = sP \in \mathbb{G}^*$.

Waters scheme

Setup.

- Choose $s \leftarrow \mathbb{Z}_p^*$.
- Choose $P_2 \leftarrow \mathbb{G}$, and set $P_1 = sP \in \mathbb{G}^*$.
- Choose $Q' \leftarrow \mathbb{G}^*$ and a random n-length vector $U = (Q_i)$ with $Q_i \leftarrow \mathbb{G}^*$.

Waters scheme

Setup.

- Choose $s \leftarrow \mathbb{Z}_p^*$.
- Choose $P_2 \leftarrow \mathbb{G}$, and set $P_1 = sP \in \mathbb{G}^*$.
- Choose $Q' \leftarrow \mathbb{G}^*$ and a random n-length vector $U = (Q_i)$ with $Q_i \leftarrow \mathbb{G}^*$.
- Set $\mathcal{M} = \mathbb{G}_T, \quad \mathcal{C} = \mathbb{G}_T \times \mathbb{G}^* \times \mathbb{G}^*$ and $\mathcal{ID} = \{0,1\}^n$.

Waters scheme

Setup.

- Choose $s \leftarrow \mathbb{Z}_p^*$.
- Choose $P_2 \leftarrow \mathbb{G}$, and set $P_1 = sP \in \mathbb{G}^*$.
- Choose $Q' \leftarrow \mathbb{G}^*$ and a random n-length vector $U = (Q_i)$ with $Q_i \leftarrow \mathbb{G}^*$.
- Set $\mathcal{M} = \mathbb{G}_T, \quad \mathcal{C} = \mathbb{G}_T \times \mathbb{G}^* \times \mathbb{G}^*$ and $\mathcal{ID} = \{0,1\}^n$.
- params $= \langle p, \mathbb{G}, \mathbb{G}_T, \widehat{t}, P, P_1, P_2, Q', U \rangle$.
- The master-key is sP_2.

Waters scheme (ii)

Extract.

- Let ID^i denote the i-th bit of ID and $\mathcal{V} \subset \{0, \ldots, n\}$ the set of i st $\mathsf{ID}^i = 1$.
- Choose $r \leftarrow \mathbb{Z}_p^*$.
- $d_{\mathsf{ID}} = \left(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i \right)^r, rP \right)$

Waters scheme (ii)

Extract.

- Let ID^i denote the i-th bit of ID and $\mathcal{V} \subset \{0, \ldots, n\}$ the set of i st $\mathsf{ID}^i = 1$.
- Choose $r \leftarrow \mathbb{Z}_p^*$.
- $d_{\mathsf{ID}} = \left(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i \right)^r, rP \right)$

Encrypt. To encrypt $M \in \mathbb{G}_T$ under the public key ID

- Choose $x \leftarrow \mathbb{Z}_p^*$.
- Set $C = \left(\widehat{t}(P_1, P_2)^x M, xP, \left(Q' \prod_{i \in \mathcal{V}} Q_i \right)^x \right)$.

Waters scheme (iii)

Decryption. Let $C = (C_1, C_2, C_3)$ a valid encryption under ID.

- Decrypt C using $d_{\mathsf{ID}} = (d_1, d_2)$ as $C_1 \dfrac{\widehat{t}(d_2, C_3)}{\widehat{t}(d_1, C_2)}$

Waters scheme (iii)

Decryption. Let $C = (C_1, C_2, C_3)$ a valid encryption under ID.

- Decrypt C using $d_{\mathsf{ID}} = (d_1, d_2)$ as $C_1 \dfrac{\widehat{t}(d_2, C_3)}{\widehat{t}(d_1, C_2)}$

  Let $d_{\mathsf{ID}} = \left(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^r, rP\right)$ and

Waters scheme (iii)

Decryption. Let $C = (C_1, C_2, C_3)$ a valid encryption under ID.

- Decrypt C using $d_{\mathsf{ID}} = (d_1, d_2)$ as $C_1 \dfrac{\widehat{t}(d_2, C_3)}{\widehat{t}(d_1, C_2)}$

  Let $d_{\mathsf{ID}} = \left(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^r, rP\right)$ and

  $$C = \left(\widehat{t}(P_1, P_2)^x M, xP, \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^x\right), \text{ then}$$

Waters scheme (iii)

Decryption. Let $C = (C_1, C_2, C_3)$ a valid encryption under ID.

- Decrypt C using $d_{\mathsf{ID}} = (d_1, d_2)$ as $C_1 \dfrac{\widehat{t}(d_2, C_3)}{\widehat{t}(d_1, C_2)}$

  Let $d_{\mathsf{ID}} = \left(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^r, rP\right)$ and

  $$C = \left(\widehat{t}(P_1, P_2)^x M, xP, \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^x\right), \text{ then}$$

$$C_1 \frac{\widehat{t}(d_2, C_3)}{\widehat{t}(d_1, C_2)} = (\widehat{t}(P_1, P_2)^x M) \frac{\widehat{t}(rP, \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^x)}{\widehat{t}(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^r, xP)} =$$

Waters scheme (iii)

Decryption. Let $C = (C_1, C_2, C_3)$ a valid encryption under ID.

- Decrypt C using $d_{\mathsf{ID}} = (d_1, d_2)$ as $C_1 \dfrac{\widehat{t}(d_2, C_3)}{\widehat{t}(d_1, C_2)}$

Let $d_{\mathsf{ID}} = \left(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^r, rP\right)$ and

$$C = \left(\widehat{t}(P_1, P_2)^x M, xP, \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^x\right), \text{ then}$$

$$C_1 \frac{\widehat{t}(d_2, C_3)}{\widehat{t}(d_1, C_2)} = (\widehat{t}(P_1, P_2)^x M) \frac{\widehat{t}(rP, \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^x)}{\widehat{t}(sP_2 \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^r, xP)} =$$

$$(\widehat{t}(P_1, P_2)^x M) \frac{\widehat{t}(P, \left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^{rx})}{\widehat{t}(P_1, P_2)^x \widehat{t}(\left(Q' \prod_{i \in \mathcal{V}} Q_i\right)^{rx}, P)} = M.$$

Security result

Theorem Let $\mathcal{A}$ an IND-ID-CPA adversary running in time t and with advantage ε making at most q_E private key extraction queries and q_D decryption queries. Then there is an algorithm $\mathcal{B}$ running in time roughly $t + \mathcal{O}(q_E n \varepsilon^{-2} \ln(\varepsilon^{-1}) \ln(q_E n))$ that has advantage at least $\frac{\varepsilon}{32 n q_E}$ against BDDH problem in $\mathbb{G}$.

Security result

Theorem Let $\mathcal{A}$ an IND-ID-CPA adversary running in time t and with advantage ε making at most q_E private key extraction queries and q_D decryption queries. Then there is an algorithm $\mathcal{B}$ running in time roughly $t + \mathcal{O}(q_E n \varepsilon^{-2} \ln(\varepsilon^{-1}) \ln(q_E n))$ that has advantage at least $\frac{\varepsilon}{32 n q_E}$ against BDDH problem in $\mathbb{G}$.

Decision Bilinear Diffie-Hellman (DBDH) Problem on $\mathbb{G}$. Given $P, aP, bP, cP \leftarrow \mathbb{G}$ as input, and $T \leftarrow \mathbb{G}_T$,; output `yes` if $T = \widehat{t}(P, P)^{abc}$ and `no` otherwise.

Some applications of IBE schemes

- IBE schemes imply secure signature schemes
- Access control
- Delegation of decryption keys
- Strong-key insulated encryption

Some applications of IBE schemes

- IBE schemes imply secure signature schemes
- Access control
- Delegation of decryption keys
- Strong-key insulated encryption

and many more... take a look at Cryptology ePrint Archive `http://eprint.iacr.org`

Some applications of IBE schemes

- IBE schemes imply secure signature schemes
- Access control
- Delegation of decryption keys
- Strong-key insulated encryption

and many more... take a look at Cryptology ePrint Archive `http://eprint.iacr.org`

It is fair to say that identity/pairing based cryptography is currently the most active research area in cryptology

Drawbacks & Open Problems

- d_{ID} must be sent over a secure channel
- The system is inherently escrowed
  - Certificate Based encryption (Gentry)
  - Certificate-Less PKC (Al-Riyami&Paterson)
- (Mostly) Suitable for small environments
- Security reductions are inefficient
- Few schemes proven secure without the ROM

Drawbacks & Open Problems

- d_{ID} must be sent over a secure channel
- The system is inherently escrowed
  - Certificate Based encryption (Gentry)
  - Certificate-Less PKC (Al-Riyami&Paterson)
- (Mostly) Suitable for small environments
- Security reductions are inefficient
- Few schemes proven secure without the ROM

The slides of this talk are available at
`http://www.cs.ru.nl/~dgalindo`