



Identity Based Encryption: An Overview

Palash Sarkar

Indian Statistical Institute

Structure of Presentation

- Conceptual overview and motivation.
- Some technical details.
- Brief algebraic background.
- Some constructions.
- News from industry.



Conceptual Overview and Motivation

Science of Encryption

Evolution

- **Classical cryptosystems.**
 - encryption and decryption keys are same.
 - both are secret.
 - **Problems:** key distribution and management.
- **Public key cryptosystems. A paradigm shift.**
 - encryption and decryption keys are different.
 - encryption key is public; decryption key is secret.
 - **Problems:** Operational issues.

Public Key Encryption (PKE)

- Alice has two keys
 - pk_A : Available in a public directory.
 - sk_A : Kept secret by Alice.
- Bob encrypts a message using pk_A .
- Alice decrypts the ciphertext using sk_A .
- **Problem:** (Wo)man in the middle.
 - Eve impersonates Alice.
 - Puts a public key pk_E in Alice's name.
 - Eve decrypts any message encrypted using pk_E .

Digital Signature Protocol

- Consists of algorithms (Setup, Sign, Verify).
- Setup generates (pk_C, sk_C) for Charles.
- pk_C is made public (placed in a public directory).
- Charles signs message M using sk_C to obtain signature σ .
- Anybody can verify the validity of (M, σ) using pk_C .

Certifying Authority (CA)

- Consider Charles to be CA.
- Alice obtains certificate.
 - Alice generates (pk_A, sk_A) ; sends pk_A to CA.
 - CA signs (Alice, pk_A) using sk_C to obtain σ ; Alice's certificate: (Alice, pk_A, σ).
- Bob sends message M to Alice.
 - Verifies (Alice, pk_A, σ) using pk_C .
 - Encrypts M using pk_A .

CA: Operational Issues

- How long will Alice's certificate be valid?
 - CA publishes certificate status information.
 - This information has to be fresh (to a day, for example).
 - Bob has to verify that Alice's certificate has not been revoked.
- Does Bob trust Alice's CA?
 - Alice and Bob may have different CAs.
 - This may lead to a chain (or tree) of CAs.
 - CAs have to certify each other.

Public Key Infrastructure

- Consists of certifying authorities and users.
- Certificate status information.
 - Certificate revocation list (CRL).
 - Online certificate status protocol (OCSP).
 - One-way hash chains.
- A major stumbling block for widespread adoption of PKE.

Identity Based Encryption

- Alice's e-mail id `alice@gmail.com` is her public key.
- Alice authenticates herself to an “authority” and obtains the private key corresponding to this id.
- Bob uses `alice@gmail.com` and some public parameters of the “authority” to encrypt a message to Alice.
- Alice decrypts using her private key.
- No CA; no certificates; no CRLs; no chain of CAs!

Hierarchical IBE (HIBE)

“authority” is called a private key generator (PKG)

- Delegate the capability for providing private keys to lower level entities.
- This creates a hierarchy.
- There are no lower level public parameters. Only the PKG has public parameters.
- Alice obtains her private key from her “local” key generation centre.
- Bob does not have to bother about who generated Alice’s private key.

IBE Problems

- Sending Alice's private key requires a secure channel.
- Inherent key escrow: Alice's private key is known to the PKG.
- How does Alice regain her privacy?
 - Basic idea: double encryption; combine a PKE and an IBE; many subtleties to take care of.
 - Examples:
 1. Certificateless encryption.
 2. Certificate based encryption.

Historical Milestones

Classical: . . . , Enigma, DES, AES.

Public key: Diffie-Hellman, 1976.

- RSA, 1978.
- El Gamal, 1984.

IBE: Proposed by Shamir, 1984.

- Cocks, 2000 (or earlier).
- Sakai-Ohgishi-Kasahara, 2000.
- Boneh-Franklin, 2001.
Led to major research effort.

Some Technical Details

Definition of IBE

Set-Up:

Input: desired security level.

Output: PP and msk for the PKG.

Key Generation:

Input: identity ID, PP and msk.

Output: d_{ID} , the secret key for ID.

Encryption:

Input: identity ID, msg M , PP.

Output: ciphertext C .

Decryption:

Input: ID, C , d_{ID} .

Output: M or bad.

Who Does What?

- **PKG runs Set-Up.**
- **PKG runs Key Generation.**
- **Bob runs Encryption.**
- **Alice runs Decryption.**

Adversary Does What?

Intuitive goals of an adversary.

- Get the master secret key of the PKG.
- Get the decryption key of Alice.
- Try to decipher a ciphertext intended for Alice.
- Try to distinguish a ciphertext from a random string.
 - Obtain the decryption keys of some other persons.
 - Ask Alice to decrypt a few other (possibly mal-formed) ciphertexts.

Modelling Paranoid Security

- **Adversarial goal: Weak.**
Distinguish a ciphertext from a random string with non-negligible probability.
 - Allowed to obtain other decryption keys.
 - Allowed to ask Alice for decryption of other ciphertexts.
- **Adversarial resources: maximum practicable.**
Probabilistic algorithm.
 - **Asymptotic setting:** polynomial time (in the security parameter) computation.
Concrete setting: relate success probability to running time.

Security Definition

Game between adversary and simulator.

Set-Up: **simulator**

- Generates PP and msk.
- Provides the adversary with PP.
- Keeps msk secret.

Phase 1: **adversarial queries.**

- **Key extraction oracle:** ask for the key of any identity.
- **Decryption oracle:** ask for the decryption of any ciphertext on any identity.
- **Restriction:** cannot ask for decryption using ID, if a key for ID has been asked earlier.

Security Definition (contd.)

Challenge:

- Adversary outputs ID^* and two equal length messages M_0 and M_1 .
- Adversary should not have asked for the private key of ID^* .
- Simulator chooses a random bit b ; encrypts M_b using ID^* to obtain C^* ; gives C^* to the adversary.

Phase 2: adversarial queries.

- Same as Phase 1.
- More restrictions:
cannot ask for the private key of ID^* ;
cannot ask for the decryption of C^* under ID^* .

Security Definition (contd.)

Guess:

- adversary outputs a bit b' ;
- adversary wins if $b = b'$.

Advantage:

$$\epsilon = |\Pr[b = b'] - 1/2|.$$

(ϵ, t) -adversary: running time t ; advantage ϵ .

Security Definition (contd.)

- Strongest definition:
Full model: adaptive-ID and CCA-secure.
- Weaker definitions:
 - Adaptive-ID and CPA-secure.
Adversary not provided with the decryption oracle.
 - Selective-ID.
Adversary has to commit to the target identity even before the protocol is set-up.
 - CPA-secure.
 - CCA-secure.



Brief Algebraic Background

Bilinear Map

$$e : G_1 \times G_1 \rightarrow G_2.$$

- G_1, G_2 are cyclic groups of same prime order p ;
- G_1 : additively written, $G_1 = \langle P \rangle$;
- G_2 : multiplicatively written.
- **Known examples:** Weil and Tate pairings.
 - G_1 : subgroup of an elliptic curve group.
 - G_2 : subgroup of the multiplicative group of a finite field.

Bilinear Map: Properties

Binlinearity:

$$e(aP, bP) = e(P, P)^{ab}.$$

Non-degeneracy: $e(P, P) \neq 1$.

Computability: $e(Q, R)$ can be “efficiently”
computed.

Hardness Assumption

Bilinear Diffie-Hellman Problem (BDH)

Instance: (P, aP, bP, cP) .

Task: compute $e(P, P)^{abc}$.

Decisional Bilinear Diffie-Hellman Problem (DBDH)

Instance: (P, aP, bP, cP, Z) .

Task: Decide between

- $Z = e(P, P)^{abc}$ (i.e., Z is real)
- Z is random.

Several variants of the DBDH assumption are also used.

Some Constructions

Boneh-Franklin IBE

- **Setup:** $\langle P \rangle = G_1$, $s \xleftarrow{\$} \mathbb{Z}_p$, $P_{\text{pub}} = sP$
 $\text{PP} = \langle P, P_{\text{pub}}, H_1(), H_2() \rangle$, $\text{msk} = s$.
- **Key-Gen:** Given ID compute $Q_{\text{ID}} = H_1(\text{ID})$,
 $d_{\text{ID}} = sQ_{\text{ID}}$.
- **Encrypt:** Choose $r \xleftarrow{\$} \mathbb{Z}_p$,
 $C = rP, M \oplus H_2(e(Q_{\text{ID}}, P_{\text{pub}})^r)$
- **Decrypt:** Given $C = \langle U, V \rangle$ and d_{ID} compute
 $V \oplus H_2(e(d_{\text{ID}}, U)) = M$.

The Pairing Magic

Public parameter: $p_{\text{pub}} = sP$.

Decryption key: $d_{\text{ID}} = sQ_{\text{ID}}$.

Encryption Mask: $e(Q_{\text{ID}}, P_{\text{pub}})^r$.

Decryption Mask: $e(Q_{\text{ID}}, P_{\text{pub}})^r$.

Correctness:

$$\begin{aligned} e(d_{\text{ID}}, U) &= e(sQ_{\text{ID}}, rP) \\ &= e(Q_{\text{ID}}, sP)^r \\ &= e(Q_{\text{ID}}, P_{\text{pub}})^r. \end{aligned}$$

BF-IBE (contd.)

- Basic construction: CPA-secure.
- Can be converted to CCA-secure protocol.
- Corrected analysis due to Galindo.
- Drawbacks.
 - Assumes all the hash functions to be random functions.
 - Has a large security degradation.
- Simple and elegant: but, really a “proof of concept”

Subsequent Work

Goal: Remove the random oracle heuristic.

- **Weaker security model:**
 - **selective-id:** Canetti-Halevi-Katz, 2003;
construction: Boneh-Boyen, 2004;
 - **generalised selective-id (model and construction):** Chatterjee-Sarkar, 2006.
- **Stronger hardness assumptions:**
the instance contains more information.
 - **DBDHE:** Boneh-Boyen, 2005;
special case (mBDDH): Kiltz-Vahlis, 2008.
 - **q -ABDHE:** Gentry, 2006.
 - **Others.**

Subsequent Work (contd.)

- Adaptive-id, CPA-secure IBE:
 - Boneh-Boyen, 2004.
 - Waters, 2005.
A very important work for several reasons.
 - Chatterjee-Sarkar (2006), Naccache (2006).
Improvement of Waters protocol.
- Adaptive-id, CPA-secure HIBE:
 - Gentry-Silverburg, 2002: uses random oracles.
 - Waters, 2005.
 - Chatterjee-Sarkar, 2006: most efficient till date.

From CPA to CCA-Security

- Canetti-Halevi-Katz, 2003: generic construction.
- Boneh-Katz, 2005: generic construction with efficiency improvement.
- Sarkar-Chatterjee, 2007: generic construction with further efficiency improvement.
- Boyen-Mei-Waters, 2005: non-generic, but applies to many protocols.

Construction Goals

- Full model security:
 - adaptive-id and CCA-security.
- Weak assumptions:
 - no random oracles;
 - DBDH assumption (basic assumption in the area).
- Efficiency:
 - desired security level;
 - trade-off between memory and time;

Current best

Sarkar-Chatterjee (2007).

Adaptive-id, CCA-secure.

DBDH assumption, without random oracles.

- Based on Chatterjee-Sarkar extension of Waters CPA-secure IBE.
- Incorporates BMW techniques to achieve CCA-security.
- Uses hybrid encryption.
- Uses a few other techniques.
- Can be used to obtain a HIBE.

Set-Up

1. Choose α randomly from \mathbb{Z}_p .
2. Set $P_1 = \alpha P$.
3. Choose $P_2, U'_1, U_1, \dots, U_l$ randomly.
4. Choose W randomly from G_1 .
5. $H_s : \{1, \dots, h\} \times G_1 \rightarrow \mathbb{Z}_p$
is randomly chosen from a UOWHF.
6. Public parameters:
 $P, P_1, P_2, U'_1, U_1, \dots, U_l$ and W .
7. Master secret key: αP_2 .

Key Generation

Identity $ID = (ID_1, \dots, ID_l)$, each ID_i is an (n/l) -bit string. (Waters' proposal $l = n$.)

(modified) Waters hash.

$$V(ID) = U'_1 + \sum_{i=1}^l ID_i U_i.$$

1. Choose r randomly from \mathbb{Z}_p .
2. $d_0 = \alpha P_2 + rV(ID)$.
3. $d_1 = rP$;
4. Output $d_{ID} = (d_0, d_1)$.

Encryption

Identity ID; message M .

1. Choose t randomly from \mathbb{Z}_p .
2. $C_1 = tP$, $B = tV(\text{ID})$.
3. $K = e(P_1, P_2)^t$.
4. $(\text{IV}, dk) = \text{KDF}(K)$.
5. $(\text{cpr}, \text{tag}) = \text{AE.Encrypt}_{dk}(\text{IV}, M)$.
6. $\gamma = \mathbf{H}_s(\mathbf{j}, \mathbf{C}_1)$; $\mathbf{W}_\gamma = \mathbf{W} + \gamma\mathbf{P}_1$; $\mathbf{C}_2 = t\mathbf{W}_\gamma$.
7. Output $(C_1, C_2, B, \text{cpr}, \text{tag})$.

Decryption

Identity ID;

ciphertext $(C_1, C_2, B, \text{cpr}, \text{tag})$;

decryption key $d_{\text{ID}} = (d_0, d_1)$.

1. $\gamma = \mathbf{H}_s(\mathbf{j}, \mathbf{C}_1)$; $\mathbf{W}_\gamma = \mathbf{W} + \gamma \mathbf{P}_1$.
2. *If $e(\mathbf{C}_1, \mathbf{W}_\gamma) \neq e(\mathbf{P}, \mathbf{C}_2)$ return \perp .*
3. $K = e(d_0, C_1) / e(B, d_1)$.
4. $(\text{IV}, dk) = \text{KDF}(K)$.
5. $M = \text{AE.Decrypt}_{dk}(\text{IV}, C, \text{tag})$.
(This may abort and return \perp).
6. Output M .

Security

$(\epsilon_{ibe}, t, q_{ID}, q_C)$ -secure.

$$\epsilon_{ibe} \leq 2\epsilon_{uowhf} + \frac{\epsilon_{dbdh}}{\lambda} + 4\epsilon_{kdf} + \epsilon_{enc} + 2q_C\epsilon_{auth}.$$

- ϵ_{xxx} denotes advantage of an adversary in breaking component XXX.
- $\lambda \approx 1/(8ql2^{n/l})$, $q = q_{ID} + q_C$.
- Security degradation (with respect to ϵ_{dbdh}) is $1/\lambda \approx 8ql2^{n/l}$.

Efficiency

Recall $e : G_1 \times G_1 \rightarrow G_2$.

- Public parameters: $(l + 4)$ elements of G_1 ; 1 element of G_2 .
- Decryption key: 2 elements of G_1 .
- Key generation: $2[\text{SM}] + 1[\text{H}_{n,l}]$.
- Encryption: $4[\text{SM}] + 1[\text{e}] + 1[\text{H}_{n,l}]$.
- Decryption: $1[\text{SM}] + 1[\text{VP}] + 2[\text{P}]$.
- Cost of symmetric operations not mentioned.

$[\text{SM}]$: scalar multiplication in G_1 ; $[\text{e}]$: exponentiation in G_2 ; $[\text{P}]$: pairing; $[\text{VP}]$: pairing based verification; $[[\text{H}_{n,l}]]$: modified Waters hash.

News From Industry

Companies and Products

- Voltage Security: USA based.
 - Secure e-mail.
 - Uses BF-IBE.
 - Boneh and his students are founders.
- Identum: UK based.
 - Secure e-mail.
 - Uses SK-IBE.
 - Smart (University of Bristol) is one of the technical advisors.

Standards

IEEE P1363.3 standard.

- Boneh-Franklin: secure under random oracle heuristic.
- Boneh-Boyen: selective-id security.
- Chen et al (modified Sakai-Kasahara): secure under random oracle heuristic.

IETF standard.

- Boneh-Boyen: selective-id security.
- others

Indian Scenario

- Market for crypto products.
 - Huge and (mostly) untapped.
 - Lack of crypto awareness; security does not come for free.
- Indian crypto industry: **lack of vision.**
 - Import and sell approach.
 - Development requires major investment; recruit and retain super specialised people; high salary levels; (higher than financial market!)
- Academic administration: **sluggish.**
Prevents meaningful industry interaction.

Thank you for your kind attention!