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 the 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.

X.509 Certificates

Structure.

- version number
- serial number
- signature algorithm ID
- issuer name
- validity period
- subject name (i.e., certificate owner)
- certificate owner’s public key
- optional fields
- the CA’s signature on all previous fields

Setting Up an SSL Session

- Hello: I am Alice (client); I am Bob (server); agree on specific cryptographic algorithms to be used during the session;
- Bob sends his certificate to Alice;
- Alice verifies certificate using CA's public key;
- Alice generates a random master secret key MS;
- Alice encrypts MS using Bob's public key and sends to Bob;
- Using MS, both Alice and Bob generate two keys K_1 and K_2.
- K_1: used for authentication;
  K_2: used for encryption.

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.

Certificate Revocation Lists

- CA periodically issues the list of revoked certificates.
  - Delta-CRL: incremental update;
  - Example: issue new CRL every month and delta-CRL every day.
- High transmission cost:
  complete list must be downloaded by any party who wants to check the status of a certificate.

OCSP

- CA maintains an online server.
- Responds to any certificate status query by generating a fresh signature on the current status.
- Reduces transmission cost to a single signature per query.
- Substantially increases computation load for the server.
    - Vulnerable to a denial-of-service attack if server is centralized;
    - If the service is distributed, then compromising any server compromises the entire system.

One-Way Hash Chains

"Novomodo" (Micali): simplified description.

- Suppose Alice's certificate is to be valid for n days.
- For Alice, CA chooses a random value X_0 and computes

$$X_1 = H(X_0), X_2 = H(X_1), \ldots, X_n = H(X_{n-1});$$

H is a one-way hash function.

- Puts X_n in Alice's certificate, i.e.,

$$(\mathsf{Alice}, pk_A, X_n, \mathsf{sign}_{sk_C}(\mathsf{Alice}, pk_A, X_n)).$$

One-Way Hash Chains (contd.)

- If Alice's certificate is valid on the i-th day, CA sends X_{n-i} to the directories; otherwise it does not.
- Bob checks freshness by reading X_{n-i} and verifying

$$X_n \stackrel{?}{=} H^i(X_{n-i}).$$

- Advantages.
  - Computational: hashing is much faster than signing.
  - Transmission: the directory's response to a status query is X_{n-i};
  - Security: the directories need not be trusted.

Identity Based Encryption

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.

Some Historical Milestones

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

Public key: Diffie-Hellman, 1976.

- RSA, 1978.
- El Gamal, 1984.
- Cramer-Shoup, 1998.

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.
- Indistinguishability of ciphertext distributions.
  - 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.**

Notion of indistinguishability.

- Let M_0 and M_1 be two distinct equal length messages.
- Let $\mathcal{C}_0$ be the set of all ciphertexts which can arise from M_0. Similarly define $\mathcal{C}_1$.
- Task: given C from $\mathcal{C}_b$, for a randomly chosen b, determine b.

Oracles.

- Allowed to obtain other decryption keys.
- Allowed to ask Alice for decryption of other ciphertexts.

Modelling Paranoid Security

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:

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

(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.

Gap DH Groups

Consider DDH in G_1.

- Instance: (P, aP, bP, Z).
- Verify

$$e(P, Z) \stackrel{?}{=} e(aP, bP).$$

- Verification succeeds iff $Z = abP$.

Thus, G is a group where it is *easy* to solve DDH but *hard* to solve CDH.

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.

DBDH Advantage

Let $\mathcal{A}$ be a probabilistic algorithm

- input: $(P, P_1, P_2, P_3, Z) \in G_1^4 \times G_2$;
- output: a bit b (denoted by $\mathcal{A} \Rightarrow b$).

Advantage of $\mathcal{A}$.

$$\begin{aligned}&\mathsf{Adv}(\mathcal{A})\\ &\quad = |\Pr[\mathcal{A} \Rightarrow 1 | Z \text{ is real}]\\ &\qquad - \Pr[\mathcal{A} \Rightarrow 1 | Z \text{ is random}|.\end{aligned}$$

$\mathsf{Adv}(t)$ is the supremum of advantages over all algorithms $\mathcal{A}$ running in time at most t.

DBDH is (ϵ, t)-hard if $\mathsf{Adv}(t) \leq \epsilon$.

Joux's Key Agreement Protocol

3-party, single-round.

- Three users U_1, U_2 and U_3;
- U_i chooses a uniform random r_i and broadcasts $X_i = r_i P$;
- U_i computes $K = e(X_j, X_k)^{r_i}$, where $\{j, k\} = \{1, 2, 3\} \setminus \{i\}$;

$$K = e(P, P)^{r_1 r_2 r_3}.$$

Some Constructions

Cocks' IBE

- $N = pq$;
- $J(N)$: set of elements with Jacobi symbol 1 modulo N;
- $QR(N)$: set of quadratic residues modulo N.

Public Parameters.

- N; $u \stackrel{\$}{\leftarrow} J(N) \setminus QR(N)$;
  u is a random pseudo-square;
- hash function $H()$ which maps identities into $J(N)$.

Master Secret Key: p and q.

Cocks' IBE (contd.)

Key Generation for ID:

- $R = H(\mathsf{ID})$;
- $r = \sqrt{R}$ or $\sqrt{uR}$ according as R is square or not;
- secret key corresponding to ID is $d_{\mathsf{ID}} = r$.

Cocks' IBE (contd.)

Encryption of a bit m using an identity ID.

- $R = H(\mathsf{ID})$; $t_0, t_1 \stackrel{\$}{\leftarrow} \mathbb{Z}_N$;
- compute $d_a = (t_a^2 + u^a R)/t_a$ and $c_a = (-1)^m \cdot (\frac{t_a}{N})$;
- ciphertext: $((d_0, c_0), (d_1, c_1))$.

Decryption of $((d_0, c_0), (d_1, c_1))$ using ID and $d_{\mathsf{ID}} = r$:

- $R = H(\mathsf{ID})$; set $a \in \{0, 1\}$ such that $r^2 = u^a R$;
- set $g = d_a + 2r$; (note $g = \left(\frac{(t_a+r)^2}{t_a}\right)$ and so, $(\frac{g}{N}) = (\frac{t_a}{N})$;)
- compute $(-1)^m$ to be $c_a \cdot (\frac{g}{N})$.

Cocks IBE: Issues

- One main problem: size of the ciphertext is very large; two elements of $\mathbb{Z}_N$ per bit.
- Boneh, Gentry and Hamburg:
  1. An IBE which encrypts a single bit. (A general description of which the Cocks-IBE is *not* an instantiation.)
  2. Reuse of randomness for encrypting more than one bit.
- Significantly reduces the size of the ciphertext.
- Trade-off: substantial increase in encryption time.
- Better balance: ongoing research work.

Boneh-Franklin IBE

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

The Pairing Magic

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

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

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

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

Correctness:

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

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.

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.
- Boyen-Mei-Waters, 2005: non-generic, but applies to many protocols.

Basic Setting

Full model security:

- adaptive-id and CCA-security.

Assumptions:

- DBDH assumption
  (basic assumption in the area);
- no random oracles.

Efficiency:

- speed of encryption/decryption/key generation;
- size of keys and public parameters;
- depends on desired security level;

Basic Setting: Protocol

Sarkar-Chatterjee (2007).

- 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.

Currently known most efficient protocol in the basic setting.

Set-Up

Pairing: $e : G_1 \times G_1 \to G_2$, $G_1 = \langle P \rangle$.

PP: $P, P_1, P_2, U'_1, U_1, \ldots, U_l$ and W.

- $P_1 = \alpha P$, where $\alpha \xleftarrow{\$} \mathbb{Z}_p$;
- $P_2, U'_1, U_1, \ldots, U_l$ and W are random elements of G_1;
- $H_s : G_1 \to \mathbb{Z}_p$ is randomly chosen from a UOWHF.

Master secret key: αP_2.

Key Generation

Identity $\mathsf{ID} = (\mathsf{ID}_1, \ldots, \mathsf{ID}_l)$, each ID_i is an (n/l)-bit string, considered to be an element of $\mathbb{Z}_{2^{n/l}}$.

<u>(modified) Waters hash.</u>

$$V(\mathsf{ID}) = U_1' + \sum_{i=1}^{l} \mathsf{ID}_i U_i.$$

(Waters' proposal: $l = n$.)

$d_{\mathsf{ID}} = (d_0, d_1)$.

- $d_0 = \alpha P_2 + rV(\mathsf{ID})$, where $r \xleftarrow{\$} \mathbb{Z}_p$.
- $d_1 = rP$.

Encryption

Input: Identity ID; message M.
Output: $(C_1, C_2, B, \mathsf{cpr}, \mathsf{tag})$.

- $C_1 = tP$, $B = tV(\mathsf{ID})$, where $t \stackrel{\$}{\leftarrow} \mathbb{Z}_p$.
- $K = e(P_1, P_2)^t$.
- $(\mathsf{IV}, dk) = \mathsf{KDF}(K)$.
- $(\mathsf{cpr}, \mathsf{tag}) = \mathsf{AE.Encrypt}_{dk}(\mathsf{IV}, M)$.
- $\gamma = H_s(C_1)$; $W_\gamma = W + \gamma P_1$.
- $C_2 = tW_\gamma$.

Decryption

Input: Identity ID; ciphertext $(C_1, C_2, B, \mathsf{cpr}, \mathsf{tag})$.
Output: Message M or bad.

- $\gamma = H_s(C_1)$; $W_\gamma = W + \gamma P_1$.
- If $e(C_1, W_\gamma) \neq e(P, C_2)$ return $\perp$.
- $K = e(d_0, C_1)/e(B, d_1)$.
- $(\mathsf{IV}, dk) = \mathsf{KDF}(K)$.
- $M = \mathsf{AE.Decrypt}_{dk}(\mathsf{IV}, C, \mathsf{tag})$.
  (This may abort and return $\perp$).

Correct Decryption

- The test $e(C_1, W_\gamma) \stackrel{?}{=} e(P, C_2)$,
  $C_1 = tP$ and $C_2 = tW_\gamma$

$$
\begin{aligned}
e(C_1, W_\gamma) &= e(tP, W_\gamma) \\
&= e(P, tW_\gamma) \\
&= e(P, C_2).
\end{aligned}
$$

Correct Decryption (contd.)

- Reconstruction of K.
  During encryption: $K = e(P_1, P_2)^t$.
  During decryption:

$$
\begin{aligned}
K &= \frac{e(d_0, C_1)}{e(B, d_1)} \\
&= \frac{e(\alpha P_2 + rV(\mathsf{ID}), tP)}{e(tV(\mathsf{ID}), rP)} \\
&= e(\alpha P_2, tP) \times \frac{e(rV(\mathsf{ID}), tP)}{e(tV(\mathsf{ID}), rP)} \\
&= e(P_1, P_2)^t.
\end{aligned}
$$

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[SM]+1[$H_{n,l}$].
- Encryption: 4[SM]+1[e]+1[$H_{n,l}$].
- Decryption: 1[SM]+1[VP]+2[P].
- Cost of symmetric operations not mentioned.

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

Security

A proof is given to show that the scheme is secure assuming

- DBDH problem is hard;
- H_s is a secure UOWHF;
- KDF is a secure key derivation function;
- AE provides both privacy and authenticity.

A rather long and complex proof is used to show this.

The techniques and ideas used in the proof have evolved gradually in several papers.

Security

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

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

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

News From the 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; (possibly higher than financial jobs!)
- Academic administration: **sluggish.** Prevents meaningful industry interaction.

Thank you for your kind attention!