In this lecture, we study the IBE scheme proposed by Boneh-Franklin in 2001, which is the first practical IBE scheme with formal proof in the random oracle model. We first give a proof of security against selective-identity and chosen-plaintext attack (CPA). We then show how to extend the proof to adaptive-identity security. For security against chosen-ciphertext attack (CCA), we discuss a generic CPA-to-CCA transformation due to Fujisaki-Okamoto. Finally, we discuss the subtle differences between using “Plug-and-Pray” argument in computational and decisional attack game.

1 Semantically-Secure Identity-Based Encryption

We start by reviewing the properties of bilinear pairing, which is the underlying cryptographic primitive of the IBE scheme to be discussed, and a related intractability assumption.

1.1 Bilinear Pairings and Related Intractability Assumption

Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of prime order $q$. The bilinear pairing is given as $\hat{e} : G_1 \times G_1 \rightarrow G_2$, which satisfies the following properties:

1. **Bilinearity**: For all $P, Q, R \in G_1$, $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$, and $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$.

2. **Non-degeneracy**: There exists $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$.

3. **Computability**: It is efficient to compute $\hat{e}(P, Q) \forall P, Q \in G_1$.

**Definition 1 (Bilinear Diffie-Hellman Problem)** The Bilinear Diffie-Hellman (BDH) problem in $(G_1, G_2, \hat{e})$ is defined as follows: Given $\langle P, rP, sP, tP \rangle$ for some $r, s, t \in Z_q^*$, compute $\hat{e}(P, P)^{rst} \in G_2$.

**Assumption 1 (Bilinear Diffie-Hellman Assumption)** The advantage of an algorithm $A$ in solving the BDH problem is defined as

$$\Pr[A(q, G_1, G_2, \hat{e}, P, rP, sP, tP) = \hat{e}(P, P)^{rst} | \langle q, G_1, G_2, \hat{e} \rangle \leftarrow G(1^k), P \leftarrow G_1^*, r, s, t \leftarrow Z_q^*]$$

where $G$ is a BDH parameter generator such that the security parameter $k$ is used to determine the size for $q$. BDH assumption says that for all probabilistic polynomial time algorithm $A$, the above probability is a negligible function in $k$. 

L12-1
1.2 Boneh-Franklin Identity-Based Encryption

Below gives the basic version of the IBE scheme proposed by Boneh and Franklin [BF01, BF03] that is CPA-secure against adaptive-identity attack.

The system parameters are: \( \{ G_1, G_2, \hat{e}(\cdot, \cdot), q, P, H_1(\cdot), H_2(\cdot), \ell \} \), where \( P \neq O \in G_1 \) is a generator of the group \( G_1 \), \( \ell \) is the maximum length of the message to be encrypted, \( H_1 \) and \( H_2 \) are two cryptographic hash function \( H_1: \{0,1\}^* \rightarrow G_1 \) and \( H_2: G_2 \rightarrow \{0,1\}^\ell \). They are modelled as random oracles in the security proof. For brevity, we omit the inclusion of the system parameters for the rest of the algorithms.

**KeyGen:** The Secret Key Authority (SKA) randomly chooses \( s \in_R \mathbb{Z}_q^* \), keeps it as the master secret key, and computes the corresponding master public key \( Q_s = sP \).

**SKExtract:** Given the mater secret key \( s \), and an identity \( ID \in \{0,1\}^* \) which an user submits to the SKA, the SKA sets the user’s public key \( Q_{ID} \) to be \( H_1(ID) \in G_1 \) and computes the user’s private signing key \( SK_{ID} \) by \( S_{ID} = sQ_{ID} \). Then SKA sends this user private key to the user via a secure channel.

**Encrypt:** Given the master public key \( Q_s \), recipient’s identity \( ID \in \{0,1\}^* \) and a message \( M \in \{0,1\}^\ell \), a random element \( r \in \mathbb{Z}_q^* \) is chosen, the ciphertext is as follows.

\[
C = \langle rP, M \oplus H_2(\hat{e}(Q_s, Q_{ID})^r) \rangle
\]

**Decrypt:** Given the user’s secret key \( SK_{ID} \) and a ciphertext parsed as \( C = \langle U, V \rangle \), the message \( M’ \) can be recovered \( V \oplus H_2(\hat{e}(U, SK_{ID})) \).

Correctness can be seen from \( \hat{e}(Q_s, Q_{ID})^r = \hat{e}(r(sP), Q_{ID}) = \hat{e}(rP, sQ_{ID}) = \hat{e}(U, SK_{ID}) \).

1.3 Semantic Security under Selective-Identity Attack

For the ease of understanding, we first provide a proof of CPA-security (aka semantic security) under a restricted model of selective-identity attack, in which the adversary is required to tell the challenger which identity to be attacked before seeing the system parameter.

Specifically, the game between the challenger \( C \) and the adversary \( A \) proceeds as follows.

1. (Identity Chosen) \( A \) gives \( C \) an identity \( ID \) to be attacked (target identity).
2. (Parameter Generation) \( C \) selects the system parameters \( param \), executes KeyGen to get \( (MPK, MSK) \), keeps MSK in secret and gives \( (param, MPK) \) to \( A \).
3. (Extract Phase 1) \( A \) can adaptively issue a polynomial number of SKExtract queries; the only restriction is that \( ID \) should not appear in any of these.
4. (Encryption Query) \( A \) gives \( C \) two equal-length messages \( M_0 \) and \( M_1 \). \( C \) returns Encrypt\((MPK, ID, M_b)\) to \( A \), where \( b \) is a random bit chosen by \( C \).

---

1. In general, the master public key \( Q_s \) may be required as input, but it is not necessary in this scheme.
5. (Extract Phase 2) Same as Phase 1, $A$ can adaptively issue a polynomial number of $SKExtract$ queries; the only restriction is that $ID$ should not appear in any of these.

6. (Guess) $A$ outputs a bit $b'$. $A$ wins the game if and only if $b' = b$.

**Theorem 1** Boneh-Franklin’s IBE scheme is CPA-secure under selective-identity attack.

**Proof:** Suppose the challenge instance of the BDH problem is $\langle P, r_P, s_P, t_P \rangle$, and $ID$ is the identity to be attacked by the adversary $A$. The challenger $C$ simulates $H_1$ as below.

All $SKExtract$ queries, except the one on $\tilde{ID}$ that the adversary is not allowed to ask, can be simulated correctly by $\tilde{t}(s_P)$. Correctness can be easily seen from $\tilde{t}(s_P) = s \tilde{t}_P = sQ_{\tilde{ID}}$.

Simulation of $H_1$

1. Input: $\tilde{ID}$
2. if $\tilde{ID} = ID$ then
3. return $t_P$;
4. else
5. $\tilde{t} \leftarrow Z_q$;
6. Store $(ID, \tilde{t})$;
7. return $\tilde{t}P$
8. end if

To process encryption query, $C$ selects a random bit $b$ and returns $\langle r_P, M_b \oplus H_2(\hat{e}(P, P)^{rst}) \rangle$.

Now we prove security using the sequence-of-games approach.

In Game 0 (the original attack game), the $H_2$ oracle is correctly implemented. In particular, we have the following steps in the simulation.

(Part of the) Initialization of $H_2$

1. $k \leftarrow \{0, 1\}^\ell$;
2. $H_2(\hat{e}(P, P)^{rst}) \leftarrow k$;

The above initialization is possible since the value of $r, s, t$ are still known in Game 0. In any case, $k$ will be used in the encryption query, i.e. $\langle r_P, M_b \oplus k \rangle$ is returned.

In Game 1 (dropping the consistency of $H_2$), the simulation is the same as that of Game 1 except the line “$H_2(\hat{e}(P, P)^{rst}) \leftarrow k$” is dropped in the aforementioned $H_2$ initialization.

If $A$ can notice any difference in these two games, we can use $A$ to solve our BDH problem. The advantage of $A$ in Game 1 is 0 as $k$ is an one-time pad that appears nowhere else except in the encryption query. Our theorem is proven under the BDH assumption. \qed

We can prove the above theorem under the decisional variant of BDH assumption (i.e. deciding between $\hat{e}(P, P)^{rst}$ and a random $G_2$ element is difficult) by using a key derivation function (KDF)\(^1\) for $H_2$, instead of assuming it as a random oracle (which is a perfect KDF).

\(^1\)Informally, given a string $x$ sampled from an arbitrary distribution, together with a uniformly distributed randomized, a KDF outputs a string of a fixed length.

L12-3
1.4 Semantic Security under Adaptive-Identity Attack

Under an adaptive-identity attack, the target identity to be attacked is chosen when the adversary issues the encryption query, instead of at the very beginning of the game. This can be done by using “plug-and-pray” argument. We assume \(\text{ID}^1, \text{ID}^2, \ldots, \text{ID}^n\) are the queries to \(H_1\), i.e. \(A\) makes \(n\) distinct \(H_1\) queries in total. \(C\) chooses \(\omega \in \{1, \ldots, n\}\) at random, plugs \(H_1(\text{ID}^\omega) \leftarrow tP\) and plugs \(H_1\) everywhere else with random \(G_1\) elements of known discrete logarithm with respect to \(P\). \(C\) then prays for \(A\) choosing \(\text{ID}^\omega\) as the target identity. If this wish comes true, \(C\)’s can embed the BDH instance to the encryption query.

2 CCA Security from CPA Asymmetric Encryption

Here we describe a generic and simple conversion due to Fujisaki and Okamoto [FO99] which integrates CPA-secure public and CCA-secure symmetric encryption schemes into a CCA-secure public encryption scheme, in the random oracle model. For the ease of understanding, we only give a simplified variant of Fujisaki-Okamoto transformation.

2.1 Motivation

The original motivation for this transformation technique is to construct a CCA-secure scheme from primitives which are secure in a weaker sense. Before their work, there exists a transformation known as OAEP (Optimal Asymmetric Encryption Padding) that converts a one-way trapdoor permutation (OWTP) into a CCA-secure public key encryption scheme. This method gives raise to “Hashed RSA” encryption (OWTP can be instantiated by RSA primitive), but not many other primitives. For example, “Hashed ElGamal” is not possible since a decisional Diffie-Hellman oracle will be required.

Note that Fujisaki-Okamoto transformation is not completely generic since it requires a “reasonable” (to be explained below) semantically secure public key encryption scheme.

2.2 Conversion

Suppose \((E, D)\) is a public key encryption scheme, and \((E, D)\) is a CCA-secure symmetric scheme (many decryption queries but only one encryption query). The later can be built from universal one-way hash function and pairwise-independent hash function. We use the notation \(E_{PK}(x; s)\) to denote a public key encryption of the message \(x\) using random coin \(s\) under the public key \(pk\). The system parameter includes two cryptographic hash \(H_3\) and \(H_4\), where \(H_3\) maps bit-strings of a certain length (says \(\{0, 1\}^f\)) to the random coin spaces of \((E, D)\) and \(H_4\) maps \(\{0, 1\}^f\) to the key space of \((E, D)\).

**Encrypt:** Given a message \(m\), pick \(\sigma \leftarrow \{0, 1\}^f\), the ciphertext is \(\langle E_{PK}(\sigma; H_3(\sigma)), E_{H_4(\sigma)}(m) \rangle\).

**Decrypt:** Given a ciphertext \((C_a, C_s)\) and a secret key \(SK\), decryption goes as below.

1. Compute \(\sigma' \leftarrow D_{SK}(C_a)\);
2. Reject if \(C_a \neq E_{PK}(\sigma'; H_3(\sigma'))\);
3. Otherwise, return \( m' \leftarrow D_{H_4(\sigma')}(C_s) \).

### 2.3 Security Proof

**Theorem 2** The above public key encryption scheme is indistinguishable under adaptive chosen ciphertext attack if \((E, D)\) is a public key encryption scheme that is indistinguishable under adaptive chosen plaintext attack and \((E, D)\) is a symmetric encryption scheme that is indistinguishable under adaptive chosen ciphertext attack.

**Proof:** Let \( S_i \) be the event that the adversary \( A \) wins Game \( i \) and \( F_i \) be the event that adversary \( A \) aborts in Game \( i \), due to any unfaithful simulation of \( C \).

In **Game 0** (original game), the simulator \( S \) implements \( H_3(\cdot) \) and \( H_4(\cdot) \) as two associative arrays. In particular, \( \sigma, r, k \) are chosen at random from the appropriate domains, and \( H_3(\sigma) \leftarrow r \) and \( H_4(\sigma) \leftarrow k \) are put into the arrays. Encryption query returns \( C = \langle C_a, C_s \rangle \leftarrow \langle E_{PK}(\sigma, r), E_k(m_b) \rangle \) where \( b \) is a randomly chosen bit.

**Game 1** modifies the decryption oracle as follows.

**Decryption Oracle**

1. **Input:** \( \langle \tilde{C}_a, \tilde{C}_s \rangle \)
2. if \( \tilde{C}_a = C_a \) then
3. \ return \( D_k(\tilde{C}_s) \);
4. end if
5. if \( A \) has queried \( H_3 \) at \( \tilde{\sigma} \) such that \( \tilde{C}_a = E_{pk}(\tilde{\sigma}, H_3(\tilde{\sigma})) \) then
6. \ return \( D_{H_4(\tilde{\sigma})}(\tilde{C}_s) \);
7. else
8. \ return \( reject \);
9. end if

**Claim 1** \( Pr[S_1] \approx Pr[S_0] \).

To see this, the difference between Game 0 and Game 1 can only be noticed when \( A \) has not queried \( H_3 \) at \( \tilde{\sigma} \) but the ciphertext should not be rejected. Given \( \tilde{C}_a \) and \( SK \), \( \tilde{\sigma} \) is uniquely determined. Assuming \((E, D)\) is \( \gamma \)-uniform (that is the aforementioned “reasonable” property we want), i.e. given \( PK \) and \( m \), there are many possible corresponding ciphertexts which are uniformly distributed (specifically, for any \( PK \), any message \( m \) and any ciphertext \( c \), the probability that a random coin chosen from the coin space will encrypt \( m \) under \( PK \) to give the same \( c \) is bounded by \( \gamma \)), the probability \( H_3(\tilde{\sigma}) \) will match any of these random coins are thus bounded.

**Game 2** drops the consistency of \( H_3 \) and \( H_4 \). Specifically, \( H_3(\sigma) \leftarrow r \) and \( H_4(\sigma) \leftarrow k \) are dropped from the simulation. This makes the simulation fails when \( A \) queries \( H_3 \) or \( H_4 \) at \( \sigma \). We denote this event by \( F_2 \). Now \( k \) is only used in the encryption of \( m_b \) but not anywhere else. From the CCA of \((E, D)\), we have \( Pr[S_2] \approx 0 \). 

In **Game 3**, we can analyze \( Pr[F_2] \). This game modifies the encryption oracle by returning \( E_{PK}(\sigma_{dummy}, r) \) instead of \( E_{PK}(\sigma, r) \) for a randomly chosen \( \sigma_{dummy} \neq \sigma \).
Define the failure event $F_3$ similar to the definition of $F_2$. From the CPA security of $(E, D)$, we have $\Pr[F_3] \approx \Pr[F_2]$. On the other hand, $\sigma$ does not appear in anywhere of the simulation at all. $\Pr[F_3] \approx 0$ and hence $\Pr[F_2] \approx 0$.

Putting all equations together, we can see that $\Pr[S_0]$ is negligible. □

3 “Plug-and-Pray” for Computational/Decisional Attack

Note that in IBE, the adversary’s probability of success is correlated to the secret keys that have been extracted (generally speaking, the more the adversary’s knowledge, the more powerful the adversary is), which in turns affects the probability that the simulator needs to abort (the more keys extracted, the higher chance the simulation fails).

Now we discuss in general the subtlety that may arise when the adversary’s probability of success is correlated with the probability that the simulator needs to abort. We use $BAD$ to denote the event that the simulator needs to abort.

3.1 Computational Task

Let $Forge_0$ and $Forge_1$ be the event that the adversary solved the computational task attacking the security of a cryptographic scheme, and the underlying computational problem is solved, respectively. Takes the signature scheme implied by the above secret key extraction algorithm of the above IBE scheme as an example, the computational task is forgery of a signature, and the underlying hard problem is computational Diffie-Hellman problem.

Consider the case that $Forge_0$ and $BAD$ are independent, we have

$$\Pr[Forge_1] = \Pr[Forge_0 \land BAD] = \Pr[Forge_0] \Pr[BAD].$$

As long as $\Pr[BAD]$ is negligible, we still solve the underlying hard problem with the help of an adversary attacking our cryptographic scheme.

If they are dependent in general, we consider a partition of the adversary’s view space such that they are independent for any particular view. Suppose there is a lower bound $\delta$ such that $\Pr[BAD|View = v] \geq \delta$ for all $v$ in the view space, we have

$$\Pr[Forge_1] = \sum_v \Pr[Forge_0 \land BAD|View = v] \Pr[View = v]$$

$$= \sum_v \Pr[Forge_0|View = v] \Pr[BAD|View = v] \Pr[View = v]$$

$$\geq \delta \sum_v \Pr[Forge_0|View = v] \Pr[View = v]$$

$$= \delta \Pr[Forge_0].$$

We can still solve the underlying hard problem given an adversary attacking our cryptographic scheme as long as $\delta$ is non-negligible.

3.2 Decisional Task

For computational task, it seems whether the dependence of those probabilities does not really matter. However, it is not the case for decisional task.
Let $\text{Guess}_0$ and $\text{Guess}_1$ be the event that the adversary made a correct guess for the decisional task attacking the security of a cryptographic scheme, and a correct guess is made for distinguishable between two possible cases of the underlying decisional problem, respectively. Takes the above scheme as an example, the decisional task is to distinguish between two possible plaintext messages and the underlying hard problem is decisional bilinear Diffie-Hellman problem.

When the simulation aborts (i.e. $\text{BAD}$), the adversary is not helping but the simulator still gives a random guess. If $\text{Guess}_0$ and $\text{BAD}$ are independent, we have

$$\Pr[\text{Guess}_1] = \Pr[\text{Guess}_0 \land \text{BAD}] + \frac{1}{2} \Pr[\text{BAD}]$$

$$\Pr[\text{Guess}_1] - \frac{1}{2} = \Pr[\text{Guess}_0] \Pr[\text{BAD}] + \frac{1}{2}(1 - \Pr[\text{BAD}]) - \frac{1}{2}$$

$$= \Pr[\text{BAD}](\Pr[\text{Guess}_0] - \frac{1}{2})$$

As long as $\Pr[\text{BAD}]$ is negligible, we still solve the underlying hard problem given an adversary attacking our cryptographic scheme.

If they are dependent in general, we consider a partition of the adversary’s view space such that they are independent for any particular view. Suppose there is a lower bound $\delta$ such that $\Pr[\text{BAD}|\text{View} = v] \geq \delta$ for all $v$ in the view space, we have

$$\Pr[\text{Guess}_1] = \sum_v \Pr[\text{Guess}_0 \land \text{BAD}|\text{View} = v] \Pr[\text{View} = v] + \frac{1}{2} \Pr[\text{BAD}]$$

$$\Pr[\text{Guess}_1] = \sum_v \Pr[\text{Guess}_0|\text{View} = v] \Pr[\text{BAD}|\text{View} = v] \Pr[\text{View} = v] + \frac{1}{2} \Pr[\text{BAD}]$$

$$\geq \delta \sum_v \Pr[\text{Guess}_0|\text{View} = v] \Pr[\text{View} = v] + \frac{1}{2} \Pr[\text{BAD}]$$

$$= \delta \Pr[\text{Guess}_0] + \frac{1}{2} \Pr[\text{BAD}]$$

$$\Pr[\text{Guess}_1] - \frac{1}{2} = \delta \Pr[\text{Guess}_0] + \frac{1}{2} \Pr[\text{BAD}] - \frac{1}{2}$$

$$\geq \delta \Pr[\text{Guess}_0] + \frac{1}{2}(1 - \Pr[\text{BAD}]) - \frac{1}{2}$$

$$= \delta \Pr[\text{Guess}_0] - \frac{1}{2} \Pr[\text{BAD}]$$

$$= \delta(\Pr[\text{Guess}_0] - \frac{1}{2}) + \frac{1}{2}(\delta - \Pr[\text{BAD}])$$

The lower bound $\delta$ for $\Pr[\text{BAD}|\text{View} = v], \forall v$ is unlikely to be larger than $\Pr[\text{BAD}]$, so we can not establish a reduction of an attacker to a solver of the underlying decisional problem as easy as in the case for computational problem.
References

