Chapter 2
Definitions and Notations

In this chapter, we present definitions and notation. We start with the definition of public key encryption schemes and their security models. This forms the basis of the corresponding notions for identity-based encryption schemes. The definition of IBE schemes is given and extended to that of HIBE schemes. Security model for HIBE schemes is defined. This security model can be specialised to that of IBE schemes by fixing the number of levels to one.

There are several variants of the security model for (H)IBE schemes. These are carefully explained and the notion of anonymity is defined. A related issue is the use of random oracles in the security analysis. We mention this briefly and discuss its relevance.

2.1 Public Key Encryption

A public key encryption (PKE) scheme is specified by three probabilistic algorithms. The run-time of each of these algorithms is upper bounded by a polynomial in a quantity called the security parameter, denoted by κ. This is formally expressed by explicitly providing 1^κ as input to the algorithms and requiring the run-times of the algorithms to be upper bounded by a polynomial in the length of this input. While this is formally appropriate, it is more convenient to simply note that the run-times are polynomially bounded in κ and avoid explicitly mentioning this.

Set-Up. This algorithm takes as input a security parameter κ. It outputs descriptions of the message space, the ciphertext space, the key space and a key pair (pk, sk) from the key space. Here pk is a public key and sk is the corresponding secret key. The pair (pk, sk) is randomly sampled from the key space. (Though it is not a definitional requirement, (pk, sk) would typically be uniformly distributed over the key space.)

Encrypt. It takes as input a message M and a public key pk and outputs a ciphertext C.

S. Chatterjee and P. Sarkar, *Identity-Based Encryption*, DOI 10.1007/978-1-4419-9383-0_2,

© Springer Science+Business Media, LLC 2011

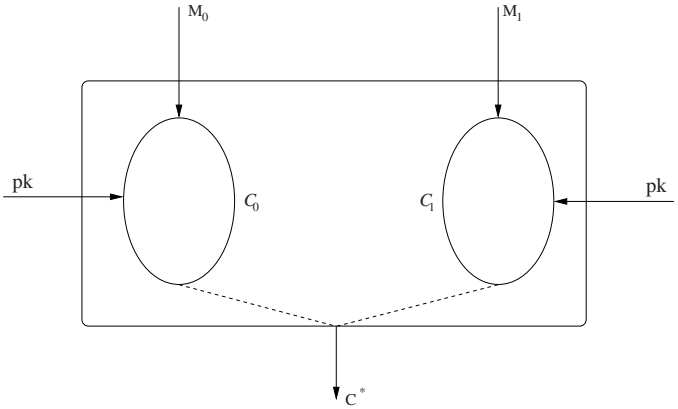

Fig. 2.1 $\mathscr{C}_0$ (resp. $\mathscr{C}_1$) corresponds to the set of possible ciphertexts that can arise when the encryption algorithm is applied to the message M_0 (resp. M_1). C^* is a uniform random choice from $\mathscr{C}_\gamma$, where γ is a uniform random bit.

Decrypt. It takes as input a ciphertext C and a private key sk and returns either a message M or the special symbol $\perp$. The symbol $\perp$ indicates that the ciphertext cannot be decrypted.

The encryption algorithm is a probabilistic algorithm and so there can be more than one ciphertext for a fixed message and a fixed public key. Equivalently, the encryption algorithm can be viewed as a sampling algorithm that given a message M and a public key pk samples from the set of possible ciphertexts which correspond to M and pk. Again the sampling will typically be done under the uniform distribution, though, it is not a definitional requirement.

A ciphertext can be said to be valid if it can be produced as an output of the encryption algorithm (on some pair of inputs M and pk) and invalid otherwise. The definition of the decryption algorithm does not require that the output has to be $\perp$ if the ciphertext is invalid; in this case, it may produce a random element of the message space as output.

For soundness, we require that if C is produced by Encrypt using pk, then the output of Decrypt on C using the corresponding secret key sk should give back M. Since the algorithms are probabilistic, the outputs are actually random variables over appropriate sets. In particular, the Set-Up algorithm can be seen to be sampling a pair of public and private keys from appropriate key spaces and the Encrypt algorithm samples from the set of possible ciphertexts which correspond to a message M and a public key pk. In principle, even though the Decrypt algorithm is allowed to be probabilistic, for most constructions, it is in fact a deterministic algorithm. We note that there are constructions, where the decryption algorithm is allowed to fail with an insignificant probability of error.

Next comes the question – how to define the security of a public key encryption scheme? A natural answer is – given a ciphertext no adversary should be able to learn any *meaningful* information about the corresponding plaintext. This intuitive

notion is formalised into what is called *semantic security* in a landmark paper by Goldwasser and Micali [99]. They also provided a technical definition of security called *indistinguishabilty* and showed that for a passive attacker these two notions are equivalent. This result has later been extended to the case of an active adversary in [98, 168]. The equivalence between the natural notion of security and the technical definition turns out to be very important. Because it is more convenient to work with the technical definition of indistingushability than the natural notion of semantic security.

This technical notion of indistinguishability of ciphertexts for a PKE scheme in the case of a passive adversary can be easily understood with the help of Figure 2.1. For $i = 0, 1$, let $\mathscr{C}_i$ be the set of ciphertexts which may arise from the message M_i under the public key pk. The encryption algorithm defines a distribution over $\mathscr{C}_i$. Suppose that a bit γ is chosen uniformly at random and a ciphertext C^* is sampled from $\mathscr{C}_\gamma$ according to the distribution defined by the encryption algorithm.

An adversary is allowed to specify the messages M_0 and M_1; the bit γ is not revealed to the adversary, but, the ciphertext C^* is given to the adversary. Now the adversary has to guess the value of γ. If the adversary is unable to do so (with probability significantly away from half), then, to the adversary, the ciphertexts arising from M_0 are indistinguishable from the ciphertexts arising from M_1. This basic idea is built into an appropriate security model as we describe below for an active adversary.

Indistinguishability against adaptive chosen ciphertext attack [21] is the strongest accepted notion of security for a public key encryption scheme. An encryption scheme secure against such an attack is said to be IND-CCA2 secure. We give an informal description of IND-CCA2 security in terms of the following game between a challenger and an adversary $\mathscr{A}$, which is a probabilistic algorithm whose runtime is bounded above by a polynomial in the security parameter. Later we provide a more detailed explanation of the security game for an IBE scheme. Figure 2.2 gives an overview of the security game for a PKE scheme.

1. Given the security parameter κ, the challenger runs the Set-Up algorithm to generate a public and private key pair (pk, sk). It gives $\mathscr{A}$ the public key pk.
2. Given the public key, $\mathscr{A}$ *adaptively* issues decryption queries, which the challenger must properly answer. By adaptively it is meant that the adversary's next query can depend on the answers to the previous queries.
3. At some point, $\mathscr{A}$ outputs two equal length messages M_0, M_1 and the challenger responds with an encryption C^* of M_γ, where γ is a random bit.
4. The adversary continues with adaptive decryption queries but not on C^*.
5. Finally, $\mathscr{A}$ outputs its guess γ' of γ and wins if $\gamma' = \gamma$.

The advantage of $\mathscr{A}$ against the encryption scheme is

$$\mathsf{Adv}_{\mathscr{A}} = \left| \Pr[\gamma = \gamma'] - \frac{1}{2} \right|.$$

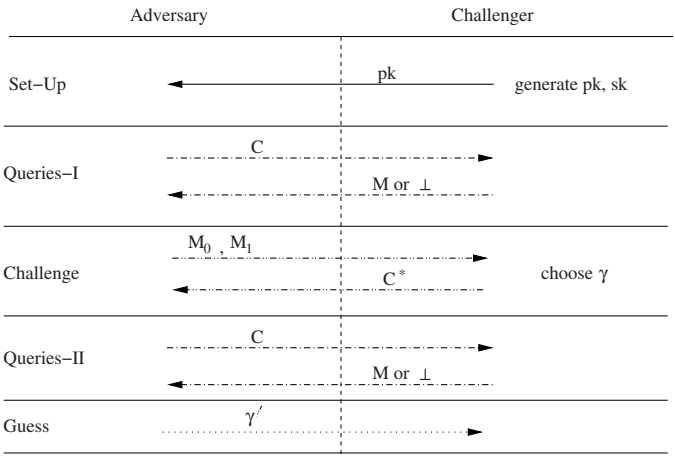

Fig. 2.2 A diagrammatic depiction of the five phases of the security model for a public key encryption scheme.

An encryption scheme is said to be (t,q,ε)-IND-CCA2 secure, if for all adversaries $\mathscr{A}$ running in time t and making at most q decryption queries, $\mathsf{Adv}_{\mathscr{A}} \leq \varepsilon$.

In case of a passive adversary, a weaker notion of security, called indistinguishability against chosen plaintext attack (in short IND-CPA security) of a public key encryption scheme is available in the literature [99, 21]. In the IND-CPA security game, the adversary is not allowed to place any decryption query. In other words, this is the scenario depicted in Figure 2.1 where the query phases depicted in Figure 2.2 are not allowed. Given a public key, the adversary simply outputs two equal length messages M_0, M_1 and the challenger responds with an encryption C^* of M_γ. The adversary wins if it can predict γ.

2.2 Identity-Based Encryption

The formal notion of an identity-based encryption scheme was developed in [155, 39]. An identity-based encryption scheme is specified by four probabilistic polynomial time (in the security parameter) algorithms: Set-Up, Key-Gen, Encrypt and Decrypt.

Set-Up: This algorithm takes as input a security parameter 1^κ, and returns the system parameters PP together with the master secret key msk. The system parameters include a description of the message space $\mathscr{M}$, the ciphertext space $\mathscr{C}$, the identity space $\mathscr{I}$ and the master public key. They are publicly known while the master secret key is known only to the private key generator (PKG). Usually, the descriptions of the different spaces are implicit in the description of the master public key and this itself is referred to as the public parameter PP.

Key-Gen: This algorithm takes as input an identity $\mathsf{id} \in \mathscr{I}$ together with the public parameters PP and the master secret key msk and returns a private key d_{id}, using the master key. The identity id is used as the public key while d_{id} is the corresponding private key.

Encrypt: This algorithm takes as input an identity $\mathsf{id} \in \mathscr{I}$, a message $M \in \mathscr{M}$ and the public parameters PP and produces as output a ciphertext $C \in \mathscr{C}$.

Decrypt: This takes as input a ciphertext $C \in \mathscr{C}$, an identity id, a corresponding private key d_{id} and the system parameters PP. It returns the message M or $\perp$ if the ciphertext cannot be decrypted.

These set of algorithms must satisfy the standard soundness requirement.

If
(PP,msk) is output by $\mathsf{Set\text{-}Up}$;
d_{id} is a private key returned by $\mathsf{Key\text{-}Gen}$ for an identity id;
C is a ciphertext produced by $\mathsf{Encrypt}$ on a message M,
using identity id and public parameters PP;
then
the output of $\mathsf{Decrypt}$ on C, id, d_{id} and PP should be M.

The comments regarding the encryption and decryption algorithms made in the context of PKE schemes are also applicable here. Additionally, similar comments apply to key generation. Given an identity and public parameters, it might be possible to have a set of corresponding decryption keys. In that case the key generation algorithm can be visualised as a strategy for sampling from this set. Note that the PKG can decrypt any message encrypted under any identity since it is the PKG who generated the private key for that identity. This is the so-called *key escrow* property of identity-based cryptography.

2.2.1 Hierarchical Identity-Based Encryption

Hierarchical identity-based encryption (HIBE) is an extension of IBE. The basic motivation for HIBE schemes is based on the following rationale. The generation of private key can be a computationally intensive task. The identity of an entity must be authenticated before issuing a private key and the private key needs to be transmitted securely to the concerned entity.

HIBE reduces the workload of the PKG by delegating the task of private key generation and hence authentication of identity and secure transmission of private key to its lower levels. However, only the PKG has a set of public parameters. The identities at different levels do not have any public parameters associated with them. Apart from being a standalone cryptographic primitive, HIBE has many interesting applications.

In contrast to IBE, for a HIBE identities are represented as vectors. So for a HIBE of maximum height h (which is denoted as h-HIBE) any identity id is a tuple

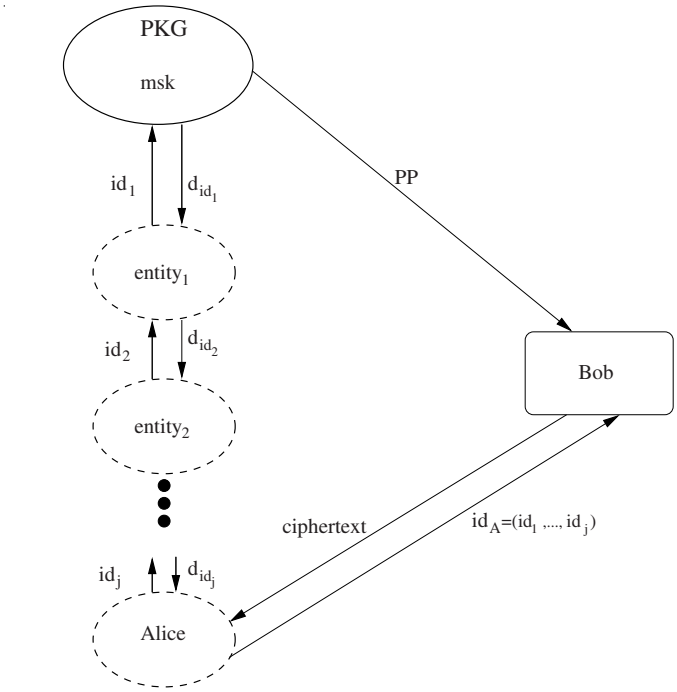

Fig. 2.3 Schematic diagram of the operation of a HIBE.

$(\mathsf{id}_1, \ldots, \mathsf{id}_\tau)$ where $1 \leq \tau \leq h$. Let, $\mathsf{id}' = \mathsf{id}'_1, \ldots, \mathsf{id}'_j$, $j \leq \tau$ be another identity tuple. We say id' is a prefix of id if $\mathsf{id}'_i = \mathsf{id}_i$ for all $1 \leq i \leq j$.

As in the case of IBE, the PKG has a set of public parameters PP and a master key msk. For all identities at the first level the private key is generated by the PKG using msk. For identities at the second level onwards, the private key can be generated by the PKG or by any of the ancestors of that identity. In the above example, the private key d_{id} of id can be generated by an entity whose identity is a prefix of id and who has obtained the corresponding private key. This is shown in Figure 2.3.

The formal notion of a HIBE scheme is an extension of the corresponding notion of an IBE scheme and was developed in [106, 94]. A HIBE scheme $\mathscr{H}$ is specified by four probabilistic polynomial time (in the security parameter) algorithms: $\mathsf{Set\text{-}Up}$, $\mathsf{Key\text{-}Gen}$, $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$.

Set-Up: This algorithm takes input a security parameter 1^κ and returns the system (or public) parameters PP together with the master secret key msk. The system parameters include a description of the message space $\mathscr{M}$, the ciphertext space $\mathscr{C}$ and the identity space $\mathscr{I}$. These are publicly known while the master key is known only to the private key generator (PKG). In case, there is some maximal level h of the HIBE, then this is also made public.

Key-Gen: This algorithm takes as input an identity tuple $\mathsf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_j)$, $j \geq 1$ and the private key $d_{\mathsf{id}|j-1}$ for the identity $(\mathsf{id}_1, \ldots, \mathsf{id}_{j-1})$ and returns a private key d_{id} using $d_{\mathsf{id}|j-1}$. If $j = 1$, then $d_{\mathsf{id}|0}$ is defined to be the master secret key msk. The identity id is used as the public key while d_{id} is the corresponding private key.

Note that by appropriately invoking the Key-Gen algorithm the PKG as well as any proper predecessor of the identity tuple $(\mathsf{id}_1, \ldots, \mathsf{id}_j)$ can produce a decryption key for this identity.

Encrypt: This algorithm takes as input the system parameters PP, an identity id and a message M and produces as output a ciphertext C. This ciphertext is the encryption of M under the identity id and the public parameters PP.

Decrypt: This algorithm takes as input the public parameters PP, an identity id, a ciphertext C and a private key d_{id} and returns the message or $\perp$ if the ciphertext is not valid.

The standard soundness requirement that holds for IBE is also applicable for HIBE. If d_{id} is a private key corresponding to the identity tuple id generated by the Key-Gen algorithm and C is the output of the Encrypt algorithm for a message $M \in \mathscr{M}$ using id as a public key and PP; then the Decrypt algorithm must return M on input d_{id} and C.

Comments on the encryption and decryption algorithms made in the context of PKE schemes are applicable here. Further, comments made on the key generation algorithm in the context of IBE schemes are also applicable here. Also note that in addition to the PKG, any proper ancestor of id can decrypt messages encrypted under id.

2.3 Security Model for (H)IBE

As we have already noted, HIBE is a generalisation of IBE i.e., an IBE can be thought of as a single level HIBE. So instead of describing the security models for IBE and HIBE separately, we only describe the security model for HIBE. The security model for IBE is obtained by setting the number of levels to one.

The basic idea of the security model for (H)IBE schemes is obtained by extending the security model for PKE schemes. As in PKE we focus on the technical notion of indistingushability (it is known [13] that in the case of IBE also this technical notion is equivalent to the more natural notion of semantic security). Just like in PKE, it is a formalisation of the adversary's inability to distinguish between ciphertexts arising out of two equal length messages M_0 and M_1. An identity is chosen by the adversary as the target identity, i.e., the adversary's goal is to compromise the security of the identity it chooses as the target identity. A random bit γ is chosen and the challenge ciphertext is produced by encrypting M_γ under the target identity. The adversary wins if it can predict γ with a probability significantly away from half.

The main difference from PKE schemes is that a coalition of valid users of an IBE scheme can possibly launch an attack against another user of the scheme. Each

valid user has a decryption key provided by the PKG for its identity. A group of users can form a coalition and attempt to compromise the security of another user. Modelling this aspect is a bit tricky, since the coalition may not be formed at the outset and may gradually grow. This modelling is done by providing the adversary with a key-extraction oracle. The adversary can query the oracle with an identity and receive a corresponding decryption key. It is allowed to place queries in an adaptive manner. Further, it can choose the target identity after making some key-extraction queries. In addition, as in PKE, the adversary may be given access to a decryption oracle. All of these is formalised in the following manner.

2.3.1 Chosen Ciphertext Attack

Recall that security against adaptive chosen ciphertext attack is the accepted notion of security for a public key encryption. This notion of security has been extended to the identity-based setting by Boneh and Franklin [39]. This is termed as IND-ID-CCA security (indistinguishability under adaptive identity and adaptive chosen ciphertext attack).

Let $\mathscr{H}$ be an h-HIBE scheme as defined in the previous section. The IND-ID-CCA security for $\mathscr{H}$ is defined [106, 94, 32] in terms of the following game between a challenger and an adversary of the HIBE. The adversary is allowed to place two types of oracle queries – decryption queries to a decryption oracle $\mathscr{O}_d$ and key-extraction queries to a key-extraction oracle $\mathscr{O}_k$. Figure 2.4 shows a schematic diagram of the security game defining the security of an IBE scheme. The notion of indistinguishability of ciphertexts is similar to the idea explained in the context of PKE schemes.

Set-Up. The challenger takes as input a security parameter 1^κ and runs the Set-Up algorithm of the HIBE. It provides $\mathscr{A}$ with the system parameters PP while keeping the master key msk to itself.

Phase 1: Adversary $\mathscr{A}$ makes a finite number of queries where each query is one of the following two types:

- key-extraction query (id): This query is placed to the key-extraction oracle $\mathscr{O}_k$. Questioned on id, $\mathscr{O}_k$ generates a private key d_{id} of id and returns it to $\mathscr{A}$.
  The Key-Gen algorithm is probabilistic and so if it is queried more than once on the same identity, then it may provide different (but valid) decryption keys. Some (H)IBE schemes can insist on storing the decryption key generated on the first query and returning the stored value on subsequent queries on the same identity. This can help in achieving a tight security reduction.
- decryption query (id, C): This query is placed to the decryption oracle $\mathscr{O}_d$. It returns the resulting plaintext or $\perp$ if the ciphertext cannot be decrypted.

$\mathscr{A}$ is allowed to make these queries adaptively, i.e., any query may depend on the previous queries as well as their answers.

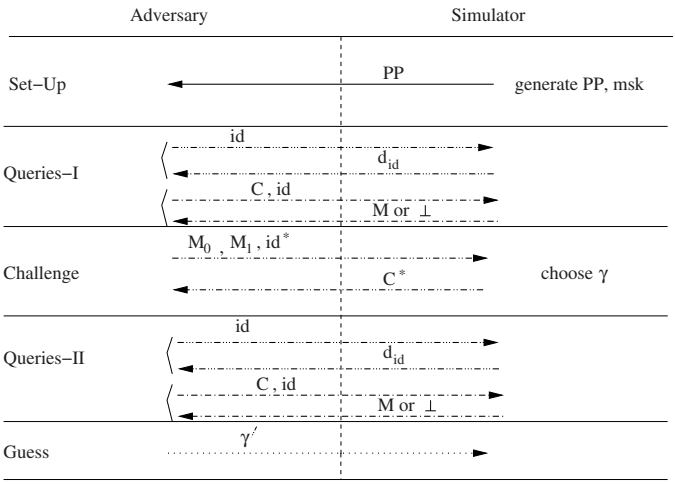

Fig. 2.4 A diagrammatic depiction of the five phases of the security model for identity-based encryption.

Challenge: When $\mathscr{A}$ decides that Phase 1 is complete, it fixes an identity id^* and two equal length messages M_0, M_1 under the (obvious) constraint that it has not asked for the private key of id^* or any prefix of id^*. The challenger chooses uniformly at random a bit $\gamma \in \{0,1\}$ and obtains a ciphertext C^* corresponding to M_γ, i.e., C^* is the output of the $\mathsf{Encrypt}$ algorithm on input $(M_\gamma, \mathsf{id}^*, \mathsf{PP})$. It returns C^* as the challenge ciphertext to $\mathscr{A}$.

Phase 2: $\mathscr{A}$ now issues additional queries just like Phase 1, with the (obvious) restriction that it cannot place a decryption query for the decryption of C^* under id^* or any of its prefixes nor a key-extraction query for the private key of id^* or any prefix of id^*. All other queries are valid and $\mathscr{A}$ can issue these queries adaptively just like Phase 1. The challenger responds as in Phase 1.

Guess: $\mathscr{A}$ outputs a guess γ' of γ.
The advantage of the adversary $\mathscr{A}$ in attacking the HIBE scheme $\mathscr{H}$ is defined as:

$$\mathsf{Adv}_{\mathscr{A}}^{\mathscr{H}} = \left|\Pr[(\gamma = \gamma')] - 1/2\right|.$$

An h-HIBE scheme $\mathscr{H}$ is said to be $(t, q_{\mathsf{id}}, q_{\mathsf{C}}, \varepsilon)$-secure against adaptive chosen ciphertext attack ($(t, q_{\mathsf{id}}, q_{\mathsf{C}}, \varepsilon)$-$\mathsf{IND\text{-}ID\text{-}CCA}$ secure) if for any t-time adversary $\mathscr{A}$ that makes at most q_{id} private key queries and at most q_{C} decryption queries, $\mathsf{Adv}_{\mathscr{A}}^{\mathscr{H}} \leq \varepsilon$. In short, we say $\mathscr{H}$ is $\mathsf{IND\text{-}ID\text{-}CCA}$ secure or when the context is clear, simply CCA-secure.

Shi and Waters [157] consider a more general security definition where the distribution of the keys depend on the actual delegation path. We do not consider this model in this work, since, for the schemes that we describe, the keys are uniformly distributed.

2.3.2 Chosen Plaintext Attack

Security reduction of (H)IBE protocols available in the literature generally concentrate on proving security in a weaker model. This is called security against adaptive identity chosen plaintext attack or IND-ID-CPA security [39]. The corresponding game is similar to the game defined above, except that the adversary is *not* allowed access to the decryption oracle $\mathcal{O}_d$. The adversary is allowed to place adaptive private key extraction queries to the key-extraction oracle $\mathcal{O}_k$ and everything else remains the same. For the sake of completeness, we give a description of the IND-ID-CPA game for an h-HIBE $\mathscr{H}$ below.

Set-Up The challenger takes as input a security parameter 1^κ and runs the Set-Up algorithm of the HIBE. It provides $\mathscr{A}$ with the system parameters PP while keeping the master key msk to itself.

Phase 1: Adversary $\mathscr{A}$ makes a finite number of key-extraction queries to $\mathcal{O}_k$. For a private key query corresponding to an identity id, the key-extraction oracle generates the private key d_{id} of id and returns it to $\mathscr{A}$. $\mathscr{A}$ is allowed to make these queries adaptively, i.e., any query may depend on the previous queries as well as their answers.

Challenge: At this stage $\mathscr{A}$ fixes an identity, id^* and two equal length messages M_0, M_1 under the (obvious) constraint that it has not asked for the private key of id^* or any of its prefixes. The challenger chooses uniformly at random a bit $\gamma \in \{0,1\}$ and obtains a ciphertext (C^*) corresponding to M_γ, i.e., C^* is the output of the Encrypt algorithm on input $(M_\gamma, \mathsf{id}^*, \mathsf{PP})$. It returns C^* as the challenge ciphertext to $\mathscr{A}$.

Phase 2: $\mathscr{A}$ now issues additional queries just like Phase 1, with the (obvious) restriction that it cannot place a key-extraction query for the private key of id^* or any prefix of id^*. All other queries are valid and $\mathscr{A}$ can issue these queries adaptively just like Phase 1.

Guess: $\mathscr{A}$ outputs a guess γ' of γ.
Like the IND-ID-CCA game, the advantage of the adversary $\mathscr{A}$ in attacking the HIBE scheme $\mathscr{H}$ is defined as

$$\mathsf{Adv}_{\mathscr{A}}^{\mathscr{H}} = \left|\Pr[\gamma = \gamma'] - 1/2\right|.$$

An h-HIBE scheme $\mathscr{H}$ is said to be (t, q, ε) secure against adaptive chosen plaintext attack if for any t-time adversary $\mathscr{A}$ that makes at most q private key extraction queries, $\mathsf{Adv}_{\mathscr{A}}^{\mathscr{H}} \leq \varepsilon$. In short we say $\mathscr{H}$ is (t, q, ε)-IND-ID-CPA secure or simply CPA-secure if the context is clear.

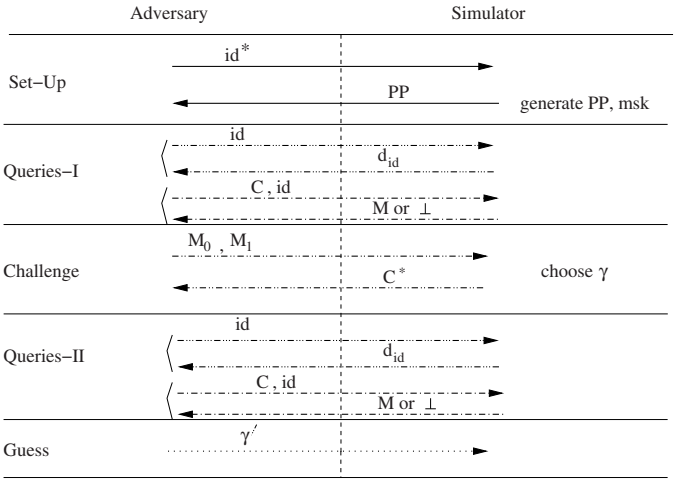

Fig. 2.5 A diagrammatic depiction of the five phases of the selective-identity security model for identity-based encryption.

2.3.3 Selective-ID Model

A weaker definition of security for identity-based encryption schemes is the so called *selective*-ID model [53, 54]. In this model, the adversary $\mathscr{A}$ commits to a target identity before the system is set up. This notion of security is called the selective identity, chosen ciphertext security (IND-sID-CCA security in short).

Compared to the security model where the adversary can choose the target identity adaptively, this is a very restricted notion of security. Correspondingly, it is also significantly easier to argue security in this model. If the sole interest is in obtaining a secure IBE, then selective identity security model is not satisfactory. On the other hand, this model is useful in other ways. In particular, it provides a new method to convert CPA-secure IBE schemes to CCA-secure PKE schemes.

Following [53, 54, 32] we define IND-sID-CCA security for an h-HIBE in terms of the game described below. A schematic diagram for the selective-identity security model is shown in Figure 2.5.

Initialization: The adversary outputs a target identity tuple $\mathsf{id}^* = (\mathsf{id}_1^*, \ldots, \mathsf{id}_u^*)$, $1 \leq u \leq h$ on which it wishes to be challenged.

Set-Up: The challenger sets up the HIBE and provides the adversary with the system public parameters PP. It keeps the master key msk to itself.

Phase 1: Adversary $\mathscr{A}$ makes a finite number of queries where each query is either a decryption or a key-extraction query. In a decryption query, it provides the ciphertext as well as the identity under which it wants the decryption. Similarly, in a key-extraction query, it asks for the private key of the identity it provides. Further, $\mathscr{A}$ is allowed to make these queries adaptively, i.e., any query may depend on the

previous queries as well as their answers. The only restriction is that it cannot ask for the private key of id^* or any of its prefixes.

Challenge: At this stage, $\mathscr{A}$ outputs two equal length messages M_0, M_1 and gets a ciphertext C^* corresponding to M_γ encrypted under the a priori chosen identity id^*, where γ is chosen by the challenger uniformly at random from $\{0,1\}$.

Phase 2: $\mathscr{A}$ now issues additional queries just like Phase 1, with the (obvious) restriction that it cannot ask for the decryption of C^* under id^* or any of its prefixes nor the private key of id^* or any prefix of id^*.

Guess: $\mathscr{A}$ outputs a guess γ' of γ.
The advantage of the adversary $\mathscr{A}$ in attacking the HIBE scheme $\mathscr{H}$ is defined as:

$$\mathsf{Adv}_{\mathscr{A}}^{\mathscr{H}} = \left|\Pr[\gamma = \gamma'] - 1/2\right|.$$

The HIBE scheme $\mathscr{H}$ is said to be $(t, q_{\mathsf{id}}, q_{\mathsf{C}}, \varepsilon)$-secure against selective identity, adaptive chosen ciphertext attack (in short, $(t, q_{\mathsf{id}}, q_{\mathsf{C}}, \varepsilon)$-IND-sID-CCA secure) if for any t-time adversary $\mathscr{A}$ that makes at most q_{id} private key extraction queries and at most q_{C} decryption queries, $\mathsf{Adv}_{\mathscr{A}}^{\mathscr{H}} \leq \varepsilon$.

Note that, in the above game the adversary has to commit to an identity tuple even before the appropriate spaces are defined by the Set-Up algorithm.

We may restrict the adversary from making any decryption query. An h-HIBE scheme $\mathscr{H}$ is said to be $(t, q_{\mathsf{id}}, \varepsilon)$-secure against selective identity, adaptive chosen plaintext attack (in short, $(t, q_{\mathsf{id}}, \varepsilon)$-IND-sID-CPA secure) if for any t-time adversary $\mathscr{A}$ that makes at most q_{id} private key queries, $\mathsf{Adv}_{\mathscr{A}}^{\mathscr{H}} \leq \varepsilon$.

2.3.4 Anonymous (H)IBE

There is another aspect which is sometimes added to the security definition of (H)IBE. This concerns the anonymity of the ciphertext. Informally, the idea is as follows. For certain applications, it may be a security concern that the identity of the intended recipient is revealed from the ciphertext. In other words, looking only at C (and PP) it may be possible to determine the identity id which has been used to generate C from some message. Roughly speaking, if for a (H)IBE scheme this is not possible, then the scheme is said to be anonymous. The formal definition of anonymity is obtained by modifying the above security game.

1. The Set-Up and the query phases 1 and 2 of the security game remain unchanged.
2. In the challenge stage, the adversary submits two identities id_0 and id_1 along with two equal length messages M_0 and M_1.
3. The simulator chooses two independent and uniform random bits γ_1 and γ_2 and provides the adversary with C^* which is the encryption of M_{γ_1} to the identity id_{γ_2}.

4. In the guess stage, the adversary outputs two bits γ_1' and γ_2' and wins if $\gamma_1 = \gamma_1'$ and $\gamma_2 = \gamma_2'$.
5. The adversary's advantage is defined to be

$$\left| \Pr\left[(\gamma_1 = \gamma_1') \wedge (\gamma_2 = \gamma_2') \right] - \frac{1}{4} \right|.$$

Parameterization of an adversary is done in the manner it is done for the (H)IBE schemes without anonymity considerations.

In any security game for (H)IBE schemes, the key-extraction oracle is always provided to the adversary. Additionally, the different variants of the security game are based on the following three points.

1. Whether the decryption oracle is provided to the adversary or not.
2. Whether the adversary can choose the target identity adaptively, i.e., after making queries to the key-extraction oracle; or whether the adversary has to commit to the target identity before the scheme is set-up giving the selective-identity game.
3. Whether the (H)IBE scheme is defined for the anonymity game or not.

The convention is that if anonymity is required, then it is specifically mentioned. For the other two points, there are four options leading to four different sets of restrictions on the adversary. Among these, the most restricted adversary is selective-identity and not having access to the decryption oracle which leads to the weakest (among the above four) notion of security for a (H)IBE scheme. The most powerful adversary is adaptive-identity and can make decryption queries leading to the strongest notion of security for a (H)IBE scheme. Several other security notions such as one-wayness, non-malleability, semantic security and multiple-target-multiple-challenge CCA security have been formulated in the context of IBE and it is shown [13] that IND-ID-CCA security implies all other notions of security. In this sense IND-ID-CCA security can be considered as the "right" notion of security in the identity-based setting.

As we will discuss later, there are generic methods to convert a CPA-secure (H)IBE scheme to a CCA-secure (H)IBE scheme. Non-generic methods which apply to a fairly large number of schemes are also known. Due to these results, almost all schemes start out by first obtaining CPA-security. Among CPA-secure schemes, the issue is whether the scheme satisfies selective-identity or adaptive identity security. We will be seeing both kinds of schemes later.

2.3.5 Use of Random Oracles

The security analyses of (H)IBE schemes essentially show that if an adversary can win a security game (one among the variants described above), then it is possible to use such an adversary to solve some problem which is conjectured to be computationally hard. This is the usual reductionist technique used in complexity theory. We

will be seeing some of these conjectured hard problems later. For the moment, we would like to discuss a different issue.

During the security reduction (or proof), it may be necessary to assume that certain functions described in the scheme are uniform random functions. In other words, if a set of distinct inputs are provided to such a function, then the outputs are independent and uniformly distributed. Clearly, such an assumption cannot hold for any practical function that may be used to instantiate the scheme. Alternatively, this can be viewed as an idealisation of some functions that are actually used in practice.

This is the so-called "random oracle" assumption used to argue about the security of many cryptographic primitives and not only (H)IBE schemes. This technique was formally introduced in [24] and has been criticised later in [52]. A rebuttal of the criticism and a robust defence of the random oracle technique has been given in [124].

Clearly, it would be better if one could build schemes where one does not have to use random oracles and there are indeed such schemes. At the same time, one should note that schemes for which the proofs require "random oracles" may also be useful in two situations: when no other scheme is known which does not require "random oracle proofs" (under the same kind of hardness assumption) and when the use of random oracles improves efficiency.

2.4 Structure of Security Proofs

We provide a few words on the structure of security proofs. All proofs are reductions. Suppose a cryptographic protocol is built upon several other smaller protocols. Then the assurance provided by a reduction is of the following form.

If

(smaller protocols are secure
and)
some problem Π is computationally hard

then

the main protocol is secure.

The argument is established through a contradiction. One starts with the assumption that there is an adversary who can break the main protocol with some non-negligible advantage in the given security model. This adversary is then used as a blackbox to construct an algorithm that either solves the underlying hard computational problem Π or breaks one of the smaller protocols with non-negligible probability of success. This contradicts the original hypothesis. As mentioned earlier, the proof itself may assume some hash functions to be uniform random functions.

A convenient way to structure the reductionist proof is to consider a sequence of games [160, 26]. To prove, for example, the indistinguishability of the encryption of two equal length plaintexts we construct a sequence of games of the following form.

$$\underline{\text{A Game Sequence}}$$
$$G_0,$$
$$G_1,$$
$$\vdots$$
$$G_k$$

- Let X_i be the event that $\gamma = \gamma'$ in Game G_i. We consider

$$\begin{array}{l}\mathsf{Pr}[X_0],\\ \mathsf{Pr}[X_0]-\mathsf{Pr}[X_1],\\ \vdots\\ \mathsf{Pr}[X_{k-1}]-\mathsf{Pr}[X_k]\\ \mathsf{Pr}[X_k].\end{array}$$

In the above sequence, the following points are to be noted

1. G_0 is the game which defines the security of the protocol and so

$$\mathsf{Adv}(\mathscr{A}) = |\mathsf{Pr}[\gamma = \gamma'] - 1/2| = |\mathsf{Pr}[X_0] - 1/2|.$$

2. G_k is designed such that the bit γ is statistically hidden from the adversary. So,

$$\mathsf{Pr}[X_k] = 1/2.$$

3. Games G_{i-1} and G_i differ:
   a. the difference is not too much;
   b. the adversary should not be able to notice whether he is playing Game G_{i-1} or Game G_i.
4. More precisely, $\mathsf{Pr}[X_{i-1}] - \mathsf{Pr}[X_i]$ is bounded above by
   a. either, the advantage of an adversary in breaking one of the smaller protocols;
   b. or, the advantage of solving problem Π.

$$\begin{aligned}\mathsf{Adv}(\mathscr{A}) &= |\mathsf{Pr}[X_0] - 1/2|\\ &= |\mathsf{Pr}[X_0] - \mathsf{Pr}[X_k]|\\ &\le |\mathsf{Pr}[X_0] - \mathsf{Pr}[X_1]|\\ &\quad + |\mathsf{Pr}[X_1] - \mathsf{Pr}[X_2]|\\ &\quad + \cdots\\ &\quad + |\mathsf{Pr}[X_{k-1}] - \mathsf{Pr}[X_k]|.\end{aligned}$$

If the adversary has a non-negligible advantage then there must be at least two consecutive games, X_{i-1} and X_i such that $|\mathsf{Pr}[X_{i-1}] - \mathsf{Pr}[X_i]|$ is non-negligible – which contradicts the original hypothesis.

2.5 Conclusion

This chapter introduced the necessary formalism. Later chapters describing different constructions of IBE schemes will be based on the formal notions introduced here. Apart from this minimal requirement, this chapter also provided background intuition behind the various definitions. This intuition will be useful in working through the proofs and constructions given later.

http://www.springer.com/978-1-4419-9382-3

Identity-Based Encryption
Chatterjee, S.; Sarkar, P.
2011, XI, 180 p., Hardcover
ISBN: 978-1-4419-9382-3