Chapter 10
Sakai-Kasahara IBE

Hamza Mutaher and Mahmoud E. Hodeish

Abstract Public key cryptography (PKC) provides a very robust encryption in networking and electronic communication. The strength of PKC comes from the idea of paired keys that are independent (but mathematically dependent). The encryption-decryption process of PKC requires both parties of communication, i.e., sender and receiver, to provide each other with its public key and the digital certificate of authority, and each party has to keep a directory to store all parties' public keys so these requirements are considered as drawbacks of PKC. To overcome these drawbacks, the identity-based encryption (IBE) came to existence. IBE is a form of PKC which uses a third-party server to distribute the public parameters to all the parties and extract the private key from the arbitrary public key. To encrypt the message, the sender will use the receiver public key, and to decrypt the message, the receiver will use the extracted private key. In this chapter, we discuss the Sakai-Kasahara IBE and how it differs from other IBE schemes. The additive, multiplicative, and full schemes of IBE are explained along with the encryption and decryption process. The security of this scheme is also discussed and proved.

Keywords Cryptography · IBE · Encryption · Decryption · Additive · Multiplicative · Security

H. Mutaher (✉)
Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India

M. E. Hodeish
Faculty of Computer & Information Technology, Al-Razi University, Sana'a, Yemen

Department of Computer, Faculty of Education-Zabid, Hodeidah University, Hodeidah, Yemen

K. A. B. Ahmad et al. (eds.), *Functional Encryption*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-030-60890-3_10

1 Introduction

Identity-based encryption (IBE) or the so-called ID-based encryption is a scheme that uses public key encryption in which any bit string can be represented as a public key in which the public key of a user can be some unique meaningful identity like name and email address.

The motivation of introducing the IBE scheme is to solve such problems of traditional public key systems like the necessity for directories and digital certificates to manage public keys and the expensive computations of generating public-private key pairs. However, Shamir [1] was the first who introduced the concept of IBE that eliminates the use of directories and digital certificates. He considered the receiver identity as the representation of the public key. Despite solving some of the related problems of identity-based signature, IBE proved much more challenging.

The Cocks IBE scheme [2] is one of the encryption algorithms. This type of encryption algorithm encrypts the plaintext into ciphertext. The assurance of this algorithm depends on the durability of the quadratic residuosity problem and the computational difficulty of integer factorization as well. The system authority of this algorithm generates a modulus m which is universally available. To create this modulus, system authority calculates two primes p and q; thus the modulus m will be the product of this calculation, where both primes p and q must be congruent to 3 mod 4. This system ensures the use of a universally available hash function to the text that needs to be encrypted to represent it as a value to a modulo m. Therefore, when the user A wants to get encrypted data, he/she needs to send any of his/her identities such as (username or email address) to the system authority. Mutually, user A will receive a private key from the system authority. The user B who seeks to send an encrypted data to user A will be able to deliver it by knowing only the public identity of user A and universal public parameters where there is no need to know the public key.

On the other hand, there is another type of IBE algorithm called Boneh-Franklin IBE [3]. It is an algorithm-based identity that encrypts the data for security. It is considered as the first secure and practical scheme of IBE and it is an example of an IBE full domain hash scheme family. This scheme maps the identity to the elliptic curve to accomplish the process of encryption and decryption. Modular exponentiation is required to start the process of mapping between the identity and the point in the elliptic curve. The expensive calculation is considered as a drawback for the performance of the full domain hash scheme.

The Boneh-Boyen IBE scheme [4] is also used to encrypt the identity of the users. In this scheme, the sender and receiver have to use the same value to encrypt and decrypt the identity where the receiver also uses its private key at the decryption side. In this scheme, the user identity is hashed to an integer to be used in the process of encryption and decryption. The hashed integer avoids the calculation of modular exponentiation and it is considered more rapid than the full domain hash scheme.

This chapter aims to discuss in detail the Sakai-Kasahara IBE scheme [5, 6]. This scheme depends on the bilinear pairing and elliptic curve to provide security solutions. The private key is the system element that is responsible to decrypt the

ciphertext. It is one of the security solutions that belong to the exponent inversion scheme family. The encryption and decryption procedures are applied through a hashed integer on an identity in a form of string. Such algorithms like Boneh-Franklin IBE use the full domain hashed scheme which is considered slower than Sakai-Kasahara IBE. Due to the avoidance of modular exponentiation, Sakai-Kasahara IBE is quite faster than Boneh-Franklin IBE [7] which is going to be discussed in this chapter. Before the discussion of Sakai-Kasahara IBE, we have to explain the procedure that occurs in the IBE system to encrypt and decrypt the message.

When the sender *A* wants to send an encrypted message to receiver *B*, he/she simply encrypts the message using *B's* public identity, for example email address, and there is no need for *A* to check the *B's* public key certificate. When *B* receives the encrypted message, he/she will communicate the private key generator (*PKG*), also called system authority, which will send him/her a private key to allow him/her to decrypt the message. Note that *A* and *B* have to authenticate themselves to the PKG before starting the message exchange procedure; see Fig. 10.1.

IBE has four major operations explained as follows:

1. Setup of parameters: *PKG* will generate public parameters θ and master secret *S* and distribute public parameters to both *A* and *B*.
2. Extraction of the private key: *PKG* will use the master secret *S* to extract the private key S_{ID_B} which corresponds to an arbitrary public identity of string ID_B.
3. Encryption: The sender *A* will encrypt the message using the receiver *B's* public identity ID_B.
4. Decryption: The receiver *B* will decrypt the message using the corresponding private key S_{ID_B} that has been sent by the *PKG*.

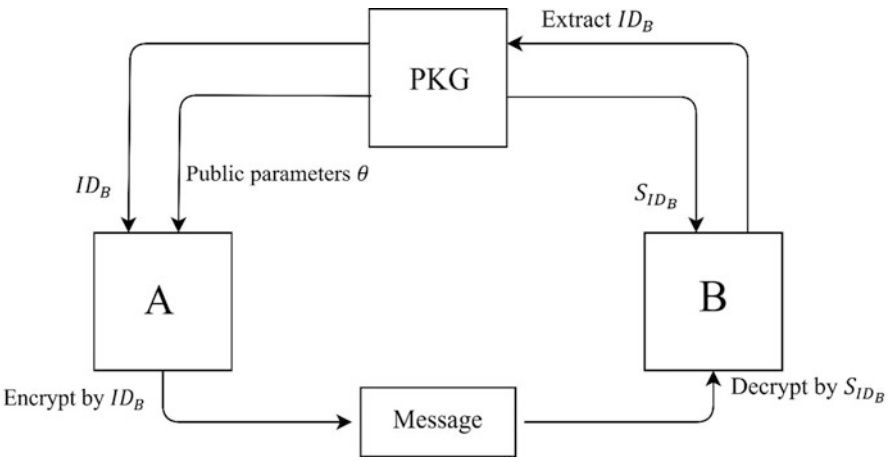

Fig. 10.1 IBE operations

This chapter is divided into two main parts; the first one discusses the encryption and decryption of Sakai-Kasahara IBE basic scheme with its additive notation and multiplicative notation. The second part discusses encryption and decryption of the Sakai-Kasahara IBE full scheme with its security proof. Both parts will explain the setup of the parameters to accomplish the encryption and decryption process.

2 Sakai-Kasahara IBE (Basic Scheme: Additive Notation)

The S-K IBE basic scheme is less secure than the S-K IBE full scheme but easier to understand. In the basic scheme, two parties need to exchange encrypted messages safely. Both parties must agree on a unique shared secret to encrypt the message in plaintext form. The first party (sender) calculates the shared secret from its public parameters and identity of the second party (receiver). The receiver gets the shared secret by calculating the ciphertext and its private key.

2.1 Setup of Parameters

This scheme deals with additive notation; thus $E(F_q)$ is an elliptic curve group, σ_1, σ_2 are two elements of the elliptic curve, and $\sigma_1 + \sigma_2$ indicates the $E(F_q)$ group operation to be applied to the group elements σ_1, σ_2 and multiply σ by integer s which is indicated as $s\sigma$.

To implement this scheme, we need to define some basic essential elements shown in Table 10.1 and explained as follows:

1. Security parameters to define the level of the bit durability which will be provided by the encryption process.
2. Define G_1 and G_T groups.
3. Pair $\hat{e} : G_1 \times G_1 \rightarrow G_T$.
4. Define $p \mid \#E(F_q)$, where E/F_q denotes an elliptic curve along with embedding degree k and p is prime.
5. Define the size of G_1 and G_T groups by the security parameters.
6. Let $\sigma \in E(F_q)[p]$, where σ is a random point on the elliptic curve.
7. Let G_1 and G_T be a cyclic group of order σ such as $G_1 = \langle\sigma\rangle$ and $G_T = \langle\hat{e}(\sigma, \sigma)\rangle$.
8. Define a cryptographic hash function one h_1 to map the string of the identity to an integer such as $h_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_p$.
9. Define a cryptographic hash function two h_2 to hash the element of G_T such as $h_2 : G_T \rightarrow \{0, 1\}^n$, so we can associate the plaintext with it, where n is the bit string of the plaintext.
10. Define S integer as a master secret such as $S \in \mathbb{Z}_p$, which is shown in Table 10.1.

The public parameters of this scheme are $\left(G_1, G_T, \hat{e}, \sigma, s\sigma, h_1, h_2, v\right)$.

Table 10.1 Parameters of Sakai-Kasahara IBE scheme

Element	Description
p	Prime
q	Prime power
E/F_q	Elliptic curve
G_1	Cyclic group
G_T	Cyclic group
$\hat{e}$	Pairing
n	Positive integer
σ	A point on elliptic curve
$s\sigma$	A point on elliptic curve
h_1	A cryptographic one-way hash function
h_2	A cryptographic one-way hash function
h_3	A cryptographic one-way hash function
h_4	A cryptographic one-way hash function
v	Element of $F_{q^k}^*$
S	Master secret $S \in \mathbb{Z}_p$
$Priv_{ID} = \frac{1}{S+q_{ID}}\sigma$	A private key for additive notation
$Priv_{ID} = \sigma^{1/(S+q_{ID})}$	A private key for multiplicative notation and full scheme

Algorithm 11.1: Parameters _Setup ()

Input: A security parameter, an elliptic curve E, and a plaintext length n

Output: Public parameters $\theta_1 = \left(G_1, G_T, \hat{e}, \sigma, s\sigma, h_1, h_2, v\right)$ and a master secret S.

Procedure:

Begin

1. Select a prime$p$ and a prime power q with $p \mid \#E(F_q)$ which meets the security parameter requirement.
2. Pick up a random $\sigma \in E(F_q)[p]$ and let $G_1 = \langle\sigma\rangle$.
3. Embed the degree k to $F_{q^k}^*$ and pair $\hat{e} : G_1 \times G_1 \to F_{q^k}^*$.
4. Let $G_T = \left\langle\hat{e}\left(\sigma, \sigma\right)\right\rangle$.
5. Pick up a random $S \in \mathbb{Z}_p$.
6. Use cryptographic hash functions:
   (a) $h_1 : \{0,1\}^* \to \mathbb{Z}_p$
   (b) $h_2 : G_T \to \{0,1\}^n$

End

2.2 Extraction of the Private Key

The extraction of the private key is the responsibility of the receiver party. After listing out the security parameters, elements, and master key, the receiver party extracts the private keys as follows:

1. Map the *ID* to the integer $q_{ID} \in \mathbb{Z}_p$ by calculating $q_{ID} = h_1(ID)$.
2. Use master secret S to calculate the private key such as $Priv_{ID} = \frac{1}{S+q_{ID}}\sigma$, where the calculation of $\frac{1}{S+q_{ID}}$ occurs in $\mathbb{Z}_p^*$.

Algorithm 11.2: Private_Key_Extraction ()

Input: An identity *ID*, public parameters $\theta_1 = \left(G_1, G_T, \hat{e}, \sigma, s\sigma, h_1, h_2, v\right)$, and a master secret S

Output: A private ID $Priv_{ID}$

Procedure:

Begin

1. Calculate $Priv_{ID} = 1/S + q_{ID}$.

End

2.3 Sakai-Kasahara IBE Encryption

In this section, the sender needs to encrypt the message $M \in \{0,1\}^n$ and send it to the receiver along with identity *ID*, so the sender will perform some steps to encrypt the message as follows:

1. Map the identity to an integer and hash it using hash function one as $q_{ID} = h_1(ID)$.
2. Pick up a random number $R \in \mathbb{Z}_p$.
3. Calculate $L = R\,(s\sigma + q_{ID^\sigma}) = R\,(S + q_{ID^\sigma})$.
4. Calculate $\lambda = h_2(v)^R$.
5. Calculate $\omega = M \oplus \lambda$.
6. Define $C = (L, \omega)$ as a ciphertext.

Algorithm 11.3: Encryption ()

Input: A plaintext message $M \in \{0,1\}^n$, a string *ID*, public parameters $\theta_1 = \left(G_1, G_T, \hat{e}, \sigma, s\sigma, h_1, h_2, v\right)$, and a master secret S

Output: A ciphertext $C = (L, \omega)$

Procedure:

Begin

1. Calculate $q_{ID} = h_1(ID)$.
2. Pick up a random number: $R \in \mathbb{Z}_p$.
3. Calculate $L = R\,(s\sigma + q_{ID^\sigma}) = R\,(S + q_{ID^\sigma})$.

4. Calculate $\lambda = h_2(v)^R$.
5. Calculate $\omega = M \oplus \lambda$.
6. Calculate $C = (L, \omega)$.

End

2.4 Sakai-Kasahara IBE Decryption

In the section, the receiver needs to decrypt the message that has been sent by the sender to get the plaintext by performing the following steps:

1. Calculate $\lambda = h_2\left(\hat{e}\left(L, Priv_{ID}\right)\right)$.
2. Calculate $M = (\omega \oplus \lambda)$.

Note that

$$\hat{e}\left(L, Priv_{ID}\right) = \hat{e}\left(R\left(S + q_{ID}\right)\sigma, \frac{1}{S + q_{ID}}\sigma\right) = \hat{e}(\sigma, \sigma)^R$$

So, step 5 of the encryption section and step 2 of the decryption section are calculating the same λ that permits the receiver to decrypt the ciphertext correctly.

Algorithm 11.4 Decryption ()

Input: A ciphertext $C = (L, \omega)$, public parameters$\theta_1 = \left(G_1, G_T, \hat{e}, \sigma, s\sigma, h_1, h_2, v\right)$, and a private key $Priv_{ID}$

Output: A plaintext message$M$

Procedure:

Begin

1. Calculate $\lambda = h_2\left(\hat{e}\left(L, Priv_{ID}\right)\right)$.
2. Calculate $M = (\omega \oplus \lambda)$.

End

3 Sakai-Kasahara IBE (Basic Scheme: Multiplicative Notation)

The S-K IBE basic scheme is less secure than the S-K IBE full scheme but easier to understand. In the basic scheme, two parties need to exchange encrypted messages safely. Both parties must agree on a unique shared secret to encrypt the message in plaintext form. The first party (sender) calculates the shared secret from its public parameters and identity of the second party (receiver). The receiver gets the shared secret by calculating the ciphertext and its private key.

3.1 Setup of Parameters

This scheme deals with multiplicative notations; thus $E(F_q)$ is an elliptic curve group and σ_1, σ_2 are two elements of the elliptic curve; then we consider $\sigma_1\sigma_2$ to point out $E(F_q)$ group operation to be applied to the group elements σ_1,σ_2 and multiply σ by integer s indicated as σ^s.

To implement this scheme, we need to define some basic essential elements shown in Table 10.1 and explained as follows:

1. Security parameters to define the level of the bit durability which will be provided by the encryption process.
2. Define G_1 and G_T groups.
3. Pair $\hat{e} : G_1 \times G_1 \to G_T$.
4. Define $p \mid \#\ E(F_q)$, where E/F_q denotes an elliptic curve along with the embedding degree k and $p$is prime.
5. Define the size of G_1 and G_T groups by the security parameters.
6. Let $\sigma \in E(F_q)[p]$ where σ is a random point on the elliptic curve.
7. Let G_1 and G_T be a cyclic group of order σ such as $G_1 = \langle\sigma\rangle$ and $G_T = \langle\hat{e}(\sigma, \sigma)\rangle$.
8. Define a cryptographic hash function one h_1 to map the string of the identity to an integer such as$h_1 : \{0, 1\}^* \to \mathbb{Z}_p$.
9. Define a cryptographic hash function two h_2 to hash the element of G_T such as$h_2 : G_T \to \{0, 1\}^n$, so we can associate the plaintext with it, where n is the bit string of the plaintext.
10. Define S integer as a master secret such as $S \in \mathbb{Z}_p$, which is shown in Table 10.1.

The public parameters of this scheme are $\left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, v\right)$.

Algorithm 11.5 Parameters_ Setup ()

Input: A security parameter, an elliptic curve E, and a plaintext length n

Output: Public parameters $\theta_2 = \left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, v\right)$ and a master secret S

Procedure:

Begin

1. Select a prime p and a prime power q with $p \mid \#\ E(F_q)$ which meets the security parameter requirement.
2. Pick up a random $\sigma \in E(F_q)[p]$ and let $G_1 = \langle\sigma\rangle$.
3. Embed the degree k to $F^*_{q^k}$ and pair $\hat{e} : G_1 \times G_1 \to F^*_{q^k}$.
4. Let $G_T = \langle\hat{e}(\sigma, \sigma)\rangle$.
5. Pick up a random $S \in \mathbb{Z}_p$.
6. Use cryptographic hash functions:
   (a) $h_1 : \{0, 1\}^* \to \mathbb{Z}_p$
   (b) $h_2 : G_T \to \{0, 1\}^n$

End

3.2 Extraction of the Private Key

The extraction of the private key is the responsibility of the receiver party. After listing out the security parameters, elements, and master key, the receiver party extracts the private keys as follows:

1. Map the *ID* to the integer $q_{ID} \in \mathbb{Z}_p$ by calculating$q_{ID} = h_1(ID)$.
2. Use master secret S to calculate the private key such as $Priv_{ID} = \sigma^{1/(S+q_{ID})}$.

Algorithm 11.6 Private_Key_Extraction ()

Input: An identity*ID*, public parameters$\theta_2 = \left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, v\right)$, and a master secret S

Output: A private ID $Priv_{ID}$

Procedure

Begin

1. Calculate $Priv_{ID} = \sigma^{1/(S+q_{ID})}$.

End

3.3 Sakai-Kasahara IBE Encryption

In this section, the sender needs to encrypt the message $M \in \{0,1\}^n$ and send it to the receiver along with identity*ID*, so the sender will perform some steps to encrypt the message as follows:

1. Map the identity to an integer and hash it using hash function one as $q_{ID} = h_1(ID)$.
2. Pick up a random number $R \in \mathbb{Z}_p$.
3. Calculate $L = R\left(\sigma^S \sigma^{q_{ID}}\right)^R = \sigma^{R(S+q_{ID})}$.
4. Calculate $\lambda = h_2(v)^R$.
5. Calculate $\omega = M \oplus \lambda$.
6. Define $C = (L, \omega)$ as a ciphertext.

Algorithm 11.7 Encryption ()

Input: A plaintext message $M \in \{0,1\}^n$, a string *ID*, public parameters $\theta_2 = \left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, v\right)$, and a master secret S

Output: A ciphertext $C = (L, \omega)$

Procedure

Begin

1. Calculate $q_{ID} = h_1(ID)$.
2. Pick up a random number: $R \in \mathbb{Z}_p$.
3. Calculate $L = R\left(\sigma^S \sigma^{q_{ID}}\right)^R = \sigma^{R(S+q_{ID})}$.

4. Calculate $\lambda = h_2(v)^R$.
5. Calculate $\omega = M \oplus \lambda$.
6. Calculate $C = (L, \omega)$.

End

3.4 Sakai-Kasahara IBE Decryption

In the section, the receiver needs to decrypt the message that has been sent by the sender to get the plaintext by performing the following steps:

1. Calculate $\lambda = h_2\left(\hat{e}\,(L,\, Priv_{ID})\right)$.
2. Calculate $M = (\omega \oplus \lambda)$.

Note that

$$\hat{e}\,(L,\, Priv_{ID}) = \hat{e}\left(\sigma^{R(S+q_{ID})}, \sigma^{1/(S+q_{ID})}\right) = \hat{e}(\sigma, \sigma)^R$$

So, step 5 of the encryption section and step 2 of the decryption section are calculating the same λ that permits the receiver to decrypt the ciphertext correctly.

Algorithm 11.8 Decryption ()
Input: A ciphertext$C = (L, \omega)$, public parameters
$\theta_2 = \left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, v\right)$, and a private key $Priv_{ID}$
Output: A plaintext message M
Procedure
Begin

1. Calculate $\lambda = h_2\left(\hat{e}\,(L,\, Priv_{ID})\right)$.
2. Calculate $M = (\omega \oplus \lambda)$.

End

4 Sakai-Kasahara IBE (Full Scheme)

The basic scheme is insecure to chosen ciphertext attack: if an attacker wants to get the plaintext back, the attacker will decrypt the ciphertext $C(L, \omega \oplus \varepsilon)$ to get the plaintext $M \oplus \varepsilon$, and the attacker reconstructs M as $M = (M \oplus \varepsilon)$. The full scheme is intended to overcome this insecurity by adding the Fujisaki-Okamoto transformation technique [8] to the basic scheme [7].

4.1 Setup of Parameters

The setup of parameters in the full scheme is similar to the basic scheme along with some extra parameters. We need extra hash function parameters to impose the security against chosen ciphertext attack. Principally, we need two hash functions $h_3 : \{0,1\}^n \rightarrow \mathbb{Z}_p$ and $h_4 : \{0,1\}^n \rightarrow \{0,1\}^n$ and need to add these two hash functions into the list of public parameters of the full scheme. The master secret remains the same as in the basic scheme. The public parameters of this scheme are $\left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, h_{3,} h_{4,} v\right)$.

Algorithm 11.9 Parameters _Setup ()

Input: A security parameter, an elliptic curve E, and a plaintext length n

Output: Public parameters $\theta_{3=} \left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, h_{3,} h_{4,} v\right)$ and a master secret S

Procedure

Begin

1. Select a prime p and a prime power q with $p \mid \# E(F_q)$ which meets the security parameter requirement.
2. Pick up a random $\sigma \in E(F_q)[p]$ and let $G_1 = \langle \sigma \rangle$.
3. Embed the degree $k$to $F_{q^k}^*$ and pair $\hat{e} : G_1 \times G_1 \rightarrow F_{q^k}^*$.
4. Let $G_T = \left\langle \hat{e}\left(\sigma, \sigma\right) \right\rangle$.
5. Pick up a random $S \in \mathbb{Z}_p^*$.
6. Use cryptographic hash functions:
   (a) $h_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$
   (b) $h_2 : G_T \rightarrow \{0,1\}^n$
   (c) $h_3 : \{0,1\}^n \rightarrow \mathbb{Z}_p$
   (d) $h_4 : \{0,1\}^n \rightarrow \{0,1\}^n$

End

4.2 Extraction of the Private Key

The extraction of the private key in the full scheme occurs as follows:

1. Map the ID to the integer $q_{ID} \in \mathbb{Z}_p$ by calculating $q_{ID} = h_1(ID)$.
2. Use master secret S to calculate the private key such as $Priv_{ID} = \sigma^{1/(S+q_{ID})}$.

Note that the extraction of the private key in the full scheme is similar to the extraction of the private key in the basic scheme.

Algorithm 11.10 Private_Key_Extraction ()

Input: An identity ID, public parameters $\theta_{3=} \left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, h_{3,} h_{4,} v\right)$, and a master secret S

Output: A private ID $Priv_{ID}$
Procedure
Begin

1. Calculate $Priv_{ID} = \sigma^{1/(S+q_{ID})}$.

End

4.3 Sakai-Kasahara IBE Encryption

In this section, the sender needs to encrypt the message $M \in \{0,1\}^n$ and send it to the receiver along with identity*ID*, so the sender will perform some steps to encrypt the message as follows:

1. Map the identity to an integer and hash it using hash function one as $q_{ID} = h_1(ID)$.
2. Pick up a random number $\tau \in \mathbb{Z}_p$.
3. Calculate $R = h_3(\tau, M)$.
4. Calculate $L = \left(\sigma^S \sigma^{q_{ID}}\right)^R = \sigma^{R(S+q_{ID})}$.
5. Calculate $\lambda = \tau \oplus h_2(v)^R$.
6. Calculate $\omega = M \oplus h_4(\tau)$.
7. Calculate $\delta = h_4(M)$.
8. Define $C = (L, \omega, \delta)$ as a ciphertext.

Algorithm 11.11 Encryption ()
Input: A plaintext message $M \in \{0,1\}^n$, a string ID, public parameters $\theta_{3=}\left(G_1, G_T, \hat{e}, n, \sigma, \sigma^s, h_1, h_2, h_{3,} h_{4,} v\right)$, and a master secret$S$
Output: A ciphertext $C = (L, \omega, \delta)$
Procedure
Begin

1. Calculate $q_{ID} = h_1(ID)$.
2. Pick up a random number: $\tau \in \mathbb{Z}_p$.
3. Calculate $R = h_3(\tau, M)$.
4. Calculate $L = \left(\sigma^S \sigma^{q_{ID}}\right)^R = \sigma^{R(S+q_{ID})}$.
5. Calculate $\lambda = \tau \oplus h_2(v)^R$.
6. Calculate $\omega = M \oplus h_4(\tau)$.
7. Calculate $\delta = h_4(M)$.
8. Calculate $C = (L, \omega, \delta)$.

End

4.4 Sakai-Kasahara IBE Decryption

In this section, the receiver needs to decrypt the message that has been sent by the sender to get the plaintext by performing the following steps:

1. Calculate $q_{ID} = h_1(ID)$.
2. Calculate $N = \hat{e}\,(L,\, Priv_{ID})$.
3. Calculate $\tau = \lambda \oplus h_2(N)$.
4. Calculate $M = \delta \oplus h_4(\tau)$.
5. Calculate $R = h_3(\tau, M)$.
6. If $L \neq \left(\sigma^{q_{ID}}\sigma^{S}\right)^{R}$, then an error has occurred, so exit.
7. Else assign the plaintext to M.

Algorithm 11.12 Decryption ()

Input: A ciphertext $C = (L, \omega, \delta)$, public parameters $\theta_{3=}\left(G_1, G_T, \hat{e}, n, \sigma, \sigma^{s}, h_1, h_2, h_3, h_4, v\right)$, and a private key $Priv_{ID}$

Output: A plaintext message M

Procedure

Begin

1. Calculate $q_{ID} = h_1(ID)$.
2. Calculate $N = \hat{e}\,(L,\, Priv_{ID})$.
3. Calculate $\tau = \lambda \oplus h_2(N)$.
4. Calculate $M = \delta \oplus h_4(\tau)$.
5. Calculate $R = h_3(\tau, M)$.
6. If $L \neq \left(\sigma^{q_{ID}}\sigma^{S}\right)^{R}$ exit.
7. Else plaintext $=M$.

End

5 Security of the Sakai-Kasahara IBE Scheme

In this section, we are going to prove that the S-K IBE scheme is secure against the adversary $\bar{E}$ using the random oracle model (ROM); therefore we have to define the one-way hash function (OWH) before we start the analysis.

Definition 1.1: The OWH function $f : \{0,1\}^{*} \rightarrow \{0,1\}^{n}$ that is considered to be infeasible to invert is that which can take any input $x \in \{0,1\}^{*}$ of arbitrary length and give an arbitrary-length output value $y = f(x) \in \{0,1\}^{n}$ which is called digest or hash value. While using the hash function, we have to consider the following properties:

1. $y = f(x) \in \{0,1\}^{n}$ is irreversible.

$$y = h(x) \neq h\left(x^{'}\right).$$

2. It is impossible to get $h(x^{'})$ if$x \neq x^{'}$.

Theorem 1.1: We assume that the OWH function closely operates as a random oracle. According to our assumption, the Sakai-Kasahara IBE scheme is provably secure against an adversary $\bar{E}$ to derive the message M.

Proof 1.1: We assume that the adversary $\bar{E}$ can derive the message M that has been sent from the sender to the receiver. To find out the message M, an adversary $\bar{E}$ has to use the experimental algorithm

$$EXPT_{HASH,\phi}^{h_3(\tau,M)}$$

The probability of success of the experimental algorithm is defined as

$$\left|SUCCESS_{HASH,\phi}^{h_3(\tau,M)} = \Pr\left[EXPT_{HASH,\phi}^{h_3(\tau,M)} = 1\right] - 1\right|$$

where Pr denotes the probability of success of $EXPT_{HASH,\phi}^{h_3(\tau,M)}$. The experimental algorithm is dependent on the advantage function that is defined as

$$ADVAT_{HASH,\phi}^{h_3(\tau,M)}(et, qR) = \max_{\phi}\left\{SUCCESS_{HASH,\phi}^{h_3(\tau,M)}\right\}$$

where *max* is specified by three factors:

1. Overall $\bar{A}$
2. The number of queries (qR) obtained from the execution time (et)
3. Reveal oracle

We can say that the S-K IBE scheme is vulnerable to the adversary $\bar{E}$ to derive the message M if

$$ADVAT_{HASH,\phi}^{h_3(\tau,M)}(et) \leq \varepsilon, \forall \varepsilon > 0.$$

Contemplating Algorithm 11.1, the adversary $\bar{E}$ can derive the message M if and only if it can invert the OWH function. According to Definition 11.1, the OWH function is infeasible to be inverted by cause of

$$ADVAT_{HASH,\phi}^{h_3(\tau,M)}(et, qR) \leq \varepsilon$$

Since $ADVAT_{HASH,\phi}^{h_3(\tau,M)}(et, qR)$ depends on $ADVAT_{HASH,\phi}^{h_3(\tau,M)}(et)$, the S-K IBE scheme is provably secure against the adversary $\bar{E}s$ to derive the message M.

References

1. Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques* (pp. 47–53). Berlin, Heidelberg: Springer.
2. Cocks, C. C. (1973). *A note on non-secret encryption* (p. 20). CESG Memo.
3. Boneh, D., & Franklin, M. (2003). Identity-based encryption from the Weil pairing. *SIAM J Comp, 32*(3), 586–615.
4. Boneh, D., & Boyen, X. (2004). Efficient selective-ID secure identity-based encryption without random oracles. In *International conference on the theory and applications of cryptographic techniques* (pp. 223–238). Berlin, Heidelberg: Springer.
5. Sakai, R., Ohgishi, K., & Kasahara, M. (2000). *Cryptosystems based on pairing. The 2000 Symposium on Cryptography and Information Security, Japan* (Vol. 45, pp. 26–28).
6. Chen, L., Cheng, Z., Malone-Lee, J., & Smart, N. P. (2005). *An efficient ID-KEM based on the Sakai-Kasahara key construction* (p. 224). IACR Cryptology ePrint Archive.
7. Martin, L. (2008). *Introduction to identity-based encryption*. Artech house.
8. Fujisaki, E., & Okamoto, T. (1999). Secure integration of asymmetric and symmetric encryption schemes. In *Annual international cryptology conference* (pp. 537–554). Berlin, Heidelberg: Springer.