IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 26, 2019, accepted April 19, 2019, date of publication May 7, 2019, date of current version May 21, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2915373*

# Identity-Based Revocation From Subset Difference Methods Under Simple Assumptions

**KWANGSU LEE[1], (Member, IEEE), AND JONG HWAN PARK[2]**
[1]Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea
[2]Department of Computer Science, Sangmyung University, Seoul 03016, South Korea

Corresponding author: Kwangsu Lee (kwangsu@sejong.ac.kr)

This work was supported in part by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean Government (MSIT) (A Study on Functional Encryption: Construction, Security Analysis, and Implementation) under Grant 2016-6-00600.

**ABSTRACT** Identity-based revocation (IBR) is a specific kind of broadcast encryption that can effectively send ciphertext to a set of receivers. In IBR, a ciphertext is associated with a set of revoked users instead of a set of receivers and the maximum number of users in the system can be an exponential value in the security parameter. In this paper, we reconsider the general method of Lee *et al.* (ESORICS 2014) that constructs a public-key revocation (PKR) scheme by combining the subset difference (SD) method of Naor, Naor, and Lotspiech (CRYPTO 2001) and a single revocation encryption (SRE) scheme. Lee *et al.* left it as an open problem to construct an SRE scheme under the standard assumption without random oracles. In this paper, we first propose a selectively secure SRE scheme under the standard assumption without random oracles. We also propose a fully secure SRE scheme under simple static assumptions without random oracles. Next, we present an efficient IBR scheme that supports fast decryption by combining the SD method and our SRE scheme. The security of our IBR scheme depends on that of the underlying SRE scheme. Finally, we implemented our SRE and IBR schemes and measured the performance.

**INDEX TERMS** Broadcast encryption, identity-based revocation, subset cover framework, bilinear maps.

## I. INTRODUCTION

Public-key broadcast encryption (PKBE) is a special type of public-key encryption (PKE) such that any user can create a compact ciphertext for a dynamic changing set of receivers. PKBE can be used for secure group communication systems, pay-TV systems, content distribution systems, and secure file systems. Public-key revocation (PKR) is a variation of PKBE where a ciphertext is associated with a set $R$ of revoked users instead of a set $S$ of receivers and a user can decrypt the ciphertext if he is not revoked in the ciphertext. PKBE can be extended to identity-based broadcast encryption (IBBE) where a user is mapped to any identity string and the total number of users in the system can be an exponential value in the security parameter. We also can define identity-based revocation (IBR) by associating a ciphertext with a set of revoked users $R$ instead of a set of receivers $S$.

The associate editor coordinating the review of this manuscript and approving it for publication was Cristina Rottondi.

One method to build a collusion-resistant PKBE scheme is to use bilinear groups. Boneh *et al.* [1] proposed the first PKBE scheme with short ciphertexts in bilinear groups and proved its selective security under $q$-type assumption. After their work, other PKBE, IBBE, and IBR schemes were proposed in bilinear groups [2]–[6]. Another method to build a secure PKBE scheme is to combine the subset cover framework of Naor *et al.* [7] and an identity-based encryption (IBE) scheme [8]. Naor *et al.* [7] showed that a PKR scheme can be obtained from the complete subtree (CS) method and an IBE scheme and Dodis and Fazio [9] showed that an efficient PKR scheme can be derived from the subset difference (SD) method and an hierarchical IBE (HIBE) scheme. Recently, Lee *et al.* [10] showed that an improved PKR scheme can be derived by combining the SD method with a single revocation encryption (SRE). An SRE scheme allows to broadcast to a group of users labeled by a group label $GL$, but it has the option of revoking a single user $ML$ from that group. Compared to PKR schemes that are directly built on bilinear groups, PKR schemes derived from the subset cover framework provide short public parameters and efficient operations in the decryption algorithm.

The PKR scheme of Lee *et al.* [10] that combines the SD method with an SRE scheme is interesting since it achieves the asymptotically optimal bound in the SD method and it supports fast decryption. The SD method is one instance of the subset cover framework of Naor *et al.* [7] and it can be used to build an efficient revocation system where a ciphertext is associated to a set of subsets that covers all receivers by excluding revoked users. In SD, a subset is defined by a subtree $\mathcal{T}_{i,j}$ that is related with two nodes $v_i$ and $v_j$ in a binary tree. That is, $\mathcal{T}_{i,j}$ is defined as a set of leaf nodes in $\mathcal{T}_i$ but not in $\mathcal{T}_j$ where the root node of $\mathcal{T}_i, \mathcal{T}_j$ is $v_i, v_j$ respectively. In SRE, a ciphertext is associated with labels $(GL, ML)$ and a private key is associated with labels $(GL', ML')$ and a ciphertext can be decrypted if $GL = GL'$ and $ML \neq ML'$ [10]. To construct an improved PKR scheme, Lee *et al.* [10] observed that a subset $\mathcal{T}_{i,j}$ in the SD method can be directly mapped to labels $(GL, ML)$ in the SRE scheme. Although the PKR scheme of Lee et al. can reduce the size of public keys and private keys compared to the PKR scheme of Dodis and Fazio, their SRE scheme is proven to be secure under $q$-type assumption in the random oracle model. Thus, they left it as an interesting problem to build an SRE scheme under standard assumptions without random oracles.

### A. OUR CONTRIBUTIONS

In this paper, we give affirmative answers to the above interesting problem. We obtain the following results:

#### 1) SRE WITH SELECTIVE SECURITY

We first propose an SRE scheme in prime-order bilinear groups and prove its selective security under a standard assumption without random oracles. In SRE, a ciphertext and a private key are associated with labels $(GL, ML)$ and $(GL', ML')$ respectively and the ciphertext can be decrypted if $GL = GL'$ and $ML \neq ML'$. Previously, Lee *et al.* [10] attempted to derive an SRE scheme by carefully modifying the IBR scheme of Lewko *et al.* [6]. However, this directly derived SRE scheme essentially used random oracles to support the equality of labels. The main idea of our SRE scheme under the standard assumption is that we can combine an IBE scheme and a simple IBR scheme since the IBE scheme can be directly used to support the equality $GL = GL'$ and the simple IBR scheme can be directly used to support the inequality $ML \neq ML'$. In this case, an SRE can be proven to be secure under the standard assumption since both an IBE scheme and a simple IBR scheme can be proved to be secure under the decisional bilinear Diffie-Hellman (DBDH) assumption.

#### 2) SRE WITH FULL SECURITY

Our first SRE scheme is just secure in the selective model in which an adversary should submit the target labels before it receives public parameters. To construct an SRE scheme with full security, we propose an SRE scheme in composite-order bilinear groups and prove its full security under simple static assumptions. The structure of our second SRE scheme is similar to that of our first scheme except that it uses composite-order bilinear groups to use the dual system encryption technique of Waters [11], Lewko and Waters [12]. To prove the full security of our second SRE scheme, we carefully analyze our SRE scheme and shows that the information theoretic argument of the dual system encryption technique still holds in our SRE scheme.

#### 3) IBR FROM SUBSET DIFFERENCE

To construct an IBR scheme by combining the SD method with an SRE scheme, we follow the design principle of Lee *et al.* [10]. As mentioned before, an SRE scheme can be integrated with the SD method since a subset $\mathcal{T}_{i,j}$ in SD can be directly mapped to the labels $(GL, ML)$ in SRE. Lee *et al.* only proposed a PKR scheme in which the maximum number of users is fixed to be a polynomial value in the security parameter since their SRE scheme is proven under a $q$-type assumption where $q$ is related to the maximum number of users in the systems. However, our scheme can be identity-based one by extending the depth of a binary tree since our SRE scheme can support any label strings. Additionally, our IBR scheme provides better efficiency since it adopts the hybrid approach that encrypts a session key by using an SRE scheme and encrypts a message by using a symmetric-key encryption scheme. The security of our IBR scheme follows that of the underlying SRE scheme. Our IBR scheme can be also integrated with the efficient layered SD (LSD) scheme [13]. The detailed comparisons of IBR schemes are given in Table 1. We also implemented our SRE and IBR schemes in Charm and measured the performance these implementations.

### B. RELATED WORK

Broadcast encryption, introduced by Fiat and Naor [15], is symmetric-key encryption where a trusted center which knows all private keys of all users can create a ciphertext for a set of receivers. Fiat and Naor proposed broadcast encryption schemes in the bounded collusion security model. The full literature of broadcast encryption is extensive and it is beyond the scope of this paper. We will only review some papers that are relevant to our work. Naor *et al.* [7] proposed the general methodology named the subset cover framework for revocation systems. The complete subtree (CS) and subset difference (SD) methods in binary trees are two important instances of the subset cover framework. The subset cover framework can be extended to trace-and-revoke by incorporating the tracing functionality that can trace a traitor of the system. After their work, other improved method was proposed [13], [16].

In public-key broadcast encryption (PKBE), any user can create a ciphertext for a set of receivers $S$ by using a public key whereas only the center can create a ciphertext in (symmetric-key) broadcast encryption. As mentioned before, public-key revocation (PKR) is a variation of PKBE where a ciphertext is

**TABLE 1.** Comparisons of identity-based revocation schemes.

| Scheme | PP Size | CT Size | SK Size | Decrypt Cost | Model | Assumption |
|---|---|---|---|---|---|---|
| NNL [7] | $O(1)$ | $O(r \log \frac{N}{r})$ | $O(\log N)$ | 1P | RO, AD | BDH |
| DF [9] | $O(\log N)$ | $O(r)$ | $O(\log^{2.5} N)$ | 2P | SE | $q$-Type |
| LSW1 [6] | $O(1)$ | $O(r)$ | $O(1)$ | 3P + 2$r$E | SE | $q$-Type |
| LSW2 [6] | $O(1)$ | $O(r)$ | $O(1)$ | 2$r$P + $r$E | SE | DLIN, DBDH |
| AL1 [14] | $O(n)$ | $O(1)$ | $O(n)$ | 2P + $r$E | SE, BR | DLIN, DBDH |
| AL2 [14] | $O(n)$ | $O(r/n)$ | $O(n)$ | $2r/n$P + $r/n$E | SE | DLIN, DBDH |
| CLR [23] | $O(n)$ | $O(1)$ | $O(1)$ | 2P + $n$E | SE, BR | Static |
| Ours | $O(1)$ | $O(r)$ | $O(\log^{1.5} N)$ | 3P + 2E | SE | DBDH |
| Ours | $O(1)$ | $O(r)$ | $O(\log^{1.5} N)$ | 3P + 2E | AD | Static |

We let $\lambda$ be a security parameter, $N$ be the number of maximum users, $r$ be the number of revoked users, and $n$ be a bound parameter. We count the number of group elements to measure the size of parameters. We use symbols P for a pairing operation and E for an exponentiation. We also use symbols RO for random oracle model, AD for adaptive model, SE for selective model, and BR for bounded revocation where the maximum number of revocation is bounded.

associated with a set of revoked users $R$. Naor and Pinkas [17] introduced revocation systems and proposed a PKR scheme by using a polynomial-based secret key sharing method in the bounded collusion model. Boneh *et al.* [1] proposed a fully collusion-resistant PKBE scheme in bilinear groups that achieves short ciphertexts. After that, many PKBE scheme in bilinear groups were presented [2], [5], [6], [18]–[20]. The subset cover framework also can be used to build a PKR scheme by combining it with an IBE, HIBE, or SRE scheme [7], [9], [10].

Identity-based broadcast encryption (IBBE) is a special type of PKBE where the maximum number of users in the system can be an exponential value in the security parameter since the size of public parameters is not linearly dependent on the number of users. Delerablée [3], Sakai, and Furukawa [4] independently proposed the first IBBE schemes with short ciphertexts and keys, but the maximum size of a receiver set should be fixed in public parameters. After that, ideal IBBE schemes that have constant size ciphertexts and keys with short public parameters were proposed in multilinear maps [21], [22], but many multilinear map candidates were broken.

Identity-based revocation (IBR) is a variation of IBBE where a set of revoked users $R$ is specified in a ciphertext instead of a set of receivers $S$. Although a PKBE scheme can be easily converted to a PKR scheme, an IBBE scheme cannot be converted to an IBR scheme since the maximum number of users is an exponential value. Lewko *et al.* [6] proposed IBR schemes with ciphertext size $O(r)$ and short keys in bilinear maps where $r$ is the number of revoked users and proved their selective security under standard assumptions. Attrapadung and Libert [14] proposed an a non-zero inner-product encryption (NIPE) scheme with constant size ciphertexts and a generalized NIPE scheme for IBR. Recently, Chen *et al.* [23] proposed improved NIPE schemes with short ciphertexts and short keys. However, all IBR schemes directly derived from NIPE schemes have limitations that the maximum number of revoked users in ciphertexts is bounded.

## II. PRELIMINARIES

In this section, we define single revocation encryption (SRE) and identity-based revocation (IBR) and their security models.

### A. SINGLE REVOCATION ENCRYPTION

Before we define IBR, we first define SRE. The concept of SRE was introduced by Lee *et al.* [10] and this SRE scheme is a new public-key encryption scheme that can be combined with the subset difference method to construct an efficient IBR scheme. In SRE, each user is associated with a group label $GL'$ and a member label $ML'$ and he is given a private key for the labels $(GL', ML')$. A sender can create a ciphertext for a specific group label $GL$ excluding one revoked member label $ML$. A receiver who has a private key for labels $(GL', ML')$ can decrypt the ciphertext for $(GL, ML)$ if he belongs to the same group but he is not revoked. That is, $GL' = GL$ and $ML' \neq ML$. The formal syntax of SRE is given as follows:

*Definition 1 (Single Revocation Encryption): An SRE scheme for group and member labels consists of four algorithms **Setup**, **GenKey**, **Encrypt**, and **Decrypt**, which are defined as follows*:

- ***Setup***$(1^\lambda)$. *The setup algorithm takes as input a security parameter* $1^\lambda$. *It outputs a master key MK and public parameters PP.*
- ***GenKey***$((GL, ML), MK, PP)$. *The key generation algorithm takes as input labels* $(GL, ML)$, *the master key MK, and public parameters PP. It outputs a private key SK for the labels* $(GL, ML)$.
- ***Encrypt***$((GL, ML), M, PP)$. *The encryption algorithm takes as input labels* $(GL, ML)$, *a message* $M \in \mathcal{M}$, *and public parameters PP. It outputs a ciphertext CT for* $(GL, ML)$ *and* $M$.
- ***Decrypt***$(CT, SK, PP)$. *The decryption algorithm takes as input a ciphertext CT for labels* $(GL, ML)$, *a private*

*key SK for labels $(GL', ML')$, and public parameters PP. It outputs an encrypted message M or $\perp$.*

*The correctness property of SRE is defined as follows: For all $MK, PP$ generated by **Setup**, all $(GL, ML)$, any $SK_{(GL',ML')}$ generated by **GenKey**, and any M, it is required that*

- *If $(GL = GL') \wedge (ML \neq ML')$, **Decrypt**(**Encrypt**($(GL, ML), M, PP$), $SK_{(GL',ML')}, PK$) = M.*
- *If $(GL \neq GL') \vee (ML = ML')$, **Decrypt**(**Encrypt**($(GL, ML), M, PP$), $SK_{(GL',ML')}, PK$) = $\perp$.*

The security model of SRE was defined by Lee *et al.* [10] and we follow their definition of chosen-plaintext attack (CPA) security of SRE. In the CPA security, an adversary can adaptively obtain a private key for labels $(GL, ML)$ many times. In the challenge step, the adversary submits challenge labels $(GL^*, ML^*)$ with some restrictions and two challenge messages and then he receives a challenge ciphertext that is an encryption of one of the challenge messages. The adversary may obtain additional private keys and finally outputs a guess of the challenge ciphertext. The detailed description of the security model is given as follows:

*Definition 2 (IND-CPA Security): The security of SRE is defined in terms of the indistinguishability under chosen plaintext attacks (IND-CPA). The security game is defined as the following game between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$:*

1) ***Setup**: $\mathcal{C}$ runs **Setup**($1^\lambda$) to generate a master key MK and public parameters PP. It keeps MK to itself and gives PP to $\mathcal{A}$.*
2) ***Query 1**: $\mathcal{A}$ adaptively requests private keys for labels $(GL_1, ML_1), \ldots, (GL_{q_1}, ML_{q_1})$. In response, $\mathcal{C}$ gives the corresponding private keys $SK_1, \ldots, SK_{q_1}$ to $\mathcal{A}$ by running **GenKey**($(GL_i, ML_i), MK, PP$).*
3) ***Challenge**: $\mathcal{A}$ submits challenge labels $(GL^*, ML^*)$ and two messages $M_0^*, M_1^*$ with the equal length subject to the restriction: for all $(GL_i, ML_i)$ of private key queries, it is required that $(GL_i \neq GL^*)$ or $(GL_i = GL^*) \wedge (ML_i = ML^*)$. $\mathcal{C}$ flips a random coin $\mu \in \{0, 1\}$ and gives the challenge ciphertext $CT^*$ to $\mathcal{A}$ by running **Encrypt**($(GL^*, ML^*), M_\mu^*, PP$).*
4) ***Query 2**: $\mathcal{A}$ may continue to request private keys for labels $(GL_{q_1+1}, ML_{q_1+1}), \ldots, (GL_q, ML_q)$.*
5) ***Guess**: $\mathcal{A}$ outputs a guess $\mu' \in \{0, 1\}$ of $\mu$, and wins the game if $\mu = \mu'$.*

*The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{SRE}(\lambda) = \left|\Pr[\mu = \mu'] - \frac{1}{2}\right|$ where the probability is taken over all the randomness of the game. A SRE scheme is secure under chosen plaintext attacks if for all PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the above game is negligible in the security parameter $\lambda$.*

*Remark 3:* The selective security of SRE is a weaker security model in which an adversary initially submits the challenge labels $(GL^*, ML^*)$ before receiving public parameters.

### B. IDENTITY-BASED REVOCATION

IBR is a special type of PKBE where a ciphertext is associated with a revoked set $R$ of users instead of a receiver set $S$ of users and each user is specified by a unique identifier string $ID$. In IBR, a center generates a private key for a user $ID$ by using his master key and gives it to the user. A sender can create a ciphertext for receivers that excludes the set of revoked users $R$ and a receiver with $ID$ can decrypt the ciphertext if $ID \notin R$. The formal syntax of IBR is given as follows:

*Definition 4 (Identity-Based Revocation): An identity-based revocation (IBR) scheme for the identity $\mathcal{I}$ consists of four algorithms **Setup**, **GenKey**, **Encrypt**, and **Decrypt**, which are defined as follows*:

- ***Setup**($1^\lambda$). The setup algorithm takes as input a security parameter $1^\lambda$. It outputs a master key MK and public parameters PP.*
- ***GenKey**($ID, MK, PP$). The key generation algorithm takes as input an identity $ID \in \mathcal{I}$, the master key MK, and the public parameters PP. It outputs a private key $SK_{ID}$.*
- ***Encrypt**($R, M, PP$). The encryption algorithm takes as input a revoked set R of users, a message $M \in \{0, 1\}^m$, and the public parameters PP. It outputs a ciphertext $CT_R$ for R and M.*
- ***Decrypt**($CT_R, SK_{ID}, PP$). The decryption algorithm takes as input a ciphertext $CT_R$ for a revoked set R, a private key $SK_{ID}$ for an identity ID, and the public parameters PP. It outputs a message M or $\perp$.*

*The correctness property of IBR is defined as follows: For all $MK, PK$ generated by **Setup**, all $ID, R$, any $SK_{ID}$ generated by **GenKey**, and any M, it is required that*

- *If $ID \notin R$, **Decrypt**(**Encrypt**($R, M, PP$), $SK_{ID}, PP$) = M.*
- *If $ID \in R$, **Decrypt**(**Encrypt**($R, M, PP$), $SK_{ID}, PP$) = $\perp$.*

The security model of IBR is similar to that of IBBE and we follow the security definition of Lewko *et al.* [6]. In the CPA security, an adversary can request a private key of a user with $ID$ many times. In the challenge step, the adversary submits a challenge revoked set $R^*$ and two challenge messages with some restrictions and receives a challenge ciphertext that is the encryption of one challenge message. The adversary further can request private keys of other users and finally outputs the guess of the challenge message. The detailed description of the security is described as follows:

*Definition 5 (IND-CPA Security): The indistinguishability property of IBR under a chosen plaintext attack is defined in terms of the following game between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$:*

1) ***Setup**: $\mathcal{C}$ runs **Setup**($1^\lambda$) to generate a master key MK and public parameters PP. It keeps MK to itself and gives PP to $\mathcal{A}$.*
2) ***Query 1**: $\mathcal{A}$ may adaptively request private keys for users $ID_1, \ldots, ID_{q_1} \in \mathcal{I}$. In response, $\mathcal{C}$ gives the corresponding private keys $SK_{ID_1}, \ldots, SK_{ID_{q_1}}$ to $\mathcal{A}$ by running **GenKey**($ID_i, MK, PP$).*
3) ***Challenge**: $\mathcal{A}$ submits a challenge revoked set $R^*$ of users and two messages $M_0^*, M_1^*$ with the equal length*

*subject to the restriction: for all $ID_i$ of private key queries, $ID_i \in R^*$. $\mathcal{C}$ flips a random coin $\mu \in \{0,1\}$ and gives the challenge ciphertext $CT^*$ to $\mathcal{A}$ by running* ***Encrypt***$(R^*, M_\mu^*, PP)$.

4) ***Query 2***: *$\mathcal{A}$ may continue to request private keys for users $ID_{q_1+1}, \ldots, ID_q \in \mathcal{I}$.*
5) ***Guess***: *$\mathcal{A}$ outputs a guess $\mu' \in \{0,1\}$ of $\mu$, and wins the game if $\mu = \mu'$.*

*The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{IBR}(\lambda) = \left|\Pr[\mu = \mu'] - \frac{1}{2}\right|$ where the probability is taken over all the randomness of the game. An IBR scheme is secure under chosen plaintext attacks if for all PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the above game is negligible in the security parameter $\lambda$.*

*Remark 6:* The selective security of IBR is a weaker security model in which an adversary initially submits the challenge revoked set $R^*$ before receiving public parameters.

## III. SRE WITH SELECTIVE SECURITY

In this section, we propose a selectively secure single revocation encryption (SRE) scheme in prime-order bilinear groups and prove its security under the standard assumption.

### A. BILINEAR GROUPS OF PRIME ORDER

Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order $p$. Let $g$ be a generator of $\mathbb{G}$. The bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has the following properties:

1) Bilinearity: $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
2) Non-degeneracy: $\exists g$ such that $e(g, g)$ has order $p$, that is, $e(g, g)$ is a generator of $\mathbb{G}_T$.

We say that $\mathbb{G}, \mathbb{G}_T$ are bilinear groups if the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map $e$ are all efficiently computable.

### B. COMPLEXITY ASSUMPTIONS

To prove the security of our SRE scheme, we use the well-known standard DBDH assumption. The DBDH assumption was introduced by Boneh and Franklin [8] and widely used to prove the security of IBE, HIBE, and ABE schemes.

*Assumption 7 (Decisional Bilinear Diffie-Hellman, DBDH): Let $(p, \mathbb{G}, \mathbb{G}_T, e)$ be a description of the bilinear group of prime order $p$ with the security parameter $\lambda$. Let $g$ be a generator of $\mathbb{G}$. The DBDH assumption is that if the challenge values $D = ((p, \mathbb{G}, \mathbb{G}_T, e), \ g, g^a, g^b, g^c)$ and $T$ are given, no PPT algorithm $\mathcal{B}$ can distinguish $T = T_0 = e(g, g)^{abc}$ from $T = T_1 = e(g, g)^d$ with more than a negligible advantage. The advantage of $\mathcal{B}$ is defined as $Adv_{\mathcal{B}}^{DBDH}(\lambda) = \left|\Pr[\mathcal{B}(D, T_0) = 0] - \Pr[\mathcal{B}(D, T_1) = 0]\right|$ where the probability is taken over the random choice of $a, b, c, d \in \mathbb{Z}_p$.*

### C. CONSTRUCTION

The previous SRE scheme of Lee *et al.* [10] was proven to be secure under $q$-type assumption in the random oracle model. To construct an SRE scheme that is secure under the standard assumption without the random oracle model, we inspect the correctness property of SRE. In SRE, a ciphertext is associated with group and member labels $(GL, ML)$ and a private key is associated with labels $(GL', ML')$. The correctness property requires that the group labels should be equal but the member labels should be not equal to decrypt the ciphertext by using the private key. That is, $GL = GL'$ and $ML \neq ML'$. We observe that an IBE scheme can be used for equality and a simple IBR scheme where the number of revoked users is just one can be used for inequality. More specifically, the IBE (BB-IBE) scheme of Boneh and Boyen [24] can be used to support the equality of group labels where a private key is structured as $(g^\alpha (u^{GL} h)^r, g^{-r})$. The simple IBR (LSW-IBR) scheme of Lewko *et al.* [6] can be used to support the inequality of member labels and a private key is described as $(g^\alpha w^r, (w^{ML} v)^r, g^{-r})$. By combining the BB-IBE scheme and the simple LSW-IBR scheme, we can derive an SRE scheme with the private key structure of $(g^\alpha (u^{GL} h)^{r_1} w^{r_2}, (w^{ML} v)^{r_2}, g^{-r_1}, g^{-r_2})$.

Let $\mathcal{H}$ be a family of collision resistant hash function $H$. Our SRE scheme in prime-order bilinear groups is described as follows:

**SRE.Setup**$(1^\lambda)$**:** It first generates a bilinear group $\mathbb{G}$ of prime order $p$. Let $g$ be a generator of $\mathbb{G}$. It chooses a random $\alpha \in \mathbb{Z}_p$ and random $u, h, w, v \in \mathbb{G}$. It also chooses a random hash function $H$ from $\mathcal{H}$. It outputs $MK = \alpha$ and $PP = ((p, \mathbb{G}, \mathbb{G}_T, e), g, u, h, w, v, H, \Omega = e(g, g)^\alpha)$.

**SRE.GenKey**$((GL, ML), MK, PP)$**:** It selects random $r_1$, $r_2 \in \mathbb{Z}_p$ and outputs a private key $SK_{(GL,ML)} = \big(K_0 = g^\alpha (u^{GL} h)^{r_1} w^{r_2}, K_1 = (w^{ML} v)^{r_2}, K_2 = g^{-r_1}, K_3 = g^{-r_2}\big)$.

**SRE.Encrypt**$((GL, ML), M, PP)$**:** It chooses a random $t \in \mathbb{Z}_p$ and outputs a ciphertext $CT_{(GL,ML)} = \big(C = H(\Omega^t) \oplus M, C_0 = g^t, C_1 = (u^{GL} h)^t, C_2 = (w^{ML} v)^t\big)$.

**SRE.Decrypt**$(CT_{(GL,ML)}, SK_{(GL',ML')}, PP)$**:** If $(GL = GL') \wedge (ML \neq ML')$, then it computes $A = e(C_0, K_0) \cdot e(C_1, K_2) \cdot \big(e(C_0, K_1) \cdot e(C_2, K_3)\big)^{-1/(ML'-ML)}$ and outputs a message $M = C \oplus H(A)$. Otherwise, it outputs $\perp$.

### D. CORRECTNESS

The correctness of the above SRE scheme is easily verified by the following equation.

$$
\begin{aligned}
&e(C_0, K_0) \cdot e(C_1, K_1) \cdot (e(C_0, K_2) \cdot e(C_2, K_3))^{-1/(ML'-ML)} \\
&\quad = e(g^t, g^\alpha (u^{GL} h)^{r_1} w^{r_2}) \cdot e((u^{GL} h)^t, g^{-r_1}) \\
&\qquad \cdot \Big(e(g^t, (w^{ML'} v)^{r_2}) \cdot e((w^{ML} v)^t, g^{-r_2})\Big)^{-1/(ML'-ML)} \\
&\quad = e(g^t, g^\alpha w^{r_2}) \cdot \Big(e(g, w)^{tr_2 \cdot (ML'-ML)}\Big)^{-1/(ML'-ML)} \\
&\quad = e(g, g)^{\alpha t}.
\end{aligned}
$$

### E. SECURITY ANALYSIS

To prove the security of our SRE scheme in the selective model, we use the partitioning method that was used in the security proof of IBE and its extensions. Since our SRE scheme is derived from the BB-IBE scheme and the simple LSW-IBR scheme [6], [24], we may try to use the partitioning proof method of BB-IBE and LSW-IBR schemes. However, the original LSW-IBR scheme is proven under a complex $q$-type assumption. To prove the security under the standard assumption, we observe that a simple variant of the LSW-IBR scheme such that a ciphertext is associated with a single $ID$ instead of a set of revoked users $R$ is enough for SRE. In this case, we can prove the simple LSW-IBR scheme under the standard DBDH assumption. Therefore, we have the following result.

*Theorem 8: The above SRE scheme is selectively secure under chosen plaintext attacks if the DBDH assumption holds.*

*Proof:* Suppose there exists an adversary $\mathcal{A}$ that breaks the security game of SRE with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the DBDH assumption using $\mathcal{A}$ is given: a challenge tuple $D = ((p, \mathbb{G}, \mathbb{G}_T, e), g, g^a, g^b, g^c)$ and $T$ where $T = e(g, g)^{abc}$ or $T = e(g, g)^d$. Then $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

**Init**: $\mathcal{A}$ initially submits challenge labels $(GL^*, ML^*)$.

**Setup**: $\mathcal{B}$ selects random exponents $y_u, y_h, y_w, y_v \in \mathbb{Z}_p$ and creates public parameters implicitly setting $\alpha = ab$ as $PP = \big((p, \mathbb{G}, \mathbb{G}_T, e), g,\ u = g^a g^{y_u}, h = (g^a)^{-GL^*} g^{y_h}, w = g^a g^{y_w}, v = (g^a)^{-ML^*} g^{y_v},\ \Omega = e(g^a, g^b)\big)$.

**Query 1**: $\mathcal{A}$ may adaptively request a private key query for labels $(GL, ML)$. If $(GL = GL^*) \wedge (ML \neq ML^*)$, then it aborts since it cannot create a private key. Otherwise, it handles this query as follows:

- **Case** $GL \neq GL^*$: It selects random exponents $r_1', r_2 \in \mathbb{Z}_p$ and creates a private key by implicitly setting $r_1 = -b/(GL - GL^*) + r_1'$ as
$$K_0 = (g^b)^{-(y_u GL + y_h)/(GL - GL^*)} (u^{GL} h)^{r_1'} w^{r_2},$$
$$K_1 = (g^b)^{1/(GL - GL^*)} g^{-r_1'}, \quad K_2 = (w^{ML} v)^{r_2},\ K_3 = g^{-r_2}.$$
- **Case** $GL = GL^*$ and $ML = ML^*$: It selects random exponents $r_1, r_2' \in \mathbb{Z}_p$ and creates a private key by implicitly setting $r_2 = -b + r_2'$ as
$$K_0 = (u^{GL} h)^{r_1} (g^b)^{-y_w} w^{r_2'}, \quad K_1 = g^{-r_1},$$
$$K_2 = (g^b)^{-(y_w ML + y_v)} (w^{ML} v)^{r_2'}, \quad K_3 = g^b g^{-r_2'}.$$

**Challenge**: $\mathcal{A}$ submits two messages $M_0^*, M_1^*$. $\mathcal{B}$ flips a random coin $\mu \in \{0, 1\}$ internally. Next, it implicitly sets $t = c$ and creates a challenge ciphertext as
$$C = H(T) \cdot M_\mu^*, \quad C_0 = g^c,\ C_1 = (g^c)^{y_u GL^* + y_h},$$
$$C_2 = (g^c)^{y_w ML^* + y_v}.$$

**Query 2:** Same as Query 1.

**Guess**: Finally, $\mathcal{A}$ outputs a guess $\mu'$. If $\mu = \mu'$, $\mathcal{B}$ outputs 0. Otherwise, it outputs 1. □

### F. DISCUSSIONS

#### 1) EFFICIENCY ANALYSIS

In our SRE scheme, a private key and a ciphertext consist of four group elements respectively. The decryption algorithm requires four pairing operations and one exponentiation. To improve the efficiency, we can reduce one pairing operation by restating $e(C_0, K_0) \cdot e(C_0, K_1)^{1/(ML' - ML)}$ to $e(C_0, K_0\, K_1^{1/(ML' - ML)})$. Compared to the SRE scheme of Lee *et al.* [10] that is secure in the random oracle model, our SRE scheme requires one additional group element in the private key and the ciphertext, but our SRE scheme is secure under the standard assumption without random oracles.

#### 2) CCA SECURITY

Although we proved the CPA security of our SRE scheme, the CPA security is weaker than the chosen-ciphertext attack (CCA) security. In CCA security, an adversary additionally requests the decryption of a ciphertext adaptively chosen by the adversary. To prove the CCA security, we can use the generic CHK transformation of Canetti *et al.* [25]. That is, we can use a two-level HIBE scheme instead of an IBE scheme and a one-time signature scheme to construct an SRE scheme since the two-level HIBE scheme can be converted to a CCA-secure IBE scheme by the CHK transform. To construct a CCA-secure SRE scheme with better efficiency, we may use the technique of Boyen *et al.* [26], but we should modify our SRE scheme to be a key encapsulation mechanism (KEM).

## IV. SRE WITH FULL SECURITY

In this section, we propose a single revocation encryption (SRE) scheme in composite-order bilinear groups and prove its full-model security under simple assumptions.

### A. BILINEAR GROUPS OF COMPOSITE ORDER

Let $N = p_1 p_2 p_3$ where $p_1, p_2$, and $p_3$ are distinct prime numbers. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of same composite order $N$ and $g$ be a generator of $\mathbb{G}$. The bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has the following properties:

1) Bilinearity: $\forall u, v \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_N$, $e(u^a, v^b) = e(u, v)^{ab}$.
2) Non-degeneracy: $\exists g$ such that $e(g, g)$ has order $N$, that is, $e(g, g)$ is a generator of $\mathbb{G}_T$.

We say that $\mathbb{G}$ is a bilinear group if the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map $e$ are all efficiently computable. Furthermore, we assume that the description of $\mathbb{G}$ and $\mathbb{G}_T$ includes generators of $\mathbb{G}$ and $\mathbb{G}_T$ respectively. We use the notation $\mathbb{G}_{p_i}$ to denote the subgroups of order $p_i$ of $\mathbb{G}$ respectively. Similarly, we use the notation $\mathbb{G}_{T,p_i}$ to denote the subgroups of order $p_i$ of $\mathbb{G}_T$ respectively.

### B. COMPLEXITY ASSUMPTIONS

To prove the security of our SRE scheme, we introduce simple static assumptions that were used by Lewko and Waters [12] to prove the full model security of ABE schemes by using the dual system encryption technique.

*Assumption 9 (Subgroup Decision, SD): Let $(N, \mathbb{G}, \mathbb{G}_T, e)$ be a description of the bilinear group of composite order $N = p_1 p_2 p_3$. Let $g_1, g_2, g_3$ be generators of subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. The SD assumption is that if the challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_3)$ and $T$ are given, no PPT algorithm $\mathcal{A}$ can distinguish $T = T_0 = X_1 \in \mathbb{G}_{p_1}$ from $T = T_1 = X_1 R_1 \in \mathbb{G}_{p_1 p_2}$ with more than a negligible advantage. The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{SD}(\lambda) = \big|\Pr[\mathcal{A}(D, T_0) = 0] - \Pr[\mathcal{A}(D, T_1) = 0]\big|$ where the probability is taken over random choices of $X_1 \in \mathbb{G}_{p_1}$ and $R_1 \in \mathbb{G}_{p_2}$.*

*Assumption 10 (General Subgroup Decision, GSD): Let $(N, \mathbb{G}, \mathbb{G}_T, e)$ be a description of the bilinear group of composite order $N = p_1 p_2 p_3$. Let $g_1, g_2, g_3$ be generators of subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. The GSD assumption is that if the challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_3, X_1 R_1, R_2 Y_1)$ and $T$ are given, no PPT algorithm $\mathcal{A}$ can distinguish $T = T_0 = X_2 Y_2 \in \mathbb{G}_{p_1 p_3}$ from $T = T_1 = X_2 R_3 Y_2 \in \mathbb{G}_{p_1 p_2 p_3}$ with more than a negligible advantage. The advantage of $\mathcal{B}$ is defined as $Adv_{\mathcal{A}}^{GSD}(\lambda) = \big|\Pr[\mathcal{A}(D, T_0) = 0] - \Pr[\mathcal{A}(D, T_1) = 0]\big|$ where the probability is taken over random choices of $X_1, X_2 \in \mathbb{G}_{p_1}$, $R_1, R_2, R_3 \in \mathbb{G}_{p_2}$, and $Y_1, Y_2 \in \mathbb{G}_{p_3}$.*

*Assumption 11 (Composite Diffie-Hellman, ComDH): Let $(N, \mathbb{G}, \mathbb{G}_T, e)$ be a description of the bilinear group of composite order $N = p_1 p_2 p_3$. Let $g_1, g_2, g_3$ be generators of subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ respectively. The ComDH assumption is that if the challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_2, g_3, g_1^a R_1, g_1^b R_2)$ and $T$ are given, no PPT algorithm $\mathcal{A}$ can distinguish $T = T_0 = e(g_1, g_1)^{ab}$ from $T = T_1 = e(g_1, g_1)^c$ with more than a negligible advantage. The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{ComDH}(\lambda) = \big|\Pr[\mathcal{A}(D, T_0) = 0] - \Pr[\mathcal{A}(D, T_1) = 0]\big|$ where the probability is taken over random choices of $a, b, c \in \mathbb{Z}_N$, and $R_1, R_2 \in \mathbb{G}_{p_2}$.*

### C. CONSTRUCTION

To construct a fully secure SRE scheme, we build our SRE scheme in composite-order bilinear groups instead of prime-order bilinear groups. To prove the security of SRE, we use the dual system encryption technique of Lewko and Waters [12]. Our fully secure SRE scheme has the similar structure with that of the selectively secure SRE scheme in prime order groups. Let $\mathcal{H}$ be a family of collision resistant hash function $H$. Our SRE scheme in composite-order bilinear groups is described as follows:

**SRE.Setup($1^\lambda$):** It first generates a bilinear group $\mathbb{G}$ of composite order $N = p_1 p_2 p_3$ where $p_1, p_2$, and $p_3$ are random primes. Let $g_1$ be a generator of $\mathbb{G}_{p_1}$. It chooses random $\alpha \in \mathbb{Z}_N$ and random $u, h, w, v \in \mathbb{G}_{p_1}$, $Y \in \mathbb{G}_{p_3}$. It also chooses a random hash function $H$ from $\mathcal{H}$. It outputs $MK = \alpha$ and $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y, u, h, w, v, H, \Omega = e(g_1, g_1)^\alpha\big)$.

**SRE.GenKey($(GL, ML), MK, PP$):** It selects random $r_1, r_2 \in \mathbb{Z}_N$, $Y_0, Y_1, Y_2, Y_3 \in \mathbb{G}_{p_3}$ and outputs a private key $SK_{(GL,ML)} = \big(K_0 = g^\alpha (u^{GL} h)^{r_1} w^{r_2} Y_0, K_1 = (w^{ML} v)^{r_2} Y_1, K_2 = g^{-r_1} Y_2, K_3 = g^{-r_2} Y_3\big)$.

**SRE.Encrypt($(GL, ML), M, PP$):** It chooses random $t \in \mathbb{Z}_N$ and outputs a ciphertext $CT_{(GL,ML)} = \big(C = H(\Omega^t) \oplus M, C_0 = g^t, C_1 = (u^{GL} h)^t, C_2 = (w^{ML} v)^t\big)$.

**SRE.Decrypt($CT_{(GL,ML)}, SK_{(GL',ML')}, PP$):** If $(GL = GL') \wedge (ML \neq ML')$, then it computes $A = e(C_0, K_0) \cdot e(C_1, K_2) \cdot \big(e(C_0, K_1) \cdot e(C_2, K_3)\big)^{-1/(ML'-ML)}$ and outputs a message $M = C \oplus H(A)$. Otherwise, it outputs $\perp$.

### D. SECURITY ANALYSIS

For the security proof of our SRE scheme, we use the dual system encryption technique [11], [12] that was successfully used to prove the full security of IBE, HIBE, and ABE schemes. In dual system encryption, a private key and a ciphertext can be normal type or semi-functional type. In the security proof, we use hybrid games such that a challenge ciphertext is changed from the normal type to the semi-functional type and then each private key is changed from the normal type to the semi-functional type one by one. In the final game, it is hard for an adversary to obtains the encrypted message since semi-functional private keys given to the adversary are not related with the semi-functional challenge ciphertext. The technical difficulty of the dual system encryption technique is to define nominal semi-functional private key in order to solve the paradox in the proof and to show the information theoretic argument between the nominal semi-functional private key and the semi-functional private key.

We may directly try to use the dual system encryption technique for the security proof of our SRE scheme. However, we encounter a problem to show the information theoretic argument between a nominal private key and a semi-functional private key. The main reason of the problem is that an adversary can query a private key for labels $(GL, ML)$ such that $(GL = GL^*)$ and $(ML = ML^*)$ where $(GL^*, ML^*)$ is the challenge ciphertext labels. In IBE (or HIBE), it is relatively easy to show the information theoretic argument by using a pair-wise independent hash function and the restriction of an adversary such that $ID \neq ID^*$ is only allowed in private key queries. To solve this problem, we carefully analyze our SRE scheme and show that the information theoretic argument still holds even though the adversary queries $(GL, ML)$ such that $(GL = GL^*)$ and $(ML = ML^*)$. The security proof of our SRE scheme is described as follows:

*Theorem 12: The above SRE scheme is fully secure under chosen plaintext attacks if the SD, GSD and ComDH assumptions hold.*

*Proof:* We first define the semi-functional type of private keys and ciphertexts. For the semi-functional type, we let $g_2$ denote a fixed generator of the subgroup $\mathbb{G}_{p_2}$.

**SRE.GenKeySF**. It first creates a normal private key $SK'_{GL,ML} = (K'_0, K'_1, K'_2, K'_3)$ by using $MK$. It chooses a random $R \in \mathbb{G}_{p_2}$ and outputs a semi-functional $SK_{GL,ML} = \big(K_0 = K'_0\, R, K_1 = K'_1, K_2 = K'_2, K_3 = K'_3\big)$.

**SRE.EncryptSF**. It first creates a normal ciphertext $CT'_{GL,ML} = (C', C'_0, C'_1, C'_2)$. It chooses random $\tau, \eta_1, \eta_2, \theta_1, \theta_2 \in \mathbb{Z}_N$ and outputs semi-functional $CT_{GL,ML} = \big(C_0 = C'_0\, g_2^{\tau}, C_1 = C'_1\, g_2^{(\eta_1\, GL+\eta_2)\tau}, C_2 = C'_2\, g_2^{(\theta_1\, ML+\theta_2)\tau}\big)$. Note that $\eta_1, \eta_2, \theta_1, \theta_2$ are randomly chosen once and fixed to be used in other types of private keys that will be defined later.

Note that if a semi-functional private key is used to decrypt a semi-functional ciphertext, then the decryption fails since an additional random element $e(g_2^{\tau}, R)$ is left.

The security proof consists of the sequence of hybrid games: The first game is the original security game and the last one is a game such that the adversary has no advantage. We define the games as follows:

**Game** $G_0$. This game is the original security game. In this game, all private keys and the challenge ciphertext are normal.

**Game** $G_1$. In the game $G_1$, all private keys are still normal, but the challenge ciphertext is semi-functional.

**Game** $G_2$. Next, we define a new game $G_2$. In this game, all private keys are semi-functional. For the security proof, we additionally define a sequence of sub-games $G_{1,1}, \ldots, G_{1,k}, \ldots, G_{1,q}$ where $G_1 = G_{1,0}$ and $q$ is the maximum number of private keys. In the game $G_{1,k}$, the challenge ciphertext is semi-functional, all $j$th private keys such that $j \leq k$ are semi-functional, and the remaining $j$th private keys such that $k < j$ are normal. It is obvious that $G_{1,q} = G_2$.

**Game** $G_3$. In the final game $G_3$, all private keys and the challenge ciphertext are semi-functional, but the challenge ciphertext component $C$ is random.

Let $\mathrm{Adv}_{\mathcal{A}}^{G_j}$ be the advantage of $\mathcal{A}$ in the game $G_j$. We have $\mathrm{Adv}_{\mathcal{A}}^{SRE}(\lambda) = \mathrm{Adv}_{\mathcal{A}}^{G_0}$, $\mathrm{Adv}_{\mathcal{A}}^{G_1} = \mathrm{Adv}_{\mathcal{A}}^{G_{1,0}}$, $\mathrm{Adv}_{\mathcal{A}}^{G_2} = \mathrm{Adv}_{\mathcal{A}}^{G_{1,q}}$, and $\mathrm{Adv}_{\mathcal{A}}^{G_3} = 0$. Through the following Lemmas 13, 14, and 18, we obtain the following equation

$$\mathrm{Adv}_{\mathcal{A}}^{SRE}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}}^{SD}(\lambda) + 2q\mathrm{Adv}_{\mathcal{B}}^{GSD}(\lambda) + \mathrm{Adv}_{\mathcal{B}}^{ComDH}(\lambda).$$

where $q$ is the maximum number of private key queries. This completes our proof. □

*Lemma 13: If the SD assumption holds, then no polynomial-time adversary can distinguish between $G_0$ and $G_1$ with a non-negligible advantage.*

*Proof:* Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $G_0$ and $G_1$ with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the SD assumption using $\mathcal{A}$ is given: a challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_3)$ and $T$ where $T = X_1 \in \mathbb{G}_{p_1}$ or $T = X_1\, R_1 \in \mathbb{G}_{p_1 p_2}$. Then $\mathcal{B}$ that interacts with $\mathcal{A}$ is described as follows:

**Setup:** $\mathcal{B}$ first chooses random exponents $u', h', w', v', z', \alpha \in \mathbb{Z}_N$. It sets $MK = \alpha$ and publishes $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}H, \Omega = e(g_1, g_1)^{\alpha}\big)$.

**Query 1:** To response private key queries, $\mathcal{B}$ creates normal private keys since it knows $MK$. Note that it cannot create semi-functional private keys since it does not know $g_{p_2}$.

**Challenge:** $\mathcal{A}$ submits challenge labels $(GL^*, ML^*)$ and challenge messages $M_0^*, M_1^*$. $\mathcal{B}$ flips a random coin $\mu \in \{0, 1\}$ and creates a challenge ciphertext $CT^*$ by implicitly setting $g^t$ to be the $\mathbb{G}_{p_1}$ part of $T$ as

$$\big(C = H(e(T, g)^{\alpha}) \cdot M_{\mu}^*, \quad C_0 = T,\ C_1 = (T)^{u'GL^*+h'}, \\ C_2 = (T)^{w'ML^*+v'}\big).$$

If $T = X_1$, this is a normal ciphertext. If $T = X_1\, R_1$, this is a semi-functional ciphertext since $\tau \equiv \log_{g_2}(R_1) \bmod p_2, \eta_1 \equiv u' \bmod p_2, \eta_2 \equiv h' \bmod p_2, \theta_1 \equiv w' \bmod p_2, \theta_2 \equiv v' \bmod p_2$ are not correlated with their values modulo $p_1$ by CRT.

**Query 2:** Same as Query 1.

**Guess:** $\mathcal{A}$ outputs a guess $\mu'$. If $\mu = \mu'$, then $\mathcal{B}$ outputs 1. Otherwise, it outputs 0. □

*Lemma 14: If the GSD assumption holds, then no polynomial-time adversary can distinguish between $G_{1,k-1}$ and $G_{1,k}$ with a non-negligible advantage.*

*Proof:* We additionally define two additional semi-functional private keys. Let $g_2$ denote a fixed generator of the subgroup $\mathbb{G}_{p_2}$.

**SRE.GenKeyNSF**. It first creates a normal private key $SK'_{GL,ML} = (K'_0, K'_1, K'_2, K'_3)$ by using $MK$. Next, it chooses random $\gamma_1, \gamma_2 \in \mathbb{Z}_N$ and outputs a nominal semi-functional $SK_{GL,ML} = \big(K_0 = K'_0\, g_2^{(\eta_1\, GL+\eta_2)\gamma_1+\theta_1\gamma_2}, K_1 = K'_1\, g_2^{(\theta_1\, ML+\theta_2)\gamma_2}, K_2 = K'_2\, g_2^{-\gamma_1}, K_3 = K'_3\, g_2^{-\gamma_2}\big)$.

**SRE.GenKeyTSF**. It first creates a normal private key $SK'_{GL,ML} = (K'_0, K'_1, K'_2, K'_3)$ by using $MK$. Next, it chooses a random $R \in \mathbb{G}_{p_2}$ and outputs a temporary semi-functional $SK_{GL,ML} = \big(K_0 = K'_0\, R, K_1 = K'_1\, g_2^{(\theta_1\, ML+\theta_2)\gamma_2}, K_2 = K'_2\, g_2^{-\gamma_1}, K_3 = K'_3 g_2^{-\gamma_2}\big)$.

We also define hybrid games $H_{k-1,0}, H_{k-1,1}, H_{k-1,2}$, and $H_{k-1,3}$. The games are formally defined as follows: The game $H_{k-1,0}$ is equal to the game $G_{1,k-1}$. That is, the $k$th private key is normal. The game $H_{k-1,1}$ is almost the same as $G_{1,k-1}$ except that $k$th private key is nominal semi-functional. The game $H_{k-1,2}$ is almost the same as $G_{1,k-1}$ except that $k$th private key is temporary semi-functional. The game $H_{k-1,3}$ is equal to the game $G_{1,k}$. That is, the $k$th private key is semi-functional.

Let $\mathrm{Adv}_{\mathcal{A}}^{H_j}$ be the advantage of $\mathcal{A}$ in the game $H_j$. Through the following Lemmas 15, 16 and 17, we obtain the following equation

$$\big|\mathrm{Adv}_{\mathcal{A}}^{G_{1,k-1}} - \mathrm{Adv}_{\mathcal{A}}^{G_{1,k}}\big| = \big|\mathrm{Adv}_{\mathcal{A}}^{H_{k-1,0}} - \mathrm{Adv}_{\mathcal{A}}^{H_{k-1,3}}\big| \\ \leq \mathrm{Adv}_{\mathcal{B}}^{GSD}(\lambda) + \mathrm{Adv}_{\mathcal{B}}^{GSD}(\lambda).$$

This completes our proof. □

*Lemma 15: If the GSD assumption holds, then no polynomial-time adversary can distinguish between $H_{k-1,0}$ and $H_{k-1,1}$ with a non-negligible advantage.*

*Proof:* Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $H_{k-1,0}$ and $H_{k-1,1}$ with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the GSD assumption using $\mathcal{A}$ is given: a challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_3, X_1 R_1, R_2 Y_1)$ and $T$ where $T = X_2 Y_2$ or $T = X_2 R_3 Y_2$. Then $\mathcal{B}$ that interacts with $\mathcal{A}$ is described as follows:

**Setup:** $\mathcal{B}$ first chooses random exponents $u', h', w', v', \alpha \in \mathbb{Z}_N$. It sets $MK = \alpha$ and publishes $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}, \Omega = e(g_1, g_1)^\alpha\big)$.

**Query 1**: To response the $j$th private key query, $\mathcal{B}$ proceeds as follows: If $j < k$, then it creates a semi-functional private key since it knows $MK$ and $R_2 Y_1$ is given in the assumption. If $j > k$, then it creates a normal private key since it knows $MK$. If $j = k$, then it selects random $r_1', r_2', r_3' \in \mathbb{Z}_N$, $Y_0', Y_1', Y_2', Y_3', Y_4' \in \mathbb{G}_{p_3}$ and creates a private key $SK_{(GL,ML)}$ as

$$K_0 = g^\alpha (T)^{(u'GL+h')r_1' + w'r_2' + z'r_3'} Y_0', \quad K_1 = (T)^{(w'ML+v')r_1'} Y_1', \quad K_2 = (T)^{-r_1'} Y_2', \quad K_3 = (T)^{-r_2'} Y_3'.$$

If $T = X_2 Y_2$, this is a normal private key. If $T = X_2 R_3 Y_2$, this is a nominally semi-functional private key since $\gamma_1 \equiv \log_{g_2}(R_3) r_1' \mod p_2, \gamma_2 \equiv \log_{g_2}(R_3) r_2' \mod p_2, \gamma_3 \equiv \log_{g_2}(R_3) r_3' \mod p_2, \eta_1 \equiv u' \mod p_2, \eta_2 \equiv h' \mod p_2, \theta_1 \equiv w' \mod p_2$, and $\theta_2 \equiv v' \mod p_2$.

**Challenge**: $\mathcal{B}$ flips a random coin $\mu \in \{0, 1\}$ and creates a semi-functional ciphertext by implicitly setting $g^t = X_1$ and $g_2^\tau = R_1$ as $CT^* = \big(C = H(e(X_1 R_1, g)^\alpha) \cdot M_\mu^*, C_0 = X_1 R_1, C_1 = (X_1 R_1)^{u'GL^*+h'}, C_2 = (X_1 R_1)^{w'ML^*+v'}\big)$.

**Query 2**: Same as Query 1.

**Guess**: $\mathcal{A}$ outputs a guess $\mu'$. If $\mu = \mu'$, then $\mathcal{B}$ outputs 1. Otherwise, it outputs 0. □

*Lemma 16: No polynomial-time adversary can distinguish between $H_{k-1,1}$ and $H_{k-1,2}$ with a non-negligible advantage.*

*Proof:* To argue that any adversary cannot distinguish the nominally semi-functional private key from the semi-functional private key, we show that even an unbounded adversary cannot distinguish the type of private keys.

Suppose there exists an unbounded adversary. Let $\beta$ be a random exponent in $\mathbb{Z}_{p_2}^*$ and $x \in \{0, 1\}$. This adversary can gather $x\beta + (\eta_1\, GL + \eta_2)\gamma_1 + \theta_1\gamma_2 \mod p_2, (\theta_1\, ML + \theta_2)\gamma_2 \mod p_2, -\gamma_1 \mod p_2, -\gamma_2 \mod p_2$ from the $k$th private key and $\tau \mod p_2, (\eta_1\, GL^* + \eta_2)\tau \mod p_2, (\theta_1\, ML^* + \theta_2)\tau \mod p_2$ from the challenge ciphertext. Note that the $k$-the private key is nominally semi-functional if $x = 0$, otherwise it is semi-functional by implicitly setting $\log_{g_2}(R) = \beta + (\eta_1\, GL + \eta_2)\gamma_1 + \theta_1\gamma_2 \mod p_2$. These values can be restated as a linear equation $M\vec{u} = \vec{v}$ such that

$$\begin{pmatrix} x & GL\gamma_1 & \gamma_1 & \gamma_2 & 0 \\ 0 & 0 & 0 & ML\gamma_2 & \gamma_2 \\ 0 & GL^*\tau & \tau & 0 & 0 \\ 0 & 0 & 0 & ML^*\tau & \tau \end{pmatrix} \begin{pmatrix} \beta \\ \eta_1 \\ \eta_2 \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} x\beta + (\eta_1 GL + \eta_2)\gamma_1 + \theta_1\gamma_2 \\ (\theta_1 ML + \theta_2)\gamma_2 \\ (\eta_1 GL^* + \eta_2)\tau \\ (\theta_1 ML^* + \theta_2)\tau \end{pmatrix}.$$

To distinguish whether the $k$th private key is nominally semi-functional or semi-functional, the adversary should decide whether $x = 0$ or $x = 1$ by solving the above linear equation. Let $RowSpace(M)$ be a vector space that has the row vectors of $M$ as basis vectors. If $\vec{1}^\top = (1, 0, 0, 0, 0) \notin RowSpace(M)$, then the adversary cannot distinguish the type of the $k$th private key.

For further analysis, we divide the behavior of an adversary as two types: Type-A and Type-B. Let $(GL^*, ML^*)$ be the challenge labels. An adversary is Type-A if it queries a private key for labels $(GL, ML)$ such that $GL \neq GL^*$. An adversary is Type-B if it queries a private key for labels $(GL, ML)$ such that $GL = GL^*$ and $ML = ML^*$. First, we show that a Type-A (unbounded) adversary cannot distinguish the type of private keys. From the above $M\vec{u} = \vec{v}$, we have another linear equation $M_{2,3}\vec{\eta} = \vec{v}'$ by just taking second and third columns of $M$ as follows

$$\begin{pmatrix} GL\gamma_1 & \gamma_1 \\ GL^*\tau & \tau \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} = \begin{pmatrix} (\eta_1 GL + \eta_2)\gamma_1 \\ (\eta_1 GL^* + \eta_2)\tau \end{pmatrix}.$$

To show that $\vec{1}^\top \notin RowSpace(M)$, it is enough to show that $\vec{0}^\top = (0, 0) \notin RowSpace(M_{2,3})$. For contradiction, if we suppose that $\vec{0} \in RowSpace(M_{2,3})$, then there exists $\vec{z}$ such that $\vec{z}M_{2,3} = \vec{0}^\top$ and $\vec{z} \neq \vec{0}$. However we have $\vec{z} = \vec{0}^\top M_{2,3}^{-1} = \vec{0}$ since $GL \neq GL^*$ by the restriction of the Type-A adversary. Thus, we have $\vec{0}^\top \notin RowSpace(M_{2,3})$.

Next, we show that a Type-B (unbounded) adversary cannot distinguish the type of private keys. From the above $M\vec{u} = \vec{v}$, we have another linear equation $M_{4,5}\vec{\theta} = \vec{v}''$ by just taking forth and fifth columns of $M$ as follows

$$\begin{pmatrix} \gamma_2 & 0 \\ ML\gamma_2 & \gamma_2 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} \theta_1\gamma_1 \\ (\theta_1 ML^* + \theta_2)\gamma_2 \end{pmatrix}.$$

Note that the forth row vector of $M$ can be removed since the forth row vector is a linear span of the second row vector of $M$ if $ML = ML^*$. Similar to the analysis of the Type-A adversary, we can easily show that $\vec{0}^\top \notin RowSpace(M_{4,5})$ since $M_{4,5}$ is invertable if $\gamma_2 \neq 0 \mod p_2$. This completes our proof. □

*Lemma 17: If the GSD assumption holds, then no polynomial-time adversary can distinguish between $H_{k-1,2}$ and $H_{k-1,3}$ with a non-negligible advantage.*

*Proof:* The proof of this lemma is almost the same as that of Lemma 15 except the generation of the $k$th private key. The $k$th private key for $(GL, ML)$ is generated as follows:

If $j = k$, then it selects random $r'_1, r'_2, a' \in \mathbb{Z}_N$, $Y'_0, Y'_1, Y'_2, Y'_3 \in \mathbb{G}_{p_3}$ and creates a private key $SK_{(GL,ML)}$ as

$$K_0 = g^{\alpha}(T)^{(u'GL+h')r'_1+w'r'_2}(R_2\, Y_1)^{a'} Y'_0,$$
$$K_1 = (T)^{(w'ML+v')r'_1} Y'_1,$$
$$K_2 = (T)^{-r'_1} Y'_2,\; K_3 = (T)^{-r'_2} Y'_3.$$

Note that the $k$th private key is no longer correlated with $CT^*$ since $K_0$ is re-randomized by $(R_2\, Y_1)^{a'}$. If $T = X_2\, Y_2$, this is a semi-functional private key. If $T = X_2\, R_3 Y_2$, this is a temporary semi-functional private key. □

*Lemma 18: If the ComDH assumption holds, then no polynomial-time adversary can distinguish between $G_2$ and $G_3$ with a non-negligible advantage.*

*Proof:* Suppose there exists an adversary $\mathcal{A}$ that distinguish $G_2$ from $G_3$ with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the ComDH assumption using $\mathcal{A}$ is given: a challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_2, g_3, g_1^a R_1, g_1^b R_2)$ and $T$ where $T = e(g_1, g_1)^{ab}$ or $T = e(g_1, g_1)^c$. Then $\mathcal{B}$ that interacts with $\mathcal{A}$ is described as follows:

**Setup**: $\mathcal{B}$ chooses random exponents $u', h', w', v', z' \in \mathbb{Z}_N$ and implicitly sets $\alpha = a$ from the term $g_1^a R_1$. It publishes the public parameters $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}, H, \Omega = e(g_1, g_1^a R_1)\big)$.

**Query 1**: To response private key queries, $\mathcal{B}$ creates semi-functional private keys since $g_2^a R_1$ and $g_2$ are given. Note that it cannot create normal private keys since it does not know $g_1^a \in \mathbb{G}_{p_1}$.

**Challenge**: $\mathcal{B}$ first flips a random coin $\mu \in \{0, 1\}$ and creates a challenge ciphertext $CT^* = \big(C = H(T) \cdot M^*_{\mu}, C_0 = g_1^b R_2,\; C_1 = (g_1^b R_2)^{u'GL^*+h'},\; C_2 = (g_1^b R_2)^{w'ML^*+v'}\big)$.

**Query 2**: Same as Phase 1.

**Guess**: $\mathcal{A}$ outputs a guess $\mu'$. If $\mu = \mu'$, then $\mathcal{B}$ outputs 1. Otherwise, it outputs 0. □

## V. IDENTITY-BASED REVOCATION FROM SD

In this section, we propose an identity-based revocation (IBR) scheme by combining the subset difference (SD) method with a single revocation encryption (SRE) scheme and prove its security. For the construction of an IBR scheme, we follow the design principle of Lee et al. [10]. Compared to the scheme of Lee et al., our scheme is an identity-based one whereas their scheme is a public-key based one.

### A. SUBSET DIFFERENCE SCHEME

The subset difference (SD) scheme is one instance of the subset cover framework proposed by Naor et al. [7]. The subset cover framework is a general method to construct a revocation system for a set of users $\mathcal{N}$. In this framework, a collection $\mathcal{S}$ of subsets is defined for the system and each user is assigned to a private set $PV$ that is a subset of $\mathcal{S}$ where each subset is associated with a unique key. When a center broadcasts an encrypted message for all users except a revoked set $R$ of users, it first finds cover $CV$ that is a set of subsets that can cover all users $\mathcal{N} \setminus R$ and creates ciphertexts for each subset by using their unique key. A receiver can decrypt the ciphertext if he is not revoked in $R$.

Before we describe the SD scheme, we define some notation. Let $\mathcal{BT}$ be a perfect binary tree and $v_i$ be a node in $\mathcal{BT}$. The depth $d_i$ of a node $v_i$ is the length of a path from the root node to the node $v_i$ where the root node is at depth zero. A level of $\mathcal{BT}$ is a set of all nodes at given depth. For any node $v_i \in \mathcal{BT}$, $\mathcal{T}_i$ is defined as a subtree that is rooted at $v_i$. For any two nodes $v_i, v_j \in \mathcal{BT}$ such that $v_j$ is a descendant of $v_i$, $\mathcal{T}_{i,j}$ is defined as a subtree $\mathcal{T}_i - \mathcal{T}_j$, that is, all nodes that are descendants of $v_i$ but not $v_j$. For any node $v_i \in \mathcal{BT}$, $S_i$ is defined as the set of leaf nodes in $\mathcal{T}_i$. Similarly, $S_{i,j}$ is defined as the set of leaf nodes in $\mathcal{T}_{i,j}$, that is, $S_{i,j} = S_i \setminus S_j$. For any node $v_i \in \mathcal{BT}$, we let $L_i$ be a fixed and unique label of $v_i$. The label $L_i$ of a node $v_i$ is assigned as follows: Each edge in the tree is assigned with 0 or 1 depending on whether it is connected to its left or right child node. The label $L_i$ of $v_i$ is the bitstring obtained by reading all the bits of edges in the path from the root node to the node $v_i$. For a subtree $\mathcal{T}_i$, we define the label of $\mathcal{T}_i$ as the label $L_i$ of $v_i$ where $v_i$ is the root node of $\mathcal{T}_i$. For a subtree $\mathcal{T}_{i,j}$, we also define the label of $\mathcal{T}_{i,j}$ as $(L_i, L_j)$ where $L_i, L_j$ are labels of $v_i, v_j$ of $\mathcal{T}_{i,j}$. Similarly, we can define the label of $S_i$ as the same as that of $\mathcal{T}_i$ and the label of $S_{i,j}$ as the same as that of $\mathcal{T}_{i,j}$.

As mentioned before, the SD scheme is one instance of the subset cover framework. To describe the SD scheme, we use the abstraction of Lee *et al.* [10] since their abstraction of SD is independent of a key assignment method. They defined the SD scheme as four algorithms: **Setup**, **Assign**, **Cover**, and **Match**. The setup algorithm first defines a binary tree $\mathcal{BT}$ and the collection $\mathcal{S}$ of subsets where each subset $S_{i,j}$ is associated with a subtree $\mathcal{T}_{i,j}$. The assign algorithm first assigns a user to a leaf node of $\mathcal{BT}$ and defines a private set $PV$ for the user where any two nodes $v_i, v_j$ in the path nodes from the root node to the leaf node defines a subtree $\mathcal{T}_{i,j}$ in $PV$. The cover algorithm takes as input a revoked set $R$ of users and finds a cover set $CV$ that can cover the set of receivers $\mathcal{N} \setminus R$. The final match algorithm takes as input a private set $PV$ and a cover set $CV$ and finds two matching subsets in $PV$ and $CV$ respectively. The detailed description of the SD scheme is given as follows:

**SD.Setup($N$):** Let $N = 2^n$ for simplicity. It first sets a perfect binary tree $\mathcal{BT}$ of depth $n$. Each user is assigned to a different leaf node in $\mathcal{BT}$. The collection $\mathcal{S}$ of SD is the set of all subsets $\{S_{i,j}\}$ where $v_i, v_j \in \mathcal{BT}$ and $v_j$ is a descendant of $v_i$. It outputs the tree $\mathcal{BT}$.

**SD.Assign($\mathcal{BT}$, $ID$):** Let $v_{ID}$ be the leaf node of $\mathcal{BT}$ that is assigned to the user $ID$. Let $(v_{k_0}, v_{k_1}, \ldots, v_{k_n})$ be the path from the root node $v_{k_0}$ to the leaf node $v_{k_n} = v_{ID}$. It first sets a private set $PV_{ID}$ as an empty one. For all $i, j \in \{k_0, \ldots, k_n\}$ such that $v_j$ is a descendant of $v_i$, it adds the subset $S_{i,j}$ defined by two nodes $v_i$ and $v_j$ in the path into $PV_{ID}$. It outputs the private set $PV_{ID} = \{S_{i,j}\}$.

**SD.Cover**($\mathcal{BT}, R$)**:** It first sets a subtree $T$ as $ST(R)$, and then it builds a covering set $CV_R$ iteratively by removing nodes from $T$ until $T$ consists of just a single node as follows:

1) It finds two leaf nodes $v_i$ and $v_j$ in $T$ such that the least-common-ancestor $v$ of $v_i$ and $v_j$ does not contain any other leaf nodes of $T$ in its subtree. Let $v_l$ and $v_k$ be the two child nodes of $v$ such that $v_i$ is a descendant of $v_l$ and $v_j$ is a descendant of $v_k$. If there is only one leaf node left, it makes $v_i = v_j$ to the leaf node, $v$ to be the root of $T$ and $v_l = v_k = v$.
2) If $v_l \neq v_i$, then it adds the subset $S_{l,i}$ to $CV_R$; likewise, if $v_k \neq v_j$, it adds the subset $S_{k,j}$ to $CV_R$.
3) It removes from $T$ all the descendants of $v$ and makes $v$ a leaf node.

It outputs the covering set $CV_R = \{S_{i,j}\}$.

**SD.Match**($CV_R, PV_{ID}$)**:** Let $CV_R = \{S_{i,j}\}$ and $PV_{ID} = \{S_{i,j}\}$. It finds two subsets $S_{i,j} \in CV_R$ and $S_{i',j'} \in PV_{ID}$ such that $(v_i = v_{i'}) \wedge (d_j = d_{j'}) \wedge (v_j \neq v_{j'})$ where $d_j$ is the depth of $v_j$. If it found two subsets, then it outputs $(S_{i,j}, S_{i',j'})$. Otherwise, it outputs $\perp$.

### B. CONSTRUCTION

The general method that combines the SD scheme with an SRE scheme for the construction of a revocation system was introduced by Lee *et al.* [10]. The basic idea of their method is that there exists a one-to-one mapping between a subset $S_{i,j}$ in the SD scheme and labels $(GL, ML)$ in the SRE scheme. That is, we can set $GL = L_i \| d_j$ and $ML = L_j$ where $(L_i, L_j)$ is the labels of a subtree $\mathcal{T}_{i,j}$ and $d_j$ is the depth of the node $v_j$ since the subtree $\mathcal{T}_{i,j}$ is associated with the subset $S_{i,j}$. Thus, we can derive a public-key revocation system by using an SRE scheme instead of using a pseudo-random generator. We follow the design method of Lee *et al.* [10], but we slightly modify it to use a symmetric key encryption scheme in order to improve the efficiency.

Let **SD** = (**Setup**, **Assign**, **Cover**, **Match**) be the SD scheme and **SKE** = (**Gen**, **Enc**, **Dec**) be a symmetric key encryption scheme. Our IBR scheme for the identity space $\mathcal{I} = \{0, 1\}^n$ is described as follows:

**IBR.Setup**($1^\lambda$)**:** It define a perfect binary tree $\mathcal{BT}$ by running **SD.Setup**($2^n$). Next, it obtains $MK_{SRE}$ and $PP_{SRE}$ by running **SRE.Setup**($1^\lambda$). It outputs $MK = MK_{SRE}$ and $PP = (\mathcal{BT}, PP_{SRE})$.

**IBR.GenKey**($ID, MK, PP$)**:** It obtains a private set $PV_{ID} = \{S_{i,j}\}$ by running **SD.Assign**($\mathcal{BT}, ID$). Note that we assign $ID$ to the leaf node where the label of the leaf node is equal to $ID$. Let $d_j$ be the depth of a node $v_j$ associated with a label $L_j$. For each $S_{i,j} \in PV_{ID}$, it derives $(L_i, L_j)$ from $S_{i,j}$ and obtains $SK_{SRE,S_{i,j}}$ by running **SRE.GenKey**($(L_i\|d_j, L_j), MK_{SRE}, PP_{SRE}$). It outputs $SK_{ID} = \big(PV_{ID}, \{SK_{SRE,S_{i,j}}\}_{S_{i,j} \in PV_{ID}}\big)$.

**IBR.Encrypt**($R, M, PP$)**:** It finds a covering set $CV_R = \{S_{i,j}\}$ by running **SD.Cover**($\mathcal{BT}, R$). Let $d_j$ be the depth of a node $v_j$ associated with $L_j$. Next, it chooses a session key $K \in \{0, 1\}^\lambda$. For each $S_{i,j} \in CV_R$, it derives two labels $(L_i, L_j)$ from $S_{i,j}$ and obtains $CT_{SRE,S_{i,j}}$ by running **SRE.Encrypt**($(L_i\|d_j, L_j), K, PP_{SRE}$). It obtains $C$ by running **SKE.Encrypt**($K, M$). Finally, it outputs $CT_R = \big(CV_R, C, \{CT_{SRE,S_{i,j}}\}_{S_{i,j} \in CV_R}\big)$.

**IBR.Decrypt**($CT_R, SK_{ID}, PP$)**:** If $ID \notin R$, it finds a matching tuple $(S_{i,j}, S_{i',j'})$ by running **SD.Match**($CV_R, PV_{ID}$) and obtains $K$ by running **SRE.Decrypt**($CT_{SRE,S_{i,j}}, SK_{SRE,S_{i',j'}}, PP_{SRE}$). Otherwise, it outputs $\perp$. Finally, it outputs $M$ by running **SKE.Decrypt**($K, C$).

*Remark 19:* Our revocation scheme is identity-based one whereas the revocation scheme of Lee *et al.* [10] is public-key one since the SRE scheme of Lee *et al.* is proven under a $q$-type assumption where $q$ is depends on the number of users.

*Remark 20:* In our IBR scheme, an SRE scheme is integrated with the SD scheme. It is relatively straightforward to integrate an SRE scheme with the LSD scheme instead of the SD scheme. The detailed description of the LSD scheme is given in [13].

### C. SECURITY ANALYSIS

The security model of our IBR scheme in the proof depends on the security model of the underlying SRE scheme. That is, if the underlying SRE scheme is fully (or selectively) secure, then our IBR scheme is also fully (or selectively) secure.

*Theorem 21: The above IBR scheme is selectively (or fully) secure under chosen plaintext attacks if the SRE scheme is selectively (or fully) secure under chosen plaintext attacks and the SKE scheme is secure under chosen plaintext attacks.*

*Proof:* Let $R^*$ be the set of revoked users in the challenge ciphertext and $CV_{R^*}$ be the covering set where the number of subsets in $CV_{R^*}$ is $\ell$. The challenge ciphertext is described as $CT^* = (CV_{R^*}, C^*, \{CT^*_{SRE,S_{i_k,j_k}}\}_{k=1}^{\ell})$. For the security proof, we define hybrid games $G_0, G_1, G_2, G_3$ as follows:

**Game** $G_0$. This game is the original security game defined in the security model except that the challenge bit $\mu$ is fixed to 0. In this game, all components $CT^*_{SRE,j}$ are encryption on a correct session key $K^*$ and the component $C^*$ is an encryption on the message $M_0^*$ by using the session key $K^*$.

**Game** $G_1$. In this game, all components $CT^*_{SRE,j}$ in the challenge ciphertext $CT^*$ are encryption on a random session key $Z$ that is not related to the correct one $K^*$. However, the component $C^*$ is still an encryption on the message $M_0^*$ by using the correct one $K^*$. For the security proof, we additionally define hybrid games $G_{0,0}, \ldots, G_{0,\rho}, \ldots, G_{0,\ell}$ where $G_{0,0} = G_0$ and $G_{0,\ell} = G_1$. The game $G_{0,\rho}$ is almost identical to the game $G_{0,\rho-1}$ except that $CT^*_{SRE,\rho}$ is an encryption on a random session key $Z$. Specifically, each component $CT^*_{SRE,k}$ for $k \leq \rho$ is an encryption on a random session key $Z$ and each component $CT^*_{SRE,k}$ for $\rho < k$ is an encryption on the session key $K^*$.

**Game** $G_2$. This game is similar to the game $G_1$ except that the component $C^*$ is an encryption on the message $M_1^*$ by using the session key $K^*$. That is, the challenge ciphertext $CT^*$ is an encryption on the message $M_1^*$.

**Game** $G_3$. In this game, all components $CT^*_{SRE,j}$ in the challenge ciphertext $CT^*$ are encryption on the correct session key $K^*$ instead of the random session key $Z$. Thus, this game is the original security game in the definition except that the challenge bit $\mu$ is fixed to 1. For the security proof, we also define additional hybrid games $G_{2,0}, \ldots, G_{2,\rho}, \ldots, G_{2,\ell}$ where $G_{2,0} = G_2$ and $G_{2,\ell} = G_3$. The game $G_{2,\rho}$ is almost identical to the game $G_{2,\rho-1}$ except that $CT^*_{SRE,\rho}$ is an encryption on the correct session key $K^*$. Specifically, each component $CT^*_{SRE,k}$ for $k \leq \rho$ is an encryption on the session key $K^*$ and each component $CT^*_{SRE,k}$ for $\rho < k$ is an encryption on the random session key $Z$.

Let $S_{\mathcal{A}}^{G_i}$ be the event that $\mathcal{A}$ outputs 0 in $G_i$. From Lemmas 22, 23 and 24, we obtain the following result

$$\begin{aligned} \mathrm{Adv}_{\mathcal{A}}^{IBR}(\lambda) &\leq \frac{1}{2}\left|\Pr[S_{\mathcal{A}}^{G_0}] - \Pr[S_{\mathcal{A}}^{G_3}]\right| \\ &\leq \ell \mathrm{Adv}_{\mathcal{B}}^{SRE}(\lambda) + \mathrm{Adv}_{\mathcal{B}}^{SKE}(\lambda) + \ell \mathrm{Adv}_{\mathcal{B}}^{SRE}(\lambda). \end{aligned}$$

This completes our proof. □

*Lemma 22: If the SRE scheme is secure under chosen plaintext attacks, then no polynomial time adversary can distinguish between $G_{0,\rho-1}$ and $G_{0,\rho}$ with non-negligible advantage.*

*Proof:* Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $G_{0,\rho-1}$ and $G_{0,\rho}$ with non-negligible advantage. A simulator $\mathcal{B}$ that breaks the security game of the SRE scheme is given: challenge public parameters $PP_{SRE}$. Then $\mathcal{B}$ that interacts with $\mathcal{A}$ is described as follows:

**Setup**: $\mathcal{B}$ first sets a perfect binary tree $\mathcal{BT}$ by running **SD.Setup**$(2^n)$ and gives $PP = (\mathcal{BT}, PP_{SRE})$ to $\mathcal{A}$.

**Query 1**: If $\mathcal{A}$ adaptively requests a private key query for a user $ID$, then $\mathcal{B}$ proceeds this query as follows: It first obtains a private set $PV_{ID} = \{S_{i,j}\}$ by running **SD.Assign**$(\mathcal{BT}, ID)$. For each $S_{i,j} \in PV_{ID}$, it sets labels $(GL, ML)$ from $S_{i,j}$ and requests $SK'_{SRE,S_{i,j}}$ for labels $(GL, ML)$ to the key generation oracle that simulates **SRE.GenKey**. It sets $SK_{ID} = (PV_{ID}, \{SK'_{SRE,S_{i,j}}\})$ and gives this to $\mathcal{A}$.

**Challenge**: $\mathcal{A}$ submits a challenge revoked set $R^*$ and two challenge messages $M_0^*, M_1^*$ subject to the restrictions. It sets $\mu = 0$ and proceeds as follows: It obtains two session keys $K^*$ and $Z$ by running **SKE.GenKey**$(1^\lambda)$ and computes $C^*$ by running **SKE.Encrypt**$(M_\mu^*, K^*)$. Next, it obtains a covering set $CV_{R^*} = \{S_{i_1,j_1}, \ldots, S_{i_\ell,j_\ell}\}$ by running **SD.Cover**$(\mathcal{BT}, R^*)$ and obtains each component $CT_{SRE,S_{i_k,j_k}}$ as follows:

1) For $1 \leq k \leq \rho - 1$, it computes $CT^*_{SRE,S_{i_k,j_k}}$ by running **SRE.Encrypt**$((L_{i_j} \| d_{j_k}, L_{j_k}), Z, PP_{SRE})$ where $(L_{i_k}, L_{j_k})$ is the label of $S_{i_k,j_k}$ and $d_{j_k}$ is the depth of the node $v_{j_k}$.
2) For $k = \rho$, it submits challenge labels $GL' = L_{i_\rho} \| d_{j_\rho}, ML' = L_{j_\rho}$ and two challenge messages $M'_0 = K^*, M'_1 = Z$ to the challenge oracle of SRE and receives a challenge ciphertext $CT'_{SRE}$. It simply sets $CT^*_{SRE,\rho} = CT'_{SRE}$.
3) For $\rho + 1 \leq k \leq \ell$, it computes $CT^*_{SRE,S_{i_k,j_k}}$ by running **SRE.Encrypt**$((L_{i_k} \| d_{j_k}, L_{j_k}), K^*, PP_{SRE})$ where $(L_{i_k}, L_{j_k})$ is the label of $S_{i_k,j_k}$ and $d_{j_k}$ is the depth of the node $v_{j_k}$.

It gives $CT = (CV_{R^*}, C^*, \{CT^*_{SRE,S_{i_k,j_k}}\}_{k=1}^{\ell})$ to $\mathcal{A}$.

**Query 2**: Same as Query 1.

**Guess**: Finally, $\mathcal{A}$ outputs a bit $\mu'$. $\mathcal{B}$ also outputs $\mu'$. □

*Lemma 23: If the SKE scheme is secure under chosen plaintext attacks, then no polynomial time adversary can distinguish between $G_1$ and $G_2$ with non-negligible advantage.*

*Proof:* Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $G_1$ and $G_2$ with non-negligible advantage. A simulator $\mathcal{B}$ that breaks the security game of the SKE scheme is described as follows:

**Setup**: $\mathcal{B}$ first obtains $MK$ and $PP$ by running **IBR.Setup**$(1^\lambda)$ and gives it to $\mathcal{A}$.

**Query 1**: If $\mathcal{A}$ adaptively requests a private key query for a user $ID$, then $\mathcal{B}$ creates the private key by using the master key $MK$.

**Challenge**: $\mathcal{A}$ submits a challenge revoked set $R^*$ and two challenge messages $M_0^*, M_1^*$. It also submits two challenge messages $M'_0 = M_0^*, M'_1 = M_1^*$ and receives a challenge ciphertext $CT'$ of SKE. It sets $C^* = CT'$. It chooses a random session key $Z$ and prepare all components $CT^*_{SRE,S_{i_k,j_k}}$ that are encryption on the random session key $Z$. It gives a challenge ciphertext $CT = (CV_{R^*}, C^*, \{CT^*_{SRE,S_{i_k,j_k}}\}_{k=1}^{\ell})$ to $\mathcal{A}$.

**Query 2**: Same as Query 1.

**Guess**: Finally, $\mathcal{A}$ outputs a bit $\mu'$. $\mathcal{B}$ also outputs $\mu'$. □

*Lemma 24: If the SRE scheme is secure under chosen plaintext attacks, then no polynomial time adversary can distinguish between $G_{2,\rho-1}$ and $G_{2,\rho}$ with non-negligible advantage.*

The proof is almost similar to that of Lemma 22.

## D. COMPARISONS

In this section, we compare our IBR scheme to previous IBR schemes. IBR schemes can be categorized as tree-based schemes that combine a tree data structure with an IBE or HIBE scheme [7], [9] or pairing-based schemes that directly are built from bilinear groups [6], [14].

Naor *et al.* [7] proposed an IBR scheme (NNL-IBR) by combining the tree-based CS revocation scheme with the Boneh-Franklin IBE scheme [8] to reduce the size of public parameters. The NNL-IBR scheme is secure under standard assumption in the random oracle model, but a ciphertext is consists of $O(r \log N/r)$ group elements where $r$ is the number of revoked users and $N$ is the number of maximum users. Dodis and Fazio [9] proposed an improved IBR scheme (DF-IBR) by combining the SD (or LSD) scheme with an HIBE scheme. If the DF-IBR scheme is instantiated with the Boneh-Boyen-Goh HIBE scheme [27], then the size of ciphertexts can be reduced to have $O(r)$ group elements. However, the private key of this DF-IBR scheme consists of $O(\log^{2.5} N)$ group elements.

**TABLE 2.** Benchmarks of basic group operations.

| Curve | Exp $\mathbb{G}_1$ | Exp $\mathbb{G}_2$ | Exp $\mathbb{G}_T$ | Pairing |
|---|---|---|---|---|
| SS512 | 2.63 | 2.63 | 0.74 | 3.02 |
| MNT159 | 0.83 | 7.65 | 1.91 | 6.38 |

All benchmarks are measured in milliseconds. Exp denotes the exponentiation and Pairing denotes the pairing operation.

**TABLE 3.** Benchmarks of our SRE scheme.

| Scheme | Curve | Setup | GenKey | Encrypt | Decrypt |
|---|---|---|---|---|---|
| SRE | SS512 | 20.02 | 11.02 | 13.75 | 11.40 |
| | MNT159 | 27.23 | 30.87 | 6.27 | 27.29 |

All benchmarks are measured in milliseconds. Setup, GenKey, Encrypt, and Decrypt algorithms denote the setup, key generation, encryption, and decryption algorithm respectively.

Lewko *et al.* [6] proposed IBR schemes that are directly built from bilinear groups. They presented a simple IBR scheme under $q$-type assumption and another IBR scheme under standard assumption. In their two IBR schemes, a ciphertext is consists of $O(r)$ group elements and a private key is consists of just $O(1)$ group elements. However, the number of pairing or exponentiation operations increases proportional to the number of revoked users $r$. Thus their IBR schemes are only appropriate in a situation where the number of revoked users is very small. Attrapadung and Libert [14] proposed an improved IBR scheme. They showed that the size of ciphertexts and the cost of decryption can be reduced if the size of public parameters and private keys increases.

Our IBR schemes are built by combining the SD (or LSD) revocation scheme and our SRE schemes instead of IBE or HIBE schemes. In our IBR schemes that use the LSD scheme, public parameters are consists of $O(1)$ group elements because of the underlying SRE scheme, a ciphertext and a private key are consists of $O(r)$ and $O(\log^{1.5} N)$ group elements respectively. Additionally the decryption algorithm of our IBR scheme only requires three pairing and two exponentiation operations. Thus our IBR scheme in prime-order groups provides efficient decryption compared to those of LSW-IBR and AL-IBR schemes.

## VI. IMPLEMENTATION

We implemented our identity-based revocation (IBR) schemes by using Charm [28], which is a framework for rapidly prototyping public-key cryptographic schemes by using the Python language. Charm provides three mathematical groups: integer group, elliptic curve group, and pairing group by using OpenSSL, GMP, PBC, RELIC, and MIRACL libraries that are written in the C language. Additionally, it also provides a bunch of public-key schemes and protocols. To measure the performance of our implementation, we used a desktop with Intel Core i5-4590 3.3GHz CPU and 16.0GB RAM that runs Windows 7 with Python-3.2.5.

To implement our IBR schemes, we selected two pairing groups: SS512 and MNT159. The SS512 pairing group is a super-singular elliptic curve group where the base field size is 512 bits and the embedding degree is two. In this SS512 group, the bit size of a group element in $\mathbb{G}_1$ and $\mathbb{G}_2$ is 512 bits and the bit size of a group element in $\mathbb{G}_T$ is 1024 bits. The MNT159 pairing group is an asymmetric group created by Miyaji, Nakabayashi, and Takano where the base field size is 159 bits and the embedding degree is six. In this MNT159 group. the bit size of a group element $\mathbb{G}_1$ is 159 bits and the bit size of a group element in $\mathbb{G}_2$ and $\mathbb{G}_T$ is 954 bits. The details of these groups can be found in [29]. The benchmark result of basic group operations in these pairing groups is given in Table 2. In the MNT159 group, the basic operations in $\mathbb{G}_1$ is very efficient compared with operations in $\mathbb{G}_2$ and $\mathbb{G}_T$.

By using the selected pairing groups, we implemented our SRE scheme with selective security and measured the performance of our construction. Although our SRE scheme is originally built on a symmetric pairing group, it can be easily translated to an asymmetric pairing group. In our SRE scheme, the **GenKey** algorithm requires four group exponentiations in $\mathbb{G}_2$ if additional elements are stored in a master key, the **Encrypt** algorithm requires five group exponentiations in $\mathbb{G}_1$ and one exponentiation in $\mathbb{G}_T$, and the **Decrypt** algorithm requires four pairing operations and one group exponentiation in $\mathbb{G}_T$. The benchmark result of our SRE scheme in SS512 and MNT159 groups is given in Table 3. Note that the **Encrypt** algorithm is two-times fast in the MNT159 group, but the **GenKey** and **Decrypt** algorithms are slow in the MNT159 group.

For our IBR schemes, we implemented (tree-based) SD and LSD schemes that belong to the subset cover framework of Naor *et al.* [7] and compared their performance with that of the CS scheme. A simple way to represent a binary tree is to assign a unique node number $i$ to each node $v_i$ by using the depth-first search tree traversal. In this case, the root node is assigned to 1, the child node $v_L$ of a parent node $v$ with a number $i$ is assigned to $2i$ and the right child of $v_R$ is assigned to $2i+1$. The detailed algorithms of the SD scheme is given in Section V-A and the details of the LSD scheme is

**TABLE 4.** Benchmarks of CS, SD, and LSD schemes.

| Scheme | Algorithm | $r = 10$ | $r = 50$ | $r = 100$ | $r = 200$ | $r = 300$ |
|---|---|---|---|---|---|---|
| CS | Cover | 0.51 | 2.69 | 5.32 | 7.89 | 14.42 |
| | $\|CV\|$ | 105 | 420 | 747 | 1316 | 1826 |
| SD | Cover | 3.40 | 10.65 | 15.80 | 44.66 | 85.50 |
| | $\|CV\|$ | 13 | 63 | 119 | 256 | 374 |
| LSD | Cover | 3.72 | 13.17 | 26.77 | 46.17 | 87.50 |
| | $\|CV\|$ | 17 | 95 | 201 | 392 | 589 |

All benchmarks are measured in milliseconds. We set the depth $d$ of a binary tree to be 15 and let $r$ to be the number of revoked leaf nodes.

**TABLE 5.** Benchmarks of our IBR schemes.

| Scheme | Algorithm | $r = 10$ | $r = 50$ | $r = 100$ | $r = 200$ | $r = 300$ |
|---|---|---|---|---|---|---|
| IBR-SD | GenKey | 1332 | 1329 | 1327 | 1325 | 1325 |
| (SS512) | Encrypt | 152 | 866 | 1745 | 3404 | 5377 |
| | Decrypt | 11 | 12 | 13 | 12 | 17 |
| IBR-LSD | GenKey | 668 | 665 | 677 | 661 | 662 |
| (SS512) | Encrypt | 276 | 1293 | 2640 | 5240 | 8115 |
| | Decrypt | 11 | 11 | 11 | 15 | 17 |
| IBR-LSD | GenKey | 1821 | 1855 | 1859 | 1826 | 1819 |
| (MNT159) | Encrypt | 101 | 517 | 1304 | 2380 | 3674 |
| | Decrypt | 26 | 26 | 26 | 27 | 30 |

All benchmarks are measured in milliseconds. We set the maximum number $N$ of users to be $2^{15} = 3.2 * 10^4$ and let $r$ to be the number of revoked users.

given in [13]. To measure the performance of SD and LSD schemes, we set the depth $d$ of a binary tree as 15. In this case, the number $N$ of leaf nodes is $2^{15} = 3.2 * 10^4$. Let $R$ be a set of revoked nodes and $r$ be the size of $R$. In the SD scheme, the size of a covering set $CV$ for a revoked set $R$ is theoretically bounded by $2r - 1$, but it is approximately $1.25r$ if revoked leaf nodes are randomly selected. In the LSD scheme, the size of CV is bounded by $4r - 2$, but it is about $1.98r$ on average. The benchmark result of SD and LSD schemes is given in Table 4. In SD and LSD schemes, the performance of the **Cover** algorithm that finds $CV$ for a non-revoked set is relatively slow. If the **SD.Cover** algorithm of Naor *et al.* [7] is naively implemented, then the running time of this algorithm is $O(r^4 \log^2 N)$ where $r$ is the size of $R$. For our IBR scheme, we optimized the **SD.Cover** algorithm and obtained an improved one with $O(r^2 \log^2 N)$ running time. Note that the running time of the **CS.Cover** algorithm is just $O(r \log N)$.

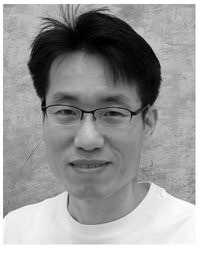
By combining the implementation of our SRE scheme and that of the SD (or LSD) scheme, we implemented our IBR-SD (or IBR-LSD) scheme and measured the performance on different pairing groups. The benchmark result of our IBR schemes is given in Table 5. First, the running time of the **Setup** algorithm is the same as that of the **SRE.Setup** algorithm. The running time of the **GenKey** algorithm is independent of the number $r$ of revoked users, but it is dependent on the depth $d = \log N$ of a binary tree where $N$ is the number of maximum users. The **IBR-SD.GenKey** algorithm performs $O(\log^2 N)$ number of the **SRE.GenKey** algorithm, but the **IBR-LSD.GenKey** algorithm performs $O(\log^{1.5} N)$ number of the **SRE.GenKey** algorithm. Thus, the **IBR-LSD.GenKey** algorithm is approximately two-times faster than the **IBR-SD.GenKey** algorithm according to the Table. The running time of the **Encrypt** algorithm increases depending on the size of $CV$. Additionally, the **Encrypt** algorithm should find $CV$ by running the **Cover** algorithm of the SD (or LSD) scheme. The running time of the **Decrypt** algorithm is very efficient since it just requires one **SRE.Decrypt** algorithm after finding a matching tuple of the SD (or LSD) scheme.

## VII. CONCLUSION

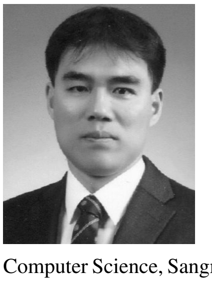
In this paper, we solved the problem of Lee *et al.* [10] to construct an SRE scheme under the standard assumption without random oracles. The main insight of our solution is that an SRE scheme can be built by combining an IBE scheme and a simple IBR scheme that are secure under the standard assumption. We first proposed an SRE scheme in prime-order bilinear groups and proved its selective security under the DBDH assumption. We next proposed another SRE scheme in composite-order bilinear groups and proved its full security under simple static assumptions. We expect that our composite-order SRE scheme can be converted into a prime-order one by following the conversion method of Lewko [30]. We proposed an IBR scheme by combining the SD (or LSD) method and our SRE scheme and proved its security. Finally, we implemented our IBR-SD and IBR-LSD schemes in Charm and measured the performance of our implementations.

## REFERENCES


[1] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Advances in Cryptology—CRYPTO*, vol. 3621. Berlin, Germany: Springer, 2005, pp. 258–275.
[2] J. H. Park, H. J. Kim, H.-M. Sung, and D. H. Lee, "Public key broadcast encryption schemes with shorter transmissions," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 401–411, Sep. 2008.
[3] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Cryptology—ASIACRYPT*, vol. 4833. Berlin, Germany: Springer, 2007, pp. 200–215.
[4] R. Sakai and J. Furukawa, "Identity-based broadcast encryption," Cryptol. ePrint Archive, Carson City, NV, USA, Tech. Rep. 2007/217, 2007. [Online]. Available: http://eprint.iacr.org/2007/217
[5] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," in *Advances in Cryptology—EUROCRYPT*, vol. 5479. Berlin, Germany: Springer, 2009, pp. 171–188.
[6] A. B. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 273–285.
[7] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology—CRYPTO*, vol. 2139. Berlin, Germany: Springer, 2001, pp. 41–62.
[8] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO*, vol. 2139. Berlin, Germany: Springer, 2001, pp. 213–229.
[9] Y. Dodis and N. Fazio, "Public key broadcast encryption for stateless receivers," in *Digital Rights Management*, vol. 2696. Berlin, Germany: Springer, 2002, pp. 61–80.
[10] K. Lee, W. K. Koo, D. H. Lee, and J. H. Park, "Public-key revocation and tracing schemes with subset difference methods revisited," in *Computer Security—ESORICS*, vol. 8713. Berlin, Germany: Springer, 2014, pp. 1–18.
[11] B. Waters, "Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions," in *Advances in Cryptology—CRYPTO*, vol. 5677. Berlin, Germany: Springer, 2009, pp. 619–636.
[12] A. B. Lewko and B. Waters, "New techniques for dual system encryption and fully secure HIBE with short ciphertexts," in *Theory of Cryptography—TCC*, vol. 5978. Berlin, Germany: Springer, 2010, pp. 455–479.
[13] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," in *Advances in Cryptology—CRYPTO*, vol. 2442. Berlin, Germany: Springer, 2002, pp. 47–60.
[14] N. Attrapadung and B. Libert, Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation," in *Public-Key Cryptography—PKC*, vol. 6056. Berlin, Germany: Springer, 2010, pp. 384–402.
[15] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology*, vol. 773. Berlin, Germany: Springer, 1993, pp. 480–491.
[16] M. T. Goodrich, J. Z. Sun, and R. Tamassia, "Efficient tree-based revocation in groups of low-state devices," in *Advances in Cryptology—CRYPTO*, vol. 3152. Berlin, Germany: Springer, 2004, pp. 511–527.
[17] M. Naor and B. Pinkas, "Efficient trace and revoke schemes," in *Financial Cryptography—FC*, vol. 1962. Berlin, Germany: Springer, 2000, pp. 1–20.
[18] D. Boneh and B. Waters, "A fully collusion resistant broadcast, trace, and revoke system," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2006, pp. 211–220.
[19] S. Garg, A. Kumarasubramanian, A. Sahai, and B. Waters, "Building efficient fully collusion-resilient traitor tracing and revocation schemes," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2010, pp. 121–130.
[20] K. Lee and D. H. Lee, "Adaptively secure broadcast encryption under standard assumptions with better efficiency," *IET Inf. Secur.*, vol. 9, no. 3, pp. 149–157, 2015.
[21] D. Boneh, B. Waters, and M. Zhandry, "Low overhead broadcast encryption from multilinear maps," in *Advances in Cryptology—CRYPTO*, vol. 8616. Berlin, Germany: Springer, 2014, pp. 206–223.
[22] M. Zhandry, "Adaptively secure broadcast encryption with small system parameters," Cryptol. ePrint Arch., Carson City, NV, USA, Tech. Rep. 2014/757, 2014. [Online]. Available: http://eprint.iacr.org/2014/757S
[23] J. Chen, B. Libert, and S. C. Ramanna, "Non-zero inner product encryption with short ciphertexts and private keys," in *Security and Cryptography for Networks—SCN*, vol. 9841. Berlin, Germany: Springer, 2016, pp. 23–41.
[24] D. Boneh and X. Boyen, "Efficient selective-id secure identity-based encryption without random oracles," in *Advances in Cryptology—EUROCRYPT*, vol. 3027. Berlin, Germany: Springer, 2004, pp. 223–238.
[25] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology—EUROCRYPT*, vol. 3027. Berlin, Germany: Springer, 2004, pp. 207–222.
[26] X. Boyen, Q. Mei, and B. Waters, "Direct chosen ciphertext security from identity-based techniques," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2005, pp. 320–329.
[27] "Hierarchical identity based encryption with constant size ciphertext," in *Advances in Cryptology—EUROCRYPT*, vol. 3494. Berlin, Germany: Springer, 2005, pp. 440–456.
[28] J. A. Akinyele *et al.*, "Charm: A framework for rapidly prototyping cryptosystems," *J. Cryptograph. Eng.*, vol. 3, no. 2, pp. 111–128, 2013.
[29] B. Lynn, "On the implementation of pairing-based cryptosystems," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2007.
[30] A. B. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Advances in Cryptology—EUROCRYPT*, vol. 7237. Berlin, Germany: Springer, 2012, pp. 318–335.


**KWANGSU LEE** received the B.S. degree in computer science from Yonsei University, South Korea, in 1998, the M.S. degree in computer science from KAIST, South Korea, in 2000, and the Ph.D. degree in information security from Korea University, in 2011. He is currently an Assistant Professor with the Department of Computer and Information Security, Sejong University, Seoul, South Korea. His research interests include public-key cryptography, cryptographic protocols, provable security, and post-quantum security.


**JONG HWAN PARK** received the B.S. degree from the Department of Mathematics and the M.S. and Ph.D. degrees from the Graduate School of Information Security, Korea University, Seoul, South Korea, in 1999, 2004, and 2008, respectively. From 2009 to 2011, he served as a Research Professor for Kyung Hee University, and from 2011 to 2013, he served as a Research Professor for Korea University. Since 2013, he has been serving as an Assistant Professor with the Department of Computer Science, Sangmyung University, Seoul, South Korea. His research interests include functional encryption, broadcast encryption, authenticated encryption, and various cryptographic protocols.