Identity-Based Revocation from Subset Difference Methods under Simple Assumptions

Kwangsu Lee* Jong Hwan Park†

Abstract

Identity-based revocation (IBR) is a specific kind of broadcast encryption that can effectively send a 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, Koo, Lee, and Park (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 work, we first propose a selectively secure SRE scheme under the standard assumption without random oracles. Next, we propose a fully secure SRE scheme under simple static assumptions without random oracles. Finally, we present an efficient IBR scheme derived from the SD method and our SRE scheme. The security of our IBR scheme depends on that of the underlying SRE scheme.

Keywords: Broadcast encryption, Identity-based revocation, Subset cover framework, Bilinear maps.

*Korea University, Korea. Email: `kwangsu.lee@korea.ac.kr`.
†Sangmyung University, Korea. Email: `jhpark@smu.ac.kr`.

1 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.

One method to build a collusion-resistant PKBE scheme is to use bilinear groups. Boneh, Gentry, and Waters [5] 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 [10, 14, 20, 25, 26]. Another method to build a secure PKBE scheme is to combine the subset cover framework of Naor, Naor, and Lotspiech [23] and an identity-based encryption (IBE) scheme [4]. Naor et al. [23] showed that a PKR scheme can be obtained from the complete subtree (CS) method and an IBE scheme and Dodis and Fazio [11] 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. [17] showed that an improved PKR scheme can be derived by combining the SD method with a single revocation encryption (SRE) scheme. 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 operation in the decryption algorithm.

The PKR scheme of Lee et al. [17] that combines the SD method with an SRE scheme is interesting since it achieves the asymptotically optimal bound in the SD method. The SD method is one instance of the subset cover framework of Naor et al. [23] 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 $T_{i,j}$ that is related with two nodes v_i and v_j in a full binary tree. That is, $T_{i,j}$ is defined as a set of leaf nodes in T_i but not in T_j where the root node of T_i, 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 $ML \neq ML'$ [17]. To construct an improved PKR scheme, Lee et al. [17] observed that a subset $T_{i,j}$ 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.

1.1 Our Contributions

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

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 $ML \neq ML'$. The main idea to build an SRE scheme under the standard assumption is that an IBE scheme can be used to support the equality $GL = GL'$ and a simple IBR scheme can be 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.

SRE with Full Security. Our first SRE scheme is just secure in the selective model. 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 with slight modification except that it uses composite-order bilinear groups to use the dual system encryption technique of Waters [21, 27]. To prove the full security of our second SRE scheme, we cannot use the dual system encryption technique of Lewko and Waters [21] that shows the information theoretic argument because of the inequality $ML \neq ML'$ in SRE. To solve this problem, we use the new proof technique of Lewko and Waters [22] that combines the selective technique with the dual system encryption technique.

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. [17]. As mentioned before, an SRE scheme can be integrated with the SD method since a subset $T_{i,j}$ in SD can be directly mapped to the labels (GL, ML) in SRE. Lee et al. only proposed a PKR scheme where 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.

1.2 Related Work

Broadcast encryption, introduced by Fiat and Naor [12], 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, Naor, and Lotspiech [23] 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 [15, 16].

In public-key broadcast encryption (PKBE), any user can create a ciphertext for a set of receivers 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 associated with a set of revoked users R. Naor and Pinkas [24] 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. [5] proposed a fully collusion-resistant PKBE scheme in bilinear groups that achieves short ciphertexts. After that, many PKBE scheme in bilinear groups were presented [6, 13, 14, 18, 20, 25]. The subset cover framework also can be used to build a PKR scheme by combining it with an IBE, HIBE, or SRE scheme [11, 17, 23].

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. A fully collusion-resistant IBBE scheme was independently proposed by Delerablée, Sakai, and Furukawa [10, 26]. Recently, IBBE schemes with short public parameters were proposed in multilinear maps [7, 28]. IBR is a variation of IBBE where a set of revoked users R is specified in a ciphertext. Note that an IBBE scheme cannot be converted to an IBR scheme since the maximum number of users is an exponential value whereas a PKBE scheme can be easily converted to a PKR scheme. Lewko et al. [20] proposed an IBR scheme with short keys in bilinear groups and an improved IBR scheme was presented by Attrapadung and Libert [2].

2 Preliminaries

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

2.1 Single Revocation Encryption

Before we define IBR, we first define SRE. The concept of SRE was introduced by Lee et al. [17] 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 2.1 (Single Revocation Encryption)**.** *An SRE scheme for the universe $\mathcal{U}$ of groups and members consists of four algorithms **Setup**, **GenKey**, **Encrypt**, and **Decrypt**, which are defined as follows:*

***Setup**($1^\lambda, \mathcal{U}$). The setup algorithm takes as input a security parameter 1^λ. 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 $(GL = GL') \wedge (ML \neq ML')$, then $\textbf{Decrypt}(\textbf{Encrypt}((GL, ML), M, PP), SK_{(GL', ML')}, PK) = M$.*
- *If $(GL \neq GL') \vee (ML = ML')$, then $\textbf{Decrypt}(\textbf{Encrypt}((GL, ML), M, PP), SK_{(GL', ML')}, PK) = \perp$.*

The security model of SRE was defined by Lee et al. [17] 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 2.2 (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^λ) 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 μ, and wins the game if $\mu = \mu'$.*

The advantage of $\mathcal{A}$ is defined as $\boldsymbol{Adv}_{\mathcal{A}}^{SRE}(\lambda) = \left|\Pr[\mu = \mu'] - \frac{1}{2}\right|$ 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 λ.

2.2 Identity-Based Revocation

IBR is a special type of PKBE where a ciphertext is associated with a set of revoked users R instead of a set of receivers S 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 2.3 (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^λ). The setup algorithm takes as input a security parameter 1^λ. 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 an encrypted 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 $ID \notin R$, then **Decrypt**(**Encrypt**$(R, M, PP), SK_{ID}, PP) = M$.*
- *If $ID \in R$, then **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. [20]. 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 2.4 (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**$(1^\lambda, N)$ 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_μ^*, 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 μ, and wins the game if $\mu = \mu'$.*

The advantage of $\mathcal{A}$ is defined as $\textbf{\textit{Adv}}_{\mathcal{A}}^{IBR}(\lambda) = \left|\Pr[\mu = \mu'] - \frac{1}{2}\right|$ 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 λ.

3 SRE with Selective Security

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

3.1 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.

3.2 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 [4] and widely used to prove the security of IBE, HIBE, and ABE schemes.

Assumption 3.1 (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 λ. 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 $\boldsymbol{Adv}_{\mathcal{B}}^{DBDH}(\lambda) = \big|\Pr[\mathcal{B}(D, T_0) = 0] - \Pr[\mathcal{B}(D, T_1) = 0]\big|$ where the probability is taken over the random choice of $a, b, c, d \in \mathbb{Z}_p$.*

3.3 Construction

The previous SRE scheme of Lee et al. [17] 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 [3] 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. [20] 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 $(g^{\alpha}(u^{GL}h)^{r_1}w^{r_2}, (w^{ML}v)^{r_2}, g^{-r_1}, g^{-r_2})$.

Our SRE scheme in prime-order bilinear groups is described as follows:

SRE.Setup(1^{λ}): This algorithm first generates a bilinear group $\mathbb{G}$ of prime order p of bit size $\Theta(\lambda)$. Let g be a generator of $\mathbb{G}$. It chooses a random exponent $\alpha \in \mathbb{Z}_p$ and a random hash function H from $\mathcal{H}$. It outputs a master key $MK = \alpha$ and public parameters as

$$PP = \Big(\ (p, \mathbb{G}, \mathbb{G}_T, e),\ g,\ u, h,\ w, v,\ H,\ \Omega = e(g,g)^{\alpha}\ \Big).$$

SRE.GenKey($(GL, ML), MK, PP$): This algorithm takes as input labels (GL, ML), the master key MK, and the public parameters PP. It selects random exponents $r_1, r_2 \in \mathbb{Z}_p$ and outputs a private key by implicitly including (GL, ML) as

$$SK_{(GL,ML)} = \Big(\ K_0 = g^{\alpha}(u^{GL}h)^{r_1}w^{r_2},\ K_1 = (w^{ML}v)^{r_2},\ K_2 = g^{-r_1},\ K_3 = g^{-r_2}\ \Big).$$

SRE.Encrypt($(GL, ML), M, PP$): This algorithm takes as input labels (GL, ML), a message $M \in \{0,1\}^m$, and the public parameters PP. It chooses a random exponent $t \in \mathbb{Z}_p$ and outputs a ciphertext by implicitly including (GL, ML) as

$$CT_{(GL,ML)} = \Big(\ C = H(\Omega^t) \oplus M,\ C_0 = g^t,\ C_1 = (u^{GL}h)^t,\ C_2 = (w^{ML}v)^t\ \Big).$$

SRE.Decrypt($CT_{(GL,ML)}, SK_{(GL',ML')}, PP$): This algorithm takes as input a ciphertext $CT_{(GL,ML)}$, a private key $SK_{(GL',ML')}$, and the public parameters PP. If $(GL = GL') \wedge (ML \neq ML')$, then it outputs a message as

$$M = C \oplus H\big(e(C_0, K_0) \cdot e(C_1, K_2) \cdot \big(e(C_0, K_1) \cdot e(C_2, K_3)\big)^{-1/(ML'-ML)}\big).$$

Otherwise, it outputs $\perp$.

3.4 Correctness

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

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

3.5 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 [3, 20], 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 3.2. *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:

Setup: $\mathcal{B}$ first guesses challenge labels (GL', ML') such that ML' is a member of GL'. It selects random exponents $y_u, y_h, y_w, y_v \in \mathbb{Z}_p$ and creates the public key implicitly setting $\alpha = ab$ as $PK = \big((p, \mathbb{G}, \mathbb{G}_T, e), g,\ u = g^a g^{y_u}, h = (g^a)^{-GL'} g^{y_h},\ w = g^a g^{y_w}, v = (g^a)^{-ML'} g^{y_v},\ \Omega = e(g^a, g^b)\big)$.

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** $GL \neq GL'$: In this case, 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

$$K_0 = (g^b)^{-(y_u GL + y_h)/(GL - GL')} (u^{GL} h)^{r_1'} w^{r_2},\ K_1 = (g^b)^{1/(GL - GL')} g^{-r_1'},\ K_2 = (w^{ML} v)^{r_2},\ K_3 = g^{-r_2}.$$

- **Case** $GL = GL'$ and $ML = ML'$: In this case, 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

$$K_0 = (u^{GL} h)^{r_1} (g^b)^{-y_w} w^{r_2'},\ K_1 = g^{-r_1},\ K_2 = (g^b)^{-(y_w ML + y_v)} (w^{ML} v)^{r_2'},\ K_3 = g^b g^{-r_2'}.$$

Challenge: $\mathcal{A}$ submits challenge labels (GL^*, ML^*) and two messages M_0^*, M_1^*. If $(GL' \neq GL^*) \vee (ML' \neq ML^*)$, then $\mathcal{B}$ aborts the simulation since it failed to guess the challenge labels. Otherwise, $\mathcal{B}$ flips a random coin $\mu \in \{0,1\}$ internally. Next, it implicitly sets $t = c$ and creates a challenge ciphertext as

$$C = H(T) \cdot M_\mu^*,\ C_0 = g^c,\ C_1 = (g^c)^{y_u GL^* + y_h},\ C_2 = (g^c)^{y_w ML^* + y_v}.$$

Query 2: Same as Query 1.
Guess: Finally, $\mathcal{A}$ outputs a guess μ'. If $\mu = \mu'$, $\mathcal{B}$ outputs 0. Otherwise, it outputs 1.

To finish the proof, we first show that hash outputs, private keys, and the challenge ciphertext are correctly distributed. In case of $GL \neq GL'$, the private key is correctly distributed since it satisfies the following equation

$$\begin{aligned}
K_0 &= g^\alpha (u^{GL} h)^{r_1} w^{r_2} = g^{ab}\big((g^a g^{y_u})^{GL} (g^a)^{-GL'} g^{y_h}\big)^{-b/(GL-GL')+r_1'} w^{r_2} \\
&= (g^b)^{-(y_u GL + y_h)/(GL-GL')} (u^{GL} h)^{r_1'} w^{r_2}, \\
K_1 &= g^{-r_1} = g^{b/(GL-GL')-r_1'} = (g^b)^{1/(GL-GL')} g^{-r_1'},\ K_2 = (w^{ML} v)^{r_2},\ K_3 = g^{-r_2}.
\end{aligned}$$

In case of $(GL = GL') \wedge (ML = ML')$, the private key is also correctly distributed as

$$\begin{aligned}
K_0 &= g^\alpha (u^{GL} h)^{r_1} w^{r_2} = g^{ab} (u^{GL} h)^{r_1} (g^a g^{y_w})^{-b+r_2'} = (u^{GL} h)^{r_1} (g^b)^{-y_w} w^{r_2'}, \\
K_1 &= g^{-r_1},\ K_2 = (w^{ML} v)^{r_2} = ((g^a g^{y_w})^{ML} (g^a)^{-ML'} g^{y_v})^{-b+r_2'} = (g^b)^{-(y_w ML + y_v)} (w^{ML} v)^{r_2'}, \\
K_3 &= g^{-r_2} = g^{b-r_2'} = g^b g^{-r_2'}.
\end{aligned}$$

Note that it cannot create a private key for (GL, ML) such that $(GL = GL') \wedge (ML \neq ML')$ since the element g^{ab} cannot be removed. The challenge ciphertext is also correctly distributed since it satisfies the following equation

$$\begin{aligned}
C &= H(e(g,g)^{\alpha t}) M_\mu^* = H(e(g,g)^{abc}) M_\mu^*,\ C_0 = g^t = g^c, \\
C_1 &= (u^{GL^*} h)^t = ((g^a g^{y_u})^{GL^*} (g^a)^{-GL'} g^{y_h})^c = (g^c)^{y_u GL^* + y_h}, \\
C_2 &= (w^{ML^*} v)^t = ((g^a g^{y_w})^{ML^*} (g^a)^{-ML'} g^{y_v})^c = (g^c)^{y_w ML^* + y_v}.
\end{aligned}$$

This completes our proof. □

3.6 Discussions

Efficiency Analysis. In our SRE scheme, a private key and a ciphertext consist of four number of 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. [17] 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 oracle model.

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. [9]. 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. [8], but we should modify our SRE scheme to be a key encapsulation mechanism (KEM).

4 SRE with Full Security

In this section, we propose an SRE scheme in composite-order bilinear groups and prove its full-model security under simple assumptions.

4.1 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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ 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.

4.2 Complexity Assumptions

To prove the security of our SRE scheme, we introduce simple static assumptions that were used by Lewko and Waters [21, 22] to prove the full model security of ABE schemes by using the dual system encryption technique.

Assumption 4.1 (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 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 $\boldsymbol{Adv}^{SD}_{\mathcal{A}}(\lambda) = \big|\Pr[\mathcal{A}(D, T_0) = 0] - \Pr[\mathcal{A}(D, T_1) = 0]\big|$ 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 4.2 (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 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 $\boldsymbol{Adv}^{GSD}_{\mathcal{A}}(\lambda) = \big|\Pr[\mathcal{A}(D, T_0) = 0] - \Pr[\mathcal{A}(D, T_1) = 0]\big|$ 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 4.3 (3-party Diffie-Hellman, 3DH)**.** *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 Assumption is that if the challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_2, g_3, g_2^a, g_2^b, g_2^c)$ and T are given, no PPT* *algorithm $\mathcal{A}$ can distinguish $T = T_0 = g_2^{abc}$ from $T = T_1 = g_2^d$ with more than a negligible advantage. The advantage of $\mathcal{A}$ is defined as $\boldsymbol{Adv}_{\mathcal{A}}^{3DH}(\lambda) = \big|\Pr[\mathcal{A}(D,T_0)=0] - \Pr[\mathcal{A}(D,T_1)=0]\big|$ where the probability is taken over random choices of $a,b,c,d \in \mathbb{Z}_N$.*

Assumption 4.4 (Composite Diffie-Hellman, ComDH)**.** *Let $(N,\mathbb{G},\mathbb{G}_T,e)$ be a description of the bilinear group of composite order $N = p_1p_2p_3$. 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 Assumption is that if the challenge tuple $D = ((N,\mathbb{G},\mathbb{G}_T,e),g_1,g_2,g_3,g_1^aR_1,g_1^bR_2)$ 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 $\boldsymbol{Adv}_{\mathcal{A}}^{ComDH}(\lambda) = \big|\Pr[\mathcal{A}(D,T_0)=0] - \Pr[\mathcal{A}(D,T_1)=0]\big|$ where the probability is taken over random choices of $a,b,c \in \mathbb{Z}_N$, and $R_1,R_2 \in \mathbb{G}_{p_2}$.*

4.3 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 [21, 22]. Our fully secure SRE scheme has the similar structure with that of the selectively secure SRE scheme in prime order groups, but this SRE scheme additionally contains one group element in private keys to prove the full security. Our SRE scheme in composite-order bilinear groups is described as follows:

SRE.Setup(1^λ): This algorithm first generates a bilinear group $\mathbb{G}$ of prime order p of bit size $\Theta(\lambda)$. Let g be a generator of $\mathbb{G}$. It chooses a random exponent $\alpha \in \mathbb{Z}_N$ and a random hash function H from $\mathcal{H}$. It outputs a master key $MK = \alpha$ and public parameters as

$$PP = \Big(\ (N,\mathbb{G},\mathbb{G}_T,e),\ g = g_1,\ Y = g_3,\ u,h,\ w,v,\ z,\ H,\ \Omega = e(g_1,g_1)^\alpha\ \Big).$$

SRE.GenKey($(GL,ML),MK,PP$): This algorithm takes as input labels (GL,ML), the master key MK, and the public parameters PP. 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 outputs a private key as

$$SK_{(GL,ML)} = \Big(\ K_0 = g^\alpha(u^{GL}h)^{r_1}w^{r_2}z^{r_3}Y_0,\ K_1 = (w^{ML}v)^{r_2}Y_1,\ K_2 = g^{-r_1}Y_2,\ K_3 = g^{-r_2}Y_3,\ K_4 = g^{-r_3}Y_4\ \Big).$$

SRE.Encrypt($(GL,ML),M,PP$): This algorithm takes as input labels (GL,ML), a message $M \in \{0,1\}^m$, and the public parameters PP. It chooses a random exponent $t \in \mathbb{Z}_N$ and outputs a ciphertext as

$$CT_{(GL,ML)} = \Big(\ C = H(\Omega^t)\oplus M,\ C_0 = g^t,\ C_1 = (u^{GL}h)^t,\ C_2 = (w^{ML}v)^t,\ C_3 = z^t\ \Big).$$

SRE.Decrypt($CT_{(GL,ML)},SK_{(GL',ML')},PP$): This algorithm takes as input a ciphertext $CT_{(GL,ML)}$, a private key $SK_{(GL',ML')}$, and the public parameters PP. If $(GL = GL') \wedge (ML \neq ML')$, then it outputs a message as

$$M = C \oplus H\Big(e(C_0,K_0)\cdot e(C_1,K_2)\cdot\big(e(C_0,K_1)\cdot e(C_2,K_3)\big)^{-1/(ML'-ML)}\cdot e(C_3,K_4)\Big).$$

Otherwise, it outputs $\perp$.

4.4 Security Analysis

For the security proof of our SRE scheme, we use the dual system encryption technique [21, 27] 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 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. That is, the information theoretic argument does not work if the labels in a private key are equal to that in the challenge ciphertext [21]. To solve this problem, we use the new dual system encryption technique of Lewko and Waters [22] that combines the partitioning technique with the dual system encryption technique. In this new proof technique, we use the partitioning technique to show the indistinguishability between a nominal private key and a semi-functional private key instead of using the information theoretic argument.

To prove the full security of our SRE scheme, we use the original and new dual system encryption techniques at the same time since our SRE scheme consists of an IBE scheme and a simple IBR scheme. To apply the two techniques at the same time, we divide the behaviour of adversaries as two types: Type-A and Type-B that will be defined later. If the adversary is Type-A, then we use the information theoretic argument to show the indistinguishability between nominal type and semi-functional type. Otherwise, we use the partitioning technique. The security proof of our SRE scheme is described as follows:

Theorem 4.5. *The above SRE scheme is fully secure under chosen plaintext attacks if the SD, GSD, 3DH, 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. This algorithm first creates a normal private key $SK'_{GL,ML} = (K'_0, K'_1, K'_2, K'_3, K'_4)$ by using MK. It chooses a random element $R \in \mathbb{G}_{p_2}$ and outputs a semi-functional private key $SK_{GL,ML} = \big(K_0 = K'_0 R, K_1 = K'_1, K_2 = K'_2, K_3 = K'_3, K_4 = K'_4\big)$.

SRE.EncryptSF. This algorithm first creates a normal ciphertext $CT'_{GL,ML} = (C', C'_0, C'_1, C'_2, C'_3)$. It chooses random exponents $\tau, \eta_1, \eta_2, \theta_1, \theta_2, \kappa \in \mathbb{Z}_N$ and outputs semi-functional ciphertext $CT_{GL,ML} = \big(C_0 = C'_0 g_2^{\tau}, C_1 = C'_1 g_2^{(\eta_1 GL + \eta_2)\tau}, C_2 = C'_2 g_2^{(\theta_1 ML + \theta_2)\tau}, C_3 = C'_3 g_2^{\kappa\tau}\big)$. Note that $\eta_1, \eta_2, \theta_1, \theta_2, \kappa$ 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 $\mathbf{G}_0$. This game is the original security game. In this game, all private keys and the challenge ciphertext are normal.

Game $\mathbf{G}_1$. In the game $\mathbf{G}_1$, all private keys are still normal, but the challenge ciphertext is semi-functional.

Game $\mathbf{G}_2$. Next, we define a new game $\mathbf{G}_2$. In this game, all private keys are semi-functional. For the security proof, we additionally define a sequence of sub-games $\mathbf{G}_{1,1}, \ldots, \mathbf{G}_{1,k}, \ldots, \mathbf{G}_{1,q}$ where $\mathbf{G}_1 = \mathbf{G}_{1,0}$ and q is the maximum number of private keys. In the game $\mathbf{G}_{1,k}$, the challenge ciphertext is semi-functional, all jth private keys such that $j \leq k$ are semi-functional, and the remaining jth private keys such that $k < j$ are normal. It is obvious that $\mathbf{G}_{1,q} = \mathbf{G}_2$.

Game $\mathbf{G}_3$. In the final game $\mathbf{G}_3$, all private keys and the challenge ciphertext are semi-functional, but the challenge ciphertext component C is random.

Let $\mathbf{Adv}_{\mathcal{A}}^{G_j}$ be the advantage of $\mathcal{A}$ in the game $\mathbf{G}_j$. We have $\mathbf{Adv}_{\mathcal{A}}^{SRE}(\lambda) = \mathbf{Adv}_{\mathcal{A}}^{G_0}$, $\mathbf{Adv}_{\mathcal{A}}^{G_1} = \mathbf{Adv}_{\mathcal{A}}^{G_{1,0}}$, $\mathbf{Adv}_{\mathcal{A}}^{G_2} = \mathbf{Adv}_{\mathcal{A}}^{G_{1,q}}$, and $\mathbf{Adv}_{\mathcal{A}}^{G_3} = 0$. Through the following Lemmas 4.6, 4.7, and 4.13, we obtain the following equation

$$\begin{aligned}
&\mathbf{Adv}_{\mathcal{A}}^{SRE}(\lambda) \\
&\leq \left|\mathbf{Adv}_{\mathcal{A}}^{G_0} - \mathbf{Adv}_{\mathcal{A}}^{G_1}\right| + \sum_{k=1}^{q} \left|\mathbf{Adv}_{\mathcal{A}}^{G_{1,k-1}} - \mathbf{Adv}_{\mathcal{A}}^{G_{1,k}}\right| + \left|\mathbf{Adv}_{\mathcal{A}}^{G_2} - \mathbf{Adv}_{\mathcal{A}}^{G_3}\right| \\
&\leq \mathbf{Adv}_{\mathcal{B}}^{SD}(\lambda) + q(2\mathbf{Adv}_{\mathcal{B}}^{GSD}(\lambda) + \mathbf{Adv}_{\mathcal{B}}^{3DH}(\lambda)) + \mathbf{Adv}_{\mathcal{B}}^{ComDH}(\lambda).
\end{aligned}$$

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

Lemma 4.6. *If the SD assumption holds, then no polynomial-time adversary can distinguish between $\mathbf{G}_0$ and $\mathbf{G}_1$ with a non-negligible advantage.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $\mathbf{G}_0$ and $\mathbf{G}_1$ 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 $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}, z = g_1^{z'}, H, \Omega = e(g_1, g_1)^{\alpha}\big)$.
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

$$CT^* = \big(C = H(e(T,g)^{\alpha}) \cdot M_{\mu}^*,\ C_0 = T,\ C_1 = (T)^{u'GL^* + h'},\ C_2 = (T)^{w'ML^* + v'},\ C_3 = (T)^{z'}\big).$$

If $T = X_1$, this is a normal ciphertext. If $T = X_1 R_1$, this is a semi-functional ciphertext since $\tau \equiv \log_{g_2}(R_1) \mod p_2, \eta_1 \equiv u' \mod p_2, \eta_2 \equiv h' \mod p_2, \theta_1 \equiv w' \mod p_2, \theta_2 \equiv v' \mod p_2$, and $\kappa \equiv z' \mod p_2$ are not correlated with their values modulo p_1 by CRT.
Query 2: Same as Query 1.
Guess: $\mathcal{A}$ outputs a guess μ'. If $\mu = \mu'$, then $\mathcal{B}$ outputs 1. Otherwise, it outputs 0. □

Lemma 4.7. *If the GSD and 3DH assumptions hold, then no polynomial-time adversary can distinguish between $\mathbf{G}_{1,k-1}$ and $\mathbf{G}_{1,k}$ 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. This algorithm first creates a normal private key $SK'_{GL,ML} = (K'_0, K'_1, K'_2, K'_3, K'_4)$ by using MK. Next, it chooses random exponents $\gamma_1, \gamma_2, \gamma_3 \in \mathbb{Z}_N$ and outputs a nominal semi-functional private key $SK_{GL,ML} = \big(K_0 = K'_0 g_2^{(\eta_1 GL+\eta_2)\gamma_1+\theta_1\gamma_2+\kappa\gamma_3}, K_1 = K'_1 g_2^{(\theta_1 ML+\theta_2)\gamma_2}, K_2 = K'_2 g_2^{-\gamma_1}, K_3 = K'_3 g_2^{-\gamma_2}, K_4 = K'_4 g_2^{-\gamma_3}\big)$.

SRE.GenKeyTSF. This algorithm first creates a normal private key $SK'_{GL,ML} = (K'_0, K'_1, K'_2, K'_3, K'_4)$ by using MK. Next, it chooses a random element $R \in \mathbb{G}_{p_3}$ and outputs a temporary semi-functional private key $SK_{GL,ML} = \big(K_0 = K'_0 R, K_1 = K'_1 g_2^{(\theta_1 ML+\theta_2)\gamma_2}, K_2 = K'_2 g_2^{-\gamma_1}, K_3 = K'_3 g_2^{-\gamma_2}, K_4 = K'_4 g_2^{-\gamma_3}\big)$.

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

To argue the indistinguishability between two games $\mathbf{H}_{k-1,1}$ and $\mathbf{H}_{k-1,2}$, 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 $ML = ML^*$.

Let $\mathbf{Adv}_{\mathcal{A}}^{H_j}$ be the advantage of $\mathcal{A}$ in the game $\mathbf{H}_j$. Through the following Claims 4.8, 4.9, 4.10, 4.11, and 4.12, we obtain the following equation

$$\begin{aligned}
&\big|\mathbf{Adv}_{\mathcal{A}}^{G_{1,k-1}} - \mathbf{Adv}_{\mathcal{A}}^{G_{1,k}}\big| = \big|\mathbf{Adv}_{\mathcal{A}}^{H_{k-1,0}} - \mathbf{Adv}_{\mathcal{A}}^{H_{k-1,3}}\big| \\
&\le \big|\mathbf{Adv}_{\mathcal{A}}^{H_{k-1,0}} - \mathbf{Adv}_{\mathcal{A}}^{H_{k-1,1}}\big| + \big|\mathbf{Adv}_{\mathcal{A}}^{H_{k-1,1}} - \mathbf{Adv}_{\mathcal{A}}^{H_{k-1,2}}\big| + \big|\mathbf{Adv}_{\mathcal{A}}^{H_{k-1,2}} - \mathbf{Adv}_{\mathcal{A}}^{H_{k-1,3}}\big| \\
&\le \mathbf{Adv}_{\mathcal{B}}^{GSD}(\lambda) + \mathbf{Adv}_{\mathcal{B}}^{3DH}(\lambda) + \mathbf{Adv}_{\mathcal{B}}^{GSD}(\lambda).
\end{aligned}$$

This completes our proof. □

Claim 4.8. *If the GSD assumption holds, then no polynomial-time adversary can distinguish between $\boldsymbol{H}_{k-1,0}$ and $\boldsymbol{H}_{k-1,1}$ with a non-negligible advantage.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $\mathbf{H}_{k-1,0}$ and $\mathbf{H}_{k-1,1}$ with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the GSD assumption using $\mathcal{A}$ is given: a challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_3, X_1R_1, R_2Y_1)$ and T where $T = X_2Y_2$ or $T = X_2R_3Y_2$. 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 $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}, z = g_1^{z'}, \Omega = e(g_1, g_1)^{\alpha}\big)$.

Query 1: To response the jth 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_2Y_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

$$\begin{aligned}
&K_0 = g^{\alpha}(T)^{(u'GL+h')r'_1+w'r'_2+z'r'_3}Y'_0,\ K_1 = (T)^{(w'ML+v')r'_1}Y'_1,\ K_2 = (T)^{-r'_1}Y'_2, \\
&K_3 = (T)^{-r'_2}Y'_3,\ K_4 = (T)^{-r'_3}Y'_4.
\end{aligned}$$

If $T = X_2Y_2$, this is a normal private key. If $T = X_2R_3Y_2$, this is a nominally semi-functional private key since $\gamma_1 \equiv \log_{g_2}(R_3)r'_1 \mod p_2, \gamma_2 \equiv \log_{g_2}(R_3)r'_2 \mod p_2, \gamma_3 \equiv \log_{g_2}(R_3)r'_3 \mod p_2, \eta_1 \equiv u' \mod p_2, \eta_2 \equiv h' \mod p_2, \theta_1 \equiv w' \mod p_2, \theta_2 \equiv v' \mod p_2$, and $\kappa \equiv z' \mod p_2$.

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 $CT^* = \big(C = H(e(X_1R_1, g)^\alpha) \cdot M^*_\mu,\ C_0 = X_1R_1,\ C_1 = (X_1R_1)^{u'GL^*+h'},\ C_2 = (X_1R_1)^{w'ML^*+v'},\ C_3 = (X_1R_1)^{z'}\big)$.

Query 2: Same as Query 1.

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

Claim 4.9. *No polynomial-time Type-A adversary can distinguish between $\mathbf{H}_{k-1,1}$ and $\mathbf{H}_{k-1,2}$ with a non-negligible advantage.*

Proof. To argue that any Type-A adversary cannot distinguish the nominally semi-functional private key from the semi-functional private key, we use the restriction of the Type-A adversary that he queries a private key for (GL, ML) such that $GL \neq GL^*$. Suppose there exists an unbounded Type-A adversary. This adversary can gather $(\eta_1 GL + \eta_2)\gamma_1 + \theta_1\gamma_2 + \kappa\gamma_3 \mod p_2, (\theta_1 ML + \theta_2)\gamma_2 \mod p_2, -\gamma_1 \mod p_2, -\gamma_2 \mod p_2, -\gamma_3 \mod p_2$ from the kth private key and $\tau \mod p_2, (\eta_1 GL^* + \eta_2)\tau \mod p_2, (\theta_1 ML^* + \theta_2)\tau \mod p_2, \kappa\tau \mod p_2$ from the challenge ciphertext. If $GL \neq GL^*$, then $\eta_1 GL + \eta_2 \mod p_2$, $\eta_1 GL^* + \eta_2 \mod p_2$ look random to the adversary since $\eta_1 x + \eta_2$ is a pair-wise independent function and $\eta_1 \mod p_2, \eta_2 \mod p_2$ are information theoretically hidden to the adversary. □

Claim 4.10. *If the 3DH assumption in a subgroup holds, then no polynomial-time Type-B adversary that requests the kth query in Query 1 can distinguish between $\mathbf{H}_{k-1,1}$ and $\mathbf{H}_{k-1,2}$ with a non-negligible advantage.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $\mathbf{H}_{k-1,1}$ and $\mathbf{H}_{k-1,2}$ with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the 3DH assumption in a subgroup using $\mathcal{A}$ is given: a challenge tuple $D = ((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_2, g_3, g_2^a, g_2^b, g_2^c)$ and T where $T = g_2^{abc}$ or $T = g_2^{abc+f}$. 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 $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}, z = g_1^{z'}, H, \Omega = e(g_1, g_1)^\alpha\big)$.

Query 1: To response the jth private key query, $\mathcal{B}$ proceeds as follows: If $j < k$, then it creates a semi-functional private key by using MK and g_2. If $j > k$, then it creates a normal private key by using MK. If $j = k$, then it first creates a normal private key $SK' = (K'_0, K'_1, K'_2, K'_3, K'_4)$. Let (GL^*, ML^*) be the labels of the kth private key since the adversary is Type-B. It implicitly sets $\eta_1 = ab + \tilde{\eta}_1, \eta_2 = -abGL^* + \tilde{\eta}_2, \theta_1 = ab + \tilde{\theta}_1, \theta_2 = -abML^* + \tilde{\theta}_2, \kappa = a + \tilde{\kappa}$ by selecting random exponents $\tilde{\eta}_1, \tilde{\eta}_2, \tilde{\theta}_1, \tilde{\theta}_2, \tilde{\kappa} \in \mathbb{Z}_N$. Note that it can only create g_2^κ. Next, it selects random $\gamma_1, \gamma'_2, \gamma'_3 \in \mathbb{Z}_N$ and creates a private key $SK_{(GL^*, ML^*)}$ by implicitly setting $\gamma_2 = c + \gamma'_2, \gamma_3 = -b\gamma'_2 + \gamma'_3$ as

$$K_0 = K'_0 \cdot g_2^{(\tilde{\eta}_1 GL^* + \tilde{\eta}_2)\gamma_1} T (g_2^c g_2^{\gamma'_2})^{\tilde{\theta}_1} (g_2^a)^{\gamma'_3} (g_2^{-\gamma'_2} g_2^{\gamma'_3})^{\tilde{\kappa}},\ K_1 = K'_1 \cdot (g_2^c)^{(\tilde{\theta}_1 ML^* + \tilde{\theta}_2)} g_2^{(\tilde{\theta}_1 ML^* + \tilde{\theta}_2)\gamma'_2},$$
$$K_2 = K'_2 \cdot g_2^{-\gamma_1},\ K_3 = K'_3 \cdot (g_2^c)^{-1} g_2^{-\gamma'_2},\ K_4 = K'_4 \cdot (g_2^b)^{\gamma'_2} g_2^{-\gamma'_3}.$$

If $T = g_2^{abc}$, this is a nominal semi-functional private key since $(\eta_1 GL^* + \eta_2)\gamma_1 = (\tilde{\eta}_1 GL^* + \tilde{\eta}_2)\gamma_1$, $\theta_1\gamma_2 + \kappa\gamma_3 = (ab + \tilde{\theta}_1)(c + \gamma'_2) + (a + \tilde{\kappa})(-b\gamma'_2 + \gamma'_3) = abc + \tilde{\theta}_1(c + \gamma'_2) + a\gamma'_3 + \tilde{\kappa}(-b\gamma'_2 + \gamma'_3)$, and $(\theta_1 ML^* + \theta_2)\gamma_2 = (\tilde{\theta}_1 ML^* + \tilde{\theta}_2)(c + \gamma'_2)$. If $T = g_2^{abc+f}$, then this is a temporary semi-functional private key. Note that it cannot create a private key for (GL, ML) such that $ML \neq ML^*$ since it cannot compute K_1 without directly knowing g_2^{abc}.

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 normal ciphertext $CT' = (C', C_0', C_1', C_2', C_3')$ for M_μ^*. Next, it chooses a random exponent $\tau \in \mathbb{Z}_N$ and creates a semi-functional ciphertext CT^* as

$$C = C',\ C_0 = C_0' \cdot g_2^{\tau},\ C_1 = C_1' \cdot g_2^{(\tilde{\eta}_1 GL^* + \tilde{\eta}_2)\tau},\ C_2 = C_2' \cdot g_2^{(\tilde{\theta}_1 ML^* + \tilde{\theta}_2)\tau},\ C_3 = C_3' \cdot g_2^{\kappa\tau}.$$

Query 2: $\mathcal{B}$ creates normal private keys by using MK.
Guess: $\mathcal{A}$ outputs a guess μ'. If $\mu = \mu'$, then $\mathcal{B}$ outputs 1. Otherwise, it outputs 0. □

Claim 4.11. *If the 3DH assumption in a subgroup holds, then no polynomial-time Type-B adversary that requests the kth query in Query 2 can distinguish between $\boldsymbol{H}_{k-1,1}$ and $\boldsymbol{H}_{k-1,2}$ with a non-negligible advantage.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $\mathbf{H}_{k-1,1}$ and $\mathbf{H}_{k-1,2}$ with a non-negligible advantage. A simulator $\mathcal{B}$ that solves the augmented 3DH assumption in a subgroup using $\mathcal{A}$ is given: a challenge tuple $D = \big((N, \mathbb{G}, \mathbb{G}_T, e), g_1, g_2, g_3, g_2^a, g_2^b, g_2^c\big)$ and T where $T = g_2^{abc}$ or $T = g_2^{abc+f}$. 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 $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}, z = g_1^{z'}, H, \Omega = e(g_1, g_1)^{\alpha}\big)$.
Query 1: To response the private key queries, $\mathcal{B}$ creates semi-functional private keys by using MK and g_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 normal ciphertext $CT' = (C', C_0', C_1', C_2', C_3')$ for M_μ^*. It implicitly sets $\eta_1 = ab + \tilde{\eta}_1, \eta_2 = -abGL^* + \tilde{\eta}_2, \theta_1 = ab + \tilde{\theta}_1, \theta_2 = -abML^* + \tilde{\theta}_2, \kappa = a + \tilde{\kappa}$ by selecting random exponents $\tilde{\eta}_1, \tilde{\eta}_2, \tilde{\theta}_1, \tilde{\theta}_2, \tilde{\kappa} \in \mathbb{Z}_N$. Note that it can only create g_2^{κ}. Next, it chooses a random exponent $\tau \in \mathbb{Z}_N$ and creates a semi-functional ciphertext CT^* as

$$C = C',\ C_0 = C_0' \cdot g_2^{\tau},\ C_1 = C_1' \cdot g_2^{(\tilde{\eta}_1 GL^* + \tilde{\eta}_2)\tau},\ C_2 = C_2' \cdot g_2^{(\tilde{\theta}_1 ML^* + \tilde{\theta}_2)\tau},\ C_3 = C_3' \cdot g_2^{\kappa\tau}.$$

Query 2: To response the jth private key query, $\mathcal{B}$ proceeds as follows: If $j < k$, then it creates a semi-functional private key by using MK and g_2. If $j > k$, then it creates a normal private key by using MK. Let (GL^*, ML^*) be the labels of the kth private key since the adversary is Type-B. If $j = k$, then it first creates a normal private key $SK' = (K_0', K_1', K_2', K_3', K_4')$. Next, it selects random $\gamma_1, \gamma_2', \gamma_3' \in \mathbb{Z}_N$ and creates a private key $SK_{(GL^*, ML^*)}$ by implicitly setting $\gamma_2 = c + \gamma_2', \gamma_3 = -b\gamma_2' + \gamma_3'$ as

$$K_0 = K_0' \cdot g_2^{(\tilde{\eta}_1 GL^* + \tilde{\eta}_2)\gamma_1} T (g_2^c g_2^{\gamma_2'})^{\tilde{\theta}_1} (g_2^a)^{\gamma_3'} (g_2^{-\gamma_2'} g_2^{\gamma_3'})^{\tilde{\kappa}},\ K_1 = K_1' \cdot (g_2^c)^{(\tilde{\theta}_1 ML^* + \tilde{\theta}_2)} g_2^{(\tilde{\theta}_1 ML^* + \tilde{\theta}_2)\gamma_2'},$$
$$K_2 = K_2' \cdot g_2^{-\gamma_1},\ K_3 = K_3' \cdot (g_2^c)^{-1} g_2^{-\gamma_2'},\ K_4 = K_4' \cdot (g_2^b)^{\gamma_2'} g_2^{-\gamma_3'}.$$

If $T = g_2^{abc}$, this is a nominal semi-functional private key since $(\eta_1 GL^* + \eta_2)\gamma_1 = (\tilde{\eta}_1 GL^* + \tilde{\eta}_2)\gamma_1$, $\theta_1\gamma_2 + \kappa\gamma_3 = (ab + \tilde{\theta}_1)(c + \gamma_2') + (a + \tilde{\kappa})(-b\gamma_2' + \gamma_3') = abc + \tilde{\theta}_1(c + \gamma_2') + a\gamma_3' + \tilde{\kappa}(-b\gamma_2' + \gamma_3')$, and $(\theta_1 ML^* + \theta_2)\gamma_2 = (\tilde{\theta}_1 ML^* + \tilde{\theta}_2)(c + \gamma_2')$. If $T = g_2^{abc+f}$, then this is a temporary semi-functional private key. Note that it cannot create a private key for (GL, ML) such that $ML \neq ML^*$ since it cannot compute K_1 without directly knowing g_2^{abc}.
Guess: $\mathcal{A}$ outputs a guess μ'. If $\mu = \mu'$, then $\mathcal{B}$ outputs 1. Otherwise, it outputs 0. □

Claim 4.12. *If the GSD assumption holds, then no polynomial-time adversary can distinguish between $\boldsymbol{H}_{k-1,2}$ and $\boldsymbol{H}_{k-1,3}$ with a non-negligible advantage.*

Proof. The proof of this claim is almost the same as that of Claim 4.8 except the generation of the kth private key. The kth private key for (GL, ML) is generated as follows: If $j = k$, then it selects random $r'_1, r'_2, r'_3, a' \in \mathbb{Z}_N$, $Y'_0, Y'_1, Y'_2, Y'_3, Y'_4 \in \mathbb{G}_{p_3}$ and creates a private key $SK_{(GL,ML)}$ as

$$K_0 = g^{\alpha}(T)^{(u'GL+h')r'_1+w'r'_2+z'r'_3}(R_2Y_1)^{a'}Y'_0,\ K_1 = (T)^{(w'ML+v')r'_1}Y'_1,\ K_2 = (T)^{-r'_1}Y'_2,$$
$$K_3 = (T)^{-r'_2}Y'_3,\ K_4 = (T)^{-r'_3}Y'_4.$$

Note that the kth private key is no longer correlated with CT^* since K_0 is re-randomized by $(R_2Y_1)^{a'}$. If $T = X_2Y_2$, this is a semi-functional private key. If $T = X_2R_3Y_2$, this is a temporary semi-functional private key. □

Lemma 4.13. *If the ComDH assumption holds, then no polynomial-time adversary can distinguish between* $\mathbf{G}_2$ *and* $\mathbf{G}_3$ *with a non-negligible advantage.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguish $\mathbf{G}_2$ from $\mathbf{G}_3$ 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 $PP = \big((N, \mathbb{G}, \mathbb{G}_T, e), g = g_1, Y = g_3, u = g_1^{u'}, h = g_1^{h'}, w = g_1^{w'}, v = g_1^{v'}, z = g_1^{z'}, H, \Omega = e(g_1, g_1^a R_1)\big)$.

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 $CT^* = \big(C = H(T) \cdot M^*_\mu,\ C_0 = g_1^b R_2,\ C_1 = (g_1^b R_2)^{u'GL^*+h'},\ C_2 = (g_1^b R_2)^{w'ML^*+v'},\ C_3 = (g_1^b R_2)^{z'}\big)$.

Query 2: Same as Phase 1.

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

5 Identity-Based Revocation from SD

In this section, we propose an IBR scheme by combining the SD method with an SRE scheme and prove its security. For the construction of an IBR scheme, we follow the design principle of Lee et al. [17]. Compared to the scheme of Lee et al., our scheme is an identity-based one whereas their scheme is a public-key based one.

5.1 Subset Difference Scheme

The subset difference (SD) scheme is one instance of the subset cover framework proposed by Naor et al. [23]. 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 set of subsets 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 the set of revoked users R, it first finds cover CV that is a set of subsets that can cover all users $\mathcal{N} \setminus \mathcal{R}$ and creates ciphertexts for each subset by using their unique key. A receiver can decrypt the ciphertext if he is not revoked in the revoked set R.

Before we describe the SD scheme, we define some notation. Let $\mathcal{T}$ be a full binary tree and v_i be a node in $\mathcal{T}$. The depth d_i of a node v_i is the length of the path from the root node to the node where the root node is at depth zero. A level of $\mathcal{T}$ is a set of all nodes at given depth. For any node $v_i \in \mathcal{T}$, T_i is defined as a subtree that is rooted at v_i. For any two nodes $v_i, v_j \in \mathcal{T}$ such that v_j is a descendant of v_i, $T_{i,j}$ is defined as a subtree $T_i - T_j$, that is, all nodes that are descendants of v_i but not v_j. For any node $v_i \in \mathcal{T}$, S_i is defined as the set of leaf nodes in T_i. Similarly, $S_{i,j}$ is defined as the set of leaf nodes in $T_{i,j}$, that is, $S_{i,j} = S_i \setminus S_j$. For any node $v_i \in \mathcal{T}$, 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 the edge is connected to its left or right child node. The label L_i of $v_i \in \mathcal{T}$ 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 T_i, we define the label of T_i as the label L_i of v_i where v_i is the root node of T_i. For a subtree $T_{i,j}$, we also define the label of $T_{i,j}$ as labels (L_i, L_j) where L_i, L_j are labels of v_i, v_j of $T_{i,j}$. Similarly, we can define the label of S_i as the same as that of T_i and the label of $S_{i,j}$ as the same as that of $T_{i,j}$.

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. [17] 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 $\mathcal{T}$ and the collection $\mathcal{S}$ of subsets where each subset $S_{i,j}$ is associated with a subtree $T_{i,j}$ in $\mathcal{T}$. The assign algorithm first assigns a user to a leaf node of $\mathcal{T}$ and defines a path set PV of subsets 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 $T_{i,j}$ in PV. The cover algorithm takes as input a set of revoked users R and finds a cover set CV of subsets that can cover the set of receivers $\mathcal{N} \setminus R$. The final match algorithm takes as input a path 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 in Appendix A.

5.2 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. [17]. 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 $T_{i,j}$ and d_j is the depth of the node v_j since the subtree $T_{i,j}$ 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. [17], but we slightly modify it to use a symmetric key encryption scheme in order to improve the efficiency.

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

IBR.Setup$(1^\lambda, n)$**:** This algorithm first define a full binary tree $\mathcal{T}$ by running **SD.Setup**(2^n). Next, it obtains MK_{SRE} and PP_{SRE} by running **SRE.Setup**(1^λ). It outputs a master key $MK = MK_{SRE}$ and public parameters $PP = (\mathcal{T}, PP_{SRE})$.

IBR.GenKey(ID, MK, PP)**:** This algorithm takes as input an identity $ID \in \mathcal{I}$, the master key MK, and the public parameters PP. It first obtains a private set $PV_{ID} = \{S_{i,j}\}$ by running **SD.Assign**$(\mathcal{T}, ID)$. 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 a private key $SK_{ID} = \big(PV_{ID}, \{SK_{SRE,S_{i,j}}\}_{S_{i,j} \in PV_{ID}}\big)$.

IBR.Encrypt(R, M, PP): This algorithm takes as input a revoked set of users R, a message $M \in \{0,1\}^m$, and the public parameters PP. It first finds a covering set $CV_R = \{S_{i,j}\}$ by running **SD.Cover**($\mathcal{T}, R$). 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 an encrypted message C by running **SKE.Encrypt**(K, M). Finally, it outputs a ciphertext $CT_R = \big(CV_R, C, \{CT_{SRE,S_{i,j}}\}_{S_{i,j} \in CV_R}\big)$.

IBR.Decrypt(CT_R, SK_{ID}, PP): This algorithm takes as input a ciphertext CT_R, a private key SK_{ID}, and the public parameters 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 a session key K by running **SRE.Decrypt**$(CT_{SRE,S_{i,j}}, SK_{SRE,S'_{i,j}}, PP_{SRE})$. Otherwise, it outputs $\perp$. Finally, it outputs a message M by running **SKE.Decrypt**(K, C).

Remark 5.1. *Our revocation scheme is identity-based one whereas the revocation scheme of Lee et al. [17] 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. Another difference is that our IBR scheme encrypts a session key by using an SRE scheme and this session key is used to encrypt a message by using a symmetric-key encryption scheme.*

5.3 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 5.2. *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 ℓ. 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 $\mathbf{G}_0, \mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ as follows:

Game $\mathbf{G}_0$ This game is the original security game defined in the security model except that the challenge bit μ 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 $\mathbf{G}_1$ 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 $\mathbf{G}_{0,0}, \ldots, \mathbf{G}_{0,\rho}, \ldots, \mathbf{G}_{0,\ell}$ where $\mathbf{G}_{0,0} = \mathbf{G}_0$ and $\mathbf{G}_{0,\ell} = \mathbf{G}_1$. The game $\mathbf{G}_{0,\rho}$ is almost identical to the game $\mathbf{G}_{0,\rho-1}$ 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 $\mathbf{G}_2$ This game is similar to the game $\mathbf{G}_1$ 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 $\mathbf{G}_3$ 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 μ is fixed to 1.

For the security proof, we also define additional hybrid games $\mathbf{G}_{2,0},\dots,\mathbf{G}_{2,\rho},\dots,\mathbf{G}_{2,\ell}$ where $\mathbf{G}_{2,0} = \mathbf{G}_2$ and $\mathbf{G}_{2,\ell} = \mathbf{G}_3$. The game $\mathbf{G}_{2,\rho}$ is almost identical to the game $\mathbf{G}_{2,\rho-1}$ except that $CT^*_{SRE,\rho}$ is an encryption on the correct session key K^*. Specifically, each component $CT^*_{SRE,k}$ for $k \le \rho$ 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 $S^{G_i}_{\mathcal{A}}$ be the event that $\mathcal{A}$ outputs 0 in $\mathbf{G}_i$. From Lemmas 5.3, 5.4 and 5.5, we obtain the following result

$$\begin{aligned}
\mathbf{Adv}^{IBR}_{\mathcal{A}}(\lambda) &\le \frac{1}{2}\left|\Pr[S^{G_0}_{\mathcal{A}}] - \Pr[S^{G_3}_{\mathcal{A}}]\right| \\
&\le \frac{1}{2}\Big(\sum_{\rho=1}^{\ell}\left|\Pr[S^{G_{0,\rho-1}}_{\mathcal{A}}] - \Pr[S^{G_{0,\rho}}_{\mathcal{A}}]\right| + \left|\Pr[S^{G_1}_{\mathcal{A}}] - \Pr[S^{G_2}_{\mathcal{A}}]\right| + \sum_{\rho=1}^{\ell}\left|\Pr[S^{G_{2,\rho-1}}_{\mathcal{A}}] - \Pr[S^{G_{2,\rho}}_{\mathcal{A}}]\right|\Big) \\
&\le \ell\mathbf{Adv}^{SRE}_{\mathcal{B}}(\lambda) + \mathbf{Adv}^{SKE}_{\mathcal{B}}(\lambda) + \ell\mathbf{Adv}^{SRE}_{\mathcal{B}}(\lambda).
\end{aligned}$$

This completes our proof. □

Lemma 5.3. *If the SRE scheme is secure under chosen plaintext attacks, then no polynomial time adversary can distinguish between $\mathbf{G}_{0,\rho-1}$ and $\mathbf{G}_{0,\rho}$ with non-negligible advantage.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $\mathbf{G}_{0,\rho-1}$ and $\mathbf{G}_{0,\rho}$ 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 full binary tree $\mathcal{T}$ by running $\mathbf{SD.Setup}(2^n)$ and gives $PP = (\mathcal{T}, PP_{SRE})$ 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 $\mathbf{SD.Assign}(\mathcal{T}, ID)$. 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 $\mathbf{SKE.GenKey}(1^\lambda)$ and computes C^* by running $\mathbf{SKE.Encrypt}(M^*_\mu, K^*)$. Next, it obtains a covering set $CV_{R^*} = \{S_{i_1,j_1},\dots,S_{i_\ell,j_\ell}\}$ by running $\mathbf{SD.Cover}(\mathcal{T}, R^*)$ and obtains each component $CT_{SRE,S_{i_k,j_k}}$ as follows:

1. For $1 \le k \le \rho - 1$, it computes $CT^*_{SRE,S_{i_k,j_k}}$ by running $\mathbf{SRE.Encrypt}((L_{i_j} \| d_{j_k}, L_{j_k}), Z, PP_{SRE})$ 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 $\rho + 1 \le k \le \ell$, it computes $CT^*_{SRE,S_{i_k,j_k}}$ by running $\mathbf{SRE.Encrypt}((L_{i_k} \| d_{j_k}, L_{j_k}), K^*, PP_{SRE})$ 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 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 μ'. $\mathcal{B}$ also outputs μ'. □

Lemma 5.4. *If the SKE scheme is secure under chosen plaintext attacks, then no polynomial time adversary can distinguish between* $\mathbf{G}_1$ *and* $\mathbf{G}_2$ *with non-negligible advantage.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes between $\mathbf{G}_1$ and $\mathbf{G}_2$ 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**$(1^{\lambda}, n)$ 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 μ'. $\mathcal{B}$ also outputs μ'. □

Lemma 5.5. *If the SRE scheme is secure under chosen plaintext attacks, then no polynomial time adversary can distinguish between* $\mathbf{G}_{2,\rho-1}$ *and* $\mathbf{G}_{2,\rho}$ *with non-negligible advantage.*

The proof of this Lemma is almost similar to that of Lemma 5.3.

6 Conclusion

In this paper, we solved the problem of Lee et al. [17] to construct an SRE scheme under the standard assumption without random oracles. The important 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 [19]. Finally, we proposed an IBR scheme by combining the SD method and our SRE scheme and proved its security.

We left the following as an interesting problem. One drawback of an IBR scheme derived from the SD (or LSD) method and an SRE scheme is that the number of group elements in private keys is $O(\log^2 N)$ (or $O(\log^{1.5} N)$) where N is the maximum number of users in the system. In symmetric-key revocation systems derived from the subset cover framework, the size of private keys can be constant by increasing the cost of key derivation operations [1]. Thus, it is an interesting problem to construct an IBR scheme derived from the SD method that has shorter private keys.

References

[1] Tomoyuki Asano. A revocation scheme with minimal storage at receivers. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 433–450. Springer, 2002.

[2] Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In Phong Q. Nguyen and David Pointcheval, editors, *Public-Key Cryptography - PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 384–402. Springer, 2010.

[3] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[4] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[5] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.

[6] Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 211–220. ACM, 2006.

[7] Dan Boneh, Brent Waters, and Mark Zhandry. Low overhead broadcast encryption from multilinear maps. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014*, volume 8616 of *Lecture Notes in Computer Science*, pages 206–223. Springer, 2014.

[8] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In Vijay Atluri, Catherine Meadows, and Ari Juels, editors, *ACM Conference on Computer and Communications Security*, pages 320–329. ACM, 2005.

[9] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.

[10] Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2007.

[11] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *DRM 2002*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2002.

[12] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.

[13] Sanjam Garg, Abishek Kumarasubramanian, Amit Sahai, and Brent Waters. Building efficient fully collusion-resilient traitor tracing and revocation schemes. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 121–130. ACM, 2010.

[14] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.

[15] Michael T. Goodrich, Jonathan Z. Sun, and Roberto Tamassia. Efficient tree-based revocation in groups of low-state devices. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 511–527. Springer, 2004.

[16] Dani Halevy and Adi Shamir. The lsd broadcast encryption scheme. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2002.

[17] Kwangsu Lee, Woo Kwon Koo, Dong Hoon Lee, and Jong Hwan Park. Public-key revocation and tracing schemes with subset difference methods revisited. In Miroslaw Kutylowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014*, volume 8713 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2014.

[18] Kwangsu Lee and Dong Hoon Lee. Adaptively secure broadcast encryption under standard assumptions with better efficiency. *IET Inf. Secur.*, 9(3):149–157, 2015.

[19] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, 2012.

[20] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *IEEE Symposium on Security and Privacy*, pages 273–285. IEEE Computer Society, 2010.

[21] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In Daniele Micciancio, editor, *Theory of Cryptography - TCC 2010*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.

[22] Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 180–198. Springer, 2012.

[23] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 41–62. Springer, 2001.

[24] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *Financial Cryptography - FC 2000*, volume 1962 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2000.

[25] Jong Hwan Park, Hee Jean Kim, H.-M. Sung, and Dong Hoon Lee. Public key broadcast encryption schemes with shorter transmissions. *IEEE Trans. Broadcast.*, 54(3):401–411, 2008.

[26] Ryuichi Sakai and Jun Furukawa. Identity-based broadcast encryption. Cryptology ePrint Archive, Report 2007/217, 2007. `http://eprint.iacr.org/2007/217`.

[27] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.

[28] Mark Zhandry. Adaptively secure broadcast encryption with small system parameters. Cryptology ePrint Archive, Report 2014/757, 2014. `http://eprint.iacr.org/2014/757`.

A SD Scheme

We describe the SD scheme that is defined in [17] for the completeness of our paper.

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

SD.Assign($\mathcal{T}, ID$)**:** This algorithm takes as input the tree $\mathcal{T}$ and a user ID. Let v_{ID} be the leaf node of $\mathcal{T}$ that is assigned to the user ID. Let $(v_{k_0}, v_{k_1}, \dots, v_{k_n})$ 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 $i, j \in \{k_0, \dots, k_n\}$ 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_u. It outputs the private set $PV_{ID} = \{S_{i,j}\}$.

SD.Cover($\mathcal{T}, R$)**:** This algorithm takes as input the tree $\mathcal{T}$ and a revoked set R of users. 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})**:** This algorithm takes input as a covering set $CV_R = \{S_{i,j}\}$ and a private set $PV_{ID} = \{S'_{i,j}\}$. It finds two subsets $S_{i,j}$ and $S'_{i',j'}$ such that $S_{i,j} \in CV_R$, $S'_{i',j'} \in PV_{ID}$, $i = i'$, $d_j = d_{j'}$, and $j \neq j'$ 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 $\perp$.