ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

New Constructions of Revocable Identity-Based Encryption from Multi linear Maps

[1]SRIRAM PRADHAN, *Gandhi Institute of Excellent Technocrats, Bhubaneswar, India*
[2]SUPRIYA PRIYADARSINI SAHOO, *Rayagada Institute of Technology and Management, Rayagada, Odisha, India*

***Abstract*—A revocable identity-based encryption (RIBE) provides an efficient revocation method in IBE that a trusted authority periodically broadcasts an update key for nonrevoked users and a user can decrypt a ciphertext if he is not revoked in the update key. Boldyreva, Goyal, and Kumar (CCS 2008) defined RIBE and proposed an RIBE scheme that uses a treebased revocation encryption scheme to revoke users' private keys. In this paper, we devise a new technique for RIBE and propose RIBE schemes with a constant number of private key elements. We achieve the following results. We first devise a new technique for RIBE that combines a hierarchical IBE (HIBE) scheme and a public-key broadcast encryption (PKBE) scheme using multilinear maps. In contrast to the previous technique for RIBE, our technique uses a PKBE scheme in bilinear maps for revocation to achieve short private keys and update keys. Following our new technique for RIBE, we propose an RIBE scheme in three-leveled multilinear maps that combines the HIBE scheme of Boneh and Boyen (EUROCRYPT 2004) and the PKBE scheme of Boneh, Gentry, and Waters (CRYPTO 2005). The private key and update key of our scheme possess a constant number of group elements. Next, we propose another RIBE scheme with reduced public parameters and short keys by combining the HIBE scheme of Boneh and Boyen and the PKBE scheme of Boneh, Waters, and Zhandry (CRYPTO 2014), which uses multilinear maps. Compared with our first RIBE scheme, our second RIBE scheme requires high-leveled multilinear maps.**

***Index Terms*—Identity-based encryption, key revocation, broadcast encryption, multilinear maps.**

I. INTRODUCTION

PROVIDING an efficient revocation mechanism in cryptosystems for a large number of users is very important since it can prevent a user from accessing sensitive data in cryptosystems by revoking the private key of a user when the private key is revealed or expired. In public-key encryption (PKE), which employs the public-key infrastructure (PKI), there are many studies that deal with the certificate revocation problem [1]–[4]. In identity-based encryption (IBE) [5], [6], a natural approach for this revocation problem is that atrusted authority periodically renews a user's private key for his identity at a current time period and then a sender creates a ciphertext for both a receiver identity and a current time period. However, this approach has some problems: the trusted authority should always be online to renew the user's private keys, users should always renew their private keys regardless of whether their private keys are revoked, and a secure channel should be established between the trusted authority and a user to transmit a renewed private key.

An IBE scheme that provides an efficient revocation mechanism (RIBE) was proposed by Boldyreva*et al.* [7]. In RIBE, each user receives a (long-term) private key $SK_{I\,D}$ for his identity $I\,D$ from a trusted authority, and the trusted authority periodically broadcasts an update key $UK_{T,R}$at a current time T by including a revoked identity set R. If a user has a private key $SK_{I\,D}$ that is not revoked by the revoked identity set R of the update key $UK_{T,R}$, then he can derive his (short-term) decryption key $DK_{ID,T}$from his private key $SK_{I\,D}$ and the update key $UK_{T,R}$. This decryption key can be used to decrypt a ciphertext $CT_{I\,D,T}$for a receiver identity $I\,D$ and a time period T. The main advantage of this approach is that the trusted authority can be offline because the authority only need to broadcast the update key periodically. To build an RIBE scheme, Boldyreva*et al.* [7] used the tree-based revocation encryption scheme of Naor*et al.* [8] for revocation and the ABE scheme of Sahai and Waters [9] for encryption of an identity and a time period. Other RIBE schemes also follow this design approach that uses the tree-based revocation encryption scheme for revocation [10]–[12]. This design approach, however, has an inherent limitation in that the number of private key elements and update key elements cannot be constant since a private key is associated with path nodes in a tree and an update key is associated with covering nodes in the tree [8]. Therefore, in this paper, we ask the following questions about RIBE: "Can we build an RIBE scheme with a constant number of private key elements and update key elements? Can we devise a new technique for efficient RIBE that is different from the previous approach?"

A. Our Results

In this work, we give affirmative answers to both of the above questions. That is, we first devise a new technique for RIBE that is quite different from the previous technique, and we propose two RIBE schemes

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

with a constant number of private key elements. The following is our results:

New Techniques for Revocable IBE: Previous RIBE schemes [7], [10], [11] use IBE (or ABE) schemes for the main encryption functionality and the tree-based revocation encryption of Naor*et al.* [8] for the revocation functionality. As mentioned, the inherent limitation of the tree-based revocation encryption scheme is that the number of private key elements and update key elements cannot be constant. To achieve an RIBE scheme with a constant number of private key elements and update key elements, we observe that PKBE schemes [13], [14] in bilinear groups can be directly used for delivering a partial key of IBE to non-revoked users because these broadcast schemes have short private keys and short ciphertexts. That is, the private key $SK_{ID,T}$ of a two-level HIBE scheme with an identity ID and a time period T is divided into two partial keys SK_{ID} and SK_{T}. A user's actual key consists of SK_{ID} and the private key of PKBE, and a trusted authority broadcasts an update key $UK_{T,R}$ that is the encryption of SK_T, which excludes revoked users R. If the user is not revoked, then he can derive $SK_{ID,T}$ of HIBE combining SK_{ID} in his actual key and SK_T in $UK_{T,R}$. However, this simple RIBE scheme is vulnerable under a simple attack–that is, if an adversary corrupts a user ID at time T , then he can obtain a partial key SK_{ID} and a PKBE key for ID. The adversary then can decrypt a previous ciphertext $CT_{ID,T'}$ such that $T' < T$ by obtaining a partial key SK_T from $UK_{T,R}$ since the PKBE key that was obtained at time T can still be applied to decrypt $UK_{T,R}$ at time T. To overcome the simple attack, we set the private key SK of RIBE by binding the private key of HIBE and the private key of PKBE, and set the update key UK of RIBE by binding the private key of HIBE and the ciphertext of PKBE. However, this RIBE scheme possesses another problem–a decryption key derived from a private key and an update key by performing a pairing operation cannot be used to decrypt a ciphertext since the decryption key is the result of the pairing operation in bilinear groups. To solve this new problem, we use multilinear maps that were recently proposed by Garg *et al.* [15]. The detailed techniques are discussed below in this section.

RIBE With Shorter Private Keys and Update Keys: We first propose an RIBE scheme with a constant number of private key elements and update key elements by applying our new technique for RIBE on the three-leveled multilinear maps. For a concrete RIBE construction, we use the PKBE scheme of Boneh, Gentry, and Waters (BGW-PKBE) [13] for revocation and the HIBE scheme of Boneh*et al.* (BB-HIBE) [16] for encryption of an identity ID and a time T. The public parameters, the private key, the update key, and the ciphertext of our RIBE scheme just consist of $O(N + \lambda)$, $O(1)$, $O(1)$, and $O(1)$ group elements, respectively. As far as we know, our RIBE scheme is the first one that achieves a constant number of private key elements and update key elements. To prove the security of our RIBE scheme, we introduce a new complexity assumption called the Multilinear Diffie-Hellman Exponent (MDHE) assumption that is a natural multilinear version of the Bilinear Diffie-Hellman Exponent (BDHE) assumption of Boneh*et al.* [13]. We prove the security of our scheme in the selective revocation list model, where an adversary should initially submit a challenge identity, a challenge time, and the revoked set of identities at the challenge time.

RIBE With Reduced Pubic Parameters: The number of group elements in the public parameters of our first RIBE scheme is proportional to the maximum number of users. To overcome this problem, we propose another RIBE scheme with reduced public parameters on $O(\log N)$ leveled multilinear maps by employing the PKBE scheme of Boneh*et al.* (BWZ-PKBE) [17]. The interesting feature of the BWZ-PKBE scheme is that the public key just consists of $O(\log N)$ group elements whereas the public key of the BGW-PKBE scheme [13] consists of $O(N)$ group elements. Additionally, the BWZ-PKBE scheme has a similar structure to the BGW-PKBE scheme except that it uses $O(\log N)$-leveled multilinear maps. Because of this structural similarity, we can build an RIBE scheme based on the BWZ-PKBE scheme by following our new technique for RIBE. We prove the security of our second RIBE scheme in the selective revocation list model by using the compressed MDHE (cMDHE) assumption. Although the number of group elements in public parameters is reduced, our second RIBE scheme is not a truly identity-based one since the maximum size of the receiver set is restricted to being polynomial in the BWZ-PKBE scheme. A detailed comparison between our RIBE schemes and other RIBE schemes is given in Table I. Note that the bit size of private keys and update keys in our RIBE schemes is not constant since the bit size of group elements in leveled multilinear maps is not constant [32].

B. Our Technique

To devise an RIBE scheme with a constant number of private key elements and update key elements, we use the BGW-PKBE scheme [13] for revocation instead of using the revocation encryption of Naor*et al.* [8]. The revocation encryption of the NNL framework mainly uses a tree for broadcasting, and it is hard to provide a constant number of RIBE private key elements since the private key of the NNL framework is associated with path nodes in the tree and the update key is associated with subset covering nodes in the tree [8]. The BGW-PKBE scheme, by contrast, can provide a constant

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

number of RIBE private key elements since the PKBE scheme has a constant number of private key elements.

For our RIBE construction, we use the BGW-PKBE scheme [13] for revocation and the two-level HIBE scheme of Boneh and Boyen [16] for encryption of an identity ID and a time period T. As mentioned before, the simple approach is vulnerable under a simple attack. To address this problem, we first set the RIBE private key as $SK_{ID} = g^{\alpha d \gamma} F(ID)^{r1}, g^{r1}$, which is a careful combination of the PKBE private key $SK_{BE,d} = g^{\alpha d \gamma}$ and the HIBE private key $SK_{HIBE,ID} = (g^{a} F(ID)^{r1}, g^{r1})$, where an index d is associated with the identity ID and $F(\cdot)$ is a function from identities to group elements. That is, we replace the master key part g^{a} of the HIBE private key component with the PKBE private key component. Next, we set the RIBE update key as $UK_{T,R} = ((g^{\gamma} \prod_{j \in \mathcal{N} \setminus R} g^{\alpha^{N+1-j}})^{\beta} H(T)^{r_2}, g^{r2})$, which is a careful combination of the PKBE ciphertext $CT_{BE,R} = (g_{\gamma,T} = g^{\alpha^{N+1} \beta}, g^{\beta})$ and the HIBE private key $SK_{HIBE,T} = (g^{a} H(T)^{r_2}, g^{r_2})$, where R is a revocation set, T is an update time period, and $H(\cdot)$ is a function from times to group elements. That is, we replace the master key part g^{a} of the HIBE private key component with the PKBE ciphertext component. If a user with a private key SK_{ID} is not revoked in an update key $UK_{T,R}$ at a time T, then he can derive a decryption key $DK_{ID,T} = g^{\alpha^{N+1} \beta} F(ID)^{r1} H(T)^{r2}, g^{r1}, g^{r2}$ for his identity ID and the time T. This decryption key can be used to decrypt a ciphertext $CT_{ID,T} = e(g^{\alpha^{N+1}}, g^{\beta})^{s} \cdot M, g^{s}, F(ID)^{s}, H(T)^{s}$.

However, there is a major problem with this idea. That is, a session key that is derived from the ciphertext and the private key of PKBE in bilinear groups is an element in G_T, and this session key cannot be used for pairing in bilinear groups. This means that the RIBE decryption key $DK_{ID,T}$, which is related with the session key of PKBE, cannot be used to decrypt a RIBE ciphertext $CT_{ID,T}$ since the pairing operation can no longer be applicable. To address this problem, we use three-leveled multilinear maps [15]. Note that bilinear maps correspond to two-leveled multilinear maps. In our RIBE scheme, which uses three-leveled multilinear maps, a private key SK_{ID} is in G_1, an update key $UK_{T,R}$ is in G_1, a decryption key $DK_{ID,T}$ is in G_2, and a ciphertext $CT_{ID,T}$ is in G_1. The ciphertext $CT_{ID,T}$ in G_1 and the decryption key $DK_{ID,T}$ in G_2 can be used to derive a session key by using a bilinear map $e_{1,2}(\cdot,\cdot)$, which is additionally provided by three-leveled multilinear maps. Therefore, we can build an RIBE scheme with a constant number of private key elements and update key elements from three-leveled multilinear maps. This technique also applies to the BWZ-PKBE scheme [17].

TABLE I
COMPARISON OF REVOCABLE IDENTITY-BASED ENCRYPTION SCHEMES

Scheme	PP Size	SK Size	UK Size	Model	Maps	Assumption
BF [6]	$O(1)$	$O(1)$	$O(N-r)$	Full	BLM	RO, BDH
BGK [7]	$O(1)$	$O(\log N)$	$O(r \log(N/r))$	Selective	BLM	DBDH
LV [10]	$O(\lambda)$	$O(\log N)$	$O(r \log(N/r))$	Full	BLM	DBDH
SE [11]	$O(\lambda)$	$O(\log N)$	$O(r \log(N/r))$	Full	BLM	DBDH
LLP [18]	$O(1)$	$O(\log^{1.5} N)$	$O(r)$	Full	BLM	Static
Our-1	$O(N+\lambda)$	$O(1)$	$O(1)$	SelectiveRL	MLM	MDHE
Our-2	$O(\log N+\lambda)$	$O(1)$	$O(1)$	SelectiveRL	MLM	cMDHE

Let λ be a security parameter, N be the maximum number of users, and r be the maximum number of revoked users. Sizes for public parameters (PP), private keys (SK), and update keys (UK) count the number of group elements. BLM stands for bilinear maps and MLM stands for multilinear maps. The bit sizes of elements in bilinear maps, elements in three-leveled multilinear maps, and elements in k-leveled multilinear maps are $O(\lambda)$, $O(\lambda \log^2 \lambda)$, and $O(k^2 \lambda \log^2(k\lambda))$, respectively.

C. Related Work

Identity-Based Encryption and Its Extensions: IBE, introduced by Shamir [5], can solve the key management problem of PKE since it uses an identity string as a public key instead of using a random value. The first IBE scheme was proposed by Boneh and Franklin [6] by using bilinear groups, and many other IBE schemes have been proposed in bilinear maps [16], [19]. Another importance of IBE is that it has many surprising extensions such as hierarchical IBE (HIBE), attribute-based encryption (ABE), predicate encryption (PE), and functional encryption (FE). HIBE was introduced by Horwitz and Lynn [20] and it additionally provides private key delegation functionality [16], [21]. ABE was introduced by Sahai and Waters [9] and it can provide access controls on ciphertexts by associating a ciphertext with attributes and a private key with a policy [22]. PE can provide searches on encrypted data by hiding attributes in ciphertexts [23]. Recently, the concept of FE, which includes all the extensions of IBE was introduced by Boneh *et al.* [24], and it was shown that FE schemes for general circuits can be constructed [25].

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

Revocation in IBE: As mentioned, providing an efficient revocation mechanism that can revoke a user whose private key is revealed is a very important issue in cryptosystems. In PKE, which employs the public-key infrastructure (PKI), the certificate revocation problem was widely studied [1]–[4]. In IBE, there are some works that deal with the key revocation problem [6], [7], [10], [11], [26]. We can categorize the revocation methods for IBE in the following two ways. In the first revocation method, a trusted authority periodically broadcasts a revoked user set R, and a sender creates a ciphertext by additionally including a receiver set S that excludes the revoked user set R [26]. That is, this method conceptually combines an IBE scheme with a PKBE scheme. Though this method is simple to construct and does not require a user to update his private key, the sender should check the validity of the revoked list and the sender has the responsibility for the revocation. Ideally, the sender should proceed as in any IBE scheme and encrypt a message without worrying about potential revoked users.

With the second revocation method, a sender creates a ciphertext for a receiver identity ID and a time T, and a receiver periodically updates his private key on a time T from a trusted authority if he is not revoked at the time T. That is, this method can revoke a user by preventing the user from obtaining his key components from the authority.

Boneh and Franklin [6] proposed a revocable IBE scheme by representing a user's identity as $ID\|T$ with a user periodically receiving his private key at a time T by communicating with the authority. However, this RIBE scheme is impractical for a large number of users since all users should be connected to the authority to receive their private keys. To improve the efficiency of RIBE, Boldyreva*et al.* [7] proposed a new RIBE scheme, in which a trusted authority periodically broadcasts an update key for a time T and non-revoked users by using the revocation encryption of Naor*et al.* [8]. After that, many other RIBE schemes were proposed by following this design principle [10]–[12], [18]. The key revocation is also an important issue in ABE. Sahai *et al.* [27] proposed a revocable-storage ABE (RS-ABE) scheme for cloud storage by extending the idea of RIBE schemes, and Lee *et al.* [28] proposed an improved RS-ABE scheme and a revocable-storage PE scheme.

II. PRELIMINARIES

In this subsection, we first define revocable identity-based encryption (RIBE) and its security model, and then we review multilinear maps and complexity assumptions for our RIBE schemes.

A. Revocable Identity-Based Encryption

Revocable identity-based encryption (RIBE) is an extension of identity-based encryption (IBE) in that a user with an identity ID can be revoked later if his credential is expired [7]. In RIBE, each user receives his (long-term) private key that is associated with an identity ID from a key generation center. After that, the key generation center periodically broadcasts an update key for the non-revoked set of users where the update key is associated with a time T and a revoked set R. If a user is not revoked in the update key, then he can derive his (short-term) decryption key for his identity ID and the current time T from the private key and the update key. Using the decryption key for ID and T, the user can decrypt a ciphertext for a receiver identity ID_c and a time T_c if $ID = ID_c$ and $T = T_c$. The following is the syntax of RIBE.

Definition 2.1 (Revocable IBE): A revocable IBE (RIBE) scheme that is associated with the identity space I, the time space T, and the message space M, consists of seven algorithms **Setup**, **GenKey**, **UpdateKey**, **DeriveKey**, **Encrypt**, **Decrypt**, and **Revoke**, which are defined as follows:

- *Setup(1*security parameter 1^λ*, N):* The setup algorithm takes as input a$\lambda$and the maximum number of users N. It outputs a master keyMK, an (empty) revocation list RL, a state ST, and public parameters PP.
- *GenKey(ID, MK, ST, PP):* The private key generation algorithm takes as input an identity $ID \in$ I, the master key MK, the state ST, and public parameters PP. It outputs a private key SK_{ID} for ID and an updated state ST.
- *UpdateKey(T, RL, MK, ST, PP):* The update key generation algorithm takes as input an update time $T \in$ T, the revocation list RL, the master keyMK, the state ST, and the public parameters PP. It outputs an update key $UK_{T,R}$for T and R where R is a revoked identity set at the time T.
- *DeriveKey(SK_{ID},$UK_{T,R}$, PP):* The decryption key derivation algorithm takes as input a private key SK_{ID}, an update key $UK_{T,R}$, and the public parameters PP. It outputs a decryption key $DK_{ID,T}$or $\perp$.
- *Encrypt(ID, T, M, PP):* The encryption algorithm takes as input an identity $ID \in$ I, a time T, a message $M \in$ M, and the public parameters PP. It outputs a ciphertext $CT_{ID,T}$for ID and T.
- *Decrypt($CT_{ID,T}$, $DK_{ID,T}$, PP):* The decryption algorithm takes as input a ciphertext $CT_{ID,T}$, a decryption key $DK_{ID,T}$, and the public parameters PP. It outputs an encrypted message M or $\perp$ • *Revoke(ID, T, RL, ST):* The revocation algorithm takes as input an identity ID to be revoked and a revocation time T, a revocation list RL, and a state ST. It outputs an updated revocation list RL.

The correctness property of RIBEis defined as follows: For all MK, RL, ST, and PP generated by **Setup**$(1^\lambda, N)$,

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

SK_{ID} generated by **GenKey**(ID, MK, ST, PP) for any ID, $UK_{T,R}$ generated by **UpdateKey**(T, RL, MK, ST, PP) for any T and RL, CT_{ID_c,T_c} generated by **Encrypt**(ID_c, T_c, M, PP) for any ID_c, T_c, and M, it is required that

- If $(ID \notin R)$, then **DeriveKey**$(SK_{ID}, UK_{T,R}, PP) = DK_{ID,T}$.
- If $(ID \in R)$, then **DeriveKey**$(SK_{ID}, UK_{T,R}, PP) = \perp$ with all but negligible probability.
- If $(ID_c = ID) \wedge (T_c = T)$, then **Decrypt**$(CT_{ID_c,T_c}, DK_{ID,T}, PP) = M$.
- If $(ID_c = ID) \vee (T_c = T)$, then **Decrypt**$(CT_{ID,T}, DK_{ID,T}, PP) = \perp$ with all but negligible probability.

The security property of RIBE was formally defined by Boldyreva *et al.* [7]. Recently Seo and Emura [11] refined the security model of RIBE by considering decryption key exposure attacks. In this paper, we consider the selective revocation list security model of the refined security model. In the selective revocation list security game, an adversary initially submits a challenge identity ID_*, a challenge time T_*, and a revoked identity set R_* at the time T_*, and then he can adaptively request private key, update key, and decryption key queries with restrictions. In the challenge step, the adversary submits two challenge messages M_0^*, M_1^*, and then he receives a challenge ciphertext CT_* that is an encryption of M_{b*} where b is a random coin used to create the ciphertext. The adversary may continue to request private key, update key, and decryption key queries. Finally, the adversary outputs a guess for the random coin b. If the queries of the adversary satisfy the non-trivial conditions and the guess is correct, then the adversary wins the game. The following is the formal definition of the selective revocation security.

Definition 2.2 (Selective Revocation List Security): The selective revocation list security property of RIBE under chosen plaintext attacks is defined in terms of the following experiment between a challenger C and a PPT adversary A:

1) *Init:* A initially submits a challenge identity $ID^* \in$ I, a challenge time $T^* \in$ T, and a revoked identity set $R^* \subseteq$ I at the time T_*.
2) *Setup:* C generates a master key MK, a revocation list RL, a state ST, and public parameters PP by running **Setup**$(1^\lambda, N)$. It keeps MK, RL, ST to itself and gives PP to A.
3) *Phase 1:* A adaptively requests a polynomial number of queries. These queries are processed as follows:
   - If this is a private key query for an identity ID, then it gives the corresponding private key SK_{ID} to A by running **GenKey**(ID, MK, ST, PP) with the restriction: If $ID = ID_*$, then the revocation query for ID_* and T must be queried for some $T \le T_*$.
   - If this is an update key query for a time T, then it gives the corresponding update key $UK_{T,R}$ to A by running **UpdateKey**(T, RL, MK, ST, PP) with the restriction: If $T = T_*$, then the revoked identity set of RL at the time T_* should be equal to R_*.
   - If this is a decryption key query for an identity ID and a time T, then it gives the corresponding decryption key $DK_{ID,T}$ to A by running **DeriveKey**$(SK_{ID}, UK_{T,R}, PP)$ with the restriction: The decryption key query for ID_* and T_* cannot be queried.
   - If this is a revocation query for an identity ID and a revocation time T, then it updates the revocation list RL by running **Revoke**(ID, T, RL, ST) with the restriction: The revocation query for a time T cannot be queried if the update key query for the time T was already requested.

   Note that A is allowed to request the update key query and the revocation query in non-decreasing order of time, and an update key $UK_{T,R}$ implicitly includes a revoked identity set R derived from RL.
4) *Challenge:* A submits two challenge messages $M_0^*, M_1^* \in \mathcal{M}$ with equal length. C flips a random coin $b \in \{0,1\}$ and gives the challenge ciphertext CT_* to A by running **Encrypt**(ID_*, T_*, M_{b*}, PP).
5) *Phase 2:* A may continue to request a polynomial number of private keys, update keys, and decryption keys subject to the same restrictions as before.
6) *Guess:* Finally, A outputs a guess $b \in \{0,1\}$, and wins the game if $b = b'$.

The advantage of Pr A where the probability is taken over all is defined as $\mathbf{Adv}^{IND}_{RIBE}{}^{\text{-}sRL}{}_{,A}{}^{\text{-}CPA}(\lambda) = [b = b'] - \frac{1}{2}$ the randomness of the experiment. A RIBE scheme is secure in the selective revocation list model under chosen plaintext attacks if for all PPT adversary A, the advantage of A in the above experiment is negligible in the security parameter λ.

Remark 2.3: The selective revocation list security model is weaker than the well-known selective security model since the adversary additionally submits the revoked identity set R_* in advance. However, this weaker model was already introduced by Boldyreva *et al.* [7] to prove the security of their revocable ABE scheme.[1]

B. Leveled Multilinear Maps

[1] Boldyreva *et al.* initially claimed that their revocable ABE scheme is secure in the selective model [7], but they later corrected it as their revocable ABE scheme is secure in the selective revocation list model [29].

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

We define generic leveled multilinear maps that are the leveled version of the cryptographic multilinear maps introduced by Boneh and Silverberg [30]. We follow the definition of Garg *et al.* [15].

Definition 2.4 (Leveled Multilinear Maps): We assume the existence of a group generator G, which takes as input a security parameter λ and a positive integer k. Let $\vec{\mathbb{G}} = (\mathbb{G}_1,\ldots,\mathbb{G}_k)$ be a sequence of groups of large prime order $p > 2^\lambda$. In addition, we let g_i be a canonical generator of $\mathbb{G}_i$ respectively. We assume the existence of a set of bilinear maps $\{e_{i,j}: \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j} \mid i, j \geq 1; i + j \leq k\}$ that have the following properties:

- Bilinearity: The map $e_{i,j}(g_a, g_b) = g_{iab+j}$: $\forall e_{a^i,j}$ $_b$ satisfies the following relation: $\in \mathbb{Z}_p$
- Non-Degeneracy: We have that $e_{i,j}(g_i, g_j) = g_{i+j}$ for each valid i, j.

We say that $\mathbb{G}$ is a multilinear group if the group operations in $\mathbb{G}$ as well as all bilinear maps are efficiently computable. We often omit the subscripts of $e_{i,j}$ and just write e.

C. Complexity Assumptions

We introduce new complexity assumptions in multilinear maps. The first assumption is the multilinear version of the well-known Bilinear Diffie-Hellman Exponent (BDHE) assumption of Boneh *et al.* [13].

Assumption 2.5 [(k,N)-MDHE)]: Let $(p,\mathbb{G},\{e_{i,j} \mid i, j \geq 1; i+ j \leq k\})$ be the description of a k-leveled multilinear group of order p. Let g_i be a generator of $\mathbb{G}_i$. The decisional (k, N)-MDHE assumption is that if the challenge tuple $D = (g_1, g_1^a, g_1^{a^2}, \ldots, g_{1^{aN}}, g_{1^{aN+2}}, \ldots, g_{1^{a2N}}, g_{1^{c1}}, \ldots, g_{1^{ck-1}}$ and Z are given, no PPT algorithm A can distinguish $^{+1}\prod_{i=1}^{k-1}$ $_{aNci} Z = Z_0 = g_k$ from a random element $Z = Z_1 \in \mathbb{G}_k$ with more than a negligible advantage. The advantage of A is defined as $\mathbf{Adv}^{(k,N)\text{-}MDHE}(\lambda) = |\Pr[\mathcal{A}(D, Z_0) = 0] - \Pr[\mathcal{A}(D, Z_1) = 0]|$ where the probability is taken over random choices of $a, c_1, \ldots, c_{k-1} \in \mathbb{Z}_p$.

For the security proof of our first RIBE scheme, we use $(3, N)$-MDHE assumption that is a specific instance of the MDHE assumption since the scheme is built on the three-leveled multilinear maps.

Assumption 2.6 [(3,N)-MDHE]: Let $(p,\mathbb{G},e_{1,1},e_{1,2},e_{2,1})$ be the description of a three-leveled multilinear group of order p. Let g_i be a generator of $\mathbb{G}_i$. The decisional $(3, N)$-MDHE assumption is that if the challenge tuple $D = g_1, g_{1^a}, g_{1^{a2}}, \ldots, g_{1^{aN}}, g_{1^{aN+2}}, \ldots, g_{1^{a2N}}, g_{1^b}, g_{1^c}$ and Z are given, no ^{N1} PPT algorithm A can distinguish $Z = Z_0 = g_3^{a+bc}$ from a random element $Z = Z_1 \in \mathbb{G}_3$ with more than a negligible advantage. The advantage of A is defined as $\mathbf{Adv}^{(3,N)\text{-}MDHE}(\lambda) = |\Pr[\mathcal{A}(D, Z_0) = 0] - \Pr[\mathcal{A}(D, Z_1) = 0]|$ where the probability is taken over random choices of $a, c \in \mathbb{Z}_p$.

The second assumption in multilinear maps is the compressed version of the BDHE assumption. Boneh *et al.* [17] introduced this compressed assumption to prove the security of their broadcast encryption in multilinear maps. [2] We slightly modify their assumption for our second RIBE scheme by adding additional one element.

Assumption 2.7 [(k,n,l)-cMDHE]: Let $(p,\mathbb{G},\{e_{i,j} \mid i, j \geq 1; i+ j \leq k\})$ be the description of a k-leveled multilinear groups of order p where $k = 2n + l - 2$. Let g_i be a generator of $\mathbb{G}_i$. The decisional (k,n,l)-cMDHE assumption is that if the challenge tuple $_n = (g_1, g_1^{a^{20}}, g_1^{a^{21}}, \ldots, g_1^{a^{2n}}, g_l^b, g_n^{c-1}$ and Z are given, no PPT $_{2n}$ algorithm $_1$ A can distinguish $Z = Z_0 = g_{2an+-l-bc2}$ from a random element $Z = Z_1 \in \mathbb{G}_{2n+l-2}$ with more than a negligible advantage. $^{(k,n,l)\text{-}MDHE}$

The advantage of A is defined as $\mathbf{Adv}(\lambda) = \Pr[A(D, Z_0) = 0] - \Pr[A(D, Z_1) =^A 0]$ where the probability is taken over random choices of $a, b, c \in \mathbb{Z}_p$.

We discuss the difficulty of our new assumptions in generic multilinear groups in Appendix A.

III. Revocable IBE With Shorter Keys

In this section, we propose an RIBE scheme with a constant number of private key elements and update key elements from three-leveled multilinear maps and prove its selective revocation list security. Essentially, we use the broadcast encryption of Boneh *et al.* [13], which uses bilinear maps.

A. Construction

Let $\mathbb{N} = \{1,\ldots, N_1\}$ where N is the (polynomial) number of users. Let $\mathbb{I} = \{0,1\}^{l}$ be the identity space and $\mathbb{T} = \{0,1\}^{l2}$ be the time space where $l_1 = 2\lambda$ and $l_2 = \lambda$ for a security parameter λ. Our RIBE scheme from three-leveled multilinear maps is described as follows:

RIBE.Setup(1^λ, N): This algorithm takes as input a security parameter 1^λ and the maximum number N of users. It generates a 3-leveled multilinear group $\mathbb{G} = (\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3)$ of prime order p. Let g_1, g_2, g_3 be generators of $\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3$ respectively. Let PP_{MLM} be the description of the multilinear group with generators.

[2] In [17], Boneh *et al.* called their new assumption as the Multilinear Diffie-Hellman Exponent (MDHE) assumption, but it is different to our MDHE assumption.

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

$\{ \quad 1,i,j\}_{1\le i\le l_1, j\in\{0,1\}}$ $\quad 1,0$ $\{ \quad 1,i,j\}_{1\le i\le l_2, j\in\{0,1\}}$
Let $\vec{\mathbb{G}} = (\quad k,0, \{ \quad k,i,j\}_{1\le i\le l_1, j\in\{0,1\}})$

1) It selects random elements $f, f, h, h \in$1.
f_k f f and h_k = $(h_{k,0}, \{h_{k,i,j}\}_{1\le i\le l_2, j\in\{0,1\}})$ for a level k.
Note that we can obtain $\vec{f}_2$ and h_2 from f_1 and h_1 by performing pairing$_1$ operations. We define $F_k(ID) = f_{k,0} \prod_{i=1}^{l} f_{k,i,ID[i]}$ and $H_k(T) = h_k^{,0} \prod_{i=1}^{l_2} h_k^{,i,T_i}$ where $ID[i]$ is a bit value at the position $i$and $T[i]$ is a bit value at the position i.
2) Next, it selects random exponents $\alpha, \beta, \gamma \in \mathbb{Z}_p$. It outputs a master key $MK = (\alpha, \beta, \gamma)$, an empty revocation list RL, an empty state ST, and public parameters PP as
$PP_{MLM}, \{g_1^{\alpha^j}\}_{1\le j, j=N+1\le 2N}, g_1^{\beta}, f_1, h_1, = g_3^{\alpha^{N+1}\beta}$.

RIBE.GenKey(ID, MK, ST, PP): This algorithm takes as input an identity $ID \in$ I, the master keyMK, the state ST, and public parameters PP.
1) It first assigns an index $d \in$ N that is not in ST to the identity ID, and updates the state ST by adding a tuple (ID,d) to ST.
2) Next, it selects a random exponent $r_1 \in \mathbb{Z}_p$and outputs a private key SK_{ID} by implicitly including ID and the index d as
$K_0 = g_1^{\alpha d \gamma} F_1(ID)^{-r_1}, K_1 = g_1^{-r_1}$.

RIBE.UpdateKey(T, RL, MK, ST, PP): This algorithm takes as input a time T, the revocation list RL, the master keyMK, the state ST, and public parameters PP.
1) It first defines the revoked set R of user identities at the time T from RL. That is, if there exists (ID, T)such that $(ID, T) \in RL$for any $T \le T$, then $ID \in R$. It defines the revoked index set $RI \subseteq$ N of the revoked identity set R by using the state ST since ST contains (ID,d). It also defines the non-revoked index set $SI =$ N $\setminus RI$.
2) Next, it selects a random exponent $r_2 \in \mathbb{Z}_p$and outputs an update key $UK_{T,R}$by implicitly including T, R, and the revoked index set RI as
$U_0 = \left(g_1^{\gamma} \prod_{j\in SI} g_1^{\alpha^{N+1-j}}\right)^{\beta} H_1(T)^{r_2}, \; U_1 = g_1^{-r_2}\Big)$.

RIBE.DeriveKey(SK_{ID},$UK_{T,R}$, PP): This algorithm takes as input a private key $SK_{ID} = (K_0, K_1)$ for an identity ID, an update key $UK_{T,R} = (U_0, U_1)$ for a time T and a revoked set R of identities, and the public parameters PP. If $ID \in R$, then it outputs $\perp$ since the identity $ID$is revoked. Otherwise, it proceeds the following steps:
1) Let d be the index of ID and RI be the revoked index set of R. Note that these are implicitly included in SK and UK respectively. It sets a non-revoked index set
$SI =$ N $\setminus RI$ and derives temporal components T_0, T_1 and T_2 as
$T_0 = e(g_1^{\alpha^d}, U_0) \cdot e\big(g_1^{\beta}, K_0 \prod_{j\in SI, j=d} g\big)$,
$T_1 = e(g_1^{\beta}, K_1), \; T_2 = e(g_1^{\alpha^d}, U_1)$.
2) Next, it chooses random exponents $r_1', r_2' \in \mathbb{Z}_p$ and re-randomizes these components as $D_0 = T_0 \cdot F_2(ID)^{r_1'} H_2(T)^{r_2'}, D_1 = T_1 \cdot g_2^{-r_1'}, D_2 = T_2 \cdot g_2^{-r_2'}$.
Note that these components are formed as
$D_0 = g_2^{\alpha+\beta} N_1 F_2(ID)^{r_1} H_2(T)^{r_2}, \quad D_1 = g_2^{-r_1}, \quad D_2 = g_2^{-r_2}$where $r_1 = \beta r_1 + r_1$and $r_2 = \alpha d r_2 + r_2$.
3) Finally, it outputs a decryption key as $DK_{ID,T} = D_0, D_1, D_2$.

RIBE.Encrypt(ID, T, M, PP): This algorithm takes as input an identity ID, a time T, a message M, and the public parameters PP. It first chooses a random exponent $s \in \mathbb{Z}_p$and outputs a ciphertext $CT_{ID,T}$by implicitly including ID and T as
$C = \Omega^s \cdot M, C^0 = g_1^s, \; C_1 = F_1(ID)^s, C^2 = H_1(T)^s\Big)$.

RIBE.Decrypt($CT_{ID,T}$, $DK_{ID,T}$, PP): This algorithm takes as input a ciphertext $CT_{ID,T} = (C, C_0, C_1, C_2)$, a decryption key $DK_{ID,T} = (D_0, D_1, D_2)$, and the public parameters PP. If $(ID = ID') \wedge (T = T)$, then it outputs the encrypted message M as $M = C \cdot \left(\prod_{i=0}^{2} e_{1,2}(C_i, D_i)^{-1}\right.$.
Otherwise, it outputs $\perp$

RIBE.Revoke(ID, T, RL, ST): This algorithm takes as input an identity ID, a revocation time T, the revocation list RL, and the state ST. If $(ID, -) \notin ST$, then it outputs $\perp$ since the private key of ID was not generated. Otherwise, it adds (ID, T) to RL. It outputs the updated revocation list RL.

B. Correctness

Let SK_{ID} be a private key for an identity IDthat is associated with an index d, and $UK_{T,R}$be an update key for a time T and a revoked identity set R. If $ID \notin R$, then the decryption key derivation algorithm first correctly derives temporal components as
$T_0 = e(g_1^{\alpha^d}, U_0) \cdot e(g_1^{\beta}, K_0 \prod_{j\in SI, j=d} g_{1\alpha^{N+1-j+d}})^{-1}$
$= e(g_1^{\alpha^d}, (g_1^{\gamma} \prod_{j\in SI} g_{1\alpha^{N+1-j}})^{\beta} H_1(T)^{r_2})$
$\times e(g_1^{\beta}, g_1^{\alpha^d \gamma} F_1(ID)^{-r_1} \cdot \prod_{j\in SI, j=d} g_{1\alpha^{N+1-j+d}})^{-1}$

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

$= e(g_1^{\beta}, g_1^{\alpha^{N+1}}) \cdot e(g_1^{\beta}, F_1(ID)^{r_1}) \cdot e(g_1^{\alpha_d}, H_1(T)^{r_2})$,
$= g_2^{\alpha^{N+1}\beta} F_2(ID)^{\beta r_1} H_2(T)^{\alpha^d r_2}$,
$T1 = e(g_1^{\beta}, K_1) = e(g_1^{\beta}, g_1^{-r_1}) = g_{2-\beta r_1}$,
$T2 = e(g_1^{\alpha^d}, U_1) = e(g_1^{\alpha^d}, g_1^{-r_2}) = g_2^{-\alpha^d r_2}$ where RI is the revoked index set of R and $SI = \mathbb{N} \setminus RI$.

Next, a decryption key is correctly derived from the temporal components by performing re-randomization as $D_0 = T_0 \cdot F_{12}(ID)^{r'_1} H_2(T)^{r_{2d}\alpha + \beta}$
$= g_{2\,N} F_2(ID)^{\beta r_1} H_2(T)^{\alpha r_{2\,d}} \cdot F_2(ID)^{r_1} H_2(T)^{r_2 \alpha + \beta F}$
$= g_{2\,N\,1} {}_{2}(ID)^{\beta r_1 + r_1} H_2(T)^{\alpha r_2 + r_2}$
$= g_{2\alpha N+1} {}^{\beta} F_2(ID)^{r_1} H_2(T)^{r_2}$,
$D_1 = T_1 \cdot g_{2-r_1} = g_{2-\beta r_{d1}-r_1} = g_{2-r_1}$,
$D_2 = T_2 \cdot g_{2-r''} = g_{2-\alpha r_2 - r''} = g_{2-r}$ where $r_1 = \beta r_1 + r_1$ and $r_2 = \alpha^d r_2 + r_2$.

Let $CT_{ID,T}$ be a ciphertext for an identity ID and a time T, and $DK_{ID,T}$ be a decryption key for an identity ID and a time T. If $(ID = ID) \wedge (T = T)$, then the decryption algorithm correctly outputs an encrypted message by the following equation.

$$\prod_{i=0}^{2} e(C_i, D_i)$$
$$= e(g_1^s, g_2^{\alpha^{N+1}\beta} F_2(ID)^{r''_1} H_2(T)^{r''_2}) \times e(F_1(ID)^s, g_2^{r''_1}) \cdot e(H_1(T)^s, g_2^{r''_2})$$
$$= e(g_1^s, g_2^{\alpha^{N+1}\beta}) \frac{e(g_1^s, F_2(ID)^{r''_1}) \cdot e(g_1^s, H_2(T)^{r''_2})}{(F_1(ID)^s, g_2^{r''_1}) \cdot e(H_1(T)^s, g_2^{r''_2})}$$
$$= e(g_1^s, g_2^{\alpha^{N+1}\beta}) = (g_3^{\alpha^{N+1}\beta})^s = \Omega^s.$$

C. Security Analysis

To prove the security of our RIBE scheme, we carefully combine the partitioning methods of the PKBE scheme of Boneh *et al.* [13] and the HIBE scheme of Boneh and Boyen [16].

Theorem 3.1: The above RIBE scheme is secure in the selective revocation list model under chosen plaintext attacks if the $(3, N)$-MDHE assumption holds where N is the maximum number of users in the system. That is, for any PPT adversary A, we have that $\mathbf{Adv}^{INDRIBE\text{-}sRL,A\text{-}CPA}(\lambda) \leq \mathcal{B}^{(3,N)}\,\mathbf{Adv}^{\text{-}MDHE}(\lambda)$.

Proof: Suppose there exists an adversary A that attacks the above RIBE scheme with a non-negligible advantage. A simulator B that solves the MDHE assumption using A is given: a challenge tuple $D = (g_1, g_1^a, g_1^{a^2}, \ldots, g_1^{a^N}, g_1^{a^{N+2}}, \ldots, g_1^{a^{2N}}, g_1^b, g_1^c)$ and Z where $Z = Z_0 = g_3^{a^{N+1}bc}$ or $Z = Z_1 \in \mathbb{G}_3$. Then B that interacts with A is described as follows:

Init: A initially submits a challenge identity ID^*, a challenge time T^*, and a revoked identity set R^* at the time T^*.

It first sets a state ST and a revocation list RL as empty one. For each $ID \in \{ID^*\} \cup R^*$, it selects an index $d \in \mathbb{N}$ such that $(-, d) \notin ST$ and adds (ID, d) to ST. Let $RI^* \subseteq \mathbb{N}$ be the revoked index set of R^* at the time T^* and SI^* be the non-revoked index set at the time T^* such that $SI^* = \mathbb{N} \setminus RI^*$.

Setup: B first chooses random exponents $f_0, \{f'_{i,j}\}_{1 \leq i \leq l_1, j \in \{0,1\}}, h'_0, \{h'_{i,j}\}_{1 \leq i \leq l_2, j \in \{0,1\}}, \vartheta \in \mathbb{Z}_p$.

It implicitly sets $\alpha = a, \beta = b, \gamma = \vartheta - \sum_{j \in SI^*} a^{N+1-j}$ and publishes the public parameters PP as

$$g_{1\alpha_i} = g_1^a\}_{1 \leq i \leq 2N, i \neq N+1}, g_1^{\beta} = g_1^b,$$
$$f_1 = \left(f_{1,0} = g_{1f_0} \prod_{i=1}^{l_1} f_{1,i,ID^*[i]}\right)^{-1},$$
$$f_{1,i,j} = (g_1^{a^N} f_{i,j}) 1 \leq i \leq l_1, j \in \{0,1\},$$
$$_1 = \left(h_{1,0} = \prod_{l_2} {}_{1,i,T^*[i]}\right)^{-1}$$
$$h^0 h, hg_1$$
$$\prod_{i=1}^{} h_{1,i,j} = (g_1^b)_{h_{i,j}} 1 \leq i \leq l_2, j \in \{0,1\},$$
$$= e\left(e(g_1^{\alpha}, g_1^{\alpha^N}), g_1^b\right) = g_3^{\alpha^{N+1}b}.$$

For notational simplicity, we define ID
$$\prod_{=1}^{2} (h'_{i,T[i]} - h'_{i,T^*[i]})$$
$$[\,] \not\equiv 0 \mod p$$
$= \sum_{i=1}^{l_1} (f_{i,ID[i]} -$
$f_{i,ID^*\,i})$ and $\Delta T = {}_{li}$. We have ID except with negligible probability if $ID = ID^*$ since there exists at least one index i such that $f_{i,ID[i]} \neq f'_{i,ID^*[i]}$ and $\{f'_{i,j}\}$ are randomly chosen. We also have $T \not\equiv 0 \mod p$ except with negligible probability if $T = T^*$.

Phase 1: A adaptively requests a polynomial number of private key, update key, and decryption key queries.

If this is a private key query for an identity ID, then B proceeds as follows:

- *Case $ID \in R^*$:* In this case, the simulator can use the partitioning method of Boneh *et al.* [13]. It first retrieves a tuple (ID, d) from ST where the index d is associated with ID. Note that the tuple (ID, d) exists since all identities in R^* were added to ST in the initialization step. Next, it selects a random exponent $r_1 \in \mathbb{Z}_p$ and creates a private key SK_{ID} as

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

$K^0 = (g_1^{a^d})^{\theta} \quad g_1^{a^{N+1-j+d}})^{-1} F_1(ID)^{-r_1}, \; K_1 = g_1^{-r_1}.$
$j \in SI$ $*$

- *Case* $ID \notin R^*$: In this case, we have $ID = ID_*$ from the restriction of Definition 2.2 and the simulator can use the partitioning method of Boneh and Boyen [16]. It first selects an index $d \in \mathbb{N}$ such that $(-,d) \notin ST$ and adds (ID,d) to ST. Next, it selects a random exponents $r_1' \in \mathbb{Z}_p$ and creates a private key SK_{ID} by implicitly setting $r_1 = -a/ID + r_1'$ as

$K^0 = g_1^{a^d\theta} \prod_{j \in SI^* \setminus \{d\}} g_1^{-a^{N+1-j+d}} (g_1^a)^{f_0/ID} F_1(ID)^{-r_1},$

$K^1 = (g_1^a)^{-1/ID} g_1^{r_1}.$

If this is an update key query for a time T, then B defines a revoked identity set R at the time T from RL and proceeds as follows:

- *Case* $T = T_*$: In this case, the simulator can use the partitioning method of Boneh and Boyen [16]. It first sets a revoked index set RI of R by using ST. $SI = \quad \in \mathbb{Z}$ It also sets $\mathbb{N} \setminus RI$. Next, it selects a random exponent r_{2p} and creates an update key $UK_{T,R}$ by implicitly setting $r_2 = -a(-$ $j \in SI_* \setminus SI$ a^{N+1-j} $+$ $j \in SI \setminus SI_*$ $a + r_2'$

$U^0 = (g_1^b)^{\theta} \Big(\prod_{j \in SI^* \setminus SI} g_{1-aN+1-j} \quad g_{1aN}$
$j \in SI \setminus SI_*$

$H_1(T)r_2,$

$_1 = \Big(\prod_{\in \; *} U \, g_{1-aN+1-j} g_1^{a} \Big)_{N+1-j-1/T}$

$g_{1r_2}. \; _{j \; SI \setminus SI \, j \in SI \setminus SI_*}$

- *Case* $T = T_*$: In this case, we have $R = R_*$ and the simulator can use the partitioning method of Boneh *et al.* [13]. For each $ID \in R_*$, it adds (ID, T_*) to RL if $(ID, T) \notin RL$ for any $T \leq T_*$. Next, it selects a random exponent $r_2 \in \mathbb{Z}_p$ and creates an update key $UK_{T,R}$ as

$U^0 = (g_1^b)^{\theta} H_1(T^*)^{r_2}, \; U_1 = g_1^{-r_2}.$

If this is a decryption key query for an identity ID and a time T, then B proceeds as follows:

- *Case* $ID = ID_*$: In this case, the simulator can use the partitioning method of Boneh and Boyen [16]. If $(ID,-) \notin ST$, then it selects an index $d \in \mathbb{N}$ such that $(-,d) \notin ST$ and adds (ID,d) to ST.
Next, it selects random exponents $r_1', r_2 \in \mathbb{Z}_p$ and creates a decryption key $DK_{ID,T}$ by implicitly setting $r_1 = (-a/ID + r_1')b$ as

$D^0 = e\big((g_1^a)^{-f_0'/ID} F_1(ID)^{r_1}, g_{1b} \cdot H_2(T)^{r_2},$
$D^1 = e\big((g_1^a)^{-1}{}_{/ID} g_{1r_1}, g_{1b}, D_2 = g_{2r_2}.$

- *Case* $ID = ID_*$: In this case, we have $T = T_*$ from the restriction of Definition 2.2, and the simulator can use the partitioning method of Boneh and Boyen [16]. It selects random exponents $r_1, r_2' \in \mathbb{Z}_p$ and creates a decryption key $DK_{ID,T}$ by implicitly setting $r_2 = (-a/T + r_2')a^N$ as

$D^0 = e\big((g_1^a)^{-h_0'/T} H_1(T)^{r_2}, g_1^a \big) \cdot$ $_N$
$F_2(ID)^{r_1},$

$D^1 = g_2^{r_1}, \; D_2 = e\big((g_1^a)^{-1}{}_{/T} g_1{}^2, g_1^a \big)_{rN}.$

Challenge: A submits two challenge messages M_0^*, M_1^*. $\mathcal{B}$ chooses a random bit $\delta \in \{0,1\}$ and creates the challenge ciphertext CT_* by implicitly setting $s = c$ as

$C = Z \cdot M_\delta^*, \; C_0 = g_1^c, \; C_1 = (g_1^c)^{f_0'}, \; C_2 = (g_1^c)^{h_0'}.$

Phase 2: Same as Phase 1.

Guess: Finally, A outputs a $\delta' \in \{0,1\}$. $\mathcal{B}$ guesses 0 if $\delta = \delta$ or 1 otherwise.

To finish the proof, we first show that the distribution of the simulation is correct from Lemma 3.2. Let η be a random bit for Z_η. From the above simulation, we have $\Pr[\delta = \delta' | \eta = 0] = \frac{1}{2} + \mathbf{Adv}^{IND}_{RI \; BE^{-sRL}, A^{-CPA}}(\lambda)$ since the distribution of the simulation is correct, and we also have $\Pr[\delta = \delta' | \eta = 1] = \frac{1}{2}$ since δ is completely hidden to A. Therefore we can obtain the following equation

$^{/T}$
$\mathbf{Adv}^{(3,N)\text{-}MDHE}(\lambda)$

$= |\Pr[\mathcal{B}(D, Z_0) = 0] - \Pr[\mathcal{B}(D, Z_1) = 0]|$ B
$\geq \big|\Pr[\delta = \delta' | \eta = 0]\big| - \big|\Pr[\delta = \delta' | \eta = 1]\big| = \frac{1}{2} +$

$\mathbf{Adv}^{sRL\text{-}CPA}_{INDRI\,BE\text{-},\mathcal{A}}(\lambda) - \frac{1}{2} = \mathbf{Adv}^{sRL\text{-}CPA}_{INDRI\,BE\text{-},\mathcal{A}}(\lambda).$

This completes our proof. ■

Lemma 3.2: The distribution of the above simulation is correct if $Z = Z_0$, and the challenge ciphertext is independent of δ in the adversary's view if $Z = Z_1$.

Proof: The distribution of public parameters is correct since random exponents $f_0', \{f_{i,j}'\}, h_0', \{h_{i,j}'\}, \theta \in \mathbb{Z}_p$ are chosen.

We show that the distribution of private keys is correct. In case of $ID \in R_*$, we have that the private key is correctly distributed from the setting $\gamma = \theta - \sum_{j \in SI^*} a^{N+1-j}$ as the following equation

$K_0 = g_1^{a^d \gamma} F_1(ID)^{-r_1} = g_1^{a^d(\theta - \sum_{j \in SI^*} a^{N+1-j})} F_1(ID)^{-r_1}$
$= g_1^{a^d\theta} \Big(\prod_{j \in SI^*} g_1^{a^{N+1-j+d}} \Big)^{-1} F_1(ID)^{-r_1}.$

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

In case of $ID \notin R_*$, we have that the private key is correctly distributed from the setting $\gamma = \vartheta - \sum_{j \in SI_*} a^{N+1-j}$ and $r_1 = -a/ID + r'_1$ as the following equation

$$K_0 = g_1^{a^d \gamma} F_1(ID)^{-r_1}$$
$$= g_1^{a^d \theta} \prod_{j \in SI_*} g^{-a^{N+1-j+d}} \left(f_{1,0} \prod_{i=1}^{l} f_{1,i,ID[i]}^{-r_1}\right)$$
$$= g_1^{a^d \theta} \prod_{j \in SI_*} g_{1-aN+1-j+d}^{-a^{N+1}} \cdot g_1^{\left(g_1^{f'_0} g_{1N}^{a} ID a/ID\right)^{-r'_1}}$$
$$= g_1^{a^d \theta} \prod_{j \in SI_* \setminus \{d\}}^{\{d\}} g_1^{-a^{N+1-j+d}} (g_1^a)^{f_0/ID F_1(ID)-r_1},$$
$$K_1 = g_1^{r_1} = (g_1^a)^{-1/ID} g_{1r_1}.$$

Next, we show that the distribution of update keys is correct. In case of $T = T_*$, we have that the update key is correctly distributed from the setting $\gamma = \theta - \sum_{j \in SI_*} a^{N+1+j} + \sum_{j \in SI \setminus SI_*} a^{N+1-j}$ and $r_2 = -(-\sum_{j \in SI_* \setminus SI} a^{N+1-j})/T + r_2$ as the following equation

$$= \left(g_1^{\gamma} \prod_{j \in SI} U_0 \prod_{SI} g_1^{a^{N+1-j}}\right)^{\beta} H_1(T)^{r_2}$$
$$= \left(g_1^{\theta} \left(\prod_{j \in SI_*} g_1^{a^{N+1-j}}\right)^{-1} \prod_{j \in SI}^{t} g_1^{a^{N+1-j} b} h_{1,0} \prod_{i=1} h_{1,i,T[i]}^{r_2}\right.$$
$$= (g_1^b)^{\theta} \prod_{j \in SI \setminus SI_*} g_1^{-a^{N+1-j}} \prod_{j \in SI \setminus SI_*} g_1^{a^{N+1-j} b}$$
$$\times g_1^{h_0} g_{1bT} \left(\sum_{j \in SI \setminus SI_*} a^{N+1-j} - \sum_{j \in SI \setminus SI_*} a^{N+1-j}\right)/\Delta T + r_2$$
$$= (g_1^b)^{\theta} \left(\prod_{j \in SI_*} g_{1-aN+1-j} \prod_{SI \setminus SI_*} g_{1-aN+1-j}\right)$$
$$\times H_1(T)^{r_2},$$
$$U_1 = g_1^{r_2} = \left(\prod_{j \in SI_*} U \prod_{SI \setminus SI_*} g_{1-aN+1-j}\right.$$

In case of $T = T_*$, we have that the update key is correctly distributed from the setting $\gamma = \theta - \sum_{j \in SI_*} a^{N+1-j}$ as the following equation

$$U_0 = \left(g_1^{\gamma} \prod_{j \in SI_*} g_1^{a^{N+1-j}}\right)^{\beta} \cdot H_1(T_*)^{r_2}$$
$$= \left(g_1^{\theta} g_1^{a^{N+1-j}}\right)^{-1} \cdot \prod_{j \in SI_*} g_1^{a^{N+1-j} b} H_1(T_*)^{r_2}$$
$$= (g_1^b)^{\theta} H_1(T^*)^{r_2}.$$

We show that the distribution of decryption keys is correct. In case of $ID = ID_*$, the decryption key is correctly distributed from the setting $\log_{g_2} F_2(ID) = \alpha^N ID$ and $r_1 = (-\alpha/ID + r'_1)b$ as the following equation

$$D_0 = g_2^{\alpha^{N+1} \beta} F_2(ID)^{r_1} H_2(T)^{r_2}$$
$$= g_2^{a^{N+1} b} \left(f_{2,0} \prod_{i=1}^{l} f_{2,i,ID[i]}\right)^{(-a/ID + r'_1)b} H_2(T)^{r_2}$$
$$= e\left(g_1^{a^{N+1}}, \left(g_1^{f'_0} g_{1N}^{a\, ID - a/ID + r'_1}, g_1^b\right)\right) \cdot H_2(T)^{r_2}$$
$$= e\left((g_1^a)^{-f'_0/ID} F_1(ID)^{r_1}, g_1^b\right) \cdot H_2(T)^{r_2},$$
$$D_1 = g_2^{r_1} = e(g_1, g_1)^{(-a/ID)b} = e\left((g_1^a)^{-1/ID} g_{1r_1}, g_{1b}\right).$$

In case of $ID = ID_*$, the decryption key is correctly distributed from the setting $\log_{g_2} H_2(T) = bT$ and $r_2 = (-a/T + r'_2)a^N$ as the following equation

$$D_0 = g_2^{\alpha^{N+1} \beta} F_2(ID)^{r_1} H_2(T)^{r_2}$$
$$= g_2^{a^{N+1} b} F_2(ID)^{r_1} (u_{2,2}^T h_{2,2})^{(-a/T + r'_2)a^N}$$
$$= e\left(g_1^{ab} \left(g_1^{b \Delta T} g_1^{h'_2}\right)^{-a/T + r'_2}, g_1^{a^N}\right) \cdot F_2(ID)^{r_1}$$
$$= e\left((g_1^a)^{-h'_0/T} H_1(T)^{r_2}, g_1^a\right) \cdot F_2(ID)^{r_1},$$
$$D_2 = g_2^{r_2} = e(g_1, g_1)^{(-a/T + r'_2)a^N}$$
$$= e\left((g_1^a)^{-1/T} g_1^2, g_1^a\right)^{r_N}$$

Finally, we show that the distribution of the challenge ciphertext is correct. If $Z = Z_0 = g_3^{a^{N+1} bc}$ is given, then the challenge ciphertext is correctly distributed as the following equation

$$C = \Omega^s \cdot M_\delta^* = g_3^{a^{N+1} bs} \cdot M_\delta^* = Z_0 \cdot M_\delta^*,$$
$$C_0 = g_1^s = g_{1c},$$
$$C_1 = \left(g_1^{f_0} \prod_{i=1}^{l} f_{1,i,ID^*[i]}^{-1}\right)^c = (g_1^c)^{f_0},$$
$$C_2 = \left(g_1^{h_0} \prod_{i=1}^{t} h_{1,i,T^*[i]}^{-1}\right)^c = (g_1^c)^{h_0}.$$

Otherwise, the component C of the challenge ciphertext is independent of δ in the A's view since Z_1 is a random element in G_3. This completes our proof. ■

D. Discussions

Graded Encoding Systems: The candidate multilinear maps of Garg *et al.* [15] are different from the leveled multilinear maps in Section II-B. The main difference is that the encoding of a group element is randomized in the GGH framework whereas the encoding is deterministic in the leveled multilinear maps. This means that it is not trivial to check whether two strings encode the same element. Thus, additional procedures for this checking are essentially required in the GGHframework. In the full version of this paper [31], we define the graded encoding system of Garg *et al.* [15] and translate our RIBE scheme for small universe

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

in that the total number of identities is limited to polynomial number into the graded encoding system.

Asymptotic Analysis: The number of group elements in public parameters, a private key, an update key, and a ciphertext of our RIBE scheme is $O(N + \lambda)$, $O(1)$, $O(1)$, and $O(1)$ respectively, where N is the maximum number of users. Although our RIBE scheme provides efficient asymptotic parameters, except for the public parameters, it is not actually efficient since the underlying multilinear maps are not yet practically efficient. Let $\lambda = 80$ and $k = 3$. The multilinear maps of Garg *et al.* [15] have the following asymptotic parameters such that the bit size of the public parameters is $O(k^3\lambda^5 \log(k\lambda)) \approx 7.1 * 10^{11}$ and the bit size of group elements is $O(k^2\lambda^3) \approx 4.6 * 10^6$.

To improve the efficiency, we may use the multilinear maps of Langlois *et al.* [32] such that the bit size of the public parameters is $O(k^3\lambda \log^2(k\lambda)) \approx 1.8 * 10^5$ and the bit size of the group elements is $O(k^2\lambda \log^2(k\lambda)) \approx 4.6 * 10^4$. Note that the bit size of the group elements in bilinear groups is 160.

Chosen-Ciphertext Security: Security against chosenciphertext attacks (CCA security) is similar to security against chosen-plaintext attacks (CPA security) except that an adversary can request a ciphertext decryption query. To provide CCA security, we can use the general transformation of Canetti *et al.* [33] since the structure of our RIBE scheme is similar to that of the BB-HIBE scheme [16]. That is, we can modify our RIBE scheme to support three-level hierarchies by providing additional elements, and then the modified RIBE scheme is easily converted to a CCA-secure RIBE scheme since this tree-level HIBE scheme with CPA security is converted to a two-level HIBE scheme with CCA security.

IV. Revocable IBE With Shorter Parameters

In this section, we propose an RIBE scheme with short public parameters and short keys from multilinear maps and prove its selective revocation list security. To achieve shorter size of public parameters, we use the BWZ-PKBE scheme [17] that uses multilinear maps since it has short public parameters and the structure of it is almost similar to that of the BGW-PKBE scheme [13].

A. Construction

We set $N = 2^n - 2$ for some integer n. Note that N should be polynomial in the security parameter[1] λ. Let $\mathsf{N} = \{1, \ldots, 2N\}$. Let $\mathsf{I} = \{0,1\}^l$ be the identity space and $\mathsf{T} = \{0,1\}^l$ be the time space where $l_1 = 2\lambda$ and $l_2 = \lambda$. We suppose that an index d that is assigned to an identity ID has a Hamming weight l. Our RIBEscheme from $2n+l-2$-leveled multilinear maps is described as follows:

RIBE.Setup(1^λ, N): This algorithm takes as input a security parameter 1^λ and the maximum number N of users. It generates a $2n + l - 2$-leveled multilinear group $\vec{\mathbb{G}} = (\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_{2n+l-2})$ of prime order p respectively. Let PP_{MLM} be the description of the multilinear group with generators. Let g_i be generators of $\mathbb{G}_i$.

$$\vec{f}_k = \left(f_{k,0}, \{f_{k,i,j}\}_{1 \le i \le l_1, j \in \{0,1\}}\right) \in \mathbb{G}_n^{\{-1\} \cup \{0,1\}^{\le k} \times \{1\} \cup \{0,1\}^{\le l_1} \times \{0,1\}}$$

1) It selects random elements $f_{n-1,0}$, $f_{n1,i,j\,1 \le i \le l,j}$ $_{0,1}$, $h_{n1,0}$, $h_{n1,i,j\,1 \le i \le l,j}$ $_{0,1} \in \mathbb{G}_{n-1}$. Let $h_k = h_{k,0}, \{h_{k,i,j}\}_{1 \le i \le l_2, j \in \{0,1\}}$ and for a level $k \ge n - 1$. Note that we can obtain f_k and h_k from f_{n-1} and $\vec{h}_{n-1}$ by performing pairing operations.[1] We define $F_k(ID) = f_{k,0} \prod_{i=1} f_{k,i,ID[i]}$ and value at the position $H_k(T) = h_k^{\prime 0} \prod_{i=1}^{l_2} h_{k,i,T[i]}^{\prime i}$ and $T[i]$ where $T[i]$ is a bit value at the $ID[i]$ is a bit position i.
2) Next, it selects random exponents $\alpha, \beta, \gamma \in \mathbb{Z}_p$. It outputs a master key $MK = (\alpha, \beta, \gamma)$, an empty revocation list RL, an empty state ST, and public parameters PP as

$$PP_{MLM}, g_1^{\alpha^{2i}}{}_{0 \le i \le n}, g_1^{\beta}, f_{n-1}, h_{n-1}, g_{2n+l-2}^{\alpha^{2^n-1}\beta}\Big).$$

RIBE.GenKey(ID, MK, ST, PP): This algorithm takes as input an identity $ID \in \mathsf{I}$, the master key MK, the state ST, and public parameters PP.

1) It first assigns an index $d \in \{0,1\}^n$ of Hamming weight l that is not in ST to the identity ID and updates the state ST by adding a tuple (ID, d) to ST.
2) It computespairing operations on the elements that are given in $g_n^{\alpha - d}{}_1$ by performing multiplications and PP.
3) Next, it selects a random exponent $r_1 \in \mathbb{Z}_p$ and outputs a private key SK_{ID} by implicitly including ID and the index d as

$$K_0 = g_{n\alpha - d\gamma 1} F_{n-1}(ID)^{-r_1}, K_1 = g_{n--r11}.$$

RIBE.UpdateKey(T, RL, MK, ST, PP): This algorithm takes as input a time $T \in \mathsf{T}$, the revocation list RL, the master key MK, the state ST, and public parameters PP.

1) It first defines the revoked set R of user identities at the time T from RL. That is, if there exists (ID, T') such that $(ID, T') \in RL$ for any $T' \le T$, then $ID \in R$. It defines the revoked index set $RI \subseteq \mathsf{N}$ of the revoked identity set R by using the state ST since ST contains (ID, d). It also defines the non-revoked index set $SI = \mathsf{N} \setminus RI$. Note that $|SI|$ is polynomial since N is polynomial. $\{g_{n-1}^{\alpha^{2^n-1-j}}\}$
2) It computes $_{j \in SI}$ by performing multiplications and pairing operations on the elements that are given in PP.

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

3) Next, it selects a random exponent $r_2 \in Z_p$ and outputs an update key $UK_{T,R}$ by implicitly including T, R, and the revoked index set RI as

$$U_0 = \left(g_{n-1}^{\gamma} \prod_{j \in SI} g_{n-1}^{\alpha^{2^n-1-j}}\right)^{\beta} H_{n-1}(T)^{r_2}, \quad U_1 = g_{n-1}^{-r_2}$$

RIBE.DeriveKey(SK_{ID}, $UK_{T,R}$, PP): This algorithm takes as input a private key $SK_{ID} = (K_0, K_1)$ for an identity ID, an update key $UK_{T,R} = (U_0, U_1)$ for a time T and a revoked set R of identities, and the public parameters PP. If $ID \in R$, then it outputs $\perp$ since the identity ID is revoked. Otherwise, it proceeds the following steps:

1) Let d be the index of ID and RI be the revoked index set of R. Note that these are implicitly included in SK and UK respectively. It sets a non-revoked index set $SI = \mathbb{N} \setminus RI$. Note that $|SI|$ is polynomial since N is polynomial. $\{g_{n-1}^{\alpha^{2^n-1-j+d}}\}$
2) It computes $g_l^{\alpha d}{}_{,j \in SI, j=d}$ by performing multiplications and pairing operations on the elements that are given in PP. Using these elements, it derives temporal components T_0, T_1 and T_2 as

$$T_0 = e(g_l^{\alpha d}, U_0) \cdot e\left(g_l^{\beta}, K_0 \Big/ g_{n-1}^{\alpha^{2^n-1-j+d}}\right)^{-1},$$
$$T_1 = e(g_l^{\beta}, K_1), \quad T_2 = e(g_l^{\alpha^d}, U_1).$$

3) Next, it chooses random exponents $r_1, r_2 \in Z_p$ and re-randomizes these components as $D_0 = T_0 \cdot F_{n+l-1}(r_1 ID_2)^{r_1} H_{n+l-1}(T)^{r_2}$, $D_1 = T_1 \cdot g_{n+rl-1}^{r_1}$,
4) Finally, $D_2 = T_2 \, g_{it_{n+l-1}}$ outputs a decryption key as $DK_{ID,T} = D_0, D_1, D_2$.

RIBE.Encrypt(ID, T, M, PP): This algorithm takes as input an identity ID, a time T, a message M, and the public parameters PP. It chooses a random exponent $s \in Z_p$ and outputs a ciphertext $CT_{ID,T}$ by implicitly including ID and T as

$$C = {}^{s} \cdot M, \; C_0 = g_n{}^{s}{}_{-1}, \; C_1 = F_{n-1}(ID)^s, \; C_2 = H_{n-1}(T)^s.$$

RIBE.Decrypt($CT_{ID,T}$, $DK_{ID,T}$, PP): This algorithm takes as input a ciphertext $CT_{ID,T} = (C, C_0, C_1, C_2)$, a decryption key $DK_{ID,T} = (D_0, D_1, D_2)$, and the public parameters PP. If $(ID = ID') \wedge (T = T)$, then it outputs the encrypted message M as $M = C \cdot \prod_{i=0}^{2} e(C_i, D_i)^{-1}$. Otherwise, it outputs $\perp$.

RIBE.Revoke(ID, T, RL, ST): This algorithm takes as input an identity ID, a revocation time T, the revocation list RL, and the state ST. If $(ID, -) \notin ST$, then it outputs $\perp$ since the private key of ID was not generated. Otherwise, it adds (ID, T) to RL. It outputs the updated revocation list RL.

B. Correctness

We first show that some elements that are needed for the scheme can be easily computed from the elements in PP. We use the following claim of Boneh, Waters, and Zhandry.

Claim [17]: Using group multiplications and pairing operations on the $g_1^{\alpha^{2i}}$ for $i \in [0,n]$, it is possible to compute $g_l^{\alpha^j}$ for $j \in [1, 2^n - 2]$ of weight exactly $2n-1$ j l, $g_{n-1}^{\alpha^{2^n-1-j}}$ for $j \in [1, 2^n - 2]$ of weight exactly l, and $g_n^{\alpha_{-1}^{---+}}$ for $j, u \in [1, 2^n - 2]$, $j \neq u$ of weight exactly l.

The correctness of decryption keys and the decryption algorithm is almost similar to that of Section III. We omit this since the lack of space.

C. Security Analysis

The proof of security is almost similar to that in Theorem 3.1.

Theorem 4.1: The above RIBE scheme is secure in the selective revocation list model under chosen plaintext attacks if the (k,n,l)-cMDHE assumption holds where $N = 2^n - 2$ is the maximum number of users and $k = 2n+l-2$. That is, for any PPT adversary A, we have that $\mathbf{Adv}^{IND\text{-}sRL\text{-}CPA}_{RIBE,A}(\lambda) \leq \mathbf{Adv}^{(k,n,l)\text{-}cMDHE}_{B}(\lambda)$.

Proof: Suppose there exists an adversary A that attacks the above RIBE scheme with a non-negligible advantage. A simulator B that solves the cMDHE assumption using A is given: a challenge tuple $D = (g_1, g_1^{a^0}, g_1^{a^1}, \ldots, g_1^{a^{2n}}, g_1^{a^{2n+2}}, g_1^{a^{2n+3}}, \ldots, g_1^{a^{2n+l}}, g_1^{b}, g_1^{c}, \ldots, g_n^{a^{n-1}bc})$ and Z where $Z = Z_0 = g_{2n+l}^{a^{2n+l-1}bc}$ or $Z = Z_1 \in G_{2n+l}$. Then B that interacts with A is described as follows:

Init: A initially submits a challenge identity ID^*, a challenge time T^*, and a revoked identity set R^* at the time T^*.

It first sets a state ST and a revocation list RL as empty one. For each $ID \in \{ID^*\} \cup R^*$, it selects an index $d \in \mathbb{N}$ with Hamming weight l such that $(-,d) \notin ST$ and adds (ID, d) to ST. Let $RI^* \subseteq \mathbb{N}$ be the revoked index set of R^* at the time T^* and SI^* be the non-revoked index set on the time T^* such that $SI^* = \mathbb{N} \setminus RI^*$.

Setup: B first chooses random exponents f_0, $\{f'_{i,j}\}_{1 \leq i \leq l_1, j \in \{0,1\}}$, h'_0, $\{h'_{i,j}\}_{1 \leq i \leq l_2, j \in \{0,1\}}$, $\vartheta^n \in Z_p$. It implicitly sets $\alpha = a$, $\beta = b$, $\gamma = c$ and publishes the public parameters $PP = \vartheta$ as

$$\left\{g_1^{\alpha^2{}_i} = g_1^{a^{2i}}\right\}_{0 \leq i \leq 2n}, \; g_{l_1} = g^{lb},$$

$$f_{n-1} = f_{n-1,0} = g_n^{f_0 \prod_{i=1}^{n-1} f_{n-1,i,ID^*[i]}}{}^{-1},$$

$$f_{n-1,i,j} = (g_{n-1}^{a^{2^n-2}})_{f_{i,j}}{}_{1 \leq i \leq l_1, j \in \{0,1\}},$$

$$h_{n-1} = h_{n-1,0} = g_{n-1}^{h_0} \prod_{i=1}^{l_2} h_{n-1,i,T^*[i]}^{-1},$$

$$h_{n-1,2,i,j_2} = (g_{n-1}^{b})_{h_{i,j}1 \le i \le 2l_{n-2},1 j \in \{0,1\}},$$

$$= e\left(e(g_1^{a}, g_{n-1}^{a^{n-}}), g_{lb}, g_{n-2}\right) = g_{2\alpha n+-l-b2}.$$

For notational simplicity, we define $^{l}ID = \sum_{i=1}^{l_1}(f_{i,ID}[i] - f_{i,ID^*})$ and $T = {}_{i} {}^{[i]} \not\equiv 0 \mod {}^{2}_{=1}(h'_{i,T[i]} - h'_{i,T^*[i]})$. We have ID except with negligible probability if $ID = ID^*$ since there exists at least one index i such that $f_{i,ID[i]} T = \equiv f_{i,ID}0^*$ and $\{f'_{i}{}_{j}\}_{[i \bmod]}$ p except with negligible probability, are randomly chosen. We also if $T = T_*$.

Phase 1: A adaptively requests a polynomial number of private key, update key, and decryption key queries.

If this is a private key query for an identity ID, then B proceeds as follows:

- *Case* $ID \in R_*$: It first retrieves a tuple (ID,d) from ST where the index d is associated with ID. Note that the tuple (ID,d) exists since all identities in R_* were added to ST in the initialization step. Next, it selects a random exponent $r_1 \in \mathbb{Z}_p$ and creates a private key SK_{ID} as

$$_{0} = \left(g_{n-1}^{a^{d}}\right)\left(\prod_{j \in SI^*}{}^{2^n-1-j+d}_{-1}\right)^{-1} F_{n-1}(ID)^{-}$$

$$\vartheta ar_1, Kg_n$$

$$K^{1} = g_{n-1}^{-r_1}.$$

- *Case* $ID \notin R^*$: In this case, we have $ID = ID^*$ from the restriction of Definition 2.2. It first selects an index $d \in \mathbb{N}$ such that $(-,d) \notin ST$ and adds (ID,d) to ST. Next, it selects a random exponents $r'_1 \in \mathbb{Z}_p$ and creates a private key SK_{ID} by implicitly setting $r_1 = -a/ID + r'_1$ as

$$= g_{n-1}^{a^{d}\theta} \prod_{j \in SI^*} \quad K0g_{n--a12n-1-j+d}$$

$$(g_{na-1})\, f0/ID$$

$$\backslash\{^{d}_{}\}$$

$$F_{n-1}(ID)^{-r'_1},$$

$$K^{1} = (g_{n-1}^{a})^{-1}{}_{/ID} g_{nr-1}^{1}.$$

If this is an update key query for a time T, then B defines a revoked identity set R at the time T from RL and proceeds as follows:

- *Case* $T = T_*$: It first sets a revoked index set RI of R by using ST. It also sets $SI = \mathbb{N} \setminus RI$. Next, it selects a random exponent $r_2 \in \mathbb{Z}_p$ and creates an update key $_nUK_{T,R}$ by implicitly setting $^{2}_{n-}{}^{1}_{-j})/r_2 T = -as(-$ $_{j \in SI_*\backslash SI} a^{2}{}^{-1}_{-j} +$ $_{j \in SI\backslash SI^*} a + r'_2$

$$U0 = (g_{n-1}^{b})^{\theta}$$

$$-a^{2^{n-1}-}{}_{j} \quad g_{n-1} \quad g_{n}a-2^{n} \quad 1 \quad j \quad -h'1 \quad -- \quad) \quad {}^{0}_{/T}$$

$$j \in SI^*\backslash SIj \in SI\backslash SI_*$$

$$\times H_{n-1}(T)^{r}{}_{2,1\,j} \qquad {}^{2n}_{-1-j})^{-1}$$

$$U \quad {}^{1} = \Big(\prod_{j \in SI^*\backslash SI} g_{n--a12n--j \in SI\backslash SI^*} g_{na-1}$$

$$/T$$

$$r^2$$

$$\times g_{n-1}.$$

- *Case* $T = T_*$: In this case, we have $R = R_*$. For each $ID \in R_*$, it adds (ID, T_*) to RL if $(ID, T') \notin RL$ for any $T \le T_*$. Next, it selects a random exponent $r_2 \in \mathbb{Z}_p$ and creates an update key $UK_{T,R}$ as

$$U0 = (g_{n-1}^{b})^{\theta} H_{n-1}(T^*)^{r_2}, \quad U_1 = g_{n-1}^{-r_2}.$$

If this is a decryption key query for an identity ID and a time T, then B proceeds as follows:

- *Case* $ID = ID_*$: If $(ID,-) \in ST$, then it selects an index $d \in \mathbb{N}$ such that $(-,d) \notin ST$ and adds (ID,d) to ST. Next, it selects random exponents $r'_1, r_2 \in \mathbb{Z}_p$ and creates a decryption key $DK_{ID,T}$ by implicitly setting $r_1 = (-a/ID + r'_1)b$ as

$$D \quad {}_{0} = e\left((g_{n-1}^{a})^{-f'_0}{}_{/ID} F_{n-1}(ID)_{r1}, g_{lb} \cdot\right.$$

$$H_{n+l-1}(T)_{r2},$$

$$D^{1} = e\left((g_{n-1}^{a})^{-1}{}_{/ID} g_{n} r^{1}{}_{-1}, g_{lb}\right), D^{2} = g_{n+l-1}^{r_2}.$$

- *Case* $ID = ID_*$: In this case, we have $T = T_*$ from the restriction of Definition 2.2. It selects random exponents $r^{1}, r'_2 \in \mathbb{Z}_p$ and creates a decryption key $DK_{ID,T}$ by implicitly setting $r_2 = (-a/T + r'_2)a^{2^n-2}$ as

$$D \quad {}_{0} = e\left((g_l^{a})^{-h'_0}{}_{/T} H_l(T)^{\,{}'_{2}}, g_{n-1}{}^{n-}_{r\,a2\,2} \cdot F_{n+nl-21}(I\right.$$

$$D)_{r1}, D^{1} = g_{n+l-1}^{r_1}, \quad D_2 = e\left((g_l^{a})^{-1}{}_{/T} g_{lr2}, g_{na-2}\right.$$

$$1-.$$

Note that it can computes $H_l(T)$ since g_l^{b} is given in the assumption.

Challenge: A submits two challenge messages M_{0}^{*}, M_1^{*}. $\mathcal{B}$ chooses a random bit $\delta \in \{0,1\}$ and creates the challenge ciphertext CT_* by implicitly setting $s = c$ as

$$C = Z \cdot M_{\delta}^{*}, \quad C_0 = g_{n-1}^{c}, \quad C_1 = (g_{n-1}^{c})^{f'_0}, \quad C_2 = (g_{n-1}^{c})^{h'_0}.$$

Phase 2: Same as Phase 1.

Guess: Finally, A outputs a guess $\delta' \in \{0,1\}$. $\mathcal{B}$ outputs 0 if $\delta = \delta$ or 1 otherwise.

To finish the proof, we should show that the distribution of the simulation is correct. We omit the analysis of the distribution since the analysis is almost similar to that of Lemma 3.2 except that it uses multilinear maps and the Claim IV-B. This completes our proof. ■

D. Discussions

Asymptotic Analysis: The number of group elements in public parameters, a private key, an update key, and a ciphertext of our second RIBE scheme is $O(\log N + \lambda)$,

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

$O(1)$, $O(1)$, and $O(1)$ respectively, where N is the maximum number of users. However, our RIBE scheme requires k-leveled multilinear maps where $k \approx 2.5\log N$ since the BWZ-PKBE scheme requires $1.5\log N$-leveled multilinear maps [17]. If we use the improved multilinear maps of Langlois *et al.* [32], the bit size of the group elements in k-leveled multilinear maps is $O(k^2\lambda\log^2(k\lambda))$. Let $\lambda = 80$ and $N = 2^{20}$. The bit size of the group elements in $2.5\log N$-leveled multilinear maps is approximately $2.4 * 10^5$. In the BGK-RIBE scheme [7], the bit size of a private key and an update key is approximately $3.2 * 10^2$ and $1.6 * 10^6$ respectively, where the number of revoked users is $r = 2^{10}$. Therefore, our second RIBE scheme equipped with the currently best leveled multilinear maps [32] does not provide better parameters except the bit size of update keys. However, we expect that the parameters of leveled multilinear maps will be improved in the near future.

V. Conclusion

In this paper, we devised a new technique for RIBE that uses multilinear maps to combine an IBE scheme with a PKBE scheme. Following our technique, we first proposed an RIBE scheme with a constant number of private key elements and update key elements by combining the HIBE scheme of Boneh and Boyen [16] and the BGW-PKBE scheme [13], and then we proved its security in the selective revocation list model. Next, we proposed another RIBE scheme that reduces the number of public parameters from $O(N + \lambda)$ to $O(\log N + \lambda)$ group elements by using the BWZ-PKBE scheme [17], which has short public parameters. We expect that our technique will open a new direction to build an efficient RIBE scheme and its extensions.

There are many interesting unsolved problems in RIBE. The first one is to construct an RIBE scheme with short parameters and short keys that is secure in the adaptive security model instead of in the selective revocation list model. The second one is to construct a revocable HIBE (RHIBE) scheme with better parameters. RHIBE provides the private key delegation functionality and the revocation functionality for each user. The RHIBE scheme of Seo and Emura [11] has $O(l^2 \log N)$ number of private key elements and $O(r \log(N/r))$ number of update key elements where l is the depth of hierarchy, N is the maximum number of users, and r is the maximum number of revoked users. The third one is to build an RIBE scheme with a constant number of private key elements and update key elements that can handle the exponential number of users in the system. Recall that our second RIBE scheme cannot handle the exponential number of users since the size of receiver set in the BWZ-PKBE scheme is restricted to being polynomial.

Appendix A Security in Generic Multilinear Groups

In this section, we introduce the definition of generic multilinear groups and discuss the difficulty of our new assumptions in generic multilinear groups.

$= O(n2$ *A. Generic Multilinear*
$_2$ *Groups*

We define the generic multilinear groups by following the generic group model [34], [35]. Let k be the target integer. Let $\xi : \mathsf{Z}_p \times \mathsf{Z} \to \{0,1\}^m$ be a random injective encoding that maps elements of the additive group Z_p and an integer Z into strings of length m. We define the groups $\mathsf{G}_i = \{\xi(x,i) | x \in \mathsf{Z}_p\}$. We are given oracles to compute the multiplication and pairing operations. That is, an algorithm in the generic multilinear groups is given the following oracles:

***Encode**(x, i):* If i is a non-negative integer such that $i \leq k$, then it returns $\xi(x,i)$. Otherwise it returns $\perp$. Note that the generator g_i for the group G_i can be obtained as **Encode**$(1,i)$. ***Mult**(ξ_1,ξ_2,b):* If $\xi_1 = \xi(x_1,i)$ and $\xi_2 = \xi(x_2, j)$ where $i = j$, then it returns $\xi(x_1 + (-1)^b x_2, i)$. Otherwise, it returns $\perp$.

***Pair**(ξ_1,ξ_2):* If $\xi_1 = \xi(x_1,i)$ and $\xi_2 = \xi(x_2, j)$ where $i + j \leq k$, then it returns $\xi(x_1 \cdot x_2, i + j)$. Otherwise it returns $\perp$.

B. Analysis of New Assumptions

The master theorem of Boneh*et al.* [35] is widely used to prove the difficulty of an assumption in generic bilinear groups. It is relatively straightforward to extend the master theorem of Boneh*et al.* in generic multilinear groups as pointed by Boneh*et al.* [17]. The master theorem informally states that if the target polynomial is independent of given polynomials in the assumption, then the advantage of an adversary in generic groups is bounded by q^2d/p, where q is the maximum number of queries, d is the maximum degree of polynomials that the adversary can obtain by performing pairing operations, and p is in Z_p.

In the (k, N)-MDHE assumption, the target polynomial f is $a_{N+1} \; k_i=-1 \; 1 \; c_i$ where a and(pairings) to obtain$c_i$are variables. We need two$a^{N}{}_{+}{}^{1}$ since polynomial multiplications
$a^{N}{}_{+}{}^{1}$ is not directly given in the assumption, and we need $k-1$
polynomial multiplications to obtain$k+1$ polynomial multiplications (pairings) to obtain the target$^{k}{}_{i=}{}^{-}{}_{1}{}^{1}$ c_i. Thus, we need
polynomial, but this is not allowed in the k-leveled multilinear maps. Therefore, the target polynomial is independent of given polynomials. We have the degree of polynomials $d = O(kN)$ since the adversary can obtain elements with high-degree $a^{kN}$by performing pairing operations. For λ-bit security, we can set $p \approx 2^{\lambda}$ since N is a polynomial value in a security parameter. The difficulty of the assumption of Boneh*et al.* [17] is

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

already given in generic multilinear groups. We can also follow their analysis since our cMDHE assumption is a slight modification of their assumption. In the *(k,n,l)*-cMDHE assumption, the target polynomial f is $a^{2n-1}bc$ where a, b, and c are variables. We need n polynomial multiplications (pairings) to obtain a^{2n-1} since $\{a^{2i}\}_{i\in[0,n]}$ are only- given in the assumption. Thus the target polynomial $a^{2n-1}bc$ should reside in $2n+l-1$-level since b is a polynomial in the l-level and c is a polynomial in the $n-1$-level, but this is not allowed in the $2n+l-2$-leveled multilinear maps. Therefore, the target polynomial is independent of given polynomials. We have $d^{n})$ since the assumption includes elements $_{n}{}^{3\lambda}$ with high degree a . For λ-bit security, we can set $p\approx 2$ instead of $p \approx 2^{\lambda}$.

REFERENCES

[1] S. Micali, "Efficient certificate revocation," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. MIT/LCS/TM-542b, 1996.

[2] W. Aiello, S. Lodha, and R. Ostrovsky, "Fast digital identity revocation (extended abstract)," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1462, H. Krawczyk, Ed. Berlin, Germany: Springer-Verlag, 1998, pp. 137–152.

[3] M. Naor and K. Nissim, "Certificate revocation and certificate update," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 561–570, Apr. 2000.

[4] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2656, E. Biham, Ed. Berlin, Germany: Springer-Verlag, 2003, pp. 272–293.

[5] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 196, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer-Verlag, 1984, pp. 47–53.

[6] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2139, J. Kilian, Ed. Berlin, Germany: Springer-Verlag, 2001, pp. 213–229.

[7] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, 2008, pp. 417–426.

[8] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2139, J. Kilian, Ed. Berlin, Germany: Springer-Verlag, 2001, pp. 41–62.

[9] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.

[10] B. Libert and D. Vergnaud, "Adaptive-ID secure revocable identitybased encryption," in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 5473, M. Fischlin, Ed. Berlin, Germany: Springer-Verlag, 2009, pp. 1–15.

[11] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction," in *Public-Key Cryptography* (Lecture Notes in Computer Science), vol. 7778, K. Kurosawa and G. Hanaoka, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 216–234.

[12] J. H. Seo and K. Emura, "Efficient delegation of key generation and revocation functionalities in identity-based encryption," in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 7779, E. Dawson, Ed. Berlin, Germany: Springer-Verlag, 2013, pp. 343–358.

[13] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3621, V. Shoup, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 258–275.

[14] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 5479, A. Joux, Ed. Berlin, Germany:
Springer-Verlag, 2009, pp. 171–188.

[15] S. Garg, C. Gentry, and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 7881, T. Johansson and P. Q. Nguyen, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 1–17.

[16] D. Boneh and X. Boyen, "Efficient selective-ID secure identitybased encryption without random oracles," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3027, C. Cachin and J. L. Camenisch, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 223–238.

[17] D. Boneh, B. Waters, and M. Zhandry, "Low overhead broadcast encryption from multilinear maps," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 8616, J. A. Garay and R. Gennaro, Eds. Berlin, Germany: Springer-Verlag, 2014, pp. 206–223.

[18] K. Lee, D. H. Lee, and J. H. Park. (2014). "Efficient revocable identity-based encryption via subset difference methods," Cryptol. ePrint Arch., Tech. Rep. 2014/132. [Online]. Available: http://eprint.iacr.org/2014/132

[19] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 114–127.

[20] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2332, L. R. Knudsen, Ed. Berlin, Germany: Springer-Verlag, 2002, pp. 466–481.

[21] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2501, Y. Zheng, Ed. Berlin, Germany: Springer-Verlag, 2002, pp. 548–566.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[23] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 4392, S. P. Vadhan, Ed. Berlin, Germany: Springer-Verlag, 2007, pp. 535–554.

[24] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 6597, Y. Ishai, Ed. Berlin, Germany: Springer-Verlag, 2011, pp. 253–273.

[25] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Functional encryption with bounded collusions via multi-party computation," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 7417,
R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 162–179.

[26] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Pairing-Based Cryptography* (Lecture Notes in Computer Science), vol. 5671, H. Shacham and B. Waters, Eds. Berlin, Germany: Springer-Verlag, 2009, pp. 248–265.

[27] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 199–217.

[28] K. Lee, S. G. Choi, D. H. Lee, J. H. Park, and M. Yung, "Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 8269, K.

ISSN: 2278-4632
Vol-10 Issue-09 No.03 September 2020

Sako and P. Sarkar, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 235–254.

[29] A. Boldyreva, V. Goyal, and V. Kumar. (2012). "Identity-based encryption with efficient revocation," Cryptol. ePrint Arch., Tech. Rep. 2012/052. [Online]. Available: http://eprint.iacr.org/2012/052

[30] D. Boneh and A. Silverberg, "Applications of multilinear forms to cryptography," *Contemp. Math.*, vol. 324, no. 1, pp. 71–90, 2003.

[31] S. Park, K. Lee, and D. H. Lee. (2013). "New constructions of revocable identity-based encryption from multilinear maps," Cryptol. ePrint Arch.,
Tech. Rep. 2013/880. [Online]. Available: http://eprint.iacr.org/2013/880

[32] A. Langlois, D. Stehlé, and R. Steinfeld, "GGHLite: More efficient multilinear maps from ideal lattices," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 8441, P. Q. Nguyen and E. Oswald, Eds. Berlin, Germany: Springer-Verlag, 2014, pp. 239–256.

[33] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3027, C. Cachin and J. L. Camenisch, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 207–222.

[34] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1233, W. Fumy, Ed. Berlin, Germany: Springer-Verlag, 1997, pp. 256–266.

[35] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 440–456.