Fully Collusion-Resistant Trace-and-Revoke Scheme in Prime-Order Groups

Jong Hwan Park, Hyun Sook Rhee, and Dong Hoon Lee

Abstract: **A trace-and-revoke scheme is a type of broadcast encryption scheme for content protection on various platforms such as pay-per-view TV and DVD players. In 2006, Boneh and Waters (BW) presented a fully collusion-resistant trace-and-revoke scheme. However, a decisive drawback of their scheme is to require composite-order groups. In this paper, we present a new trace-and-revoke scheme that works in prime-order groups. Our scheme is fully collusion-resistant and achieves ciphertexts and private keys of size $O(\sqrt{N})$ for N users. For the same level of security, our scheme is better than the BW scheme in all aspects of efficiency. Some superior features include 8.5 times faster encryption, 12 times faster decryption, and 3.4 times shorter ciphertexts. To achieve our goal, we introduce a novel technique where, by using asymmetric bilinear maps in prime-order groups, the cancellation effect same as in composite-order groups can be obtained.**

Index Terms: **Bilinear maps, broadcast encryption, content distribution system, trace-and-revoke scheme.**

I. INTRODUCTION

A trace-and-revoke scheme [1] is a type of broadcast encryption scheme for content protection on various platforms such as pay-per-view TV and DVD players. One typical method to protect content is to encrypt messages that are broadcast by a content distributor and to let only authorized users decrypt resultant ciphertexts. If a user subscribes to a pay-per-view TV system, the user will be given a distinct set of decryption keys that could be stored into a device, e.g., a set-top box. The set of decryption keys, which enables to watch programs that the TV system offers, contains a user-specific identifier such as a user index $i \in \{1, \cdots, N\}$ where N is the total number of users. Using those indices, a content distributor (adopting a trace-and-revoke scheme) not only specifies some subset of users $S \subseteq \{1, \cdots, N\}$ who are authorized to recover encrypted content, but also traces the source identifiers $\{i\}$ of decryption keys that build a pirate decoder against the pay-per-view TV system.

Basically, a trace-and-revoke scheme provides two functionalities of revocation [2] and traitor-tracing [3]. When the scheme is first set up, a content distributor encrypts contents to a subset of users $S \subseteq \{1, \cdots, N\}$ and broadcasts ciphertexts using the revocation functionality. Later, when a pirate decoder is found, the tracing scheme interacts with the pirate decoder and identifies a set of users $T \subseteq \{1, \cdots, N\}$ whose decryption keys contribute to building the pirate decoder. Once a user $u \in T$ is revealed as being responsible for the pirate decoder, u is revoked from future broadcasts and the content distributor can take legal action against the user u.

A trace-and-revoke scheme is said to be fully collusion-resistant if the scheme is still secure against any number of colluders who wish to break revocation or traitor-tracing mechanism. Several trace-and-revoke schemes [4]–[6] were suggested in a fully collusion-resistant manner, and other schemes [1], [7]–[9] were designed for the case where the number of collusions is only fewer than t. The latter schemes become insecure as soon as more than t revoked users work together for breaking the revocation or a pirate decoder uses more than t user keys for attacking the traitor-tracing. Naturally, the content distributor wants to obtain a trace-and-revoke scheme that is able to resist arbitrary collusion.

In constructing trace-and-revoke schemes, one of main challenges is to reduce ciphertext size. This is because, similar to broadcast encryption schemes [2], a normal trace-and-revoke scheme should be suitable for applications with a large number of users. For instance, a pay-per-view TV system with $N = 10^6$ users has to send an encrypted content to almost all users in the end even thought each user wants to see the content at a different time. In this case, the total transmission cost regarding 10^6 users is an important factor to the system networks. Other challenge is to obtain a tracing time as short as possible. For a given pirate decoder, the traitor-tracing algorithm tries to identify a set of traitors $T \subseteq \{1, \cdots, N\}$, under the consideration that each user could be a potential traitor in T. This means that the tracing process is performed per each user i for $i = 1$ to N. In this case, for a large N, the total tracing time should be within a reasonable bound such as a few days or one weak. Otherwise, a trace-and-revoke scheme must end up being applied to broadcast systems with smaller N.

Until now, there has been only one trace-and-revoke scheme [10] that is fully collusion-resistant and simultaneously achieves ciphertexts and private keys of size $O(\sqrt{N})$. The scheme [10] by Boneh and Waters (BW) introduced a new elegant technique to obtain a different class of trace-and-revoke scheme, but a drawback is that the BW scheme relies on *composite-order* groups equipped with bilinear maps (i.e., pairings). In general, a composite order should be at least 1024 bits long at current Rivest-Shamir-Adleman (RSA) security levels. This large size of the group makes the BW scheme impractical. For instance, we cannot take advantage of representing a group el-

Manuscript received April 9, 2010; approved for publication by Chae Hoon Lim, Division I Editor, October 26, 2010 .

This work was supported by a grant from the College of Applied Science, Kyung Hee University Research Professor fellowship, Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (no. 2010-0003388), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (no. 2010-0029121).

J. H. Park is with the Department of Applied Mathematics, College of Applied Science, Kyung Hee University, Youngin, Korea, email: jonghpark@khu.ac.kr.

H. S. Rhee and D. H. Lee are with the Graduate School of Information Security, Korea University, Seoul, Korea, and D. H. Lee is the corresponding author, email: hyunsook.rhee@gmail.com, donghlee@korea.ac.kr.

1229-2370/11/$10.00 © 2011 KICS

ement as about 170 bits, which is the main merit in pairing-based groups, and thus the ciphertext size in composite-order groups becomes longer than one in prime-order groups. Also, one exponentiation in composite-order groups is about 25 times slower than one in prime-order groups, and one pairing operation in composite-order groups is about 30 times costlier than one in prime-order groups [11]. These two operations also make encryption/decryption/tracing algorithms in composite-order groups slower than those in prime-order groups.

A natural direction of research is to construct a trace-and-revoke scheme that works in prime-order groups, while preserving the elegant efficiency of the BW scheme. Clearly, there has been no known result of such a scheme. In this paper, we present a fully collusion-resistant trace-and-revoke scheme that achieves ciphertexts and private keys of size $O(\sqrt{N})$. Our new scheme is based on prime-order groups, and more efficient than the BW scheme in terms of all efficiency aspects for the same level of security. More precisely, the ciphertext is approximately 3.4 times shorter, encryption is roughly 8.5 times faster, and decryption is about 12 times faster. These results also allows for a faster tracing time than the BW scheme. To achieve the efficiency results, we introduce a novel technique for achieving the same cancellation effect as in composite-order groups by using asymmetric bilinear maps in prime-order groups. Our technique is to generate positive and negative pairing values and cancel them out in a natural way, which is new and simple in settings where user indices are arranged in a $\sqrt{N} \times \sqrt{N}$ matrix.

Related work: Broadcast encryption scheme [2] allows a content distributor to send encrypted messages to a set of legitimate users, and disallows revoked users to recover the messages even if they collude. Several broadcast encryption schemes [4]–[6], [12], [13] were secure against arbitrary number of collusion, and particularly only two schemes [12], [13], which are all pairing-based, have ciphertext size sub-linear in the number of revoked users.

On the other hand, a tractor tracing scheme [3] is a type of detection scheme that aims at identifying at least one of traitors whose secret key is used to create a pirate decoder. Two traitor tracing schemes [11], [14] have been proposed to resist arbitrary collusion and also to achieve ciphertexts of size $O(\sqrt{N})$. The scheme [14] was constructed in composite-order groups, but the scheme [11] in prime-order groups.

A trace-and-revoke scheme provides two functions of the broadcast encryption as well as the traitor tracing. Several trace-and-revoke schemes [1], [7]–[9] were designed for resisting any r ($< t$) collusions, and other trace-and-revoke schemes [4]–[6] were fully collusion-resistant. Recently, BW [10] suggested a trace-and-revoke scheme by using an algebraic property of composite-order groups. The BW scheme is the first one that is fully collusion-resistant and also achieves ciphertexts and private keys of size $O(\sqrt{N})$. However, as mentioned above, their scheme was constructed using composite-order groups.

II. PRELIMINARIES

A. Augmented Broadcast Encryption

The goal of this paper is to construct a fully collusion-resistant trace-and-revoke scheme. However, as in the approach of BW [10], we first build a simple primitive called augmented broadcast encryption (ABE) scheme, and then extend it to implement a trace-and-revoke scheme.

An ABE scheme consists of three algorithms: $\mathbf{Setup}_{\mathrm{ABE}}$, $\mathbf{Encrypt}_{\mathrm{ABE}}$, and $\mathbf{Decrypt}_{\mathrm{ABE}}$.

- $(\mathsf{PK}, K_1, K_2, \cdots, K_N) \leftarrow \mathbf{Setup}_{\mathrm{ABE}}(\lambda, N)$: The setup algorithm takes as input a security parameter λ and the number N of users in the scheme. The setup algorithm outputs a public key PK for the scheme and private keys $(K_1, \cdots, K_N)$ where K_u is given to the user u.
- $\mathsf{CT} \leftarrow \mathbf{Encrypt}_{\mathrm{ABE}}(S, \mathsf{PK}, i, M)$: The encryption algorithm takes as input a subset of users $S \subseteq \{1, \cdots, N\}$, the public key PK, an integer i satisfying $1 \leq i \leq N+1$, a message M. The encryption algorithm outputs a ciphertext CT. This algorithm encrypts the message to a set $S \cap \{i, \cdots, N\}$.
- $M \leftarrow \mathbf{Decrypt}_{\mathrm{ABE}}(\mathsf{CT}, K_j, S)$: The decryption algorithm takes as input a ciphertext CT, a private key K_j for user j, and a subset of users $S \subseteq \{1, \cdots, N\}$. This algorithm outputs a message M or $\perp$.

Correctness: The ABE scheme must satisfy the following correctness property: For all subsets $S \subseteq \{1, \cdots, N\}$, all $i, j \in \{1, \cdots, N+1\}$ (where $j \leq N$), and all messages M:

$$\begin{aligned}&\text{Let } (\mathsf{PK}, K_1, K_2, \cdots, K_N) \leftarrow \mathbf{Setup}_{\mathrm{ABE}}(\lambda, N) \text{ and}\\ &\qquad \mathsf{CT} \leftarrow \mathbf{Encrypt}_{\mathrm{ABE}}(S, \mathsf{PK}, i, M).\\ &\text{If } j \in S \text{ and } j \geq i,\\ &\qquad \text{then } M \leftarrow \mathbf{Decrypt}_{\mathrm{ABE}}(\mathsf{CT}, K_j, S).\end{aligned}$$

B. Security for ABE

Following [10], we define chosen-plaintext security for ABE by describing two games that consist of message hiding and index hiding.

Index hiding: This property requires that an adversary cannot distinguish between an encryption to index i and one to index $i+1$ without the key K_i. Additionally, it says that an adversary cannot distinguish between an encryption to index i and one to index $i+1$ when i is not in the target set S even with the key K_i. The game takes as input an index $i \subseteq \{1, \cdots, N\}$ which is given to both the challenger and the adversary. The game between the adversary and the challenger proceeds as follows.

- **Setup**: The challenger runs the setup algorithm and gives the adversary PK and the set of private keys $\{K_j \text{ s.t. } j \neq i\}$.
- **Query:** The adversary outputs a bit $s \in \{0, 1\}$. If $\tilde{s} = 1$, the challenger sends K_i to the adversary. Otherwise the challenger does nothing.
- **Challenge:** The adversary gives the challenger a set $S \subseteq \{1, \cdots, N\}$ and a message M. The only restriction is that if $\tilde{s} = 1$ then $i \notin S$. The challenger picks a random bit $\beta \in \{0, 1\}$ and sends a ciphertext $\mathsf{CT} \leftarrow \mathbf{Encrypt}_{\mathrm{ABE}}(S, \mathsf{PK}, i + \beta, M)$ to the adversary.
- **Guess:** The adversary returns a guess $\beta' \in \{0, 1\}$ of β.

The advantage of the adversary $\mathcal{A}$ is $\mathbf{Adv}_{\mathcal{A},IH}[i] = |\Pr[\beta' = \beta] - \frac{1}{2}|$.

Message hiding: This property requires that an adversary can not break semantic security when encryption is performed on input $i = N+1$. The game between the adversary and the challenger proceeds as follows.

- **Setup**: The challenger runs the setup algorithm and sends the generated public key PK and the secret keys K_1, K_2, $\cdots$, K_N to the adversary.
- **Challenge:** The adversary outputs a set $S \subseteq \{1, \cdots, N\}$ and two equal length messages M_0, M_1. The challenger picks a random bit $\beta \in \{0,1\}$ and sends a ciphertext $\mathsf{CT} \leftarrow \mathbf{Encrypt}_{\mathrm{ABE}}(S, \mathsf{PK}, N+1, M_\beta)$ to the adversary.
- **Guess:** The adversary returns a guess $\beta' \in \{0,1\}$ of β.

The advantage of the adversary $\mathcal{A}$ is $\mathbf{Adv}_{\mathcal{A},MH} = |\Pr[\beta' = \beta] - \frac{1}{2}|$.

Definition 1: We say that an N-user ABE scheme is secure if for all polynomial time adversaries $\mathcal{A}$ we have that $\mathbf{Adv}_{\mathcal{A},IH}[i]$ for all $i \in \{1, \cdots, N\}$ and $\mathbf{Adv}_{\mathcal{A},MH}$ are negligible in the security parameter λ.

C. A Trace-and-Revoke Scheme Using ABE

Given a secure ABE scheme defined above, BW [10] showed that one can obtain a secure trace-and-revoke scheme $\mathcal{TR} = (\mathbf{Setup}_{\mathrm{ABE}}, \mathbf{Encrypt}, \mathbf{Decrypt}_{\mathrm{ABE}}, \mathbf{Trace}^{\mathcal{D}})$, where two new encryption and tracing algorithms are defined as follows.

- $\mathbf{Encrypt}(S, \mathsf{PK}, M) := \mathbf{Encrypt}_{\mathrm{ABE}}(S, \mathsf{PK}, 1, M)$. This means that the message M can be decrypted by users in $S \cap \{1, \cdots, n\}$, that is, users in S.
- Given a pirate decoder $\mathcal{D}$ that will decrypt all ciphertexts encrypted for a certain set $S_{\mathcal{D}}$, the goal of the $\mathbf{Trace}^{\mathcal{D}}$ is to detect from $\mathcal{D}$ at least one of users $u \in S_{\mathcal{D}}$ whose keys were used to construct $\mathcal{D}$. More precisely, $\mathbf{Trace}^{\mathcal{D}}(S_{\mathcal{D}}, \mathsf{PK}, \epsilon)$ takes as input a certain set $S_{\mathcal{D}}$, public key PK, and a given $\epsilon > 0$ (where $\epsilon = 1/f(\lambda)$ for some polynomial f). Here, ϵ is the at least probability with which $\mathcal{D}$ decrypts ciphertexts validly. Then, the tracing algorithm $\mathbf{Trace}^{\mathcal{D}}(S_{\mathcal{D}}, \mathsf{PK}, \epsilon)$ works as follows.

```
1. Initialize set $T$ to the empty set.
2. For $i = 1$ to $N$, do the following:
   (a) For $8\lambda(N/\epsilon)^2$ times, the algorithm repeats the following steps:
       i. Sample $M$ from the finite message space at random.
       ii. Let $C \xleftarrow{R} \mathbf{Encrypt}_{\mathrm{ABE}}(S_{\mathcal{D}}, \mathsf{PK}, i, M)$.
       iii. Call oracle $\mathcal{D}$ on input $C$, and compare the output of $\mathcal{D}$ to $M$.
   (b) Let $\widehat{p}_i$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
   (c) If $\widehat{p}_i - \widehat{p}_{i+1} \geq \epsilon/(4N)$, then add $i$ to set $T$.
3. Output the set $T$.
```

Regarding the security of the trace-and-revoke scheme, we consider two games: **Message hiding game** and **tracing game**. The former is similar to that of ABE, and the latter is to ensure that the tracing algorithm successfully traces any pirate decoder $\mathcal{D}$. We refer to [10] for the formal treatment of these security games. BW [10] showed that a secure ABE scheme implies a trace-and-revoke scheme that is secure against adaptive adversaries under both the security games.

In the tracing game, an adversary outputs $(\mathcal{D}, S_{\mathcal{D}})$ where $\mathcal{D}$ will decrypt all ciphertexts encrypted for the set $S_{\mathcal{D}}$. Then, the $\mathbf{Trace}^{\mathcal{D}}$ interacts with $\mathcal{D}$ as described in the above routine, and tries to detect one of the keys $u \in S_{\mathcal{D}}$ that was used to build $\mathcal{D}$. Once such u is extracted, the tracing algorithm sets $S' = S_{\mathcal{D}} \setminus \{u\}$ and encrypts messages to S'. If $\mathcal{D}$ can still decrypt ciphertexts, the tracing algorithm will be again performed against $(\mathcal{D}, S')$, thereby to extract another of the pirate's keys in $S_{\mathcal{D}}$. In this way, we can run the tracing algorithm repeatedly until $\mathcal{D}$ does not work, by further shrinking S'.

D. Asymmetric Bilinear Maps and Complexity Assumptions

We briefly review asymmetric bilinear maps and define complexity assumptions necessary for our security proofs.

Asymmetric bilinear maps: We follow the standard notation in [15] and [16]. Let G_1, G_2, and G_T be three (multiplicative) cyclic groups of prime order p. Let $g \in G_1$ and $h \in G_2$. Let $e : G_1 \times G_2 \rightarrow G_T$ be a function that has the following properties.

1. Bilinear: For all $g \in G_1$, all $h \in G_2$, and $a, b \in Z_p$, we have $e(g^a, h^b) = e(g, h)^{ab}$.
2. Non-degenerate: If g generates G_1 and h generates G_2, then $e(g, h)$ generates G_T.
3. Computable: There is an efficient algorithm to compute the map e.

Throughout the paper, we assume that no efficiently computable isomorphism exists between G_2 and G_1. In fact, there exists isomorphisms between two groups since they are cyclic groups of the same order. However, according to [17] and [18], we can obtain such groups where computing these isomorphisms is presumably as hard as computing discrete logarithms. Based on the nonexistence of such efficient isomorphisms, the following complexity assumptions hold:

Asymmetric decision 3-party Diffie-Hellman problem: The asymmetric decision 3-party Diffie-Hellman problem [19] is defined as follows. Given $(g,\ g^a,\ g^b,\ h,\ h^a,\ h^{ab},\ h^c,\ T) \in G_1^3 \times G_2^5$ as input, determine whether $T = h^{abc}$ or T is random in G_2.

External Diffie-Hellman (XDH) problem: The XDH problem [17], [18], [20]–[23] states that the decision Diffie-Hellman (DDH) problem is hard in G_1. More precisely, the XDH problem is defined as follows: Given $(g, g^a, g^b, h, T) \in G_1^3 \times G_2 \times G_1$ as input, determine whether $T = g^{ab} \in G_1$ or T is random in G_1.

Definition 2: We say that the {asymmetric decision 3-party Diffie-Hellman, XDH} assumption holds in $G_1 \times G_2$ if the advantage of any polynomial time algorithm in solving the {asymmetric decision 3-party Diffie-Hellman, XDH} problem is negligible.

III. OUR ABE CONSTRUCTION

We assume that the number N of users in the ABE scheme is equal to m^2 for some positive integer m. As stated in [14], if N is not a square, then we can add some dummy users in order to construct the perfect square. For a user $i \in \{1, \cdots, N\}$, let x and y be two positive integers such that $i = (x-1)m+y$, where $1 \leq x \leq m$ and $1 \leq y \leq m$. Then, the user i is identified as an entry (x, y) of an $m \times m$ matrix. Also, the private key for the user i, i.e., (x, y), is uniquely generated by using secret values involved with both row x and column y.

Based on the $m \times m$ matrix, a ciphertext consists of row ciphertexts $(R_{x,1},\ R_{x,2},\ R_{x,3},\ R_{x,4},\ B_x)$ for each row $x \in$

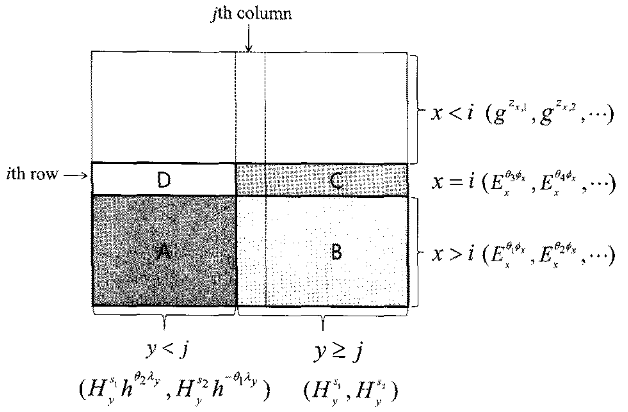

*j*th column
*i*th row
D
C
A
B
$x < i \ (g^{z_{x,1}}, g^{z_{x,2}}, \cdots)$
$x = i \ (E_x^{\theta_3\phi_x}, E_x^{\theta_4\phi_x}, \cdots)$
$x > i \ (E_x^{\theta_1\phi_x}, E_x^{\theta_2\phi_x}, \cdots)$
$y < j$
$(H_y^{s_1} h^{\theta_2\lambda_y}, H_y^{s_2} h^{-\theta_1\lambda_y})$
$y \geq j$
$(H_y^{s_1}, H_y^{s_2})$

Fig. 1. Row and column ciphertext components when encrypting to position (i, j).

$\{1, \cdots, m\}$ and column ciphertexts $(C_{y,1}, C_{y,2})$ for each column $y \in \{1, \cdots, m\}$. Thus, the total ciphertext contains $7m$ group elements. In decryption, each user needs only row and column ciphertexts corresponding to its own entry (x, y).

A. Our Approach

The basic idea behind our ABE construction is the same as that of BW [10], where two private keys of broadcast encryption scheme [12][1] and traitor-tracing scheme [14] are multiplied in order to prevent colluding users from decomposing other valid keys. In our construction, the technique of broadcast encryption scheme is the same as that of [10], but the technique of traitor-tracing scheme is new and different from that of [10]. The main difference between two tracing schemes is that the BW-ABE scheme must be based on composite-order groups, but our tracing scheme is based on prime-order groups. Nevertheless, the effect obtained is same.

We explain such difference more concretely. When encrypting messages to a certain set S and an index (i, j), the encryption algorithm of ABE outputs a ciphertext that can be decrypted by any user in S as well as parts A, B, and C in Fig. 1. Since column ciphertexts for part C and part D are correlated with *common* row ciphertexts for $x = i$, we can see that the column ciphertexts for part C should be constructed differently from those for part D. That is, there must be different factors in the column ciphertexts between part C and D. In our construction, such factors are $(h^{\theta_2\lambda_y}, h^{-\theta_1\lambda_y})$ for $y < j$ in Fig. 1, where all exponents are randomly selected. These two elements will work to prevent users in part D from successfully decrypting ciphertexts. However, such factors should not make any difference for users in both parts A and B, even when correlated with common row ciphertexts $(E_x^{\theta_1\phi_x}, E_x^{\theta_2\phi_x}, \cdots)$ for $x > i$ (where all exponents are randomly chosen). In fact, solving this paradoxical problem is the key for providing the tracing property in the $m \times m$ matrix setting. Fortunately, we can see that $(h^{\theta_2\lambda_y}, h^{-\theta_1\lambda_y})$ for $y < j$ are *canceled out* when corresponding ciphertexts components are paired in bilinear maps. Such cancellation can be checked via the following equation.

$$\begin{aligned} e(E_x^{\theta_1\phi_x},\ H_y^{s_1} h^{\theta_2\lambda_y}) e(E_x^{\theta_2\phi_x},\ H_y^{s_2} h^{-\theta_1\lambda_y}) \\ = e(E_x^{\theta_1\phi_x},\ H_y^{s_1}) e(E_x^{\theta_2\phi_x},\ H_y^{s_2}) \end{aligned}$$

where the left-hand side indicates computation for part A and the right-hand side indicates computation for part B. The important point is that the additional factors $h^{\theta_2\lambda_y}$ and $h^{-\theta_1\lambda_y}$, which need to hinder decryption of part D, are canceled out by generating positive and negative parings values. Such a cancellation effect can be similarly obtained in [10] and [14] by using composite-order groups. Let $G = G_p \times G_q$ be a composite-order group for two primes p and q, and let $g_p \in G_p$ and $g_q \in G_q$. Briefly speaking, the column ciphertexts are generated as $(g_p g_q)^{s_y} g_p^{\lambda_y}$ for $y < j$ and $(g_p g_q)^{s_y}$ for $y \geq j$ where all exponents are randomly chosen. Note that $g_p^{\lambda_y}$ makes difference like $(h^{\theta_2\lambda_y}, h^{-\theta_1\lambda_y})$ in our construction. If row ciphertexts for $x > i$ are generated by $g_q^{\phi_x}$, then the additional factor $g_p^{\lambda_y}$ is cancelled out by virtue of the property such that $e(g_p, g_q) = 1$. Finally, in any case, row ciphertexts for $x < i$ are generated in dummy forms by using randomly selected exponents such as $z_{x,1}$ and $z_{x,2}$ in Fig. 1.

B. Scheme

- **Setup**$_{\text{ABE}}(\lambda, N = m^2)$: Given a security parameter $\lambda \in Z^+$ and the number N of total users in the system, the setup algorithm runs a generation algorithm $\mathcal{G}(\lambda)$ (for bilinear groups and pairings) to obtain a tuple (p, G_1, G_2, G_T, e). The algorithm picks random group elements $g \in G_1$, $h \in G_2$, and random exponents $\{r_i, c_i, \alpha_i, \beta_i\}_{i=1}^m$ in Z_p. The algorithm sets

$$\begin{aligned} E_1 &= g^{r_1}, \cdots, E_m = g^{r_m}, \\ H_1 &= h^{c_1}, \cdots, H_m = h^{c_m}, \\ U_1 &= g^{\beta_1}, \cdots, U_m = g^{\beta_m}, \\ V_1 &= h^{\beta_1}, \cdots, V_m = h^{\beta_m}, \\ \Lambda_1 &= e(g, h)^{\alpha_1}, \cdots, \Lambda_m = e(g, h)^{\alpha_m}. \end{aligned}$$

The public key PK with the description of (p, G_1, G_2, G_T, e) is given by

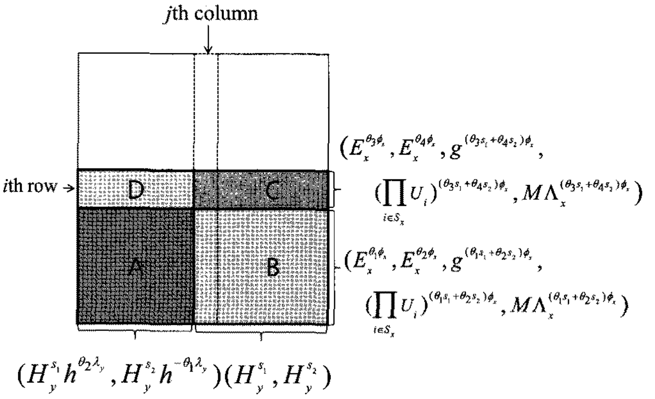
$$\begin{aligned} \mathsf{PK} = \Big(g,\ h,\ \{E_i,\ H_i,\ U_i,\ V_i,\ \Lambda_i\}_{i=1}^m\Big) \\ \in G_1^{2m+1} \times G_2^{2m+1} \times G_T^m. \end{aligned}$$

To generate the private key $K_{(x,y)}$ for user (x, y), the algorithm picks a random exponent $\sigma_{x,y} \in Z_p$ and generates the key as follows.

$$\begin{aligned} K_{(x,y)} &= (d'_{x,y},\ d''_{x,y},\ d_1, \cdots, d_{y-1},\ d_{y+1}, \cdots, d_m) \\ &= \Big(h^{\alpha_x} h^{r_x c_y} V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots,\ V_{y-1}^{\sigma_{x,y}}, \\ &\qquad V_{y+1}^{\sigma_{x,y}}, \cdots,\ V_m^{\sigma_{x,y}}\Big) \in G_2^{m+1}. \end{aligned}$$

- **Encrypt**$_{\text{ABE}}(S, \mathsf{PK}, (i, j), M)$: To encrypt a message $M \in G_T$ to the recipients who are in S and simultaneously who

[1] Strictly speaking, the original broadcast encryption scheme [12] is slightly modified, rather than straightforwardly applied. Similar portion of broadcast encryption scheme in [10] was later suggested by the works [24], [25].



jth column
ith row →
D
C
A
B
$(E_x^{\theta_3\phi_x}, E_x^{\theta_4\phi_x}, g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},$
$(\prod_{i\in S_x} U_i)^{(\theta_3 s_1+\theta_4 s_2)\phi_x}, M\Lambda_x^{(\theta_3 s_1+\theta_4 s_2)\phi_x})$
$(E_x^{\theta_1\phi_x}, E_x^{\theta_2\phi_x}, g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},$
$(\prod_{i\in S_x} U_i)^{(\theta_1 s_1+\theta_2 s_2)\phi_x}, M\Lambda_x^{(\theta_1 s_1+\theta_2 s_2)\phi_x})$
$(H_y^{s_1} h^{\theta_2\lambda_y}, H_y^{s_2} h^{-\theta_1\lambda_y})(H_y^{s_1}, H_y^{s_2})$

Fig. 2. Row and column ciphertexts on the encryption to position (i, j).

have row x such that $x > i$ or both row $x = i$ and column y such that $y \geq j$. The encryption algorithm picks random exponents s_1, s_2, θ_1, θ_2, θ_3, θ_4, $\{\phi_x\}_{x=i}^{m}$, $\{\lambda_y\}_{y=1}^{j-1}$, and $\{z_{x,1}, z_{x,2}, z_{x,3}, z_{x,4}\}_{x=1}^{i-1}$ in Z_p, under a constraint that $\theta_2\theta_3 - \theta_1\theta_4 \neq 0$ in Z_p.
Let S_x denote the set of all values y such that the user (x, y) is in the set S. For each row x the algorithm constructs row ciphertexts $(R_{x,1}, R_{x,2}, R_{x,3}, R_{x,4}, B_x)$ as follows.

If $x < i$:

$$R_{x,1} = g^{z_{x,1}}, \quad R_{x,2} = g^{z_{x,2}}, \quad R_{x,3} = g^{z_{x,3}},$$
$$R_{x,4} = \big(\prod_{i\in S_x} U_i\big)^{z_{x,3}}, \quad B_x = \Lambda_x^{z_{x,4}}.$$

If $x = i$:

$$R_{x,1} = E_x^{\theta_3\phi_x} = g^{r_x\theta_3\phi_x}, \quad R_{x,2} = E_x^{\theta_4\phi_x} = g^{r_x\theta_4\phi_x},$$
$$R_{x,3} = g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},$$
$$R_{x,4} = \big(\prod_{i\in S_x} U_i\big)^{(\theta_3 s_1+\theta_4 s_2)\phi_x},$$
$$B_x = M\Lambda_x^{(\theta_3 s_1+\theta_4 s_2)\phi_x} = Me(g,h)^{\alpha_x(\theta_3 s_1+\theta_4 s_2)\phi_x}.$$

If $x > i$:

$$R_{x,1} = E_x^{\theta_1\phi_x} = g^{r_x\theta_1\phi_x}, \quad R_{x,2} = E_x^{\theta_2\phi_x} = g^{r_x\theta_2\phi_x},$$
$$R_{x,3} = g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},$$
$$R_{x,4} = \big(\prod_{i\in S_x} U_i\big)^{(\theta_1 s_1+\theta_2 s_2)\phi_x},$$
$$B_x = M\Lambda_x^{(\theta_1 s_1+\theta_2 s_2)\phi_x} = Me(g,h)^{\alpha_x(\theta_1 s_1+\theta_2 s_2)\phi_x}.$$

For each column y, the algorithm creates column ciphertexts $(C_{y,1}, C_{y,2})$ as follows.

$$\text{If } y < j: \quad C_{y,1} = H_y^{s_1} h^{\theta_2\lambda_y} = h^{c_y s_1} h^{\theta_2\lambda_y},$$
$$C_{y,2} = H_y^{s_2} h^{-\theta_1\lambda_y} = h^{c_y s_2} h^{-\theta_1\lambda_y}.$$
$$\text{If } y \geq j: \quad C_{y,1} = H_y^{s_1} = h^{c_y s_1},$$
$$C_{y,2} = H_y^{s_2} = h^{c_y s_2}.$$

The algorithm outputs the ciphertext

$$\mathsf{CT} = \Big(\{R_{x,1}, R_{x,2}, R_{x,3}, R_{x,4}\}_{x=1}^{m}, \{C_{y,1}, C_{y,2}\}_{y=1}^{m}, \{B_x\}_{x=1}^{m}\Big) \in G_1^{4m} \times G_2^{2m} \times G_T^{m}.$$

- **Decrypt**$_{\text{ABE}}$(CT, $K_{(x,y)}$, S): Assume that user i $(= (x-1)m + y)$ belongs to S and thus S_x. To decrypt a ciphertext CT using the private key $K_{(x,y)}$, the decryption algorithm first computes a temporary key

$$K'_{(x,y)} = d'_{x,y} \prod_{\substack{k\in S_x \\ k\neq y}} d_k.$$

Next, the algorithm outputs

$$M = \frac{e(R_{x,1}, C_{y,1})e(R_{x,2}, C_{y,2})e(R_{x,4}, d''_{x,y})}{e(R_{x,3}, K'_{(x,y)})} B_x.$$

C. Correctness

We show that the ciphertext generated by **Encrypt**$_{\text{ABE}}$(S, PK, (i, j), M) algorithm are correctly decrypted by the targeted users. More precisely, the resultant ciphertext must be correctly decrypted by any user in the set S and in the three parts A, B, and C in Fig. 2. We consider each decryption process that is performed in four different parts A, B, C, and D. Note that users in D are not belonging to the set of the targeted users, and thus they must not recover the message M correctly.

IV. SECURITY

In this section, we prove that our proposed ABE scheme is secure under the asymmetric decision 3-party Diffie-Hellman assumption and the XDH assumption.

A. Index Hiding

Theorem 1: Under the asymmetric decision 3-party Diffie-Hellman assumption and the XDH assumption, there is no probabilistic polynomial time adversary that can distinguish between an encryption to two adjacent recipients in the index hiding game for any (i, j) where $1 \leq i, j \leq m$ with non-negligible probability.

A full proof of Theorem 1 follows in Lemma 1 and Lemma 2. As in [11] and [14], these two lemmas are given by two possible cases that include an index hiding game where the adversary tries to distinguish between ciphertexts encrypted to (i, j) and $(i, j + 1)$ when $1 \leq j < m$, and an index hiding game where the adversary tries to distinguish between ciphertexts encrypted to (i, m) and $(i + 1, 1)$ when $1 \leq i < m$.

Lemma 1: Under the asymmetric decision 3-party Diffie-Hellman assumption, there is no probabilistic polynomial time adversary that can distinguish between an encryption to recipient (i, j) and $(i, j + 1)$ in the index hiding game for any (i, j) where $1 \leq i \leq m$ and $1 \leq j < m$ with non-negligible probability.

Proof: Suppose that there exists an adversary $\mathcal{A}$ which can distinguish between an encryption to recipient (i, j) and $(i, j + 1)$ in the index hiding game with advantage ϵ. We then build an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the asymmetric decision 3-party Diffie-Hellman problem. On input $(g, g^a, g^b, h, h^a, h^{ab}, h^c, T) \in G_1^3 \times G_2^5$, the goal of $\mathcal{B}$ is to output 1 if $T = h^{abc}$ and 0 otherwise.

The index (i, j) is given to both $\mathcal{A}$ and $\mathcal{B}$. In simulation, $\mathcal{A}$ will eventually behave in one of two different ways:

Case I: $\mathcal{A}$ will output a bit $\widetilde{s} = 0$ and specify a target set S where $(i, j) \in S$. In this case, $\mathcal{B}$ needs to generate all secret keys, except the key $K_{(x,y)}$.

Case II: $\mathcal{A}$ will output a bit $\widetilde{s} = 1$ and specify a target set S where $(i, j) \notin S$. In this case, $\mathcal{B}$ needs to generate all secret keys.

At this point $\mathcal{B}$ does not know how $\mathcal{A}$ will behave among the two cases. Thus, $\mathcal{B}$ needs to guess which case it will be in. Since $\mathcal{B}$'s guess will be independent of which case $\mathcal{A}$ selects, $\mathcal{B}$ will be able to continue the simulation with probability $1/2$ in **Query** phase. We describe how $\mathcal{B}$ will behave in each case. We note that this proof approach follows from [10].

Case I

- **Setup:** $\mathcal{B}$ selects random exponents $\{r_i, c_i, \alpha_i, \beta_i\}_{i=1}^m$ in Z_p. It sets up the public key elements as

$$E_1 = g^{r_1},\ E_2 = g^{r_2}, \cdots, E_i = (g^a)^{r_i}, \cdots, E_m = g^{r_m},$$
$$H_1 = h^{c_1},\ H_2 = h^{c_2}, \cdots, H_j = (h^c)^{c_j}, \cdots, H_m = h^{c_m},$$
$$U_1 = g^{\beta_1},\ U_2 = g^{\beta_2}, \cdots, U_j = g^{\beta_j}, \cdots, U_m = g^{\beta_m},$$
$$V_1 = h^{\beta_1},\ V_2 = h^{\beta_2}, \cdots, V_j = h^{\beta_j}, \cdots, V_m = h^{\beta_m},$$
$$\Lambda_1 = e(g, h)^{\alpha_1},\ \Lambda_2 = e(g, h)^{\alpha_2}, \cdots, \Lambda_m = e(g, h)^{\alpha_m}.$$

Next, $\mathcal{B}$ chooses random $\sigma_{x,y} \in Z_p$ for all $(x, y) \neq (i, j)$ and computes the private key $K_{(x,y)}$ for user (x, y) as

$$K_{(x,y)} = \left(d'_{x,y},\ d''_{x,y},\ d_1, \cdots, d_{y-1}, d_{y+1}, \cdots, d_m\right)$$
$$= \begin{cases} \left(h^{\alpha_x} h^{r_x c_y} V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}},\right. \\ \qquad \left. V_{y+1}^{\sigma_{x,y}}, \cdots, V_m^{\sigma_{x,y}}\right),\ x \neq i,\ y \neq j, \\ \left(h^{\alpha_x} (h^a)^{r_x c_y} V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}},\right. \\ \qquad \left. V_{y+1}^{\sigma_{x,y}}, \cdots, V_m^{\sigma_{x,y}}\right),\ x = i,\ y \neq j, \\ \left(h^{\alpha_x} (h^c)^{r_x c_y} V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}},\right. \\ \qquad \left. V_{y+1}^{\sigma_{x,y}}, \cdots, V_m^{\sigma_{x,y}}\right),\ x \neq i,\ y = j. \end{cases}$$

The public key and private keys have an identical distribution to that in the actual construction, because all the exponents are chosen uniformly at random.

- **Query:** $\mathcal{A}$ gives a bit $\widetilde{s}$ as the query. If $\widetilde{s} = 1$, the simulation aborts. Otherwise, $\mathcal{B}$ continues the simulation.
- **Challenge:** $\mathcal{A}$ outputs a message $M \in G_T$ and a target set S. Then, $\mathcal{B}$ picks random exponents s_1, θ_1, θ_2, θ_3, θ_4, $\{\phi_x\}_{x=i}^m$, $\{\lambda_y\}_{y=1}^{j-1}$, and $\{z_{x,1}, z_{x,2}, z_{x,3}, z_{x,4}\}_{x=1}^{i-1}$ in Z_p, under a constraint that $\theta_2\theta_3 - \theta_1\theta_4 \neq 0$ in Z_p. Additionally, $\mathcal{B}$ obtains s_2 under equation $\theta_3 s_1 + \theta_4 s_2 = 0$ in Z_p. Notice that s_2 itself is not distributed identically to that of real scheme, but this will not be a problem in simulation (which we mention below).

  For each column y, $\mathcal{B}$ constructs column ciphertexts $(C_{y,1}, C_{y,2})$ as follows.

$$\text{If } y < j:\ \ C_{y,1} = (h^{ab})^{\frac{c_y\theta_2}{c_j}} h^{c_y s_1} h^{\theta_2 \lambda_y},$$
$$C_{y,2} = (h^{ab})^{-\frac{c_y\theta_1}{c_j}} h^{c_y s_2} h^{-\theta_1 \lambda_y}.$$
$$\text{If } y = j:\ \ C_{y,1} = T^{\theta_2} (h^c)^{c_j s_1},$$
$$C_{y,1} = T^{-\theta_1} (h^c)^{c_j s_2}.$$
$$\text{If } y > j:\ \ C_{y,1} = (h^{ab})^{\frac{c_y\theta_2}{c_j}} h^{c_y s_1},$$
$$C_{y,2} = (h^{ab})^{-\frac{c_y\theta_1}{c_j}} h^{c_y s_2}.$$

Here, $\mathcal{B}$ sets $\widetilde{s}_1 = s_1 + \theta_2 ab/c_j$ and $\widetilde{s}_2 = s_2 - \theta_1 ab/c_j$. Note that $\widetilde{s}_1$ and $\widetilde{s}_2$ include two random exponents ab and s_1 that are enough for random distribution of $\widetilde{s}_1$ and $\widetilde{s}_2$. Thus, the dependence of s_2 (not $\widetilde{s}_2$) is not a problem. If T corresponds to h^{abc}, then

$$C_{j,1} = T^{\theta_2} (h^c)^{c_j s_1} = (h^{abc})^{\theta_2} (h^c)^{c_j s_1}$$
$$= (h^{c_j c})^{s_1 + \frac{\theta_2 ab}{c_j}} = H_j^{\widetilde{s}_1},$$
$$C_{j,2} = T^{-\theta_1} (h^c)^{c_j s_2} = (h^{abc})^{-\theta_1} (h^c)^{c_j s_2}$$
$$= (h^{c_j c})^{s_2 - \frac{\theta_1 ab}{c_j}} = H_j^{\widetilde{s}_2}.$$

On the other hand, if T is random, i.e., $T = h^{abc} h^r$ for some (unknown) $r \in Z_p$, then

$$C_{j,1} = T^{\theta_2} (h^c)^{c_j s_1} = H_j^{\widetilde{s}_1} h^{\theta_2 r},$$
$$C_{j,2} = T^{-\theta_1} (h^c)^{c_j s_2} = H_j^{\widetilde{s}_2} h^{-\theta_1 r}$$

where the exponent r serves as a random exponent $\lambda_i \in Z_p$. Without loss of generality, we assume that the target set S is divided into subsets S_x for $x = 1, \cdots, m$. For each row x the algorithm constructs row ciphertexts $(R_{x,1}, R_{x,2}, R_{x,3}, R_{x,4}, B_x)$ as follows.

$$\text{If } x < i:\ \ R_{x,1} = g^{z_{x,1}},\ \ R_{x,2} = g^{z_{x,2}},\ \ R_{x,3} = g^{z_{x,3}},$$
$$R_{x,4} = \Big(\prod_{k \in S_x} U_k\Big)^{z_{x,3}},\ \ B_x = \Lambda_x^{z_{x,4}}.$$
$$\text{If } x = i:\ \ R_{x,1} = g^{r_x \theta_3 \phi_x},\ \ R_{x,2} = g^{r_x \theta_4 \phi_x},$$
$$R_{x,3} = (g^b)^{(\theta_3\theta_2 - \frac{\theta_4\theta_1\phi_x}{c_j})},$$
$$R_{x,4} = (g^b)^{(\sum_{k \in S_x} \beta_k)(\theta_3\theta_2 - \frac{\theta_4\theta_1\phi_x}{c_j})}$$
$$B_x = Me(g^b, h)^{\alpha_x(\theta_3\theta_2 - \frac{\theta_4\theta_1\phi_x}{c_j})}.$$
$$\text{If } x > i:\ \ R_{x,1} = g^{r_x \theta_1 \phi_x},\ \ R_{x,2} = g^{r_x \theta_2 \phi_x},$$
$$R_{x,3} = g^{(\theta_1 s_1 + \theta_2 s_2)\phi_x},$$
$$R_{x,4} = g^{(\sum_{k \in S_x} \beta_k)(\theta_1 s_1 + \theta_2 s_2)\phi_x},$$
$$B_x = Me(g, h)^{\alpha_x(\theta_1 s_1 + \theta_2 s_2)\phi_x}.$$

When $x = i$, $\mathcal{B}$ sets $\widetilde{\theta}_3 = \theta_3 b$, $\widetilde{\theta}_4 = \theta_4 b$, and $\widetilde{\phi}_i = \phi_i/ab$. Then, the row ciphertexts can be computed as

$$R_{i,1} = (g^{r_i a})^{\frac{\theta_3 b \phi_i}{ab}} = g^{r_i \theta_3 \phi_i},$$
$$R_{i,2} = (g^{r_i a})^{\frac{\theta_4 b \phi_i}{ab}} = g^{r_i \theta_4 \phi_i},$$
$$R_{i,3} = g^{\left[\theta_3 b(s_1 + \frac{\theta_2 ab}{c_j}) + \theta_4 b(s_2 - \frac{\theta_1 ab}{c_j})\right] \frac{\phi_i}{ab}}$$
$$= (g^b)^{(\theta_3 s_1 + \theta_4 s_2)\frac{\phi_i}{ab}} (g^{ab^2})^{\frac{(\theta_3\theta_2 - \theta_4\theta_1)}{c_j}(\frac{\phi_i}{ab})}$$
$$= (g^b)^{(\theta_3\theta_2 - \theta_4\theta_1)\frac{\phi_i}{c_j}},$$

- **Part A:** Recipients with entry (x, y) such that $i < x \leq m$ and $1 \leq y < j$.

$$\begin{aligned}
&\frac{e(R_{x,1}, C_{y,1})e(R_{x,2}, C_{y,2})e(R_{x,3}, d''_{x,y})}{e(R_{x,3}, K'_{(x,y)})} B_x \\
&= \frac{e(g^{r_x\theta_1\phi_x},\ h^{c_y s_1}h^{\theta_2\lambda_y})e(g^{r_x\theta_2\phi_x},\ h^{c_y s_2}h^{-\theta_1\lambda_y})e\big(\big(\prod_{i\in S_x} U_i\big)^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\alpha_x}h^{r_x c_y}\prod_{i\in S_x} V_i^{\sigma_{x,y}})} B_x \\
&= \frac{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{r_x c_y})e\big(\big(\prod_{i\in S_x} g^{\beta_i}\big)^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\alpha_x})e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{r_x c_y})e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ \prod_{i\in S_x} h^{\beta_i\sigma_{x,y}})} B_x \\
&= \frac{1}{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\alpha_x})} Me(g,h)^{\alpha_x(\theta_1 s_1+\theta_2 s_2)\phi_x} \\
&= M.
\end{aligned}$$

- **Part B:** Recipients with entry (x, y) such that $i < x \leq m$ and $j \leq y \leq m$.

$$\begin{aligned}
&\frac{e(R_{x,1}, C_{y,1})e(R_{x,2}, C_{y,2})e(R_{x,3}, d''_{x,y})}{e(R_{x,3}, K'_{(x,y)})} B_x \\
&= \frac{e(g^{r_x\theta_1\phi_x},\ h^{c_y s_1})e(g^{r_x\theta_2\phi_x},\ h^{c_y s_2})e\big(\big(\prod_{i\in S_x} U_i\big)^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\alpha_x}h^{r_x c_y}\prod_{i\in S_x} V_i^{\sigma_{x,y}})} B_x \\
&= \frac{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{r_x c_y})e\big(\big(\prod_{i\in S_x} g^{\beta_i}\big)^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\alpha_x})e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{r_x c_y})e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ \prod_{i\in S_x} h^{\beta_i\sigma_{x,y}})} B_x \\
&= \frac{1}{e(g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\ h^{\alpha_x})} Me(g,h)^{\alpha_x(\theta_1 s_1+\theta_2 s_2)\phi_x} \\
&= M.
\end{aligned}$$

$$\begin{aligned}
R_{i,4} &= \Big(\prod_{k\in S_x} g^{\beta_k}\Big)^{\left[\theta_3 b(s_1+\frac{\theta_2 ab}{c_j})+\theta_4 b(s_2-\frac{\theta_1 ab}{c_j})\right]\frac{\phi_i}{ab}} \\
&= g^{(\sum_{k\in S_x}\beta_k)b(\theta_3\theta_2-\theta_4\theta_1)\frac{\phi_i}{c_j}} \\
&= (g^b)^{(\sum_{k\in S_x}\beta_k)(\theta_3\theta_2-\theta_4\theta_1)\frac{\phi_x}{c_j}}, \\
B_i &= Me(g,h)^{\alpha_i b(\theta_3\theta_2-\theta_4\theta_1)\frac{\phi_i}{c_j}} \\
&= Me(g^b,h)^{\alpha_i(\theta_3\theta_2-\theta_4\theta_1)\frac{\phi_i}{c_j}}.
\end{aligned}$$

Next, when $x > i$, $\mathcal{B}$ sets ϕ_x as it is. Recall that θ_1 and θ_2 are already determined during the computation of the column ciphertexts. Then, it is sufficient to check that

$$\begin{aligned}
R_{x,3} &= g^{(\theta_1 s_1+\theta_2 s_2)\phi_x} = g^{\left[\theta_1(s_1+\frac{\theta_2 ab}{c_j})+\theta_2(s_2-\frac{\theta_1 ab}{c_j})\right]\phi_x} \\
&= g^{(\theta_1\bar{s}_1+\theta_2\bar{s}_2)\phi_x}.
\end{aligned}$$

If T is h^{abc}, then the challenge ciphertext indicates the encryption to (i, j); and if T is random, then the challenge ciphertext indicates the encryption to $(i, j+1)$.

► **Guess:** $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$, and then $\mathcal{B}$ outputs the same value to the asymmetric decision 3-party Diffie-Hellman challenger. It is easy to see that $\mathcal{B}$'s advantage in the reduction is straightforwardly taken from $\mathcal{A}$'s advantage.

Case II

► **Setup:** $\mathcal{B}$ selects random exponents $\{r_i, c_i, \alpha_i, \beta_i\}_{i=1}^m$ in Z_p. It sets up the public key elements as

$$\begin{aligned}
&E_1 = g^{r_1},\ E_2 = g^{r_2}, \cdots, E_i = (g^a)^{r_i}, \cdots, E_m = g^{r_m}, \\
&H_1 = h^{c_1},\ H_2 = h^{c_2}, \cdots, H_j = (h^c)^{c_j}, \cdots, H_m = h^{c_m}, \\
&U_1 = g^{\beta_1},\ U_2 = g^{\beta_2}, \cdots, U_j = (g^a)^{\beta_j}, \cdots, U_m = g^{\beta_m}, \\
&V_1 = h^{\beta_1},\ V_2 = h^{\beta_2}, \cdots, V_j = (h^a)^{\beta_j}, \cdots, V_m = h^{\beta_m}, \\
&\Lambda_1 = e(g,h)^{\alpha_1},\ \Lambda_2 = e(g,h)^{\alpha_2}, \cdots,\ \Lambda_m = e(g,h)^{\alpha_m}.
\end{aligned}$$

Next, $\mathcal{B}$ chooses random $\sigma_{x,y} \in Z_p$ for all (x, y) and computes the private key $K_{(x,y)}$ for user (x, y) as

$$\begin{aligned}
K_{(x,y)} &= \big(d'_{x,y},\ d''_{x,y},\ d_1, \cdots, d_{y-1}, d_{y+1}, \cdots, d_m\big) \\
&= \begin{cases}
\big(h^{\alpha_x}h^{r_x c_y}V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}}, \\
\quad V_{y+1}^{\sigma_{x,y}}, \cdots, V_m^{\sigma_{x,y}}\big),\ x \neq i,\ y \neq j, \\
\big(h^{\alpha_x}(h^a)^{r_x c_y}V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}}, \\
\quad V_{y+1}^{\sigma_{x,y}}, \cdots, V_m^{\sigma_{x,y}}\big),\ x = i,\ y \neq j, \\
\big(h^{\alpha_x}(h^c)^{r_x c_y}V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}}, \\
\quad V_{y+1}^{\sigma_{x,y}}, \cdots, V_m^{\sigma_{x,y}}\big),\ x \neq i,\ y = j, \\
\big(h^{\alpha_x}(h^a)^{\beta_y\sigma_{x,y}},\ h^{\sigma_{x,y}}(h^c)^{-\frac{r_x c_y}{\beta_y}}, \\
\quad h^{\beta_1\sigma_{x,y}}(h^c)^{-\frac{\beta_1 r_x c_y}{\beta_y}}, \cdots, h^{\beta_{y-1}\sigma_{x,y}}(h^c)^{-\frac{\beta_{y-1} r_x c_y}{\beta_y}}, \\
\quad h^{\beta_{y+1}\sigma_{x,y}}(h^c)^{-\frac{\beta_{y+1} r_x c_y}{\beta_y}}, \cdots, \\
\quad h^{\beta_m\sigma_{x,y}}(h^c)^{-\frac{\beta_m r_x c_y}{\beta_y}}\big),\ x = i,\ y = j.
\end{cases}
\end{aligned}$$

- **Part C:** Recipients with entry (x, y) such that $x = i$ and $j \leq y \leq m$.

$$\begin{aligned}
&\frac{e(R_{x,1}, C_{y,1})e(R_{x,2}, C_{y,2})e(R_{x,3}, d''_{x,y})}{e(R_{x,3}, K'_{(x,y)})}B_x \\
&= \frac{e(g^{r_x\theta_3\phi_x},\ h^{c_y s_1})e(g^{r_x\theta_4\phi_x},\ h^{c_y s_2})e\big(\big(\prod_{i\in S_x} U_i\big)^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\alpha_x} h^{r_x c_y} \prod_{i\in S_x} V_i^{\sigma_{x,y}})}B_x \\
&= \frac{e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{r_x c_y})e\big(\big(\prod_{i\in S_x} g^{\beta_i}\big)^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\alpha_x})e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{r_x c_y})e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ \prod_{i\in S_x} h^{\beta_i\sigma_{x,y}})}B_x \\
&= \frac{1}{e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\alpha_x})}Me(g,h)^{\alpha_x(\theta_3 s_1+\theta_4 s_2)\phi_x} \\
&= M.
\end{aligned}$$

- **Part D:** Recipients with entry (x, y) such that $x = i$ and $1 \leq y < j$.

$$\begin{aligned}
&\frac{e(R_{x,1}, C_{y,1})e(R_{x,2}, C_{y,2})e(R_{x,3}, d''_{x,y})}{e(R_{x,3}, K'_{(x,y)})}B_x \\
&= \frac{e(g^{r_x\theta_3\phi_x},\ h^{c_y s_1}h^{\theta_2\lambda_y})e(g^{r_x\theta_4\phi_x},\ h^{c_y s_2}h^{-\theta_1\lambda_y})e\big(\big(\prod_{i\in S_x} U_i\big)^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\alpha_x} h^{r_x c_y} \prod_{i\in S_x} V_i^{\sigma_{x,y}})}B_x \\
&= \frac{e(g,h)^{r_x\phi_x\lambda_y(\theta_2\theta_3-\theta_1\theta_4)}e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{r_x c_y})e\big(\big(\prod_{i\in S_x} g^{\beta_i}\big)^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\sigma_{x,y}}\big)}{e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\alpha_x})e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{r_x c_y})e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ \prod_{i\in S_x} h^{\beta_i\sigma_{x,y}})}B_x \\
&= \frac{e(g,h)^{r_x\phi_x\lambda_y(\theta_2\theta_3-\theta_1\theta_4)}}{e(g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\ h^{\alpha_x})}Me(g,h)^{\alpha_x(\theta_3 s_1+\theta_4 s_2)\phi_x} \\
&= e(g,h)^{r_x\phi_x\lambda_y(\theta_2\theta_3-\theta_1\theta_4)}M.
\end{aligned}$$

Recall that $\theta_2\theta_3 - \theta_4\theta_1 \neq 0$ in Z_p, in which case the resultant value in the part D is not equal to the message M.

In case when $x = i$ and $y = j$, $\mathcal{B}$ implicitly uses a random $\widetilde{\sigma}_{i,j} = \sigma_{i,j} - r_i c_j c/\beta_j \in Z_p$. Then, the first element in $K_{(i,j)}$ is verified as

$$\begin{aligned}
h^{\alpha_i}h^{(r_i a)(c_j c)}V_j^{\widetilde{\sigma}_{i,j}} &= h^{\alpha_i}h^{(r_i c_j)(ac)}\big(h^{\beta_j a}\big)^{(\sigma_{i,j}-\frac{r_i c_j c}{\beta_j})} \\
&= h^{\alpha_i}\big(h^a\big)^{\beta_j\sigma_{i,j}}.
\end{aligned}$$

The public key and private keys have an identical distribution to that in the actual construction, because all the exponents are chosen uniformly at random.

- ▶ **Query:** $\mathcal{A}$ gives a bit $\widetilde{s}$ as the query. If $\widetilde{s} = 0$, the simulation aborts. Otherwise, $\mathcal{B}$ continues the simulation.
- ▶ **Challenge:** $\mathcal{A}$ outputs a message $M \in G_T$ and a target set S. Then, $\mathcal{B}$ picks random exponents s_1, θ_1, θ_2, θ_3, θ_4, $\{\phi_x\}_{x=i}^{m}$, $\{\lambda_y\}_{y=1}^{j-1}$, and $\{z_{x,1}, z_{x,2}, z_{x,3}, z_{x,4}\}_{x=1}^{i-1}$ in Z_p, under a constraint that $\theta_2\theta_3 - \theta_1\theta_4 \neq 0$ in Z_p. As before, $\mathcal{B}$ obtains s_2 under equation $\theta_3 s_1 + \theta_4 s_2 = 0$ in Z_p.
For each column y, $\mathcal{B}$ constructs column ciphertexts $(C_{y,1}, C_{y,2})$ as follows.

$$\begin{aligned}
\text{If } y < j:\quad & C_{y,1} = (h^{ab})^{\frac{c_y\theta_2}{c_j}}h^{c_y s_1}h^{\theta_2\lambda_y}, \\
& C_{y,2} = (h^{ab})^{-\frac{c_y\theta_1}{c_j}}h^{c_y s_2}h^{-\theta_1\lambda_y}. \\
\text{If } y = j:\quad & C_{y,1} = T^{\theta_2}(h^c)^{c_j s_1}, \\
& C_{y,1} = T^{-\theta_1}(h^c)^{c_j s_2}.
\end{aligned}$$

$$\begin{aligned}
\text{If } y > j:\quad & C_{y,1} = (h^{ab})^{\frac{c_y\theta_2}{c_j}}h^{c_y s_1}, \\
& C_{y,2} = (h^{ab})^{-\frac{c_y\theta_1}{c_j}}h^{c_y s_2}.
\end{aligned}$$

As in case I, $\mathcal{B}$ sets $\widetilde{s}_1 = s_1 + \theta_2 ab/c_j$ and $\widetilde{s}_2 = s_2 - \theta_1 ab/c_j$. If T corresponds to h^{abc}, then

$$\begin{aligned}
C_{j,1} &= T^{\theta_2}(h^c)^{c_j s_1} = (h^{abc})^{\theta_2}(h^c)^{c_j s_1} \\
&= (h^{c_j c})^{s_1+\frac{\theta_2 ab}{c_j}} = H_j^{\widetilde{s}_1}, \\
C_{j,2} &= T^{-\theta_1}(h^c)^{c_j s_2} = (h^{abc})^{-\theta_1}(h^c)^{c_j s_2} \\
&= (h^{c_j c})^{s_2-\frac{\theta_1 ab}{c_j}} = H_j^{\widetilde{s}_2}.
\end{aligned}$$

On the other hand, if T is random, i.e., $T = h^{abc}h^r$ for some (unknown) $r \in Z_p$, then

$$\begin{aligned}
C_{j,1} &= T^{\theta_2}(h^c)^{c_j s_1} = H_j^{\widetilde{s}_1}h^{\theta_2 r}, \\
C_{j,2} &= T^{-\theta_1}(h^c)^{c_j s_2} = H_j^{\widetilde{s}_2}h^{-\theta_1 r}
\end{aligned}$$

where the exponent r serves as a random exponent $\lambda_i \in Z_p$. For each row x the algorithm constructs row ciphertexts $(R_{x,1}, R_{x,2}, R_{x,3}, R_{x,4}, B_x)$ as follows.

$$\begin{aligned}
\text{If } x < i:\quad & R_{x,1} = g^{z_{x,1}},\quad R_{x,2} = g^{z_{x,2}},\quad R_{x,3} = g^{z_{x,3}}, \\
& R_{x,4} = \Big(\prod_{k\in S_x} U_k\Big)^{z_{x,3}},\quad B_x = \Lambda_x^{z_{x,4}}.
\end{aligned}$$

$$\begin{aligned}\text{If } x = i: \quad & R_{x,1} = g^{r_x\theta_3\phi_x}, \qquad R_{x,2} = g^{r_x\theta_4\phi_x},\\ & R_{x,3} = (g^b)^{(\theta_3\theta_2-\theta_4\theta_1)\frac{\phi_x}{c_j}},\\ & R_{x,4} = (g^b)^{(\sum_{k\in S_x}\beta_k)(\theta_3\theta_2-\theta_4\theta_1)\frac{\phi_x}{c_j}}\\ & B_x = Me(g^b,h)^{\alpha_x(\theta_3\theta_2-\theta_4\theta_1)\frac{\phi_x}{c_j}}.\\ \text{If } x > i: \quad & R_{x,1} = g^{r_x\theta_1\phi_x}, \qquad R_{x,2} = g^{r_x\theta_2\phi_x},\\ & R_{x,3} = g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\\ & R_{x,4} = (\prod_{k\in S_x} U_k)^{(\theta_1 s_1+\theta_2 s_2)\phi_x}\\ & B_x = Me(g,h)^{\alpha_x(\theta_1 s_1+\theta_2 s_2)\phi_x}.\end{aligned}$$

When $x = i$, $\mathcal{B}$ sets $\widetilde{\theta}_3 = \theta_3 b$, $\widetilde{\theta}_4 = \theta_4 b$, and $\widetilde{\phi}_i = \phi_i/ab$. Observe that $j \notin S_i$ since $(i,j) \notin S$. This means that $\sum_{k\in S_x}\beta_k$ does not contain the value $\beta_j a$ and $R_{i,4}$ can avoid the term g^{ab}. Also, when $x > i$, the exponent of $R_{x,3}$ becomes $(\theta_1\widetilde{s}_1 + \theta_2\widetilde{s}_2)\phi_x = (\theta_1 s_1 + \theta_2 s_2)\phi_x$ which is known to $\mathcal{B}$. Thus, $R_{x,4}$ can be computed regardless of whether j is included into each subset S_x.

- ► **Guess:** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$, and then $\mathcal{B}$ outputs the same value to the asymmetric decision 3-party Diffie-Hellman challenger.

  If T is h^{abc}, then the challenge ciphertext indicates the encryption to (i,j); and if T is random, then the challenge ciphertext indicates the encryption to $(i, j+1)$. Then, it is easy to see that $\mathcal{B}$'s advantage in the reduction is straightforwardly taken from $\mathcal{A}$'s advantage. □

Lemma 2: Under the asymmetric decision 3-party Diffie-Hellman assumption and the XDH assumption, there is no probabilistic polynomial time adversary that can distinguish between an encryption to recipient (i,m) and $(i+1,1)$ in the index hiding game for any $1 \le i < m$ with non-negligible probability.

To prove Lemma 2, we need to create complicated hybrid games as in [11] and [14]. With the **Encrypt**$_{\text{ABE}}$ algorithm described in the previous section, we refer to rows with ciphertexts generated with random exponents as 'less-than rows,' rows with ciphertexts involved with exponents (θ_3, θ_4) as 'target rows,' and rows with ciphertexts involved with exponents (θ_1, θ_2) as 'greater-than rows.' In addition, 'encrypt to column j' means that column ciphertexts $(C_{y,1}, C_{y,2})$ for all $y \ge j$ are formed without attached components $(h^{\theta_2\lambda_y}, h^{-\theta_1\lambda_y})$. Using these terminologies, the hybrid games are described as follows.

- Γ_1: Encrypt to column m, row i is the target row, row $i+1$ is the less-than row.
- Γ_2: Encrypt to column $m+1$, row i is the target row, row $i+1$ is the greater-than row.
- Γ_3: Encrypt to column $m+1$, row i is the less-than row, row $i+1$ is the greater-than row.
- Γ_4: Encrypt to column 1, row i is the less-than row, row $i+1$ is the greater-than row.
- Γ_5: Encrypt to column 1, row i is the less-than row, row $i+1$ is the target row.

Note that Γ_1 corresponds to the encryption to (i,m) and Γ_5 corresponds to the encryption to $(i+1,1)$. The following claims show that it is computationally infeasible for an adversary to distinguish between the sequence of games from Γ_1 to Γ_5.

Claim 1. *Under the asymmetric decision 3-party Diffie-Hellman assumption, there is no probabilistic polynomial time adversary that can distinguish between games Γ_1 and Γ_2 with non-negligible probability.*

Proof: This claim can be proved by applying the result of Lemma 1. □

Claim 2. *Under the asymmetric decision 3-party Diffie-Hellman assumption, there is no probabilistic polynomial time adversary that can distinguish between games Γ_2 and Γ_3 with non-negligible probability.*

Proof: Suppose that there exists an adversary $\mathcal{A}$ which can distinguish between games Γ_2 and Γ_3 with advantage ϵ. We then build an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the asymmetric decision 3-party Diffie-Hellman problem. On input $(g, g^a, g^b, h, h^a, h^{ab}, h^c, T) \in G_1^3 \times G_2^5$, $\mathcal{B}$ outputs 1 if $T = h^{abc}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows.

- ► **Setup:** $\mathcal{B}$ selects exponents $\{c_k, \beta_k\}_{k=1}^m$ and $\{r_k, \alpha_k\}_{k=1,k\neq i}^m$ in Z_p at random. It sets up the public key elements as

$$\begin{aligned}& E_1 = g^{r_1},\ E_2 = g^{r_2}, \cdots, E_i = g^a, \cdots, E_m = g^{r_m},\\ & H_1 = h^{c_1}(h^c)^{-1},\ H_2 = h^{c_2}(h^c)^{-1}, \cdots, H_m = h^{c_m}(h^c)^{-1},\\ & U_1 = g^{\beta_1},\ U_2 = g^{\beta_2}, \cdots, U_m = g^{\beta_m},\\ & V_1 = h^{\beta_1},\ V_2 = h^{\beta_2}, \cdots, V_m = h^{\beta_m},\\ & \Lambda_1 = e(g,h)^{\alpha_1},\ \Lambda_2 = e(g,h)^{\alpha_2}, \cdots,\\ & \qquad \Lambda_i = e(g^a, h^c), \cdots, \Lambda_m = e(g,h)^{\alpha_m}.\end{aligned}$$

  Note that $r_i = a$ and $\alpha_i = ac \in Z_p$. $\mathcal{B}$ additionally selects $\sigma_{x,y} \in Z_p$ for all (x,y) and computes the private key $K_{(x,y)}$ for user (x,y) as

$$\begin{aligned}K_{(x,y)} &= (d'_{x,y},\ d''_{x,y},\ d_1, \cdots, d_{y-1}, d_{y+1}, \cdots, d_m)\\ &= \begin{cases} \big(h^{\alpha_x}h^{r_x c_y}(h^c)^{-r_x}V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}},\\ \qquad V_{y+1}^{\sigma_{y+1}}, \cdots, V_m^{\sigma_{x,y}}\big),\ x \neq i,\\ \big((h^a)^{c_y}V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots, V_{y-1}^{\sigma_{x,y}},\\ \qquad V_{y+1}^{\sigma_{y+1}}, \cdots, V_m^{\sigma_{x,y}}\big),\ x = i.\end{cases}\end{aligned}$$

  The validity of $d'_{x,y}$ such that $x = i$ can be checked as $d'_{x,y} = h^{ac}h^{(c_y - c)a}V_y^{\sigma_{x,y}} = (h^a)^{c_y}V_y^{\sigma_{x,y}}$. $\mathcal{A}$ is given the private keys $K_{(x,y)}$ except $(x,y) \neq (i,j)$. As before, the public key and private keys have an identical distribution to that in the actual construction.
- ► **Query:** $\mathcal{A}$ gives a bit $\widetilde{s}$ as the query. If $\widetilde{s} = 1$, $\mathcal{B}$ gives out $K_{(i,j)}$.
- ► **Challenge:** $\mathcal{A}$ outputs the message $M \in G_T$. $\mathcal{B}$ picks random exponents s_1, s_2, θ_1, θ_2, θ_3, θ_4, $\{\phi_x\}_{x=i}^m$, $\{\lambda_y\}_{y=1}^m$, and $\{z_{x,1}, z_{x,2}, z_{x,3}, z_{x,4}\}_{x=1}^{i-1}$ in Z_p, under a constraint that $\theta_2\theta_3 - \theta_1\theta_4 \neq 0$ in Z_p.

  For each row x, the algorithm constructs row ciphertexts $(R_{x,1}, R_{x,2}, R_{x,3}, R_{x,4}, B_x)$ as follows.

$$\begin{aligned}\text{If } x < i: \quad & R_{x,1} = g^{z_{x,1}},\quad R_{x,2} = g^{z_{x,2}},\quad R_{x,3} = g^{z_{x,3}},\\ & R_{x,4} = (\prod_{k\in S_x} U_k)^{z_{x,3}} \quad B_x = \Lambda_x^{z_{x,4}}.\end{aligned}$$

$$\begin{aligned}\text{If } x = i:\ & R_{x,1} = (g^a)^{\theta_3\phi_x}, \qquad R_{x,2} = (g^a)^{\theta_4\phi_x},\\ & R_{x,3} = (g^b)^{(\theta_3\theta_2-\theta_4\theta_1)\phi_x} g^{(\theta_3 s_1+\theta_4 s_2)\phi_x},\\ & R_{x,4} = (g^b)^{(\sum_{k\in S_x}\beta_k)(\theta_3\theta_2-\theta_4\theta_1)\phi_x}\\ & \qquad g^{(\sum_{k\in S_x}\beta_k)(\theta_3 s_1+\theta_4 s_2)\phi_x},\\ & B_x = Me(g,T)^{(\theta_3\theta_2-\theta_4\theta_1)\phi_x}\\ & \qquad e(g^a,h^c)^{(\theta_3 s_1+\theta_4 s_2)\phi_x}.\\ \text{If } x > i:\ & R_{x,1} = g^{r_x\theta_1\phi_x}, \qquad R_{x,2} = g^{r_x\theta_2\phi_x},\\ & R_{x,3} = g^{(\theta_1 s_1+\theta_2 s_2)\phi_x},\\ & R_{x,4} = g^{(\sum_{k\in S_x}\beta_k)(\theta_1 s_1+\theta_2 s_2)\phi_x},\\ & B_x = Me(g,h)^{\alpha_x(\theta_1 s_1+\theta_2 s_2)\phi_x}.\end{aligned}$$

Define $\widetilde{s}_1 = s_1 + \theta_2 b$ and $\widetilde{s}_2 = s_2 - \theta_1 b \in Z_p$. When $x = i$, $\mathcal{B}$ sets ϕ_i as it is. Then, if $T = h^{abc}$, then the row ciphertexts can be computed as

$$\begin{aligned}R_{i,1} &= (g^a)^{\theta_3\phi_i},\ R_{i,2} = (g^a)^{\theta_4\phi_i},\\ R_{i,3} &= g^{\left[\theta_3(s_1+\theta_2 b)+\theta_4(s_2-\theta_1 b)\right]\phi_i}\\ &= (g^b)^{(\theta_3\theta_2-\theta_4\theta_1)\phi_i} g^{(\theta_3 s_1+\theta_4 s_2)\phi_i},\\ R_{i,4} &= \Big(\prod_{k\in S_x} U_k\Big)^{\left[\theta_3(s_1+\theta_2 b)+\theta_4(s_2-\theta_1 b)\right]\phi_i}\\ &= (g^b)^{(\sum_{k\in S_x}\beta_k)(\theta_3\theta_2-\theta_4\theta_1)\phi_i} g^{(\sum_{k\in S_x}\beta_k)(\theta_3 s_1+\theta_4 s_2)\phi_i},\\ B_i &= Me(g,h)^{ac\left[b(\theta_3\theta_2-\theta_4\theta_1)\phi_i+(\theta_3 s_1+\theta_4 s_2)\phi_i\right]}\\ &= Me(g,T)^{(\theta_3\theta_2-\theta_4\theta_1)\phi_i} e(g^a,h^c)^{(\theta_3 s_1+\theta_4 s_2)\phi_i}.\end{aligned}$$

On the other hand, if T is random, then B_i is also randomly distributed. Thus, row i becomes the less-than row. Next, when $x > i$, $\mathcal{B}$ sets ϕ_x as it is. Then, as before, it is sufficient to check that

$$R_{x,3} = g^{\left[\theta_1(s_1+\theta_2 b)+\theta_2(s_2-\theta_1 b)\right]\phi_x} = g^{(\theta_1\widetilde{s}_1+\theta_2\widetilde{s}_2)\phi_x}.$$

For each column y, $\mathcal{B}$ constructs column ciphertexts $(C_{y,1},\ C_{y,2})$ as follows.

$$C_{y,1} = h^{c_y s_1}(h^c)^{-s_1} h^{\theta_2\lambda_y}, \quad C_{y,2} = h^{c_y s_2}(h^c)^{-s_2} h^{-\theta_1\lambda_y}.$$

The column ciphertexts are valid under randomness $\widetilde{\lambda}_y = (c - c_y)b + \lambda_y \in Z_p$ for all $y \in \{1,\cdots,m\}$. To see this,

$$\begin{aligned}C_{y,1} &= h^{(c_y-c)(s_1+\theta_2 b)} h^{\theta_2\left[(c-c_y)b+\lambda_y\right]}\\ &= h^{c_y s_1}(h^c)^{-s_1} h^{\theta_2\lambda_y},\\ C_{y,2} &= h^{(c_y-c)(s_2-\theta_1 b)} h^{-\theta_1\left[(c-c_y)b+\lambda_y\right]}\\ &= h^{c_y s_2}(h^c)^{-s_2} h^{-\theta_1\lambda_y}.\end{aligned}$$

- **Guess:** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$, and then $\mathcal{B}$ outputs the same value to the asymmetric decision 3-party Diffie-Hellman challenger.
  If T is h^{abc}, then the challenge ciphertext corresponding to row i indicates the target row; and if T is random, then the challenge ciphertext of row i indicates the less-than row.

Thus, we see that $\mathcal{B}$'s advantage in the reduction is straightforwardly taken from $\mathcal{A}$'s advantage.

□

Claim 3. *Under the asymmetric decision 3-party Diffie-Hellman assumption, there is no probabilistic polynomial time adversary that can distinguish between games Γ_3 and Γ_4 with non-negligible probability.*

To prove Claim 3, we need to further refine our hybrid games between games Γ_3 and Γ_4. Let $\Gamma_{3,m+1}(=\Gamma_3), \Gamma_{3,m}, \cdots, \Gamma_{3,1}(=\Gamma_4)$ be the hybrid games. In the game $\Gamma_{3,j}$, all column ciphertexts $(C_{y,1}, C_{y,2})$ are well formed without attached components for all y such that $j \le y \le m$. As in the proof of Lemma 1 (and Claim 1), it is sufficient to prove the indistinguishability of games $\Gamma_{3,j}$ and $\Gamma_{3,j+1}$ for j where $1 \le j \le m$.

Proof: Suppose that there exists an adversary $\mathcal{A}$ which can distinguish between games $\Gamma_{3,j}$ and $\Gamma_{3,j+1}$ with advantage ϵ. We then build an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the asymmetric decision 3-party Diffie-Hellman problem. On input $(g, g^a, g^b, h, h^a, h^{ab}, h^c, T) \in G_1^3 \times G_2^5$, $\mathcal{B}$ outputs 1 if $T = h^{abc}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows.

- **Setup:** $\mathcal{B}$ selects random exponents $\{r_k, c_k, \alpha_k, \beta_k\}_{k=1}^m$ in Z_p. It sets up the public key elements as

$$\begin{aligned}& E_1 = g^{r_1},\ E_2 = g^{r_2}, \cdots,\ E_m = g^{r_m},\\ & H_1 = h^{c_1},\ H_2 = h^{c_2}, \cdots,\ H_j = (h^c)^{c_j}, \cdots,\ H_m = h^{c_m},\\ & U_1 = g^{\beta_1},\ U_2 = g^{\beta_2}, \cdots,\ U_m = g^{\beta_m},\\ & V_1 = h^{\beta_1},\ V_2 = h^{\beta_2}, \cdots,\ V_m = h^{\beta_m},\\ & \Lambda_1 = e(g,h)^{\alpha_1},\ \Lambda_2 = e(g,h)^{\alpha_2}, \cdots,\ \Lambda_m = e(g,h)^{\alpha_m}.\end{aligned}$$

$\mathcal{B}$ chooses random $\sigma_{x,y}$ for all (x,y) and computes the private key $K_{(x,y)}$ for user (x,y) as

$$K_{(x,y)} = (d'_{x,y},\ d''_{x,y},\ d_1,\cdots,d_{y-1},d_{y+1},\cdots,d_m) = \begin{cases}\big(h^{\alpha_x}h^{r_x c_y}V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}},\cdots,V_{y-1}^{\sigma_{x,y}},\\ \qquad V_{y+1}^{\sigma_{y+1}},\cdots,V_m^{\sigma_{x,y}}\big),\ y \neq j,\\ \big(h^{\alpha_x}(h^c)^{r_x c_y}V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}},\cdots,V_{y-1}^{\sigma_{x,y}},\\ \qquad V_{y+1}^{\sigma_{y+1}},\cdots,V_m^{\sigma_{x,y}}\big),\ y = j.\end{cases}$$

The public key and private keys have an identical distribution to that in the actual construction, because all the exponents are chosen uniformly at random.

- **Query:** $\mathcal{A}$ gives a bit $\widetilde{s}$ as the query. If $\widetilde{s} = 1$, $\mathcal{B}$ gives out $K_{(i,j)}$.
- **Challenge:** $\mathcal{A}$ outputs the message $M \in G_T$. $\mathcal{B}$ picks random exponents s_1, s_2, θ_1, θ_2, θ_3, θ_4, $\{\phi_x\}_{x=i+1}^m$, $\{\lambda_y\}_{y=1}^{j-1}$, and $\{z_{x,1}, z_{x,2}, z_{x,3}, z_{x,4}\}_{x=1}^i$ in Z_p, under a constraint that $\theta_2\theta_3 - \theta_1\theta_4 \neq 0$ in Z_p.
  For each column y, $\mathcal{B}$ constructs column ciphertexts $(C_{y,1},\ C_{y,2})$ as follows.

$$\begin{aligned}\text{If } y < j:\ & C_{y,1} = (h^{ab})^{\frac{c_y\theta_2}{c_j}} h^{c_y s_1} h^{\theta_2\lambda_y},\\ & C_{y,2} = (h^{ab})^{-\frac{c_y\theta_1}{c_j}} h^{c_y s_2} h^{-\theta_1\lambda_y}.\\ \text{If } y = j:\ & C_{y,1} = T^{\theta_2}(h^c)^{c_j s_1},\\ & C_{y,1} = T^{-\theta_1}(h^c)^{c_j s_2}.\end{aligned}$$

$$\text{If } y > j:\ C_{y,1} = (h^{ab})^{\frac{c_y\theta_2}{c_j}} h^{c_y s_1}, \qquad C_{y,2} = (h^{ab})^{-\frac{c_y\theta_1}{c_j}} h^{c_y s_2}.$$

As in the proof of Lemma 1, $\mathcal{B}$ sets $\widetilde{s}_1 = s_1 + \theta_2 ab/c_j$ and $\widetilde{s}_2 = s_2 - \theta_1 ab/c_j$. If T is h^{abc}, then

$$\begin{aligned} C_{j,1} &= T^{\theta_2}(h^c)^{c_j s_1} = (h^{abc})^{\theta_2}(h^c)^{c_j s_1} \\ &= (h^{c_j c})^{s_1 + \frac{\theta_2 ab}{c_j}} = H_j^{\widetilde{s}_1}, \\ C_{j,2} &= T^{-\theta_1}(h^c)^{c_j s_2} = (h^{abc})^{-\theta_1}(h^c)^{c_j s_2} \\ &= (h^{c_j c})^{s_2 - \frac{\theta_1 ab}{c_j}} = H_j^{\widetilde{s}_2}. \end{aligned}$$

On the other hand, if T is random, i.e., $T = h^{abc}h^r$ for some (unknown) $r \in Z_p$, then

$$\begin{aligned} C_{j,1} &= T^{\theta_2}(h^c)^{c_j s_1} = H_j^{\widetilde{s}_1} h^{\theta_2 r}, \\ C_{j,2} &= T^{-\theta_1}(h^c)^{c_j s_2} = H_j^{\widetilde{s}_2} h^{-\theta_1 r} \end{aligned}$$

where the exponent r serves as a random exponent $\lambda_i \in Z_p$. For each row x the algorithm constructs row ciphertexts $(R_{x,1}, R_{x,2}, R_{x,3}, R_{x,4}, B_x)$ as follows.

$$\begin{aligned} \text{If } x \le i:\ & R_{x,1} = g^{z_{x,1}}, \quad R_{x,2} = g^{z_{x,2}}, \quad R_{x,3} = g^{z_{x,3}}, \\ & R_{x,4} = \big(\prod_{k\in S_x} U_k\big)^{z_{x,3}}, \quad B_x = \Lambda_x^{z_{x,4}}. \\ \text{If } x > i:\ & R_{x,1} = g^{r_x\theta_1\phi_x}, \qquad R_{x,2} = g^{r_x\theta_2\phi_x}, \\ & R_{x,3} = g^{(\theta_1 s_1 + \theta_2 s_2)\phi_x}, \\ & R_{x,4} = g^{(\sum_{k\in S_x}\beta_k)(\theta_1 s_1 + \theta_2 s_2)\phi_x}, \\ & B_x = Me(g,h)^{\alpha_x(\theta_1 s_1 + \theta_2 s_2)\phi_x}. \end{aligned}$$

When $x > i$, $\mathcal{B}$ sets ϕ_x as it is. As before, it is sufficient to check that

$$\begin{aligned} R_{x,3} &= g^{(\theta_1 s_1 + \theta_2 s_2)\phi_x} = g^{\left[\theta_1(s_1 + \frac{\theta_2 ab}{c_j}) + \theta_2(s_2 - \frac{\theta_1 ab}{c_j})\right]\phi_x} \\ &= g^{(\theta_1\widetilde{s}_1 + \theta_2\widetilde{s}_2)\phi_x}. \end{aligned}$$

► **Guess:** $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$, and then $\mathcal{B}$ outputs the same value to the asymmetric decision 3-party Diffie-Hellman challenger.
If T is h^{abc}, then the challenge ciphertext indicates the encryption in the game $\Gamma_{3,j}$; and if T is random, then the challenge ciphertext indicates the encryption in the game $\Gamma_{3,j+1}$. $\mathcal{B}$'s advantage in the reduction is straightforwardly taken from $\mathcal{A}$'s advantage.

□

Claim 4. *Under the XDH assumption, there is no probabilistic polynomial time adversary that can distinguish between games Γ_4 and Γ_5 with non-negligible probability.*

Proof: Suppose that there exists an adversary $\mathcal{A}$ which can distinguish between games Γ_4 and Γ_5 with advantage ϵ. We then build an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the XDH problem. On input $(g, g^a, g^b, h, T) \in G_1^4 \times G_2$, $\mathcal{B}$ outputs 1 if $T = g^{ab}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows.

► **Setup:** $\mathcal{B}$ selects random exponents $\{r_k, c_k, \alpha_k, \beta_k\}_{k=1}^m$ in Z_p. It sets up the public key elements as

$$\begin{aligned} & E_1 = g^{r_1},\ E_2 = g^{r_2}, \cdots,\ E_m = g^{r_m}, \\ & H_1 = h^{c_1},\ H_2 = h^{c_2}, \cdots,\ H_m = h^{c_m}, \\ & U_1 = g^{\beta_1},\ U_2 = g^{\beta_2}, \cdots,\ U_m = g^{\beta_m}, \\ & V_1 = h^{\beta_1},\ V_2 = h^{\beta_2}, \cdots,\ V_m = h^{\beta_m}, \\ & \Lambda_1 = e(g,h)^{\alpha_1},\ \Lambda_2 = e(g,h)^{\alpha_2}, \cdots,\ \Lambda_m = e(g,h)^{\alpha_m}. \end{aligned}$$

$\mathcal{B}$ selects a random $\sigma_{x,y} \in Z_p$ for all (x, y) and computes the private key $K_{(x,y)}$ for user (x, y) as follows.

$$\begin{aligned} K_{(x,y)} &= \big(d'_{x,y},\ d''_{x,y},\ d_1, \cdots, d_{y-1}, d_{y+1}, \cdots, d_m\big) \\ &= \big(h^{\alpha_x} h^{r_x c_y} V_y^{\sigma_{x,y}},\ h^{\sigma_{x,y}},\ V_1^{\sigma_{x,y}}, \cdots,\ V_{y-1}^{\sigma_{x,y}}, \\ &\qquad V_{y+1}^{\sigma_{x,y}}, \cdots,\ V_m^{\sigma_{x,y}}\big),\ \forall\, x,\ y. \end{aligned}$$

The public key has an identical distribution to that in the actual construction, because all the exponents are chosen uniformly at random.

► **Query:** $\mathcal{A}$ gives a bit $\widetilde{s}$ as the query. If $\widetilde{s} = 1$, $\mathcal{B}$ gives out $K_{(i,j)}$.

► **Challenge:** $\mathcal{A}$ outputs the message $M \in G_T$. $\mathcal{B}$ picks random exponents s_1, s_2, θ_1, θ_2, θ_3, θ_4, $\{\phi_x\}_{x=i+1}^m$, and $\{z_{x,1}, z_{x,2}, z_{x,3}, z_{x,4}\}_{x=1}^i$ in Z_p, under a constraint that $\theta_2\theta_3 - \theta_1\theta_4 \neq 0$. Define $\widetilde{\theta_1} = \theta_1 a + \theta_3$ and $\widetilde{\theta_2} = \theta_2 a + \theta_4 \in Z_p$.
For each row x the algorithm constructs row ciphertexts $(R_{x,1}, R_{x,2}, R_{x,3}, R_{x,4}, B_x)$ as follows.

If $x \le i$:

$$\begin{aligned} & R_{x,1} = g^{z_{x,1}}, \quad R_{x,2} = g^{z_{x,2}}, \quad R_{x,3} = g^{z_{x,3}}, \\ & R_{x,4} = \big(\prod_{k\in S_x} U_k\big)^{z_{x,3}}, \quad B_x = \Lambda_x^{z_{x,4}}. \end{aligned}$$

If $x = i + 1$:

$$\begin{aligned} R_{x,1} &= T^{r_x\theta_1\phi_x}(g^b)^{r_x\theta_3\phi_x}, \\ R_{x,2} &= T^{r_x\theta_2\phi_x}(g^b)^{r_x\theta_4\phi_x}, \\ R_{x,3} &= T^{(\theta_1 s_1 + \theta_2 s_2)\phi_x}(g^b)^{(s_1\theta_3 + s_2\theta_4)\phi_x}, \\ R_{x,4} &= T^{(\sum_{k\in S_x}\beta_k)(\theta_1 s_1 + \theta_2 s_2)\phi_x} \\ &\quad \cdot (g^b)^{(\sum_{k\in S_x}\beta_k)(s_1\theta_3 + s_2\theta_4)\phi_x}, \\ B_x &= Me(T,h)^{\alpha_x(\theta_1 s_1 + \theta_2 s_2)\phi_x} e(g^b,h)^{\alpha_x(s_1\theta_3 + s_2\theta_4)\phi_x}. \end{aligned}$$

If $x \ge i + 2$:

$$\begin{aligned} R_{x,1} &= (g^a)^{r_x\theta_1\phi_x} g^{r_x\theta_3\phi_x}, \\ R_{x,2} &= (g^a)^{r_x\theta_2\phi_x} g^{r_x\theta_4\phi_x}, \\ R_{x,3} &= (g^a)^{(\theta_1 s_1 + \theta_2 s_2)\phi_x} g^{(s_1\theta_3 + s_2\theta_4)\phi_x}, \\ R_{x,4} &= (g^a)^{(\sum_{k\in S_x}\beta_k)(\theta_1 s_1 + \theta_2 s_2)\phi_x} \\ &\quad \cdot g^{(\sum_{k\in S_x}\beta_k)(s_1\theta_3 + s_2\theta_4)\phi_x}, \\ B_x &= Me(g^a,h)^{\alpha_x(\theta_1 s_1 + \theta_2 s_2)\phi_x} e(g,h)^{\alpha_x(s_1\theta_3 + s_2\theta_4)\phi_x}. \end{aligned}$$

For each column y, the algorithm creates column ciphertexts $(C_{y,1}, C_{y,2})$ as follows.

$$C_{y,1} = h^{c_y s_1},\ C_{y,2} = h^{c_y s_2},\ \text{for } \forall\, y.$$

When $x \geq i+2$, $\mathcal{B}$ sets ϕ_x as it is. Then, the row ciphertexts can be computed as

$$\begin{aligned} R_{x,1} &= g^{r_x(\theta_1 a+\theta_3)\phi_x} = (g^a)^{r_x\theta_1\phi_x} g^{r_x\theta_3\phi_x}, \\ R_{x,2} &= g^{r_x(\theta_2 a+\theta_4)\phi_x} = (g^a)^{r_x\theta_2\phi_x} g^{r_x\theta_4\phi_x}, \\ R_{x,3} &= g^{\left[(\theta_1 a+\theta_3)s_1+(\theta_2 a+\theta_4)s_2\right]\phi_x} \\ &= (g^a)^{(\theta_1 s_1+\theta_2 s_2)\phi_x} g^{(s_1\theta_3+s_2\theta_4)\phi_x}, \\ R_{x,4} &= \big(\prod_{k\in S_x} U_k\big)^{\left[(\theta_1 a+\theta_3)s_1+(\theta_2 a+\theta_4)s_2\right]\phi_x} \\ &\quad = (g^a)^{(\sum_{k\in S_x}\beta_k)(\theta_1 s_1+\theta_2 s_2)\phi_x} \\ &\qquad \cdot g^{(\sum_{k\in S_x}\beta_k)(s_1\theta_3+s_2\theta_4)\phi_x}, \end{aligned}$$

$$\begin{aligned} B_x &= Me(g,h)^{\alpha_x\left[a(\theta_1 s_1+\theta_2 s_2)\phi_x+(s_1\theta_3+s_2\theta_4)\phi_x\right]} \\ &= Me(g^a,h)^{\alpha_x(\theta_1 s_1+\theta_2 s_2)\phi_x} e(g,h)^{\alpha_x(s_1\theta_3+s_2\theta_4)\phi_x}. \end{aligned}$$

Next, when $x = i+1$, $\mathcal{B}$ sets $\widetilde{\phi}_{i+1} = \phi_{i+1}b \in Z_p$. If $T = g^{ab}$, then row $i+1$ is the greater-than row and thus $\mathcal{B}$ is playing game Γ_4 with $\mathcal{A}$. On the other hand, if T is random, i.e., $T = g^{(a+r)b}$ for some (unknown) non-zero $r \in Z_p$, then row $i+1$ is the target row under randomness

$$\widetilde{\theta}_3 = \theta_1(a+r)+\theta_3, \qquad \widetilde{\theta}_4 = \theta_2(a+r)+\theta_4.$$

The distribution of these two exponents is identical to that of real scheme, since two random values $\theta_1 r$ and $\theta_2 r$ are embedded into $\widetilde{\theta}_3$ and $\widetilde{\theta}_4$. In addition, $\mathcal{B}$ has that $\widetilde{\theta}_2\widetilde{\theta}_3 - \widetilde{\theta}_1\widetilde{\theta}_4 \neq 0$, because

$$\begin{aligned} &\widetilde{\theta}_2\widetilde{\theta}_3 - \widetilde{\theta}_1\widetilde{\theta}_4 \\ &\quad = (\theta_2 a+\theta_4)(\theta_1(a+r)+\theta_3) \\ &\qquad - (\theta_1 a+\theta_3)(\theta_2(a+r)+\theta_4) \\ &\quad = (\theta_1\theta_4-\theta_2\theta_3)r \end{aligned}$$

where $\theta_1\theta_4 - \theta_2\theta_3 \neq 0$ and $r \neq 0$ (as T is random). In this case, the validity of row ciphertexts can be checked as follows.

$$\begin{aligned} R_{i+1,1} &= T^{r_{i+1}\theta_1\phi_{i+1}}(g^b)^{r_{i+1}\theta_3\phi_{i+1}} \\ &= (g^{(a+r)b})^{r_{i+1}\theta_1\phi_{i+1}}(g^b)^{r_{i+1}\theta_3\phi_{i+1}} = g^{r_{i+1}\widetilde{\theta}_3\widetilde{\phi}_{i+1}}, \\ R_{i+1,2} &= T^{r_{i+1}\theta_2\phi_{i+1}}(g^b)^{r_{i+1}\theta_4\phi_{i+1}} \\ &= (g^{(a+r)b})^{r_{i+1}\theta_2\phi_{i+1}}(g^b)^{r_{i+1}\theta_4\phi_{i+1}} = g^{r_{i+1}\widetilde{\theta}_4\widetilde{\phi}_{i+1}}, \\ R_{i+1,3} &= T^{(\theta_1 s_1+\theta_2 s_2)\phi_{i+1}}(g^b)^{(s_1\theta_3+s_2\theta_4)\phi_{i+1}} \\ &= (g^{(a+r)b})^{(\theta_1 s_1+\theta_2 s_2)\phi_{i+1}}(g^b)^{(s_1\theta_3+s_2\theta_4)\phi_{i+1}} \\ &= g^{(\widetilde{\theta}_3 s_1+\widetilde{\theta}_4 s_2)\widetilde{\phi}_{i+1}}, \\ R_{i+1,4} &= T^{(\sum_{k\in S_x}\beta_k)(\theta_1 s_1+\theta_2 s_2)\phi_{i+1}} \\ &\qquad \cdot (g^b)^{(\sum_{k\in S_x}\beta_k)(s_1\theta_3+s_2\theta_4)\phi_{i+1}} \\ &= g^{(\sum_{k\in S_x}\beta_k)(\widetilde{\theta}_3 s_1+\widetilde{\theta}_4 s_2)\widetilde{\phi}_{i+1}} \\ &= \big(\prod_{k\in S_x} U_k\big)^{(\widetilde{\theta}_3 s_1+\widetilde{\theta}_4 s_2)\widetilde{\phi}_{i+1}}, \end{aligned}$$

$$\begin{aligned} B_{i+1} &= Me(T,h)^{\alpha_{i+1}(\theta_1 s_1+\theta_2 s_2)\phi_{i+1}} \\ &\qquad \cdot e(g^b,h)^{\alpha_{i+1}(s_1\theta_3+s_2\theta_4)\phi_{i+1}} \\ &= Me(g,h)^{\alpha_{i+1}\left[(a+r)b(\theta_1 s_1+\theta_2 s_2)\phi_{i+1}+b(s_1\theta_3+s_2\theta_4)\phi_{i+1}\right]} \\ &= Me(g,h)^{\alpha_{i+1}(\widetilde{\theta}_3 s_1+\widetilde{\theta}_4 s_2)\widetilde{\phi}_{i+1}}. \end{aligned}$$

Thus, $\mathcal{B}$ plays game Γ_5 with $\mathcal{A}$.

► **Guess:** $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$, and then $\mathcal{B}$ outputs the same value to the XDH challenger.

If T is g^{ab}, then the challenge ciphertext corresponds to game Γ_4; and if T is random, then it corresponds to game Γ_5. It is easy to see that $\mathcal{B}$'s advantage in the reduction is straightforwardly taken from $\mathcal{A}$'s advantage. □

B. Message Hiding

Theorem 2: No adversary can distinguish between two ciphertexts when encryption is done to the $(m+1,1)$.

Proof: The message hiding game considers an adversary who tries to break semantic security when encryption is done to the index $(m+1,1)$. However, in this encryption, all row ciphertexts will be constructed with randomly chosen exponents and thus be independent of two messages challenged upon. Thus, the adversary has 0 advantage in deciding which message has been encrypted. □

V. PERFORMANCE COMPARISON

In case of ABE scheme. We present performance comparison between the BW-ABE scheme [10] and ours. As pointed out in [14], we consider our ABE scheme as a key encapsulation mechanism (KEM) which leads to saving the ciphertext size. In the resultant KEM (using the notation of ABE scheme in Section III), the set $\{B_x\}_{i=1}^m$ is not included into a ciphertext, and instead user (x,y) can recover a key K_x through the calculation of $K_x \leftarrow e(R_{x,3}, K'_{(x,y)})/e(R_{x,1}, C_{y,1})e(R_{x,2}, C_{y,2})e(R_{x,3}, d''_{x,y})$. When encrypting to a position (i,j), the key K_x is computed as follows.

$$K_x = \begin{cases} R' \overset{R}{\leftarrow} G_T, & x < i, \\ \Lambda_x^{(\theta_3 s_1+\theta_4 s_2)\phi_x} \in G_T, & x = i, \\ \Lambda_x^{(\theta_1 s_1+\theta_2 s_2)\phi_x} \in G_T, & x > i. \end{cases}$$

Then, the key K_x is used as a symmetric key to encrypt a message encryption key K for each row x, so that all targeted recipients get to share the same key K. In reality, the encryption algorithm $\textbf{Encrypt}_{\text{ABE}}$ requires to compute $\{K_x\}_{x=1}^m$ in G_T and the ciphertext contains $\{S_{K_x}(K)\}_{x=1}^m$, instead of $\{B_x\}_{i=1}^m$, where S is a symmetric key cipher. Since $|S_{K_x(K)}| = 128$ bits (using AES) and $|B_x| \geq 1024$ bits (see Table 2), the KEM approach is clearly more transmission-efficient than the case including the set $\{B_x\}_{i=1}^m$ of elements in G_T. This approach can also be applied to the BW-ABE scheme in the same way.

Under the consideration of the KEM approach, Table 1 presents overall performance comparisons between the BW-ABE scheme and ours. For simplicity, we do not consider simple operations such as multiplication and symmetric key encryption. We note that the BW-ABE scheme and ours are fully

Table 1. Performance comparison between the BW-ABE scheme and ours for $N = m^2$.

	BW-ABE	Our ABE scheme
Group order	Composite pq	Prime p
Tracing type	Public tracing	Public tracing
Ciphertext size	$6m\,\mathsf{len}(\mathsf{G}) + m\,\mathsf{len}(\mathcal{SE})$	$4m\,\mathsf{len}(\mathsf{G}_1) + 2m\,\mathsf{len}(\mathsf{G}_2) + m\,\mathsf{len}(\mathcal{SE})$
Private key size	$(m+1)\,\mathsf{len}(\mathsf{G})$	$(m+1)\,\mathsf{len}(\mathsf{G}_2)$
Decryption cost	4 pairings	4 pairings
Encryption cost	$9m\,\mathsf{exp}(\mathsf{G}) + (m+1)\,\mathsf{exp}(\mathsf{G}_\mathsf{T})$	$4m\,\mathsf{exp}(\mathsf{G}_1) + 4m\,\mathsf{exp}(\mathsf{G}_2) + (m+1)\,\mathsf{exp}(\mathsf{G}_\mathsf{T})$

p, q: Primes, $\mathsf{len}(\mathcal{SE})$: Output size of a symmetric key encryption.
$\mathsf{len}(G, G_1, G_2)$: Size of one group element in G, G_1, and G_2, respectively.
$\mathsf{exp}(\mathsf{G}, \mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_\mathsf{T})$: Exponentiation performed in G, G_1, G_2, and G_T, respectively.

collusion-resistant and secure against adaptive adversaries. Table 1 shows that the decryption costs are determined by four pairing operations that are independent of N or m, and the encryption costs are dominated by exponentiations that are performed in G, G_1, G_2, and G_T. To evaluate performance more concretely, we use the cost results in Table 2, which was present in [11]. Here, the pairing time is estimated by using the PBC Library on its website.[2] Based on Table 2, the encryption cost of our ABE scheme is roughly $(1024^3 10)m/(4160170^2 + 4160510^2 + 1024^3)m \approx 8.5$ times faster than that of the BW-ABE scheme, and the decryption cost of our scheme is roughly $3028/256 \approx 12$ times faster than that of the BW-ABE scheme. These encryption and decryption costs will become the important efficiency factors of a tracing algorithm, which can be derived from the ABE scheme.

Table 2. Cost of different operations.

	Symmetric & order pq	Asymmetric & order p
Group order (r)	1024 bits	$\leq$ 160 bits
Group element size (b)	1024 bits in G, 1024 bits in G_T	170 bits in G_1, 510 bits in G_2, 1024 bits in G_T
Exponentiation time	$O(rb^2)$	$O(rb^2)$
Pairing time	757 ms	$\leq$ 64 ms

Table 3. Performance comparison for $N = m^2$.

	BW-ABE	Our ABE scheme
Pairing type	Symmetric & order pq	Asymmetric & order p
Ciphertext size	$6272m$ bits	$1828m$ bits
Private key size	$1024m$ bits	$512m$ bits
Decryption cost	3028 ms	$\leq$ 256 ms

ms $= 10^{-3}$ seconds.

Table 3 shows the efficiency result for $N = m^2$ when applying Table 2 to the BW-ABE scheme and ours. Here, we set the output size $\mathsf{len}(\mathcal{SE})$ of a symmetric key encryption to 128 bits by considering AES. Then, the ciphertext size of our scheme is roughly $6272m/1828m \approx 3.4$ times shorter than that of the BW-ABE scheme, and the private key size of our scheme is two times shorter than that of the BW-ABE scheme. If we set $N = 10^6$ (i.e., $m = 10^3$), then Table 3 gives a more concrete numerical result where our ABE scheme has a ciphertext of 223 kB and a private key of 62 kB. Also, a decryption can be performed within at most 256 ms.

In case of tracing algorithm. As explained in Section II, BW [10] showed that a secure ABE scheme gives a trace-and-revoke scheme, based on a general tracing method previously introduced by [4], [26], and [27]. Since their conversion is generic, we use our ABE scheme to obtain a new trace-and-revoke scheme. Now, we provide a comparison of tracing time between two tracing algorithms that are derived from the BW-ABE scheme and ours. For reader's convenience, we review the tracing algorithm $\textbf{Trace}^{\mathcal{D}}(S_{\mathcal{D}}, \mathsf{PK}, \epsilon)$, where a pirate decoder $\mathcal{D}$ will decrypt all ciphertexts encrypted for a certain set $S_{\mathcal{D}}$. For a given N, λ, and $\epsilon > 0$, the tracing algorithm works as follows.

```
1. Initialize set $T$ to the empty set.
2. For $i = 1$ to $N$, do the following:
   (a) For $8\lambda(N/\epsilon)^2$ times, the algorithm repeats the following steps:
       i. Sample $M$ from the finite message space at random.
       ii. Let $C \xleftarrow{R} \textbf{Encrypt}_{\text{ABE}}(S_{\mathcal{D}}, \mathsf{PK}, i, M)$.
       iii. Call oracle $\mathcal{D}$ on input $C$, and compare the output of $\mathcal{D}$ to $M$.
   (b) Let $\widehat{p}_i$ be the fraction of times that $\mathcal{D}$ decrypted the ciphertexts correctly.
   (c) If $\widehat{p}_i - \widehat{p}_{i+1} \geq \epsilon/(4N)$, then add $i$ to set $T$.
3. Output the set $T$.
```

For simplicity, we assume that the time for $\mathcal{D}$ to answer one oracle query is the same as the usual decryption time. Then, the total tracing time is estimated by $N(8\lambda(N/\epsilon)^2(T_E + T_D))^3$, where T_E and T_D are decryption and decryption times, respectively. According to the above performance result, our ABE scheme encrypts about 8.5 times faster and decrypts about 12 times faster than the BW-ABE scheme. As the total number N (and thus m) of users is large, T_E will be definitely greater than T_D because T_D is constant and T_E grows linearly with m. However, if we consider precomputations for calculating exponentiations[4] and extra devices for parallelizing exponentiations, we can assume that $T_D = T_E$ at a certain setting for a given N. Under this circumstance (as well as same level of pa-

[2] [Online]. Avaliable: http://crypto.stanford.edu/pbc/

[3] The asymptotic value $O(N^3)$ can be made (almost) quadratic using binary search instead of a linear scan [10].

[4] Such precomputations are possible since exponentiations in encryption can be done, based on public parameters.

rameters such as λ and ϵ), our tracing algorithm works roughly $(8.5T_E + 12T_D)/(T_E + T_D) \approx 10$ times faster than the tracing algorithm from the BW-ABE scheme. This means that if it takes 30 days for the tracing algorithm from the BW-ABE scheme to detect all traitors from $\mathcal{D}$, our tracing algorithm can do the same work in only 3 days.

VI. CONCLUSION

We constructed a new trace-and-revoke scheme that is fully collusion-resistant and works in prime-order groups. Our new scheme is publicly traceable and secure against adaptive adversaries. Our trace-and-revoke scheme was based on asymmetric bilinear maps in prime order groups, and based on cancellation effect same as in composite-order groups. To achieve the cancellation effect, we introduce a new trick to generate positive and negative pairing values and cancel them out. We proved the security of our scheme under the asymmetric decision 3-party Diffie-Hellman and XDH assumptions.

Prior to our scheme, the earlier trace-and-revoke scheme of BW [10] was the only scheme that is fully collusion-resistant, but the BW scheme was constructed over composite-order groups. By the difference of base group, our scheme can achieve shorter ciphertexts and private keys, and obtain faster encryption and decryption and tracing time. For a subsequent work, it will be interesting to design a new trace-and-revoke scheme in prime-order groups with security proven from standard assumptions such as the decisional bilinear Diffie-Hellman and the Decision Linear assumptions.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their helpful comments and suggestions for this paper.

REFERENCES

[1] M. Naor and B. Pinkas, "Effcient trace and revoke schemes," in *Proc. FC*, vol. 1962, 2000, pp. 1–20.

[2] A. Fiat and M. Naor, "Broadcast encryption," in *Proc. CRYPTO*, vol. 773, 1993, pp. 480–491.

[3] B. Chor, A. Fiat, and M. Naor, "Tracing traitors," in *Proc. CRYPTO*, vol. 839, 1994, pp. 257–270.

[4] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proc. CRYPTO*, vol. 2139, 2001, pp. 41–62.

[5] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," in *Proc. CRYPTO*, vol. 2442, 2002, pp. 47–60.

[6] M. T. Goodrich, J. Z. Sun, and R. Tamassia, "Effcient tree-based revocation in groups of low-state devices," in *Proc. CRYPTO*, vol. 2204, 2004, pp. 511–527.

[7] E. Gahai, J. Staddon, and Y. L. Yin, "Efficient methods for integrating traceability and broadcast encryption," in *Proc. CRYPTO*, vol. 1666, 1999, pp. 372–387.

[8] W. Tzeng and Z. Tzeng, "A public-key traitor tracing scheme with revocation using dynamic shares," in *Proc. PKC*, vol. 1992, 2001, pp. 207–224.

[9] Y. Dodis and N. Fazio, "Public key trace and revoke scheme secure against adaptive chosen ciphertext attack," in *Proc. PKC*, vol. 2696, 2003, pp. 100–115.

[10] D. Boneh and B. Waters, "A fully collusion resistant broadcast, trace, and revoke system," in *Proc. ACM-CCS*, ACM, 2006, pp. 211–220.

[11] S. Garg, A. Sahai, and B. Waters. Efficient fully collusion-resilient traitor tracing scheme. Cryptology ePrint Archive. Report 2009/532. [Online]. Available: http://eprint.iacr.org/2009/532/

[12] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. CRYPTO*, vol. 3621, 2005, pp. 258–275.

[13] J. H. Park, H. J. Kim, M. H. Sung, and D. H. Lee, "Public key broadcast encryption schemes with shorter transmissions," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 401–411, 2008.

[14] D. Boneh, A. Sahai, and B. Waters, "Fully collusion resistant traitor tracing with short ciphertexts and private keys," in *Proc. EUROCYYPT*, vol. 4004, 2006, pp. 573–592.

[15] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO*, vol. 2139, 2001, pp. 213–229.

[16] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. CRYPTO*, vol. 3152, 2004, pp. 41–55.

[17] M. Scott. (2002). Authenticated id-based key exchange and remote log-in with simple token and pin number. Cryptology ePrint Archive, Report 2002/164, 2002, [Online]. Available: http://eprint.iacr.org/2002/164/

[18] S. Galbraith, K. Paterson, and N. Smart, "Pairings for cryptographers," *Discrete Appl. Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.

[19] L. Ducas, "Anonymity from asymmetry: New constructions for anonymous HIBE," in *Proc. CT-RSA*, vol. 5985, 2010, pp. 148–164.

[20] S. D. Galbraith, "Supersingular curves in cryptography," in *Proc. ASIACRYPT*, vol. 2248, 2001, pp. 495–513.

[21] N. McCullagh and P. S. L. M. Barreto, "A new two-party identity-based authenticated key agreement," in *Proc. CT-RSA*, 2005, vol. 3376, pp. 262–274.

[22] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. (2005) Compact e-cash, Cryptology ePrint Archive, Report 2005/060. [Online]. Available: http://eprint.iacr.org/2005/060/

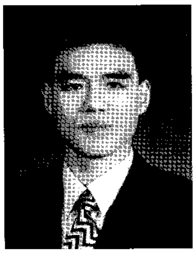
[23] G. Ateniese, J. Camenisch, and B. D. Medeiros, "Untraceable rfid tags via insubvertible encryption," in *Proc. ACM-CCS*, 2005, pp. 92–101.

[24] J. H. Park and D. H. Lee, "A new public key broadcast encryption using Boneh-Boyen-Goh's HIBE scheme," in *Proc. ISPEC*, vol. 4991, 2008, pp. 101–115.

[25] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," in *Proc. EUROCRYPT*, vol. 5479, 2009, pp. 171–188.

[26] D. Boneh and M. K. Franklin, "An effcient public key traitor tracing scheme," in *Proc. CRYPTO*, vol. 1666, 1999, pp. 338–353.

[27] A. Kiayias and M. Yung, "On crafy pirates and foxy tracers," in *Proc. ACM-DRM*, 2001, pp. 22–39.



Jong Hwan Park received the B.S. degree in the Department of Mathematics from Korea University, Seoul, Korea, in 1999, and the M.S. and Ph.D. degrees in Information Security from Korea University, Seoul, Korea, in 2004 and 2008, respectively. In 2008, he was in a postdoc position in Korea University. Since 2009, he has served as a Research Professor in the Department of Applied Mathematics from Kyung Hee University, Young-in, Korea. His research areas include pairing-based encryption, broadcast encryption, and predicate encryption.

Hyun Sook Rhee received the B.S. and the M.S. degrees in Department of Mathematics from Dankook University, Korea, in 1998 and 2000, respectively. She received the Ph.D. degree in Information Security from Korea University, Korea, in 2008. In 2008, she was in a Research Fellow position in Wollongong University, Australia. Since 2009, she has served as a Research Professor in the Center for Information Security Technologies at Korea University. Her research areas include public-key encryption, searchable encryption, and privacy enhanced technologies.

Dong Hoon Lee was born in Republic of Korea in 1959. He received the B.S. degree in Economics from the Korea University in 1984. He received the M.S. and Ph.D. degrees in Computer Science from the University of Oklahoma in 1988 and 1992, respectively. He is currently a Full Professor in the Graduate School of Information Security at Korea University, Seoul. His research areas include cryptography and information security.