Multireceiver Predicate Encryption for Online Social Networks

Chun-I Fan, *Member, IEEE*, Yi-Fan Tseng, Jheng-Jia Huang, Shih-Fen Chen, and Hiroaki Kikuchi, *Member, IEEE*

***Abstract*—Among the applications of the internet and cloud computing, online social network (OSN) is a very popular service. Since a lot of personal information is stored on the OSN platform, privacy protection on such an application has become a critical issue. Apart from this, OSN platforms need advertisement revenue to enable continued operations. However, if the users encrypt their messages, then OSN providers cannot generate accurate advertisement to users. Thus, how to achieve both privacy preserving and accurate advertisement is a worth-discussing issue. Unfortunately, none of the works on OSNs can achieve both privacy preserving and accurate advertisement simultaneously. In view of this, we propose the first multireceiver predicate encryption scheme for OSN platforms. Not only does the proposed scheme protects the users' privacy but it achieves customized advertisement as well. Compared with other predicate encryptions deployed in OSN platforms, the proposed scheme gains much shorter ciphertext. The semantic security and attribute hiding of the proposed scheme are proved under the standard model.**

***Index Terms*—Bilinear groups of composite order, inner product encryption, multi-receiver encryption, online social networks, predicate encryption.**

I. Introduction

Internet and cloud computing are thriving over the whole world in recent years. One of the most popular and diverse services is online social networks (OSNs), such as Facebook, Google, Dropbox, Twitter, and so on. A lot of personal information will be stored into OSN platforms, so that the security of OSN platforms should be guaranteed. Many works on the privacy preservation of OSNs have been proposed [1], [2], [4], [6], [8], [14]–[16], [25]–[29]. In the architecture of an OSN platform, OSN providers make profits from advertisement revenue to enable continued operations. However, protecting user privacy and producing accurate advertisement simultaneously might be a contradiction in OSN platforms due to the following reasons.

Manuscript received September 14, 2016; revised December 23, 2016; accepted April 11, 2017. Date of publication April 24, 2017; date of current version May 17, 2017. This work was supported by the Ministry of Science and Technology of the Taiwan under Grants MOST 105-2923-E-110-001-MY3 and MOST 105-2221-E-110-053-MY2. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Zhenliang Zhang. *(Corresponding author: Chun-I Fan.)*

C.-I. Fan, Y.-F. Tseng, J.-J. Huang, and S.-F. Chen are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: cifan@cse.nsysu.edu.tw; yftseng1989@gmail.com; jhengjia.huang@gmail.com; csfdora@gmail.com).

H. Kikuchi is with the Department of Frontier Media Science, Meiji University, Tokyo 164-8525, Japan (e-mail: kikn@meiji.ac.jp).

Digital Object Identifier 10.1109/TSIPN.2017.2697580

1) OSN providers extract the keywords from users' data and messages for advertisers. However, this needs users' data to be in non-encrypted forms and thus exposes the privacy of users.
2) If users encrypt the data before posting for privacy preserving, then OSN providers cannot extract the keywords from the ciphertext.

A straightforward solution to this problem would be predicate encryption (PE), which was first introduced by Katz $et\ al.$ [9] in 2008. Such encryption mechanisms provide an evaluation for encrypted messages with predicate tokens, which makes it feasible to search in ciphertext space. There are two types in PE: asymmetric predicate encryption (ASPE) [5], [9]–[11], [13], [20], [21], [31] and symmetric predicate encryption (SPE) [3], [7], [22], [24], [30], [32]. The main difference between these two types is the identity of the searcher. SPE is appropriate for the systems where the searcher is the one who encrypts the data, such as personal cloud storages. In an ASPE system, nevertheless, the searcher is not necessarily the encryptor of the data. Hence, ASPE is fitting for secure e-mail systems or credit card payment gateways. It seems that ASPE might be more suitable in solving the contradictory scenario in OSN platforms. Furthermore, the keywords of ASPE are associated with the ciphertext, which is suitable for OSN providers to produce customized advertisement efficiently. When ASPE is applied, however, the encryption procedure needs to use the parameters defined by the receiver to enable the search. This requirement will cause a great cost on communication. For instance, if a sender wants to share a file with an n-dimensional predicate vector to t receivers, then it will result in a ciphertext of $O(n \times t)$ length. In order to cope with the problems mentioned above for the OSN platform, we propose a multi-receiver predicate encryption (MRPE) scheme. The main difference between ASPE and MRPE is that, in an ASPE scheme, each user will generate his own public parameters. As we mentioned above, this would lead to the undesirable expansion of ciphertexts, since a sender should use different public parameters to execute the encryption algorithm for each receiver. In our MRPE scheme, the public parameters are defined by a third party, and the encryption process can be performed with inputting a set of receivers. Since the public parameters are independent of the receivers, the length of a cipheretxt can be compressed. This property cannot be achieved in ASPE because in an ASPE, a tuple of public parameters would correspond to a secret key. If each user shares the same public parameters in an ASPE, they will also share the same secret key. In the proposed MRPE scheme, we allow each user to choose

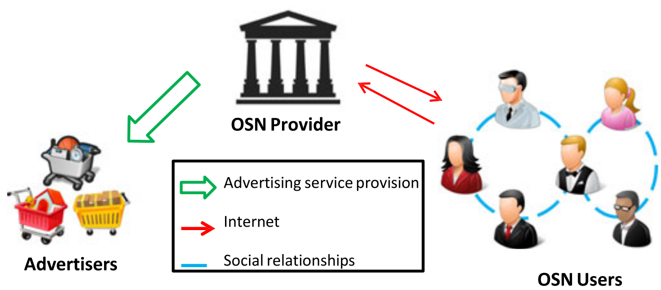

Fig. 1. The model of online social networks.

a part of his own secret value, while sharing the same public parameters. Thus, when the proposed MRPE scheme is applied to OSN, not only do the users protect their privacy but also they can search the interested ciphertext efficiently. Furthermore, the OSN provider is capable of finding corresponding keywords and producing customized advertisement, and moreover, the length of a ciphertext is $O(n+t)$ only.

II. Preliminaries

In this section, we give the definitions of the model of OSNs and asymmetric predicate encryption, and also review some hard problems and assumptions.

A. The Model of Online Social Networks

We define three distinct characters in the OSNs model: OSN providers, users, and advertisers. The relationships among each character are illustrated in Fig. 1.

1) The relationship between OSN providers and users: For those users who upload information and keep in touch with their friends in online social networks, OSN providers offer the storage for them to store, upload, share, and view the data. Besides the benefits stated above, it also provides data access control services to enable users to make access policies by themselves.
2) The relationship between OSN providers and advertisers: The OSN providers have users' data, which have been uploaded by users, and they usually contain valuable market information for advertisers who buy commercial keywords from OSN providers to send the users customized advertisements. Thus, OSN providers gain advertisement profits from the advertisers.
3) The relationship among users: By privacy setting, each sender can dynamically choose receivers and set access policy of information.

B. Asymmetric Predicate Encryption

For a positive integer N, let $\mathbb{Z}_N$ denote the set of non-negative integers smaller than N. Also, we use $\mathbb{Z}_N^n$ to represent the set of the n-dimension vectors where each component of each vector is in $\mathbb{Z}_N$.

Definition 2.1: An ASPE scheme for the class of predicates $\mathcal{F}=\{f_{\vec{v}}|\vec{v}\in\mathbb{Z}_N^n\}$ and attributes $\Sigma=\mathbb{Z}_N^n$ where $f_{\vec{v}}(\vec{x})=1$ iff $\langle\vec{v},\vec{x}\rangle=0 \bmod N$ (Reveal nothing about $\vec{x}$). It consists of probabilistic polynomial-time algorithms **Setup, GenKey, Enc,** and **Dec** as follows [9].

1) **Setup** is an algorithm that takes as input a security parameter n. It returns a secret key SK and the public key PK.
2) **GenKey** is an algorithm that takes as input SK and a predicate $\vec{v}\in\mathcal{F}$, and then returns a token $SK_{\vec{v}}$.
3) **Enc** is an algorithm that takes as input the public key PK, a keyword $\vec{x}\in\Sigma$, and a message M, and it then returns a ciphertext C. We write this as $C\leftarrow \text{Enc}_{PK}(\vec{x},M)$.
4) **Dec** is an algorithm that takes as input a ciphertext C and the token $SK_{\vec{v}}$, and then it returns a message M or distinguished symbol $\perp$.

For correctness, $(PK,SK)\longleftarrow$ **Setup**(1^n) and $SK_{\vec{v}}\longleftarrow$ **GenKey**$_{SK}(\vec{v})$, and $\vec{x}\in\Sigma$:

1) If $\langle\vec{v},\vec{x}\rangle=0$, then $\mathbf{Dec}_{SK_{\vec{v}}}(\mathbf{Enc}_{PK}(\vec{x},M))=M$.
2) If $\langle\vec{v},\vec{x}\rangle\neq 0$, then $\mathbf{Dec}_{SK_{\vec{v}}}(\mathbf{Enc}_{PK}(\vec{x},M))=\perp$.

C. Bilinear Groups of Composite Order

Let $\mathcal{G}$ be a group generator that takes as input a security parameter n and outputs a 6-tuple $(p,q,r,\mathbb{G},\mathbb{G}_T,\hat{e})$ where p,q,r are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups of composite order $N=pqr$, and $\hat{e}:\mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ is a non-degenerate bilinear map, i.e., it satisfies

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

Let $\mathbb{G}_p,\mathbb{G}_q$ and $\mathbb{G}_r$ denote the respective subgroups of order p,q,r of $\mathbb{G}$, and then $\mathbb{G}=\mathbb{G}_p\times\mathbb{G}_q\times\mathbb{G}_r$. If g is a generator of $\mathbb{G}$, then g^{pq} is a generator of $\mathbb{G}_r$, g^{pr} is a generator of $\mathbb{G}_q$, and g^{qr} is a generator of $\mathbb{G}_p$. Furthermore, if $h_p\in\mathbb{G}_p$ and $h_q\in\mathbb{G}_q$ then

$$\hat{e}(h_p,h_q)=\hat{e}((g^{qr})^{\alpha_1},(g^{pr})^{\alpha_2})=\hat{e}(g^{\alpha_1},g^{r\alpha_2})^{pqr}=1_{\mathbb{G}_T}$$

where $\alpha_1=\log_{g^{qr}}h_p$ and $\alpha_2=\log_{g^{pr}}h_q$. In the assumptions below, let $\mathbb{G}_{T_p}$ denote the subgroup of order p in $\mathbb{G}_T$.

D. Complexity Assumption and Hard Problems

In this paper, a negligible function is a function $f(n)$ with $f(n)=o(n^{-c})$ for every fixed constant c.

1) Subgroup Decision (SD) Assumption [9]: A random element in $\mathbb{G}_q$ is indistinguishable from a random element in $\mathbb{G}$. More precisely, for a given group generator $\mathcal{G}$, define the following distribution $P(\lambda)$:

$$(p,q,r,\mathbb{G},\mathbb{G}_T,\hat{e})\xleftarrow{R}\mathcal{G}(\lambda),\ N\leftarrow pqr,\ s,a,b\xleftarrow{R}\mathbb{Z}_N,$$

$$g_p\xleftarrow{R}\mathbb{G}_p,\ g_q,Q_1,Q_2\xleftarrow{R}\mathbb{G}_q,\ g_r,R_0,R_2,R_3\xleftarrow{R}\mathbb{G}_r,$$

$$\bar{Z}\leftarrow((N,\mathbb{G},\mathbb{G}_T,\hat{e}),g_p,g_r,g_qR_0,g_p^b,g_p^{b^2},g_p^ag_q,g_p^{ab}Q_1,$$

$$g_p^s,g_p^{bs}Q_2R_2)$$

Output $\bar{Z}$.

For an algorithm $\mathcal{A}$, define $\mathcal{A}$'s advantage in solving the SD problem for $\mathcal{G}$ as,

$$\text{SD-Adv}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\bar{Z},T)=1|T=g_p^{b^2 s}R_3 \in \mathbb{G}_{pr}]$$
$$-\Pr[\mathcal{A}(\bar{Z},T)=1|T=g_p^{b^2 s}g_q^{\gamma}R_3 \in \mathbb{G}]|$$

where $\bar{Z} \leftarrow P(\lambda)$ and $\gamma \in \mathbb{Z}_N$.

Definition 2.2: We say that $\mathcal{G}$ satisfies the SD assumption if for any polynomial time algorithm $\mathcal{A}$ we have that SD-Adv$_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ.

2) Bilinear Subgroup Decision (BSD) Assumption [9]: A random order p element in $\mathbb{G}_T$ is indistinguishable from a random element in $\mathbb{G}_T$ when $g_p, g_q, g_r \in \mathbb{G}$ are given. Define $P(\lambda)$:

$$(p,q,r,\mathbb{G},\mathbb{G}_T,\hat{e}) \xleftarrow{R} \mathcal{G}(\lambda),\ N \leftarrow pqr,\ s,\mu \xleftarrow{R} \mathbb{Z}_N,$$
$$g_p, h \xleftarrow{R} \mathbb{G}_p,\ g_q, Q_1, Q_2 \xleftarrow{R} \mathbb{G}_q,\ g_r \xleftarrow{R} \mathbb{G}_r$$
$$\bar{Z} \leftarrow ((N,\mathbb{G},\mathbb{G}_T,\hat{e}), g_p,\ g_q,\ g_r,\ h,\ g_p^s,\ h^s Q_1,\ g_p^{\mu} Q_2,$$
$$\hat{e}(g_p,h)^{\mu})$$

Output $\bar{Z}$.

For an algorithm $\mathcal{A}$, define $\mathcal{A}$'s advantage in solving the BSD problem for $\mathcal{G}$ as,

$$\text{BSD-Adv}_{\mathcal{G},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\bar{Z},T)=1|T=e(g_p,h)^{\mu s}$$
$$\in \mathbb{G}_{Tp}] - \Pr[\mathcal{A}(\bar{Z},T)=1|T \in \mathbb{G}_T]|$$

where $\bar{Z} \leftarrow P(\lambda)$.

Definition 2.3: We say that $\mathcal{G}$ satisfies the BSD assumption if for any polynomial time algorithm $\mathcal{A}$ we have that BSD-Adv$_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ.

3) Assumption 3: Assumption 3 is a variant of the DBDH assumption. Define $P(\lambda)$:

$$(p,q,r,\mathbb{G},\mathbb{G}_T,\hat{e}) \xleftarrow{R} \mathcal{G}(\lambda),\ N \leftarrow pqr,\ a,b,c, \xleftarrow{R} \mathbb{Z}_p,$$
$$g_p \xleftarrow{R} \mathbb{G}_p,\ g_q \xleftarrow{R} \mathbb{G}_q,\ g_r \xleftarrow{R} \mathbb{G}_r$$
$$\bar{Z} \leftarrow ((N,\mathbb{G},\mathbb{G}_T,\hat{e}), g_p,\ g_q,\ g_r,\ g_p^a,\ g_p^b,\ g_p^c,\ g_q^a)$$

Output $\bar{Z}$.

For an algorithm $\mathcal{A}$, define $\mathcal{A}$'s advantage as,

$$\text{Adv}_{\mathcal{G},\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(\bar{Z},T)=1|T=\hat{e}(g_p,g_p)^{abc} \in \mathbb{G}_{Tp}]\right.$$
$$-\Pr[\mathcal{A}(\bar{Z},T)=\ 1|T \in_R \mathbb{G}_{Tp}]|$$

where $\bar{Z} \leftarrow P(\lambda)$.

Definition 2.4: We say that $\mathcal{G}$ satisfies Assumption 3 if for any polynomial time algorithm $\mathcal{A}$, we have that Adv$_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ.

III. The Proposed Scheme

In order to solve the problems mentioned in Section I, we design a multi-receiver predicate encryption scheme that is a predicate encryption tailored for the OSN platform. In the proposed scheme, a sender shares encrypted messages with a set of authorized receivers who can decrypt them. The OSN provider can retrieve commercial keywords from the encrypted messages for advertisers, which improves the accuracy of advertisement, without revealing the contents of the messages.

A. Overview

The proposed scheme adopts a composite order group $\mathbb{G}$, whose order N is a product of three distinct primes, p, q, and r. We define three subgroups as follows:

1) $\mathbb{G}_q$: This subgroup will be used to encode the secret key and ciphertext associated with vectors $\vec{v}$ and $\vec{x}$, respectively.
2) $\mathbb{G}_p$: This subgroup will be taken to encode an equation to protect the message.
3) $\mathbb{G}_r$: This subgroup is used to hide the information of other subgroups.

Given a random element of $\mathbb{G}$, it is hard to determine which subgroup it belongs to.

B. Multi-Receiver Predicate Encryption

In this section, we first define multi-receiver predicate encryption and propose a practical scheme.

1) Definition of Multi-Receiver Predicate Encryption:

Definition 3.1: A multi-receiver predicate encryption consists of six algorithms, **Setup**, **Join**, **PredicateExtract**, **Encrypt**, **MasterSearch**, and **Decrypt**.

1) **Setup** is an algorithm that takes as input a security parameter (1^n). It returns a master key msk and system parameters $param$.
2) **Join** is an algorithm that takes as input i which is an index of user i. It returns a key pair (PK_i, SK_i). We write this as **Join**$(i) \rightarrow (PK_i, SK_i)$.
3) **PredicateExtract** is an algorithm that takes as input a predicate vector $\vec{v}$. It returns a predicate token $SK_{\vec{v}}$. We write this as **PredicateExtract**$(param, \vec{v}) \rightarrow SK_{\vec{v}}$.
4) **Encrypt** is an algorithm that takes as input $param$, a message M, and a keyword vector $\vec{x}$, and a set $\{PK_1, PK_2, \ldots, PK_t\}$ containing the public keys of t receivers. It returns a ciphertext C. We write this as **Encrypt**$_{\{PK_i\}_{i=1}^t}(param, M, \vec{x}) \rightarrow C$.
5) **MasterSearch** is an algorithm that takes as input $param$, a ciphertext C, and a predicate token $SK_{\vec{v}}$ of a predicate vector $\vec{v}$. It returns 1 or distinguished symbol ⊥. We write this as **MasterSearch**$(param, C, SK_{\vec{v}}) \rightarrow 1/\perp$.
6) **Decrypt** is an algorithm that takes as input $param$, a ciphertext C, a predicate token $SK_{\vec{v}}$ of predicate $\vec{v}$, and secret key SK_i. It returns a message M. We write this as **Decrypt**$(param, C, SK_{\vec{v}}, SK_i) \rightarrow M$.

In the following sections, we give the threat models of MRPE, which can be defined as security games.

Definition 3.2 (Security Games Against Malicious KGC) Let $\mathcal{A}$ be a polynomial-time attacker (a malicious KGC). $\mathcal{A}$ interacts with a simulator $\mathcal{S}$ in the following game.

Setup: $\mathcal{S}$ runs the **Setup** algorithm to generate $param$ and msk. Also, $\mathcal{S}$ generates a set of public keys $\{PK_i\}_{i=1}^t$. $\mathcal{S}$ then sends $param, msk, \{PK_i\}_{i=1}^t$ to $\mathcal{A}$.

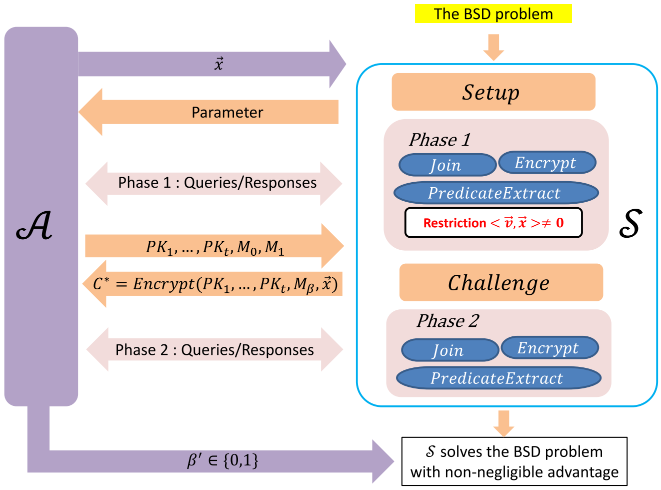

Fig. 2. The experiment of semantic security.

Phase 1: $\mathcal{A}$ requests and obtains secret keys SK_i's except for $i \in \{1, \ldots, t\}$.

Challenge: $\mathcal{A}$ submits (M_0, M_1) and a keyword vector $\vec{x}$ to $\mathcal{S}$ where M_0, M_1 are two distinct messages of the same length. $\mathcal{S}$ then randomly chooses $\beta \in \{0, 1\}$ and generates $C^* = \mathbf{Encrypt}_{\{PK_i\}_{i=1}^t}(param, M_\beta, \vec{x})$.

Phase 2: This phase is the same as that of *Phase 1*.

Guess: Eventually, $\mathcal{A}$ outputs $\beta' \in \{0, 1\}$ and wins the game if $\beta' = \beta$.

The advantage of $\mathcal{A}$ winning the game is defined as

$$\mathbf{Adv}^{\text{Malicious KGC}}(\mathcal{A}) = \left| Pr[\beta' = \beta] - \frac{1}{2} \right|.$$

A multi-receiver predicate encryption scheme is semantically secure against a malicious KGC if there exists no polynomial-time attacker that can win the above security game with non-negligible advantage.

The security goal of this game is to guarantee that none, except the users with the corresponding secret keys, can reveal the message of a ciphertext even if the attacker is a malicious KGC.

Definition 3.3 (Semantic Security) Let $\mathcal{A}$ be a polynomial-time attacker (a malicious user or an unauthenticated user). $\mathcal{A}$ interacts with a simulator $\mathcal{S}$ in the following game that is also illustrated in Fig. 2.

Initialization: $\mathcal{A}(1^n)$ outputs $\vec{x}$.

Setup: $\mathcal{S}$ runs the **Setup** algorithm to generate $param$ and msk. $\mathcal{S}$ then sends $param$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ requests and obtains secret keys SK_i's and predicate tokens $SK_{\vec{v}}$'s with the restriction that $\langle \vec{v}, \vec{x} \rangle \neq 0$ for each $\vec{v}$.

Challenge: $\mathcal{A}$ submits (M_0, M_1) and $\{PK_i\}_{i=1}^t$ to $\mathcal{S}$ where M_0, M_1 are two distinct messages of the same length and t is a positive integer. $\mathcal{S}$ then randomly chooses $\beta \in \{0, 1\}$ and generates $C^* = \mathbf{Encrypt}_{\{PK_i\}_{i=1}^t}(param, M_\beta, \vec{x})$.

Phase 2: $\mathcal{A}$ can continue querying the secret keys for any registered users and the predicate tokens for additional predicates with the same restriction as that in *Phase 1*.

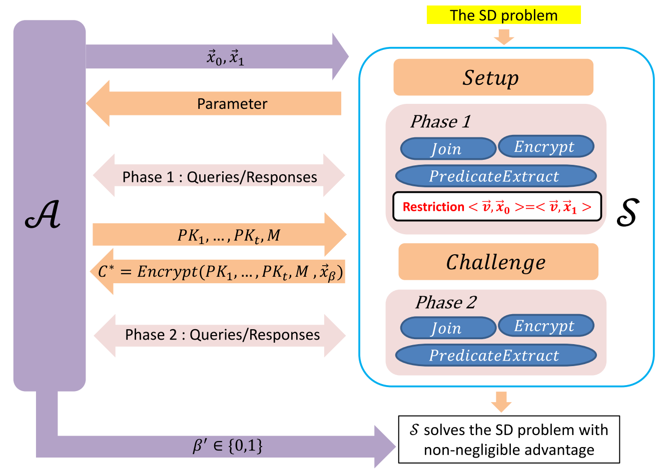

Fig. 3. The experiment of attribute hiding.

Guess: Eventually, $\mathcal{A}$ outputs $\beta' \in \{0, 1\}$ and wins the game if $\beta' = \beta$.

The advantage of $\mathcal{A}$ winning the game is defined as

$$\mathbf{Adv}^{\text{Semantic-Security}}(\mathcal{A}) = \left| Pr[\beta' = \beta] - \frac{1}{2} \right|.$$

A multi-receiver predicate encryption scheme is semantically secure if there exists no polynomial-time attacker that can win the semantic security game with non-negligible advantage.

The security goal of this game is to guarantee that no users can obtain the information of the content of a ciphertext without correct predicate tokens.

Definition 3.4 (Attribute Hiding) Let $\mathcal{A}$ be a polynomial-time attacker (a malicious user or an unauthenticated user). $\mathcal{A}$ interacts with a simulator $\mathcal{S}$ in the following game, which is also illustrated in Fig. 3.

Initialization: $\mathcal{A}(1^n)$ outputs $\vec{x}_0, \vec{x}_1$, where $\vec{x}_0 \neq \vec{x}_1$.

Setup: $\mathcal{S}$ runs the **Setup** algorithm to generate $param$ and msk. $\mathcal{S}$ then sends $param$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ requests and obtains secret keys SK_i's and predicate tokens $SK_{\vec{v}}$'s with the restriction that $\langle \vec{v}, \vec{x}_0 \rangle = \langle \vec{v}, \vec{x}_1 \rangle$ for each $\vec{v}$.

Challenge: $\mathcal{A}$ submits M and $\{PK_i\}_{i=1}^t$ to $\mathcal{S}$ where t is a positive integer. $\mathcal{S}$ then randomly chooses $\beta \in \{0, 1\}$ and generates $C^* = \mathbf{Encrypt}_{\{PK_i\}_{i=1}^t}(param, M, \vec{x}_\beta)$.

Phase 2: $\mathcal{A}$ can continue querying the secret keys for any registered users and predicate tokens for additional predicates with the same restriction as that in *Phase 1*.

Guess: $\mathcal{A}$ outputs $\beta' \in \{0, 1\}$ and wins the game if $\beta' = \beta$.

The advantage of $\mathcal{A}$ winning the game is defined as

$$\mathbf{Adv}^{\text{Attribute-Hiding}}(\mathcal{A}) = \left| Pr[\beta' = \beta] - \frac{1}{2} \right|.$$

A multi-receiver predicate encryption scheme is with attribute hiding if there exists no polynomial-time attacker that can win the attribute hiding game with non-negligible advantage.

The security goal of this game is to guarantee that no users can obtain the information about the keywords of a ciphertext even though he has the corresponding predicate tokens.

TABLE I
THE NOTATIONS

Notation	Meaning
$\mathbb{G}$	a cyclic multiplicative group of order $N = pqr$
$\mathbb{G}_p$	a subgroup of $\mathbb{G}$ with prime order p
$\mathbb{G}_q$	a subgroup of $\mathbb{G}$ with prime order q
$\mathbb{G}_r$	a subgroup of $\mathbb{G}$ with prime order r
KGC	the key generation center
M	a message
PK_i	the public key computed by user i
SK_i	the secret key computed by user i

C. The Proposed Multi-Receiver Predicate Encryption Scheme

The proposed scheme is described as follows. The notations used in the proposed scheme are defined in Table I.

1) **Setup**$(1^n) \to (param)$. KGC (OSNs) obtains $(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e})$ with $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r$. Next, it computes g_p, g_q, and g_r as generators of $\mathbb{G}_p$, $\mathbb{G}_q$, and $\mathbb{G}_r$, respectively. KGC (OSNs) performs the following operations:
   a) Choose $\mu \in \mathbb{Z}_N$, $h \in \mathbb{G}_p$, and $R_0 \in \mathbb{G}_r$ at random.
   b) Choose $h_{1,j}, h_{2,j} \in \mathbb{G}_p, R_{1,j}, R_{2,j} \in \mathbb{G}_r$ at random for $j = 1$ to n.
   c) Choose $Q_2 \in \mathbb{G}_q$ randomly.
   d) Compute
$$H_{1,j} = h_{1,j} R_{1,j},\ j = 1, \ldots, n$$
$$H_{2,j} = h_{2,j} R_{2,j},\ j = 1, \ldots, n$$
   e) Compute $Q = g_q R_0$ and $g' = g_p^{\mu} Q_2$.
   f) Set the public system parameters
$$param = (g_p, g_r, Q, g', h, N, \hat{e}, \{H_{1,j}, H_{2,j}\}_{j=1}^n)$$
   and the master secret key
$$msk = (p, q, r, g_q, h^{-\mu}, \{h_{1,j}, h_{2,j}\}_{j=1}^n).$$
2) **Join**$(i) \to (PK_i, SK_i)$. When user i joins the system, he will randomly choose the secret key $SK_i = z_i \in \mathbb{Z}_N$ and set the public key $PK_i = (g')^{z_i}$.
3) **PredicateExtract**$(param, \vec{v}) \to (SK_{\vec{v}})$. User i sends a predicate vector $\vec{v} = (v_1, \ldots, v_n)$, $v_j \in \mathbb{Z}_N$ for $j = 1$ to n, to KGC (OSNs) and KGC produces a predicate token by performing the following steps.
   a) Randomly select $r_{1,j}, r_{2,j} \in \mathbb{Z}_p$ for $j = 1$ to n.
   b) Randomly select $R_3 \in \mathbb{G}_r, Q_3 \in \mathbb{G}_q$, and $f_1, f_2 \in \mathbb{Z}_q$.
   c) Compute
$$K_0 = h^{-\mu} R_3 Q_3 \prod_{j=1}^{n} h_{1,j}^{-r_{1,j}} h_{2,j}^{-r_{2,j}}$$
$$K_{1,j} = g_p^{r_{1,j}} g_q^{f_1 v_j},\ j = 1, \ldots, n$$
$$K_{2,j} = g_p^{r_{2,j}} g_q^{f_2 v_j},\ j = 1, \ldots, n.$$
   d) Send user i the predicate token
$$SK_{\vec{v}} = (K_0, \{K_{1,j}, K_{2,j}\}_{j=1}^n).$$
4) **Encrypt**$_{\{PK_i\}_{i=1}^t}(param, M, \vec{x}) \to (C)$. A sender generates the ciphertext of message M for t selected receivers with a keyword vector $\vec{x} = (x_1, \ldots, x_n)$ by performing the following steps.
   a) Choose a message $M \in \mathbb{G}_T$ and a keyword element $x_j \in \mathbb{Z}_N$ for $j = 1$ to n and get the t receivers' public keys PK_i's, $i = 1, \ldots, t$.
   b) Randomly choose $\alpha, \beta, s \in \mathbb{Z}_N$ and $R_{4,j}, R_{5,j} \in \mathbb{G}_r$ for $j = 1$ to n.
   c) Compute
$$C'_i = M\hat{e}(PK_i, h)^s, i = 1, \ldots, t$$
$$C_0 = g_p^s$$
$$C_{1,j} = H_{1,j}^s Q^{\alpha x_j} R_{4,j},\ j = 1, \ldots, n$$
$$C_{2,j} = H_{2,j}^s Q^{\beta x_j} R_{5,j},\ j = 1, \ldots, n.$$
   d) Set $C = (\{C'_i\}_{i=1}^t, C_0, \{C_{1,j}, C_{2,j}\}_{j=1}^n)$ to be the ciphertext.
5) **MasterSearch** $(param, C, SK_{\vec{v}}) \to (1/\perp)$. If KGC (OSNs) would like to determine whether a predicate vector $\vec{v}$ matches a ciphertext $C = (\{C'_i\}_{i=1}^t, C_0, \{C_{1,j}, C_{2,j}\}_{j=1}^n)$ or not, it can take $SK_{\vec{v}} = (K_0, \{K_{1,j}, K_{2,j}\}_{j=1}^n)$ and perform as follows.
   a) Compute
$$D = \hat{e}(C_0, K_0) \prod_{j=1}^{n} \hat{e}(C_{1,j}, K_{1,j}) \hat{e}(C_{2,j}, K_{2,j}).$$
   b) If $D^p = 1_{\mathbb{G}_T}$, output 1. Otherwise, output the distinguished symbol $\perp$.

   The correctness of **MasterSearch** is shown in **Remark 1** in Appendix.
6) **Decrypt**$(param, C, SK_{\vec{v}}, SK_i) \to (M)$. After finding a matched ciphertext $C = (\{C'_i\}_{i=1}^t, C_0, \{C_{1,j}, C_{2,j}\}_{j=1}^n)$ by a predicate token $SK_{\vec{v}} = (K_0, \{K_{1,j}, K_{2,j}\}_{j=1}^n)$, a selected receiver, say i, can apply his secret key $SK_i = z_i$ to decrypt C by computing
$$M = C'_i(\hat{e}(C_0, K_0) \prod_{j=1}^{n} \hat{e}(C_{1,j}, K_{1,j}) \hat{e}(C_{2,j}, K_{2,j}))^{SK_i}.$$
   The correctness of **Decrypt** is shown in **Remark 2** in Appendix.

D. Construction of the OSN Platform

The construction of the OSN platform is presented in this section. The roles mentioned in Section II-A perform the algorithms in the proposed OSN platform. There are six algorithms in the proposed scheme for the OSN platform: **Setup, Registering, Sharing Data, Adding/Removing Friends, Advertising** and **Downloading Data**. At first, the OSN provider runs $\mathtt{Setup}$ where it generates its master secret key and the public parameters for the platform. A user runs the $\mathtt{Registering}$ algorithm to join the OSN platform. When a user joins this platform and chooses his own key pair, the OSN provider can produce predicate tokens for the user to find the matched data

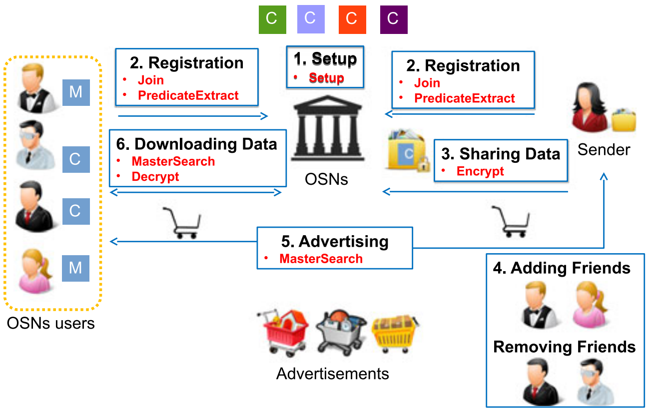

Fig. 4. The scenario of the OSN platform.

efficiently. By using the $\mathbf{SharingData}$ algorithm, a sender encrypts his data and sends them to the receivers when he wants to share them with the receivers in the OSN platform. If a sender would like to add or remove friends, he can perform the $\mathbf{Adding/RemovingFriends}$ algorithm. The OSN provider executes the $\mathbf{Advertising}$ algorithm to verify if some specified commercial keywords exist in the encrypted data of users, and the advertisers can issue customized advertisement to those users whose encrypted data contain the keywords. Finally, a user performs the $\mathbf{DownloadingData}$ algorithm to find interested data and decrypt them efficiently. The flow of the proposed construction is also illustrated in Fig. 4.

1) **Setup**
   a) The OSN provider executes **Setup**(1^n) to generate the system parameters and the master secret key. The OSN provider publishes $param$ and keeps msk secret.

2) **Registering in the OSN platform**
   a) When a user i joins the system, he performs **Join**(i) to generate his own key pair (PK_i, SK_i). Then, user i sends the index i to the OSN provider for registration and keeps SK_i as secret.
   b) User i can choose and send predicate vectors to the OSN provider to request predicate tokens. After receiving a predicate vector $\vec{v}$, the OSN provider calls

$$\mathbf{PredicateExtract}(param, \vec{v})$$

   to compute a predicate token $SK_{\vec{v}}$ for user i which is associated with $\vec{v}$ and can provide user i for an undecryptable search.

3) **Sharing Data**
   a) Let $f_w = \{i \mid \text{user } i \text{ is a friend of user } w\}$ be the index set of the friends of user w in the OSN platform. If user w would like to share with his friends in the data M assiciated with a keyword vector $\vec{x}$, he can perform

$$\mathbf{Encrypt}_{\{PK_i\}_{i\in f_w}}(param, M, \vec{x})$$

   to get the ciphertext

$$C = (\{C_i'\}_{i\in f_w}, C_0, \{C_{1,j}, C_{2,j}\}_{j=1}^n).$$

   Then, he sends the ciphertext C to his friends via the OSN platform.

4) **Adding Friends/ Removing Friends**
   a) If user w would like to add a new friend, say user i, then he should update his friend set $f_w = f_w + \{i\}$ in the OSN platform.
   b) User w can update his friend set $f_w = f_w - \{j\}$ in the OSN platform if he wants to remove user j from his friend list.

5) **Advertising**
   a) When the OSN provider would like to check if a ciphertext C matches a predicate $\vec{v}$, it can compute

$$SK_{\vec{v}} = \mathbf{PredicateExtract}(param, \vec{v})$$

   and execute

$$\mathbf{MasterSearch}(param, C, SK_{\vec{v}}).$$

   The OSN provider can just search matched ciphertexts but it cannot decrypt them. If the output is 1, the advertiser can send the advertisement corresponding to $\vec{v}$ to those users who can decrypt C. Otherwise (i.e., the output is $\perp$), C does not match $\vec{v}$.

6) **Downloading Data**
   a) If user i would like to find the ciphertexts matching $\vec{v}$ in his received ciphertexts, C_{i_j}'s, he can get

$$SK_{\vec{v}} = \mathbf{PredicateExtract}(param, \vec{v})$$

   and run

$$\mathbf{MasterSearch}(param, C_{i_j}, SK_{\vec{v}})$$

   for each C_{i_j}. Then, he downloads

$$C_{i_{\vec{v}}} = \{C_{i_j} \mid \mathbf{MasterSearch}(param, C_{i_j}, SK_{\vec{v}}) = 1\}$$

   and executes

$$\mathbf{Decrypt}(param, C_{i_j}, SK_{\vec{v}}, SK_i)$$

   for each C_{i_j} in $C_{i_{\vec{v}}}$ by using his secret key SK_i. On the other hand, the unselected receivers of a ciphertext are unable to decrypt the ciphertext uploaded by the sender.

IV. Security Proofs

A. *Malicious KGC (OSNs)*

In this section, we will analyse the security against malicious KGC (OSNs) in the multi-receiver predicate encryption.

We give a conceptual description of the proof of the semantic security against KGC as follows.

Proof: Given the instance of Assumption 3, $((N, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_p, g_q, g_r, g_p^a, g_p^b, g_p^c, g_q^a, T)$, we use an attacker $\mathcal{A}$, who is able to access the master secret key, to construct an algorithm for breaking Assumption 3.

Setup: $\mathcal{S}$ randomly chooses $\alpha \in \mathbb{Z}_N$ and computes $Q_2 = (g_q^a)^\alpha, h = g_p^b$. The remaining parts are the same as those in the scheme. Then $\mathcal{S}$ randomly chooses $z_i \in \mathbb{Z}_N$ and computes $PK_i = ((g_p^a)^\mu (g_q^a)^\alpha)^{z_i}$ for $i = 1, \ldots, t$. Finally $\mathcal{S}$ outputs $param, msk, \{PK_i\}_{i=1,\ldots,t}$ to $\mathcal{A}$.

Phase 1: The simulation of this phase is trivial since $\mathcal{S}$ knows msk.

Challenge: Upon receiving (M_0, M_1) from $\mathcal{A}$, $\mathcal{S}$ randomly choose $\beta \in \{0,1\}$. Then $\mathcal{S}$ computes $C_0 = g_p^c$ and $C_i' = M_\beta T^{\mu z_i}, i = 1, \ldots, t$. The remaining part of the challenge ciphertext can be computed as the same way as that in the scheme.

Phase 2: This phase is the same as that of Phase 1.

Guess: $\mathcal{A}$ outputs a bit β'. If $T = \hat{e}(g_p, g_p)^{abc}$, $T^{\mu z_i} = \hat{e}((g_p^a)^{\mu z_i}, g_p^b)^c = \hat{e}((g_p^a)^{\mu z_i} Q_2^{z_i}, g_p^b)^c = \hat{e}(PK_i, h)^c$. Thus, $\mathcal{S}$ simulates the game perfectly. ■

As the proof shown above, if there exists such an attacker who can make a correct guess, we can break Assumption 3 within polynomial time. Hence, the proposed scheme achieves the semantic security against the malicious KGC. That is, even the server of OSN, who is able to access the master secret key, cannot get the plaintext of a ciphertext.

B. Security Against Malicious Users

In this section, we present formal proofs of a hybrid game for semantic security and attribute hiding against malicious users in the selective models of multi-receiver predicate encryption.

Theorem 4.1: (Semantic Security) The proposed scheme is semantically secure if the BSD assumption holds.

Proof: We construct $\mathcal{S}$ that tries to break the BSD assumption. $\mathcal{S}$ uses a sequence of hybrid games, $Game_0$ and $Game_1$, such that $\mathcal{A}$ cannot distinguish Game_0 from Game_1. If $\mathcal{A}$ has advantage ϵ in distinguishing Game_0 from Game_1, then $\mathcal{S}$ has the same advantage ϵ in breaking the BSD assumption. The games are defined as follows.

$Game_0$: The challenge ciphertext is used to construct the original security game. That is, we choose $s, \alpha, \beta \in \mathbb{Z}_N$ and $R_{4,j}, R_{5,j} \in \mathbb{G}_r$ at random, and compute the ciphertext as

$$C = (\{C_i'\}_{i=1}^t = M\{\hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s,$$
$$\{C_{1,j} = H_{1,j}^s Q^{\alpha x_j} R_{4,j}, C_{2,j} = H_{2,j}^s Q^{\beta x_j} R_{5,j}\}_{j=1}^n).$$

$Game_1$: The challenge ciphertext is generated as a proper encryption on a random element of $\mathbb{G}_T$, i.e., the ciphertext is formed as above except that (C_i')'s are chosen uniformly from $\mathbb{G}_T$.

$$C = (\{C_i'\}_{i=1}^t, C_0 = g_p^s, \{C_{1,j} = H_{1,j}^s Q^{\alpha x_j} R_{4,j},$$
$$C_{2,j} = H_{2,j}^s Q^{\beta x_j} R_{5,j}\}_{j=1}^n).$$

First, $\mathcal{S}$ is given $(N = pqr, \mathbb{G}, \mathbb{G}_T, \hat{e})$ along with the elements $g_p,\ g_q,\ g_r,\ h,\ g_p^s,\ h^s Q_1,\ g_p^\mu Q_2, \hat{e}(g_p, h)^\mu$, and an element T which is either equal to $\hat{e}(g_p, h)^{\mu s}$ or is uniformly distributed in $\mathbb{G}_T$. $\mathcal{S}$ interacts with $\mathcal{A}$ as we now describe.

Initialization: $\mathcal{A}$ outputs a predicate vector $\vec{x}$ which it wishes to attack.

Setup: $\mathcal{S}$ begins by giving N to $\mathcal{A}$. $\mathcal{S}$ chooses $\delta_{1,j}, \delta_{2,j} \in \mathbb{Z}_N$ and $R_{1,j}, R_{2,j}, R_0 \in \mathbb{G}_r$ at random. It includes $(N, \mathbb{G}, \mathbb{G}_T, \hat{e})$ in the public parameters, and returns the remainder of the parameters to $\mathcal{A}$ as follows:

$$param = (g_p, g_r, Q = g_q R_0, g' = g_p^\mu Q_2, h, N,$$
$$\{H_{1,j} = h^{x_j} g_p^{\delta_{1,j}} R_{1,j}, H_{2,j} = h^{x_j} g_p^{\delta_{2,j}} R_{2,j}\}_{j=1}^n)$$

where $\mathcal{S}$ is implicitly setting $h_{1,j} = h^{x_j} g_p^{\delta_{1,j}}$ and $h_{2,j} = h^{x_j} g_p^{\delta_{2,j}}$. Note that $param$ has the appropriate distribution.

Phase 1: In this phase, $\mathcal{A}$ can issue the queries for secret keys and predicate tokens corresponding to different vectors $\vec{v}$ as long as $\langle \vec{v}, \vec{x} \rangle \neq 0$. We now describe how $\mathcal{S}$ prepares the secret key of user i and predicate token corresponding to any such vector as follows.

1) $\mathcal{A}$ submits an identity i to $\mathcal{S}$ and $\mathcal{S}$ returns the public key $PK_i = (g')^{z_i}$ and secret key $SK_i = z_i$ to $\mathcal{A}$. Then, $\mathcal{S}$ keeps (PK_i, z_i) in PK-list.
2) $\mathcal{A}$ requests the predicate token for vector $\vec{v}$. Let $k = \frac{1}{2 \cdot \langle \vec{x}, \vec{v} \rangle}$ mod N. $\mathcal{S}$ first chooses random $f_1', f_2', r_{1,j}', r_{2,j}' \in \mathbb{Z}_N$. Next, for all j's it computes:

$$K_{1,j} = (g_p^\mu Q_2)^{-k v_j} g_q^{f_1' v_j} g_p^{r_{1,j}'} = g_p^{-k v_j \mu + r_{1,j}'} g_q^{(f_1' - kd) \cdot v_j}$$
$$K_{2,j} = (g_p^\mu Q_2)^{-k v_j} g_q^{f_2' v_j} g_p^{r_{2,j}'} = g_p^{-k v_j \mu + r_{2,j}'} g_q^{(f_2' - kd) \cdot v_j}$$

where we set $d = log_{g_q} Q_2$. The simulator then chooses random $R \in \mathbb{G}_r$ and computes:

$$K_0 = QR \cdot \prod_j ((g_p^{\delta_{1,j}} h^{x_j})^{-r_{1,j}'} \cdot (g_p^\mu Q_2)^{k v_j \delta_{1,j}})$$
$$\cdot ((g_p^{\delta_{2,j}} h^{x_j})^{-r_{2,j}'} \cdot (g_p^\mu Q_2)^{k v_j \delta_{2,j}}).$$

3) $\mathcal{S}$ returns the predicate token $SK_{\vec{v}} = (K_0, \{K_{1,j}, K_{2,j}\}_{j=1}^n)$. We give the correctness of K_p (the projection of K_0 in $\mathbb{G}_p$) distribution in **Remark 4**.

Challenge: $\mathcal{A}$ sends (M_0, M_1) and $\{PK_i\}_{i=1}^t$ to $\mathcal{S}$ where M_0, M_1 are two distinct messages with the same length and t is a positive integer. $\mathcal{S}$ does the following:

1) Choose random $R_{6,j}, R_{7,j} \in \mathbb{G}_r$ and $Q_3 \in \mathbb{G}_q$ for $j = 1$ to n.
2) Choose $\beta \in \{0,1\}$ and find the corresponding z_i in PK-list for $i = 1$ to t.
3) For $i = 1$ to t, set $C_i' = M_\beta \cdot T^{z_i}$, and $C_0 = g_p^s$.
4) For $j = 1$ to n, compute

$$C_{1,j} = (g_p^s)^{\delta_{1,j}} \cdot (h^s Q_1)^{x_j} \cdot R_{6,j}$$
$$= (h^{x_j} g_p^{\delta_{1,j}})^s \cdot Q_1^{x_j} \cdot R_{6,j}$$
$$C_{2,j} = (g_p^s)^{\delta_{2,j}} \cdot (h^s Q_1)^{x_j} \cdot (Q_3)^{x_i} \cdot R_{7,j}$$
$$= (h^{x_j} g_p^{\delta_{2,j}})^s \cdot (Q_1 Q_3)^{x_j} \cdot R_{7,j}.$$

5) $\mathcal{S}$ returns the ciphertext $C^* = (\{C_i' = M_\beta T^{z_i}\}_{i=1}^t, C_0 = g_p^s, \{C_{1,j} = (h^{x_j} g_p^{\delta_{1,j}})^s \cdot Q_1^{x_j} \cdot R_{6,j}, C_{2,j} = (h^{x_j} g_p^{\delta_{2,j}})^s \cdot (Q_1 Q_3)^{x_j} \cdot R_{7,j}\}_{j=1}^n)$.

Phase 2: $\mathcal{A}$ makes queries as those in *Phase 1*.

Guess: Finally, $\mathcal{A}$ outputs $\beta' \in \{0,1\}$. If $\beta' = \beta$, then $\mathcal{S}$ outputs 1. Otherwise, $\mathcal{S}$ outputs 0.

If $T = \hat{e}(g_p, h)^{\mu s}$, the challenge ciphertext is distributed exactly as that in $Game_0$, whereas if T is chosen uniformly from $\mathbb{G}_T$, the challenge ciphertext is distributed exactly as that in

$Game_1$. It follows that under the BSD assumption, these two games are indistinguishable. Therefore, C^* is a correct ciphertext. As the construction above, $\mathcal{S}$ correctly simulates the semantic security game. If $\mathcal{A}$ wins the semantic security game with non-negligible advantage ϵ, $|Pr[\mathcal{S}(\bar{Z}, T = e(g_p, h)^{\mu s}) = 1] - Pr[\mathcal{S}(\bar{Z}, T \in \mathbb{G}_T) = 1]| \geq \epsilon$ under a correct simulation of the game, i.e., $|Pr[\mathcal{A}(\Omega) = \beta' = \beta \mid Game_0] - Pr[\mathcal{A}(\Omega) = \beta' = \beta \mid Game_1]| \geq \epsilon$, where Ω is a correct multi-receiver predicate encryption scheme.

Therefore, $\mathcal{S}$ solves the BSD problem with non-negligible advantage ϵ within polynomial time. ■

Theorem 4.1 guarantees the semantic security against malicious users. Therefore, no one can obtain any information about the encrypted content of a ciphertext unless he owns a corresponding predicate token if the proposed MRPE scheme is applied to an OSN.

Theorem 4.2: (Attribute Hiding) The proposed scheme is with attribute hiding if the SD assumption holds.

Proof: We construct $\mathcal{S}$ that tries to break the SD assumption. $\mathcal{S}$ uses a sequence of hybrid games, Game$_i$'s, such that $\mathcal{A}$ cannot distinguish Game$_i$ from Game$_{i+1}$, where the challenge ciphertext will be encrypted with a vector in the first sub-system and a different vector in the second sub-system. Let $(\vec{x}_0, \vec{x}_1)$ denote a ciphertext encrypted using vector $\vec{x}_0$ in the first sub-system $\{C_{1,j}\}_{j=1}^n$ and $\vec{x}_1$ in the second sub-system $\{C_{2,j}\}_{j=1}^n$. If $\mathcal{A}$ has advantage ϵ in distinguishing Game$_0$ from Game$_4$, then $\mathcal{S}$ has the same advantage ϵ in breaking the SD assumption. We use a series of games demonstrating that

$$(\vec{x}_0, \vec{x}_0) \approx (\vec{x}_0, \vec{0}) \approx (\vec{x}_0, \vec{x}_1) \approx (\vec{0}, \vec{x}_1) \approx (\vec{x}_1, \vec{x}_1)$$

and the games are defined as follows.

$Game_0$: The challenge ciphertext is used to construct the original security game as a proper encryption of M using $\vec{x}_0$. That is, we choose $s, \alpha, \beta \in \mathbb{Z}_N, R_{4,j}, R_{5,j} \in \mathbb{G}_r$, and $C' \in \mathbb{G}_T$ at random, and compute the ciphertext as

$$C = (\{C'_i\}_{i=1}^t = \{M\hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s,$$
$$\{C_{1,j} = H_{1,j}^s Q^{\alpha x_{0,j}} R_{4,j}, C_{2,j} = H_{2,j}^s Q^{\beta x_{0,j}} R_{5,j}\}_{j=1}^n).$$

$Game_1$: We now generate $\{C_{2,j}\}$ as if the encryption is done using $\vec{0}$. That is, we choose $s, \alpha, \beta \in \mathbb{Z}_N, R_{4,j}, R_{5,j} \in \mathbb{G}_r$, and $C' \in \mathbb{G}_T$ at random, and compute the ciphertext as

$$C = (\{C'_i\}_{i=1}^t = \{M\hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s,$$
$$\{C_{1,j} = H_{1,j}^s Q^{\alpha x_{0,j}} R_{4,j}, C_{2,j} = H_{2,j}^s R_{5,j}\}_{j=1}^n).$$

$Game_2$: We generate $\{C_{2,j}\}$ using vector $\vec{x}_1$. That is, we choose $s, \alpha, \beta \in \mathbb{Z}_N, R_{4,j}, R_{5,j} \in \mathbb{G}_r$, and $C' \in \mathbb{G}_T$ at random, and compute the ciphertext as

$$C = (\{C'_i\}_{i=1}^t = \{M\hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s,$$
$$\{C_{1,j} = H_{1,j}^s Q^{\alpha x_{0,j}} R_{4,j}, C_{2,j} = H_{2,j}^s Q^{\beta x_{1,j}} R_{5,j}\}_{j=1}^n).$$

$Game_3$: We generate $\{C_{1,j}\}$ as if the encryption is done using $\vec{0}$. That is, we choose $s, \alpha, \beta \in \mathbb{Z}_N, R_{4,j}, R_{5,j} \in \mathbb{G}_r$, and $C' \in \mathbb{G}_T$ at random, and compute the ciphertext as

$$C = (\{C'_i\}_{i=1}^t = \{M\hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s,$$
$$\{C_{1,j} = H_{1,j}^s R_{4,j}, C_{2,j} = H_{2,j}^s Q^{\beta x_{1,j}} R_{5,j}\}_{j=1}^n).$$

$Game_4$: We generate $\{C_{1,j}\}$ using vector $\vec{x}_1$. That is, we choose $s, \alpha, \beta \in \mathbb{Z}_N, R_{4,j}, R_{5,j} \in \mathbb{G}_r$, and $C' \in \mathbb{G}_T$ at random, and compute the ciphertext as

$$C = (\{C'_i\}_{i=1}^t = \{M\hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s, \{C_1$$
$$= H_{1,j}^s Q^{\alpha x_{1,j}} R_{4,j}, C_{2,j} = H_{2,j}^s Q^{\beta x_{1,j}} R_{5,j}\}_{j=1}^n).$$

First, $\mathcal{S}$ is given $(N = pqr, \mathbb{G}, \mathbb{G}_T, \hat{e})$ along with the elements $g_p, g_r, g_q R_0, h_p = g_p^b, k_p = g_p^{b^2}, g_p^a g_q, g_p^{ab} Q_1, g_p^s, g_p^{bs} Q_2 R_2$, and an element $T = g_p^{b^2 s} g_q^\beta R_3$, where β is either 0 or uniformly distributed in $\mathbb{Z}_p$. The simulator interacts with $\mathcal{A}$ as we now describe.

1) Indistinguishability Between Game$_0$ and Game$_1$: Initialization: $\mathcal{A}$ outputs two predicate vectors $\vec{x}_0$ and $\vec{x}_1$ that it wishes to attack.

Setup: $\mathcal{S}$ begins by giving N to $\mathcal{A}$, who outputs vectors $\vec{x}_0, \vec{x}_1$. $\mathcal{S}$ chooses $\mu, \delta_{1,j}, \delta_{2,j} \in \mathbb{Z}_N, R_{1,j}, R_{2,j}, R_0 \in \mathbb{G}_r$, and $h \in \mathbb{G}_p$ at random, and it includes $(N, \mathbb{G}, \mathbb{G}_T, \hat{e})$ in the public parameters and computes $h^{-\mu}$. It then returns the remainder of the parameters as follows:

$$param = (g_p, g_r, Q = g_q R_0, g' = g_p^\mu Q'_2, h, N,$$
$$\{H_{1,j} = h_p^{x_{0,j}} g_p^{\delta_{1,j}} R_{1,j}, H_{2,j} = k_p^{x_{0,j}} g_p^{\delta_{2,j}} R_{2,j}\}_{j=1}^n).$$

The simulator is implicitly setting $h_{1,j} = h_p^{x_{0,j}} g_p^{\delta_{1,j}}$ and $h_{2,j} = k_p^{x_{0,j}} g_p^{\delta_{2,j}}$. Note that $param$ has the appropriate distribution.

Phase 1: In this phase, $\mathcal{A}$ can issue the queries for secret keys and predicate tokens corresponding to different vectors $\vec{v}$'s. Here, we distinguish two cases, depending on whether $\langle \vec{v}, \vec{x}_0 \rangle$ and $\langle \vec{v}, \vec{0} \rangle$ are both zero or whether they are both non-zero. Since the vector $\vec{0}$ is orthogonal to everything, we only disscuss the fact that $\langle \vec{v}, \vec{x}_0 \rangle \neq 0$ here. We now describe how $\mathcal{S}$ prepares the secret keys and predicate tokens corresponding to any such vectors.

1) $\mathcal{A}$ sends an identity i to $\mathcal{S}$ and $\mathcal{S}$ returns the public key $PK_i = (g')^{z_i}$ and secret key $SK_i = z_i$ to $\mathcal{A}$.
2) $\mathcal{A}$ requests the predicate token for vector $\vec{v}$. Let $k = \langle \vec{x}_0, \vec{v} \rangle$. $\mathcal{S}$ first chooses random $f'_1, f'_2, r'_{1,j}, r'_{2,j} \in \mathbb{Z}_N$. Next, for $j = 1$ to n, it computes:

$$\begin{aligned} K_{1,j} &= (g_p^a g_q)^{f'_1 v_j} \cdot (g_p^{ab} Q_1)^{-f'_2 v_j} \cdot g_p^{r'_{1,j}} \\ &= g_p^{(af'_1 - abf'_2) \cdot v_j + r'_{1,j}} \cdot g_q^{(f'_1 - df'_2) \cdot v_j} \\ K_{2,j} &= (g_p^a g_q)^{f'_2 v_j} \cdot g_p^{r'_{2,j}} \\ &= g_p^{af'_2 v_j + r'_{2,i}} \cdot g_q^{f'_2 v_j} \end{aligned}$$

where we set $d = log_{g_q} Q_1$. $\mathcal{S}$ then chooses random $R \in \mathbb{G}_r$ and computes:

$$K_0 = h^{-\mu} QR \cdot (g_p^{ab} Q_1)^{-k\cdot f_1'} \cdot \prod_{j=1}^{n} (g_p^a g_q)^{-f_1' v_j \delta_{1,j} - f_2' v_j \delta_{2,j}} \cdot (g_p^{ab} Q_1)^{f_2' v_j \delta_{1,j}} \cdot g_p^{-\delta_{1,j}\cdot r_{1,j}' - \delta_{2,j}\cdot r_{2,j}'} \cdot h_p^{-x_{0,j}\cdot r_{1,j}'} \cdot k_p^{-x_{0,j}\cdot r_{2,j}'}.$$

3) $\mathcal{S}$ returns the predicate token $SK_{\vec{v}} = (K_0, \{K_{1,j}, K_{2,j}\}_{j=1}^n)$ to $\mathcal{A}$. We provide the correctness of K_p distribution in **Remark 5**.

Challenge: $\mathcal{A}$ sends M and $\{PK_i\}_{i=1}^t$ to $\mathcal{S}$, where t is a positive integer. $\mathcal{S}$ does the following:

1) Set $C_i' = M \cdot \hat{e}(PK_i, h)^s$, where $i = 1$ to t, and $C_0 = g_p^s$.
2) For $j = 1$ to n, choose $R_{6,j}, R_{7,j} \in \mathbb{G}_r$ randomly, and compute

$$C_{1,j} = (g_p^s)^{\delta_{1,j}} \cdot (g_p^{bs} Q_2 R_2)^{x_{0,j}} \cdot R_{6,j} = (h_{1,j})^s Q_2^{x_{0,j}} R_{6,j}'$$

$$C_{2,j} = (g_p^s)^{\delta_{2,j}} \cdot T^{x_{0,j}} \cdot R_{7,j} = (h_{2,j})^s \cdot (g_q^\gamma)^{x_{0,j}} \cdot R_{7,j}.$$

3) $\mathcal{S}$ returns the ciphertext $C^* = (\{C_i'\}_{i=1}^t = \{M \cdot \hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s, \{C_{1,j} = (h_{1,j})^s Q_2^{x_{0,j}} R_{6,j}', C_{2,j} = (h_{2,j})^s \cdot (g_q^\gamma)^{x_{0,j}} \cdot R_{7,j}\}_{j=1}^n)$.

Phase 2: $\mathcal{A}$ makes queries as those in *Phase 1*.

Guess: Finally, $\mathcal{A}$ outputs $\beta' \in \{0,1\}$. If $\beta' = \beta$, then $\mathcal{S}$ outputs 1. Otherwise, $\mathcal{S}$ outputs 0.

If $T = g_p^{b^2 s} g_q^\gamma R_3$ and $\gamma = 0$, the challenge ciphertext is distributed exactly as that in $Game_1$, whereas if γ is chosen uniformly from $\mathbb{Z}_N$, the challenge ciphertext is distributed exactly as that in $Game_0$. It follows that under the SD assumption, these two games are indistinguishable. Therefore, C^* is a correct ciphertext. As the construction above, $\mathcal{S}$ correctly simulates the attribute hiding game. If $\mathcal{A}$ wins the attribute hiding game with non-negligible advantage ϵ, $|Pr[\mathcal{S}(\bar{Z}, T = g_p^{b^2 s} R_3)) = 1] - Pr[\mathcal{S}(\bar{Z}, T \in \mathbb{G}) = 1]| \geq \epsilon$ under a correct simulation of the game, i.e., $|Pr[\mathcal{A}(\Omega) = \beta' = \beta \mid \ Game_0] - Pr[\mathcal{A}(\Omega) = \beta' = \beta \mid \ Game_1]| \geq \epsilon$, where Ω is a correct multi-receiver predicate encryption scheme.

Therefore, $\mathcal{S}$ solves the SD problem with non-negligible advantage ϵ within polynomial time.

2) Indistinguishability Between $Game_1$ and $Game_2$: Initialization: $\mathcal{A}$ outputs two predicate vectors $\vec{x}_0$ and $\vec{x}_1$ that it wishes to attack.

Setup: $\mathcal{S}$ begins by giving N to $\mathcal{A}$. $\mathcal{S}$ chooses $\mu, \delta_{1,j}, \delta_{2,j} \in \mathbb{Z}_N$, $R_{1,j}, R_{2,j}, R_0 \in \mathbb{G}_r$, and $h \in \mathbb{G}_p$ at random, and it includes $(N, \mathbb{G}, \mathbb{G}_T, \hat{e})$ in the public parameters and computes $h^{-\mu}$. Then, it returns the remainder of the parameters as follows:

$$params = (g_p, g_r, Q = g_q R_0, g' = g_p^\mu Q_2', h, N, \{H_{1,j} = h^{x_{0,j}} g_p^{\delta_{1,j}} R_{1,j}, H_{2,j} = h^{x_{1,j}} g_p^{\delta_{2,j}} R_{2,j}\}_{j=1}^n).$$

The simulator is implicitly setting $h_{1,j} = h_p^{x_{0,j}} g_p^{\delta_{1,j}}$ and $h_{2,j} = k_p^{x_{1,j}} g_p^{\delta_{2,j}}$. Note that $params$ has the appropriate distribution.

Phase 1: In this phase, $\mathcal{A}$ can issue the queries for secret keys and predicate tokens corresponding to different vectors $\vec{v}$'s. The simulation for secret key queries are the same as that in Section IV-B1. Here, we distinguish two cases, depending on whether $\langle \vec{v}, \vec{x}_0 \rangle$ and $\langle \vec{v}, \vec{x}_1 \rangle$ are both zero or whether they are both non-zero. We now describe how the simulator prepares the secret keys corresponding to any such vectors.

Case 1.

1) $\mathcal{A}$ requests the predicate token for vector $\vec{v}$ where $\langle \vec{x}_0, \vec{v} \rangle = 0 = \langle \vec{x}_1, \vec{v} \rangle$. $\mathcal{S}$ first chooses random $f_1', f_2', r_1', r_{2,j}' \in \mathbb{Z}_N$. Next, for $j = 1$ to n, it computes:

$$K_{1,j} = (g_p^a g_q)^{f_1 v_j} \cdot g_p^{r_{1,j}'} = g_p^{a f_1 v_j + r_{1,j}'} \cdot g_q^{f_1 v_j}$$
$$K_{2,j} = (g_p^a g_q)^{f_2 v_j} \cdot g_p^{r_{2,j}'} = g_p^{a f_2 v_j + r_{2,j}'} \cdot g_q^{f_2 v_j}.$$

$\mathcal{S}$ then chooses random $R \in \mathbb{G}_r$ and computes:

$$K_0 = h^{-\mu} QR \cdot \prod_{j=1}^{n} (g_p^a g_q)^{-f_1 v_j \delta_{1,j} - f_2 v_j \delta_{2,j}} \cdot g_p^{-\delta_{1,j}\cdot r_{1,j}' - \delta_{2,j}\cdot r_{2,j}'} \cdot h_p^{-x_{0,j}\cdot r_{1,j}'} \cdot k_p^{-x_{1,j}\cdot r_{2,j}'}.$$

2) $\mathcal{S}$ returns the predicate token $SK_{\vec{v}} = (K_0, \{K_{1,j}, K_{2,j}\}_{j=1}^n)$ to $\mathcal{A}$. We show the correctness of K_p distribution in **Remark 6**.

Case 2.

3) $\mathcal{A}$ requests the predicate token for vector $\vec{v}$ where $\langle \vec{v}, \vec{x}_0 \rangle = c_{x_0} \neq 0$ and $\langle \vec{v}, \vec{x}_1 \rangle = c_{x_1} \neq 0$. $\mathcal{S}$ first chooses random $f_1', f_2', r_{1,j}', r_{2,j}' \in \mathbb{Z}_N$. Next, for $j = 1$ to n, it computes:

$$K_{1,j} = (g_p^a g_q)^{f_1' v_j} \cdot (g_p^{ab} Q_1)^{-c_{x_1}\cdot f_2' v_j} \cdot g_p^{r_{1,j}'} = g_p^{(a f_1' - abc_{x_1} f_2')\cdot v_j + r_{1,j}'} \cdot g_q^{(f_1' - c_{x_1} d f_2')\cdot v_j}$$
$$K_{2,j} = (g_p^a g_q)^{c_{x_0}\cdot f_2' v_j} \cdot g_p^{r_{2,j}'} = g_p^{a c_{x_0} f_2' v_j + r_{2,i}'} \cdot g_q^{c_{x_0}\cdot f_2' v_j}.$$

where we set $d = log_{g_q} Q_1$. $\mathcal{S}$ then chooses random $R \in \mathbb{G}_r$ and computes:

$$K_0 = h^{-\mu} QR \cdot (g_p^{ab} Q_1)^{-c_{x_0}\cdot f_1'} \cdot \prod_{j=1}^{n} (g_p^a g_q)^{-f_1' v_j \delta_{1,j} - f_2' c_{x_0} v_j \delta_{2,j}} (g_p^{ab} Q_1)^{f_2' c_{x_1} v_j \delta_{1,j}} \cdot g_p^{-\delta_{1,j}\cdot r_{1,j}' - \delta_{2,j}\cdot r_{2,j}'} \cdot h_p^{-x_{0,j}\cdot r_{1,j}'} \cdot k_p^{-x_{1,j}\cdot r_{2,j}'}.$$

4) $\mathcal{S}$ returns the predicate token $SK_{\vec{v}} = (K_0, \{K_{1,j}, K_{2,j}\}_{j=1}^n)$ to $\mathcal{A}$. The correctness of K_p distribution is shown in **Remark 7**.

Challenge: $\mathcal{A}$ sends M and $\{PK_i\}_{i=1}^t$ to $\mathcal{S}$, where t is a positive integer. $\mathcal{S}$ does the following:

1) Set $C_i' = M \cdot \hat{e}(PK_i, h)^s$, for $i = 1$ to t, and $C_0 = g_p^s$.
2) For $j = 1$ to n, choose $R_{6,j}, R_{7,j} \in \mathbb{G}_r$ at random, and compute

$$C_{1,j} = (g_p^s)^{\delta_{1,j}} \cdot (g_p^{bs} Q_2 R_2)^{x_{0,j}} \cdot R_{6,j} = (h_{1,j})^s Q_2^{x_{0,j}} R_{6,j}'$$

$$C_{2,j} = (g_p^s)^{\delta_{2,j}} \cdot T^{x_{1,j}} \cdot R_{7,j} = (h_{2,j})^s \cdot (g_q^\gamma)^{x_{1,j}} \cdot R_{7,j}.$$

TABLE II
PROPERTY AND PERFORMANCE COMPARISONS IN THE OSN PLATFORM

Control	Access Group	Dynamic Searching	Data Resistance	Collusion Keywords	Unified Length	Ciphertext Cost	Encryption Cost	Decryption
[9]	Yes	Yes	Yes	Yes	No	$t((2n+1)\|\mathbb{G}\| + \|\mathbb{G}_T\|) = (2nt+2t)*256$ bits	$t[(4n+2)T_s + (4n+1)T_a + (2n)T_m] \approx t(7820n+3836)$ CCs	$(2n+1)T_p + (2n+1)T_a \approx (158416n + 79208)$ CCs
[11]	Yes	Yes	Yes	Yes	No	$t(6n)\|\mathbb{G}\| = 6nt*256$ bits	$t[(6n)((2n+1)T_s + (2n-1)T_a)] \approx t(23064n^2 + 11437n)$ CCs	$(6n)T_p \approx 475200n$ CCs
[13]	Yes	Yes	Yes	Yes	No	$t((2n+3)\|\mathbb{G}\| + \|\mathbb{G}_T\|) = (2nt+4t)*256$ bits	$t[(2n+3)((n+3)T_s + (n+1)T_a) + (T_s+T_a)] \approx t(3844n^2 + 17266n + 19172)$ CCs	$(2n+3)T_p \approx (158400n + 237600)$ CCs
[21]	Yes	Yes	Yes	Yes	No	$t((4n+2)\|\mathbb{G}\| + \|\mathbb{G}_T\|) = (4nt+3t)*256$ bits	$t[(4n+2)((n+3)T_s + (n+1)T_a) + (T_s+T_a)] \approx t(7688n^2 + 26844n + 13422)$ CCs	$(4n+2)T_p \approx (316800n + 158400)$ CCs
[17]	Yes	Yes	No	No	No	$\triangle$	$\triangle$	$\triangle$
Ours	Yes	Yes	Yes	Yes	Yes	$(2n+1)\|\mathbb{G}\| + t\|\mathbb{G}_T\| = (2n+t+1)*$ 256 bits	$(4n+t+1)T_s + (2n)T_m + (4n+t)T_a + tT_p \approx (7820n + 81122t + 1914)$ CCs	$(2n+1)T_p + T_s + (2n+1)T_a \approx (158416n + 81122)$ CCs

- $|\mathbb{G}|$: the length of an element in $\mathbb{G}$
- $|\mathbb{G}_T|$: the length of an element in $\mathbb{G}_T$
- n : the dimension of a predicate vector
- t : the number of receivers
- T_p : the cost of a pairing operation
- T_m : the cost of a modular multiplication in $\mathbb{Z}_q$
- T_s : the cost of a scalar multiplication in an additive group or an exponentiation in a multiplicative group
- T_a : the cost of an addition in an additive group or a multiplication in a multiplicative group
- CCs: Clock Cycles
- $\triangle$: the performance relies upon the underlying cryptographic primitives

3) $\mathcal{S}$ returns the ciphertext $C^* = (\{C_i'\}_{i=1}^t = \{M \cdot \hat{e}(PK_i, h)^s\}_{i=1}^t, C_0 = g_p^s, \{C_{1,j} = (h_{1,j})^s Q_2^{x_{0,j}} R'_{6,j}, C_{2,j} = (h_{2,j})^s \cdot (g_q^\gamma)^{x_{1,j}} \cdot R_{7,j}\}_{j=1}^n)$.

Phase 2: $\mathcal{A}$ makes queries as those in *Phase 1*.

Guess: Finally, $\mathcal{A}$ outputs $\beta' \in \{0,1\}$. If $\beta' = \beta$, then $\mathcal{S}$ outputs 1. Otherwise, $\mathcal{S}$ outputs 0.

If $T = g_p^{b^2 s} g_q^\gamma R_3$ and $\gamma = 0$, the challenge ciphertext is distributed exactly as that in $Game_1$, whereas if γ is chosen uniformly from $\mathbb{Z}_N$, the challenge ciphertext is distributed exactly as that in $Game_2$. It follows that under the SD assumption, these two games are indistinguishable. Therefore, C^* is a correct ciphertext. As the construction above, $\mathcal{S}$ correctly simulates the attribute hiding game. If $\mathcal{A}$ wins the attribute hiding game with non-negligible advantage ϵ, $|Pr[\mathcal{S}(\bar{Z}, T = g_p^{b^2 s} R_3) = 1] - Pr[\mathcal{S}(\bar{Z}, T \in \mathbb{G}) = 1]| \geq \epsilon$ under a correct simulation of the game, i.e., $|Pr[\mathcal{A}(\Omega) = \beta' = \beta \mid Game_0] - Pr[\mathcal{A}(\Omega) = \beta' = \beta \mid Game_1]| \geq \epsilon$, where Ω is a correct multi-receiver predicate encryption scheme.

Therefore, $\mathcal{S}$ solves the SD problem with non-negligible advantage ϵ within polynomial time.

3) Completing the Proof of Hybrid Games: Game$_2$ and Game$_3$ are indistinguishable where the proof is similar to that in Section IV-B2, while Game$_3$ and Game$_4$ are indistinguishable where the proof is similar to that in Section IV-B1. This summarizes the proof of **Theorem 4.2**. ■

As the proof of **Theorem 4.2** shown above, our MRPE scheme achieves attribute hiding against malicious users. Thus, we can say that no users can obtain the information about the keywords of a ciphertext, unless he has the corresponding predicate tokens.

V. COMPARISONS

In this section, we discuss the properties of the proposed scheme, Lin *et al.*'s scheme [17], and the other ASPE schemes [9], [11], [13], [21] in the OSN platform, and analyze the security properties of theses schemes.

A. Properties Comparisons

Due to Lin *et al.*'s scheme depending on the underlying cryptographic primitives it adopts, we exclude it from performance comparison. Our scheme is the first multi-receiver predicate encryption and our works offer users not only privacy preserving but also the ability of finding the ciphertexts they desire. In the comparison of properties, we will focus on the storage cost of ciphertext size. Let $|\mathbb{G}|, |\mathbb{G}_T|$ be the length of an element in $\mathbb{G}$, $\mathbb{G}_T$, respectively, n be the dimension of a predicate vector, and t be the number of receivers. According to [12], [18], [23], [33], we can obtain that $T_p \approx 5T_e$, $T_s \approx 29T_m$, $T_e \approx 240T_m$, and $T_a \approx 0.12T_m$. Besides, by applying the result of [19], a modular multiplication over a 256-bit finite field needs 66 clock cycles. The property comparison is shown in Table II. Figs. 5 and 6 show that the growth of the clock cycles when an encryption is performed. Fig. 5 illustrates the case that a sender encrypts a message for five receivers ($t = 5$). On the other hand, in Fig. 6 we show the computation cost of an encryption under

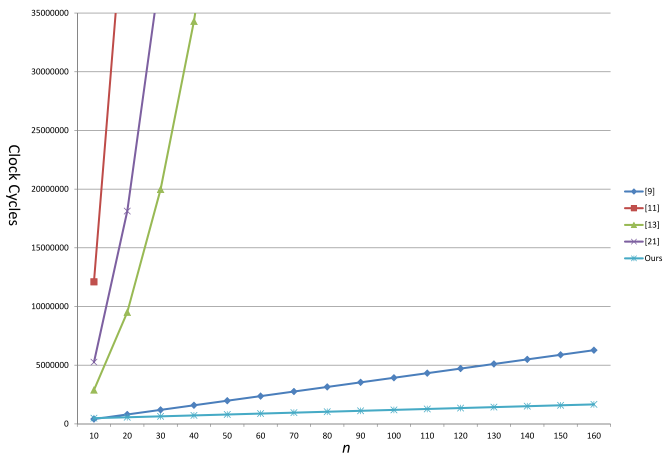

Fig. 5. Computation cost of encryption ($t = 5$).

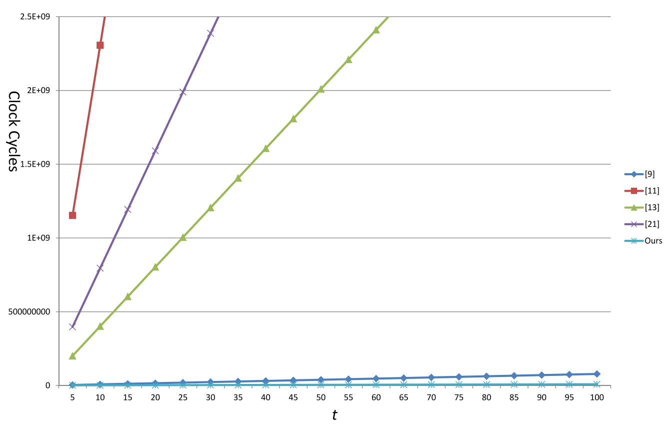

Fig. 6. Computation cost of encryption ($n = 100$).

different numbers of receivers with a fixed $n = 100$. We also demonstrate the computation cost of a decryption in Fig. 7. One can observe that the computation cost is much lower than the other works. As for the ciphertext length, we conclude the comparison results with Figs. 8 and 9. In Fig. 8 we consider the bit length of a ciphertext with a fixed $t = 5$, and Fig. 9 shows the cipheretxt length with a fixed $n = 100$. The cipheretxt length in our scheme outclasses the other schemes. Furthermore, we demonstrate the execution time for one encryption/decryption on both PC (ASUS M32CD i5-6400) and smart phone (SAMSUNG Galxy S7) in Table III.

In the following, we give the properties that are required for an OSN platform and the comparisons will be based on these properties.

• Access Control: The property means that every message in the OSN platform can be accessed by authorized parties only. An unauthorized party is unable to get any data in the OSN platform.

• Dynamic Group: The data owners can freely share the data among multiple users rather than a group with fixed members. The property of dynamic group makes it possible to renew or revoke users efficiently with neither a group manager nor computing a group key.

• Data Searching: The property means that users can perform keyword search to find interested data posted by other users

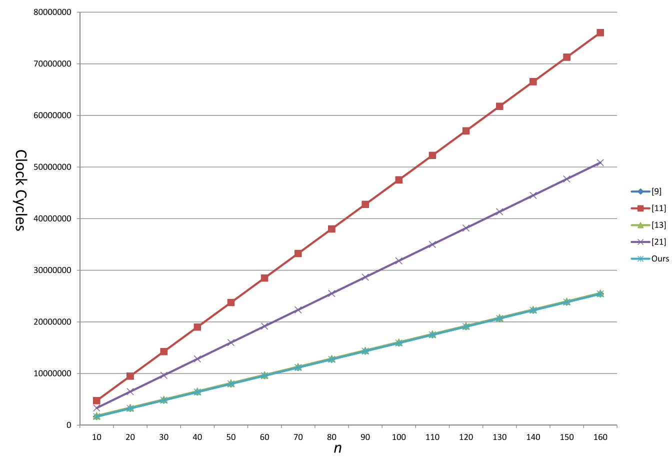

Fig. 7. Computation cost of decryption.

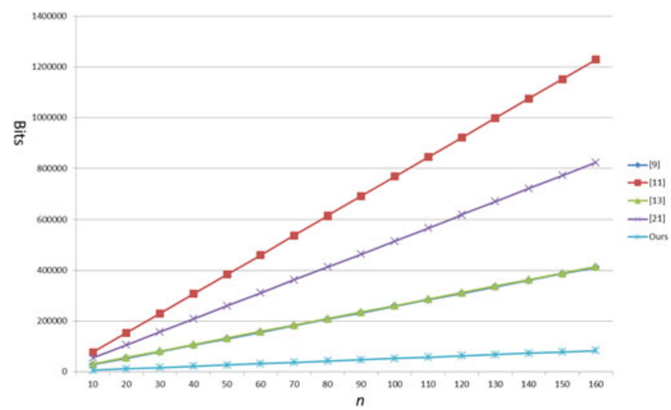

Fig. 8. Ciphertext length with $t = 5$.

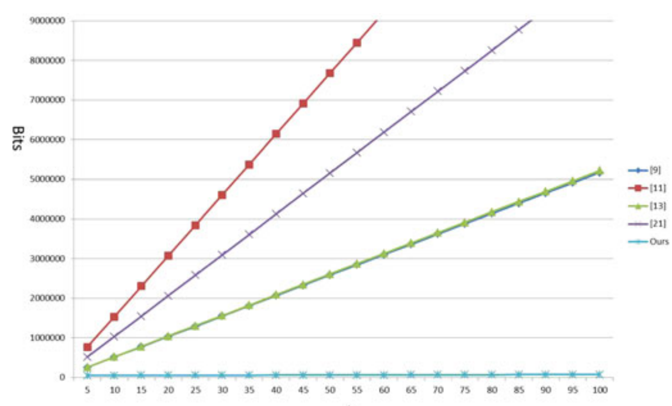

Fig. 9. Ciphertext length with $n = 100$.

TABLE III
COMPUTATION COST

	Clock Speed	Encryption	Decryption
ASUS M32CD (i5-6400)	2.7 GHz	0.00044 sec	0.0059 sec
SAMSUNG Galaxy S7	1.6 GHz	0.00074 sec	0.0099 sec

Here we set $t = 5$ and $n = 100$.

TABLE IV
SECURITY COMPARISONS

	Assumption	Security		Attribute Hiding	Order of $\mathbb{G}$
[9]	Variants of GSD	Selective	CPA	Full	Composite
[11]	DLIN	Adaptive	CPA	Full	Prime
[13]	n-eDDH	Adaptive	CPA	Weak	Prime
[21]	DLIN	Adaptive	CPA	Full	Prime
[17]	Δ	Δ	Δ	Δ	Prime
Ours	Variants of GSD	Selective	CPA	Full	Composite

- GSD: General Subgroup Decision Assumption
- DLIN: Decisional Linear Assumption
- n-eDDH: n-Extended Decisional Diffie-Hellman Assumption
- Δ: the security properties depend on the underlying cryptographic primitives.

efficiently in the OSN platform. Lin *et al.*'s scheme does not satisfy the property because a user can search on the messages he has received only.

• Collusion Resistance: Only the selected receivers can successfully decrypt the ciphertext. The property guarantees the confidentiality even if the attacker colludes with the OSN provider. Lin *et al.*'s scheme does not satisfy the property since the OSN provider has all of the re-encryption keys, so that it can re-encrypt a ciphertext for a collusive user.

• Unified Keywords: The public parameter is published by the OSN provider, not published separately by each user. In a traditional ASPE, each user chooses his own system parameters, but it would cause a great overhead on ciphertext processing.

• Ciphertext Length: Suppose that a data owner wants to share a message with an n-dimensional predicate vector among t receivers in the OSN platform. Though Lin *et al.*'s scheme achieves constant-size ciphertext by applying proxy re-encryption, it loses some important properties, such as ″ Collusion Resistance″ and ″ Data Searching″. In a traditional ASPE, the storage cost of the ciphertext is $O(t \cdot n)|\mathbb{G}| + O(1)|\mathbb{G}_T|$ as it does not unify the public parameters, but in our proposed scheme, the cost is $O(t+n)|\mathbb{G}| + O(1)|\mathbb{G}_T|$ only.

B. Security Comparisons

In this section, we discuss the security of the proposed scheme and the other schemes in the OSN platform. The underlying assumptions of our work and [9] are two variants of the subgroup decision assumption. The scheme of [17] is a general construction that uses a proxy re-encryption scheme and an SSE scheme as black boxes. Thus, the security of [17] depends on the security of the underlying cryptographic primitives. Selective security requires that the adversary has to commit the attacked predicates in advance. On the contrary, adaptive security allows the adversary to commit the target predicates after getting access to the oracles. The weak attribute hiding is defined in [13]. For the weak attribute hiding, an attacker $\mathcal{A}$ queries predicates $\vec{v}$ to the oracles with restriction that for challenge keywords $\vec{x}_0$ and $\vec{x}_1$, $\langle v, x_0\rangle \neq 0$ and $\langle v, x_1\rangle \neq 0$. The full attribute hiding is defined in [9], in which the restriction for the challenge phase is $\langle \vec{v}\cdot\vec{x}_0\rangle = \langle \vec{v}\cdot\vec{x}_1\rangle = 0$ and $M_0 = M_1$. Our scheme can achieve this. The security comparisons are summarized in Table IV.

VI. CONCLUSION

Due to the thriving nature of internet and cloud computing, OSN platforms have become a popular application. The most important issue is protecting users' privacy and generating accurate advertisement simultaneously in OSN platforms.

However, there is no scheme that can solve the problems mentioned in Section I of the OSN platforms. In order to cope with the problems, we have proposed a multi-receiver predicate encryption scheme, which can achieve both privacy preserving and customized advertisement. The proposed scheme is the first multi-receiver predicate encryption and our work supports a user to search for his interested data encrypted and shared by other users in the OSN platform. In this paper, we have proven the CPA security of our scheme in the standard model against malicious users and KGC. Moreover, the proposed scheme greatly reduces the size of ciphertext. In the future works, we will further improve it to reach constant size of ciphertext and achieve CCA security in the standard model.

APPENDIX

Remark 1: The correctness of the MasterSearch phase is demonstrated below.

$$
\begin{aligned}
D &= \hat{e}(C_0, K_0)\prod_{j=1}^{n}\hat{e}(C_{1,j}, K_{1,j})\hat{e}(C_{2,j}, K_{2,j}) \\
&= \hat{e}\left(g_p^s, h^{-\mu}R_3Q_3\prod_{j=1}^{n}h_{1,j}^{-r_{1,j}}h_{2,j}^{-r_{2,j}}\right)\prod_{j=1}^{n}\hat{e}(H_{1,j}^sQ^{\alpha x_j}R_{4,j}, \\
&\quad g_p^{r_{1,j}}g_q^{f_1v_j})\hat{e}(H_{2,j}^sQ^{\beta x_j}R_{5,j}, g_p^{r_{2,j}}g_q^{f_2v_j}) \\
&= \hat{e}\left(g_p^s, h^{-\mu}\prod_{j=1}^{n}h_{1,j}^{-r_{1,j}}h_{2,j}^{-r_{2,j}}\right)\prod_{j=1}^{n}\hat{e}(h_{1,j}^sg_q^{\alpha x_j}, g_p^{r_{1,j}}g_q^{f_1v_j}) \\
&\quad \cdot\hat{e}(h_{2,j}^sg_q^{\beta x_j}, g_p^{r_{2,j}}g_q^{f_2v_j}) \\
&= \hat{e}(g_p, h)^{-s\mu}\prod_{j=1}^{n}\hat{e}(g_q, g_q)^{(\alpha f_1+\beta f_2)x_jv_j} \\
&= \hat{e}(g_p, h)^{-s\mu}\hat{e}(g_q, g_q)^{(\alpha f_1+\beta f_2)\cdot\langle\vec{x},\vec{v}\rangle}
\end{aligned}
$$

where α, β are randomly selected in $\mathbb{Z}_N$ and f_1, f_2 are randomly chosen in $\mathbb{Z}_q$.

1) If KGC (OSNs) is successful in matching, i.e., $\langle\vec{x},\vec{v}\rangle = 0 \bmod N$, then D is in the subgroup of $\mathbb{G}_T$ with order p.
2) Otherwise, $\langle\vec{x},\vec{v}\rangle \neq 0 \bmod N$, then D is in the subgroup of $\mathbb{G}_T$ with order pq.

Remark 2: The correctness of the Decrypt phase is demonstrated in the following.

$$
\begin{aligned}
&C'_i\left(\hat{e}(C_0,K_0)\prod_{j=1}^{n}\hat{e}(C_{1,j},K_{1,j})\hat{e}(C_{2,j},K_{2,j})\right)^{SK_i}\\
&= M\hat{e}(PK_i,h)^s\cdot\left(\hat{e}(g_p,h)^{-s\mu}\prod_{j=1}^{n}\hat{e}(g_q,g_q)^{(\alpha f_1+\beta f_2)x_j v_j}\right)^{z_i}\\
&= M\hat{e}((g_p^{\mu}Q_2)^{z_i},h)^s\cdot(\hat{e}(g_p,h)^{-s\mu}\hat{e}(g_q,g_q)^{(\alpha f_1+\beta f_2)\cdot\langle\vec{x},\vec{v}\rangle})^{z_i}\\
&= M\hat{e}(g_p,h)^{s\mu z_i}\cdot(\hat{e}(g_p,h)^{-s\mu}\hat{e}(g_q,g_q)^{(\alpha f_1+\beta f_2 \mod q)\langle\vec{x},\vec{v}\rangle})^{z_i}\\
&= M\hat{e}(g_p,h)^{s\mu z_i}\cdot(\hat{e}(g_p,h)^{-s\mu})^{z_i}\\
&= M,
\end{aligned}
$$

where α,β are random elements in $\mathbb{Z}_N$ and f_1,f_2 are random elements in $\mathbb{Z}_q$.

1) If $\langle\vec{x},\vec{v}\rangle = 0 \mod N$, the output of the decryption is M.
2) If $\langle\vec{x},\vec{v}\rangle \neq 0 \mod N$, there are two cases:
   a) If $\langle\vec{x},\vec{v}\rangle \neq 0 \mod q$, the decryption will fail since the keywords and predicates do not match.
   b) If $\langle\vec{x},\vec{v}\rangle = 0 \mod q$, the calculation result is M and the decryption will success. However this condition only happens with negligible probability since it will reveal a non-trivial factor of N.

Remark 3: We now give a simple instance for illustrating how OR-gate and AND-gate can be implemented by inner-product. As a simple example, we assume the predicates $(X_1 = A, X_2 = B)$,

1) **Case 1.** We use OR-gate, where the predicates $X_1 = A$ or $X_2 = B$ can be encoded as a polynomial as follows.

$$
\begin{aligned}
&(X_1 = A)\vee(X_2 = B)\\
&\Rightarrow(X_1 - A)(X_2 - B) = X_1X_2 - AX_2 - BX_1 + AB = 0,\\
&\Rightarrow\vec{x} := \omega(X_1X_2, X_1, X_2, 1),\ \vec{v} := \sigma(1, -A, -B, AB)
\end{aligned}
$$

.

Thus, the result vector $\vec{v} := (\sigma(1,-A,-B,AB))$ can make the predicates ($X_1 = A$ or $X_2 = B$) match the corresponding ciphertexts.

2) **Case 2.** We use AND-gate, where the predicates $X_1 = A$ and $X_2 = B$ can be encoded as a polynomial as follows.

$$
\begin{aligned}
&(X_1 = A)\wedge(X_2 = B)\\
&\Rightarrow[\omega\omega'(X_1 - A) + \sigma\sigma'(X_2 - B)] = 0,\\
&\Rightarrow\vec{x} := (\omega(X_1,1),\sigma(X_2,1)),\ \vec{v} := (\omega'(1,-A),\sigma'(1,-B)).
\end{aligned}
$$

Thus, the result vector $\vec{v} := (\omega'(1,-A),\sigma'(1,-B))$ can make the predicates ($X_1 = A$ and $X_2 = B$) match the corresponding ciphertexts.

Remark 4: To see that this predicate token has the correct distribution, by construction of $\{K_{1,j},K_{2,j}\}$, the simulator is implicitly setting $f_1 = f'_1 - kd,\ f_2 = 1 - kd,\ r_{1,j} = -kv_j + r'_{1,j}$, and $r_{2,j} = -kv_j + r'_{2,j}$ for all j's. These values are all uniformly and independently distributed in $\mathbb{Z}_N$. Next, note that

$$
\begin{aligned}
&\prod_{j=1}^{n}(g_p^{\delta_{1,j}}h^{x_j})^{-r'_{1,j}}\cdot(g_p^{\mu})^{kv_j\delta_{1,j}}\\
&= \prod_{j=1}^{n} g_p^{-\delta_{1,j}r'_{1,j}+k\mu v_j\delta_{1,j}}\cdot h^{-x_j r'_{1,j}}\\
&= \prod_{j=1}^{n} g_p^{-\delta_{1,j}\cdot(r_{1,j}+k\mu v_j)+k\mu v_j\delta_{1,j}}\cdot h^{-x_j(r_{1,j}+k\mu v_j)}\\
&= \prod_{j=1}^{n}(h^{x_j}g_p^{\delta_{1,j}})^{-r_{1,j}}\cdot h^{-\mu k v_j x_j} = h^{-\mu/2}\cdot\prod_{j=1}^{n}h_{1,j}^{-r_{1,j}},
\end{aligned}
$$

using the fact that $k = \frac{1}{2\cdot\langle\vec{x},\vec{v}\rangle} \mod N$. We denote K_p as the "$\mathbb{G}_p$ part" of K_0. Thus, we can that

$$
\begin{aligned}
K_p &= \prod_{j=1}^{n}((g_p^{\delta_{1,j}}h^{x_j})^{-r'_{1,j}}\\
&\quad\cdot(g_p^{\mu})^{kv_j\delta_{1,j}})\cdot((g_p^{\delta_{2,j}}h^{x_j})^{-r'_{2,j}}\cdot(g_p^{\mu})^{kv_j\delta_{2,j}})\\
&= h^{-\mu}\cdot\prod_{j=1}^{n}h_{1,j}^{-r_{1,j}}h_{2,j}^{-r_{2,j}},
\end{aligned}
$$

and thus, K_p (and hence K_0) is distributed appropriately.

Remark 5: We denote K_p as the "$\mathbb{G}_p$ part" of K_0. To see that this predicate token has the correct distribution, by the construction of $\{K_{1,j},K_{2,j}\}$, the simulator is implicitly setting $f_1 = f'_1 - df'_2,\ f_2 = f'_2,\ r_{1,j} = r'_{1,j} + v_j\cdot(af'_1 - abf'_2)$, and $r_{2,j} = r'_{2,j} + af'_2 v_j$ for all j's. These values are all uniformly and independently distributed in $\mathbb{Z}_N$. Next, note that

$$
\begin{aligned}
K_p &= h^{-\mu}\cdot g_p^{-abkf'_1}\cdot\prod_{j=1}^{n}g_p^{-af'_1v_j\delta_{1,j}-af'_2v_j\delta_{2,j}}g_p^{abf'_2v_j\delta_{1,j}}\\
&\quad\cdot g_p^{-\delta_{1,j}r'_{1,j}-\delta_{2,j}r'_{2,j}}h_p^{-x_{0,j}r'_{1,j}}k_p^{-x_{0,j}r'_{2,j}}\\
&= h^{-\mu}\cdot h_p^{-akf'_1}\cdot\prod_{j=1}^{n}(h_p^{-x_{0,j}r'_{1,j}}g_p^{-\delta_{1,j}r'_{1,j}}g_p^{-\delta_{1,j}v_j(af'_1-abf'_2)})\\
&\quad\cdot(k_p^{-x_{0,j}r'_{2,j}}g_p^{-\delta_{2,j}r'_{2,j}}g_p^{-\delta_{2,j}af'_2v_j})\\
&= h^{-\mu}\cdot\prod_{j=1}^{n}h_p^{-ax_{0,j}v_jf'_1}\\
&\quad\cdot(h_p^{-x_{0,j}r'_{1,j}}g_p^{-\delta_{1,j}r'_{1,j}}g_p^{-\delta_{1,j}v_j(af'_1-abf'_2)})\\
&\quad\cdot(h_p^{abx_{0,j}v_jf'_2}\cdot h_p^{-abx_{0,j}v_jf'_2})\\
&\quad\cdot(k_p^{-x_{0,j}r'_{2,j}}g_p^{-\delta_{2,j}r'_{2,j}}g_p^{-\delta_{2,j}af'_2v_j}),
\end{aligned}
$$

using the fact that $\langle \vec{x_0}, \vec{v} \rangle = \sum_j x_{0,j} v_j$. Thus, we have that

$$K_p = h^{-\mu} \cdot \prod_{j=1}^{n} (h_p^{-x_{0,j} r'_{1,j}} g_p^{-\delta_{1,j} r'_{1,j}} h_p^{-x_{0,j} v_j (af'_1 - abf'_2)}$$
$$\cdot g_p^{-\delta_{1,j} v_j (af'_1 - abf'_2)})$$
$$\cdot (k_p^{-x_{0,j} r'_{2,j}} g_p^{-\delta_{2,j} r'_{2,j}} k_p^{-x_{0,j} af'_2 v_j} g_p^{-\delta_{2,j} af'_2 v_j})$$
$$= h^{-\mu} \cdot \prod_{j=1}^{n} (h_p^{x_{0,j}} g_p^{\delta_{1,j}})^{-r_{1,j}} (k_p^{x_{0,j}} g_p^{\delta_{2,j}})^{-r_{2,j}}$$
$$= h^{-\mu} \prod_{j=1}^{n} h_{1,j}^{-r_{1,j}} h_{2,j}^{-r_{2,j}},$$

and thus, K_p (and hence K_0) is distributed appropriately.

Remark 6: We denote K_p as the "$\mathbb{G}_p$ part" of K_0. To see that this predicate token has the correct distribution, by the construction of $\{K_{1,j}, K_{2,j}\}$, the simulator is implicitly setting f_1, f_2 are random, $r_{1,j} = r'_{1,j} + af_1 v_j$ and $r_{2,j} = r'_{2,j} + af_2 v_j$ for all j's. These values are all uniformly and independently distributed in $\mathbb{Z}_N$. Thus, we have that

$$K_p = h^{-\mu} \cdot \prod_{j=1}^{n} g_p^{-af_1 v_j \delta_{1,j} - af_2 v_j \delta_{2,j}} \cdot g_p^{-\delta_{1,j} r'_{1,j} - \delta_{2,j} r'_{2,j}}$$
$$\cdot h_p^{-x_{0,j} \cdot r'_{1,j}} \cdot k_p^{-x_{1,j} \cdot r'_{2,j}}$$
$$= h^{-\mu} \cdot \prod_{j=1}^{n} h_p^{-af_1 v_j x_{0,j}} k_p^{-af_2 v_j x_{1,j}} \cdot g_p^{-af_1 v_j \delta_{1,j} - af_2 v_j \delta_{2,j}}$$
$$\cdot g_p^{-\delta_{1,j} r'_{1,j} - \delta_{2,j} r'_{2,j}} \cdot h_p^{-x_{0,j} \cdot r'_{1,j}} \cdot k_p^{-x_{1,j} \cdot r'_{2,j}},$$
$$= h^{-\mu} \cdot \prod_{j=1}^{n} (h_p^{x_{0,j}} g_p^{\delta_{1,j}})^{-r_{1,j}} (k_p^{x_{1,j}} g_p^{\delta_{2,j}})^{-r_{2,j}}$$

using the fact that $\prod_{j=1}^{n} h_p^{-af_1 x_{0,j} v_j} = h_p^{-af_1 \cdot <\vec{x_0}, \vec{v}>} = 1 = \prod_{j=1}^{n} k_p^{-af_2 x_{1,j} v_j}$. Thus, K_p (and hence K_0) is distributed appropriately.

Remark 7: To see that this predicate token has the correct distribution, by the construction of $\{K_{1,j}, K_{2,j}\}$, the simulator is implicitly setting the value $f_1 = f'_1 - c_{x_1} \cdot df'_2$, $f_2 = c_{x_0} \cdot f'_2$, $r_{1,j} = r'_{1,j} + (af'_1 - c_{x_1} abf'_2) \cdot v_j$, and $r_{2,j} = r'_{2,j} + ac_{x_0} f'_2 v_j$ for all j's. These values are all uniformly and independently distributed in $\mathbb{Z}_N$. Thus, we have that

$$K_p = h^{-\mu} \cdot g_p^{-abc_{x_0} f'_1} \cdot \prod_{j=1}^{n} g_p^{-af'_1 v_j \delta_{1,j} - af'_2 c_{x_0} v_j \delta_{2,j}}$$
$$\cdot g_p^{abf'_2 c_{x_1} v_j \delta_{1,j}} \cdot g_p^{-\delta_{1,j} r'_{1,j} - \delta_{2,j} r'_{2,j}} \cdot h_p^{-x_{0,j} \cdot r'_{1,j}} \cdot k_p^{-x_{1,j} \cdot r'_{2,j}}$$
$$= h^{-\mu} \cdot \prod_{j=1}^{n} h_p^{-af'_1 v_j x_{0,j}} \cdot g_p^{-af'_1 v_j \delta_{1,j} - af'_2 c_{x_0} v_j \delta_{2,j}}$$
$$\cdot g_p^{abf'_2 c_{x_1} v_j \delta_{1,j}} \cdot (h_{1,j})^{-r'_{1,j}} \cdot (h_{2,j})^{-r'_{2,j}}$$
$$= h^{-\mu} \cdot h_p^{c_{x_0} c_{x_1} abf'_2} \cdot h_p^{-c_{x_0} c_{x_1} abf'_2} \cdot \prod_{j=1}^{n} g_p^{-af'_2 c_{x_0} v_j \delta_{2,j}}$$
$$\cdot g_p^{abf'_2 c_{x_1} v_j \delta_{1,j}} \cdot (h_{1,j})^{-r'_{1,j} - av_j f'_1} \cdot (h_{2,j})^{-r'_{2,j}}$$
$$= h^{-\mu} \cdot \prod_{j=1}^{n} h_p^{x_{0,j} v_j c_{x_1} abf'_2} k_p^{-c_{x_0} x_{1,j} v_j af'_2} g_p^{-af'_2 c_{x_0} v_j \delta_{2,j}}$$
$$\cdot g_p^{abf'_2 c_{x_1} \delta_{1,j} v_j} \cdot (h_{1,j})^{-r'_{1,j} - av_j f'_1} \cdot (h_{2,j})^{-r'_{2,j}}$$
$$= h^{-\mu} \cdot \prod_{j=1}^{n} (h_{1,j})^{-r'_{1,j} - av_j f'_1 + abf'_2 c_{x_1} v_j}$$
$$\cdot (h_{2,j})^{-r'_{2,j} - ac_{x_{0,x}} v_j f'_2}$$
$$= h^{-\mu} \prod_{j=1}^{n} (h_{1,j})^{-r_{1,j}} \cdot (h_{2,j})^{-r_{2,j}},$$

and thus, K_p (and hence K_0) is distributed appropriately.

References

[1] J. Anderson, J. Diaz, C. Bonneau, and F. Stajano, "Privacy-enabling social networking over untrusted networks," in *Proc. Workshop Online Social Netw.*, 2009, pp. 1–6.

[2] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 135–146, Aug. 2009.

[3] C. Blundo, V. Iovino, and G. Persiano, "Private-key hidden vector encryption with key confidentiality," in *Proceedings of the 8th International Conference on Cryptology and Network Security* (LNCS), vol. 5888, Berlin, Germany: Springer, 2009, pp. 259–277.

[4] S. Braghin, V. Iovino, G. Persiano, and A. Trombetta, "Secure and policy-private resource sharing in an online social network," in *Proc. Privacy, Secur., Risk Trust*, 2011, pp. 872–875.

[5] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. 4th Theory of Cryptography Conference (LNCS)*, vol. 4392, Berlin, Germany: Springer, 2007, pp. 535–554.

[6] M. Durr, M. Maier, and F. Dorfmeister, "Vegas—A secure and privacy-preserving peer-to-peer online social network," in *Proc. Social Comput./Privacy, Secur., Risk Trust*, 2012, pp. 868–874.

[7] C. I. Fan and S. Y. Huang, "Controllable privacy preserving search based on symmetric predicate encryption in cloud storage," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1716–1724, 2013.

[8] S. Jahid, P. Mittal, and N. Borisov, "Easier: Encryption-based access control in social networks with efficient revocation," in *Proc. 6th Int. Symp. Inf. Comput. Commun. Secur.*, 2011, pp. 411–415.

[9] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proceedings of Advances in Cryptology—EUROCRYPT 2008 (LNCS)*, vol. 4965, Berlin, Germany: Springer, 2008, pp. 146–162.

[10] J. Katz and A. Yerukhimovich, "On black-box constructions of predicate encryption from trapdoor permutations," in *Proc. 15th Int. Conf. Theory*

Appl. Cryptol. Inf. Secur., Adv. Cryptol., Tokyo, Japan, Dec. 6–10, 2009, pp. 197–213.

[11] Y. Kawai and K. Takashima, “Predicate- and attribute-hiding inner product encryption in a public key setting,” in *Pairing-Based Cryptography—Pairing*, vol. 8365, Berlin, Germany: Springer, 2013, pp. 113–130.

[12] N. Koblitz, A. Menezes, and S. Vanstone, “The state of elliptic curve cryptography,” *Des. Codes Cryptography*, vol. 19, pp. 173–193, 2000.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, “Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption,” in *Proc. Adv. Cryptology—EUROCRYPT 2010 (LNCS)*, vol. 6110, Berlin, Germany: Springer, 2010, pp. 62–91.

[14] M. Li, N. Cao, S. Yu, and W. Lou, “Findu: Privacy-preserving personal profile matching in mobile social networks,” in *Proc. IEEE Conf. Comput. Commun.*, 2011, pp. 2435–2443.

[15] X. Liang, M. Barua, R. Lu, X. Lin, and X. Shen, “Healthshare: Achieving secure and privacy-preserving health information sharing through health social networks,” *Comput. Commun.*, vol. 35, no. 15, pp. 1910–1920, 2012.

[16] X. Liang, R. Lu, L. Chen, X. Lin, and X. Shen, “PEC: A privacy-preserving emergency call scheme for mobile healthcare social networks,” *J. Commun. Netw.*, vol. 13, no. 2, pp. 102–112, 2011.

[17] Y. H. Lin, C. Y. Wang, and W. T. Chen, “A content privacy-preserving protocol for energy efficient access to commercial online social networks,” in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 325–341.

[18] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 2001.

[19] A. Mrabet *et al.*, “A systolic hardware architecture of montgomery modular multiplication for public key cryptosystem,” *Cryptology ePrint Archive, Report* 2016/487, 2016. [Online]. Available: http://eprint.iacr.org/2016/487

[20] J. M. González-Nieto, M. Manulis, and D. Sun, “Fully private revocable predicate encryption,” in *Australasian Conf. Inf. Security Privacy (LNCS)*, vol. 7372, Berlin, Germany: Springer, 2012, pp. 350–363.

[21] T. Okamoto and K. Takashima, “Adaptively attribute-hiding (hierarchical) inner product encryption,” in *Proc. Adv. Cryptology—EUROCRYPT 2012 (LNCS)*, vol. 7237, Berlin, Germany: Springer, 2012, pp. 591–608.

[22] G. Romain, M. Pierrick, and W. Hoeteck, “Predicate encryption for multi-dimensional range queries from lattices,” in *Proc. Public-Key Cryptography—18th IACR Int. Conf. Pract. Theory Public-Key Cryptography*, Gaithersburg, MD, USA, Mar. 30–Apr. 1, 2015, pp. 752–776.

[23] M. Scott, “Implementing cryptographic pairings,” in *Proc. Pairing-Based Cryptography*, 2007, pp. 177–196.

[24] E. Shen, E. Shi, and B. Waters, “Predicate privacy in encryption systems,” in *Theory of Cryptography*, vol. 5444, Berlin, Germany: Springer, 2009, pp. 457–473.

[25] H. Shuai and W. Zhu, “Masque: Access control for interactive sharing of encrypted data in social networks,” in *Proc. 6th Int. Conf. Netw. Syst. Secur.*, 2012, pp. 503–515.

[26] Y. Song, P. Karras, Q. Xiao, and S. Bressan, “Sensitive label privacy protection on social network data,” in *Scientific and Statistical Database Management* (Lecture Notes in Computer Science), vol. 7338, Berlin, Germany: Springer, 2012, pp. 562–571.

[27] J. Sun, X. Zhu, and Y. Fang, “A privacy-preserving scheme for online social networks with efficient revocation,” in *Proc. IEEE Conf. Comput. Commun.*, 2010, pp. 2516–2524.

[28] D. H. Tran, H. L. Nguyen, W. Zha, and W. K. Ng, “Towards security in sharing data on cloud-based social networks,” in *Proc. 8th Int. Conf. Inf. Commun. Signal Process.*, 2011, pp. 1–5.

[29] F. W. L. Philip, “Preventing sybil attacks by privilege attenuation: A design principle for social network systems,” in*Proc. IEEE Symp. Secur. Privacy*, 2011, pp. 263–278.

[30] R. Wei and D. Ye, “Delegate predicate encryption and its application to anonymous authentication,” in *Proc. Int. Symp. Inf., Comput., Commun. Secur.*, 2009, pp. 372–375.

[31] K. Xagawa, “Improved (hierarchical) inner-product encryption from lattices,” in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 7778, Berlin, Germany: Springer, 2013, pp. 235–252.

[32] M. Yoshino, N. Kunihiro, K. Naganuma, and H. Sato, “Symmetric inner-product predicate encryption based on three groups,” in *Provable Security* (Lecture Notes in Computer Science), vol. 7496, Berlin, Germany: Springer, 2012, pp. 215–234.

[33] Y. Zhang, W. Liu, W. Lou, and Y. Fang, “Securing mobile ad hoc networks with certificateless public keys,” *IEEE Trans. Depend. Sec. Comput.*, vol. 3, no. 4, pp. 386–399, Oct.–Dec. 2006.

Chun-I Fan received the M.S. degree in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1998. From 1999 to 2003, he was an Associate Researcher and a Project Leader with the Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taoyuan, Taiwan. In 2003, he joined the faculty of the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, where has been a Full Professor since 2010. His research interests include applied cryptology, cryptographic protocols, and information and communication security.

Prof. Fan is the Deputy Chairman of the Chinese Cryptology and Information Security Association, and was the Chief Executive Officer of “Aim for the Top University Plan” Office at the National Sun Yat-sen University. He was the recipient of the Best Student Paper Awards from the National Conference on Information Security in 1998, the Dragon Ph.D. Thesis Award from Acer Foundation, the Best Ph.D. Thesis Award from the Institute of Information and Computing Machinery in 1999, and the Engineering Professors Award from Chinese Institute of Engineers—Kaohsiung Chapter in 2016. He is also an Outstanding Faculty in Academic Research at the National Sun Yat-sen University.

Yi-Fan Tseng was born in Kaohsiung, Taiwan. He received the M.S. degree in computer science and engineering, in 2014, from the National Sun Yat-sen University, Kaohsiung, Taiwan, where he is currently working toward the Ph.D. degree. His research interests include cloud computing and security, network and communication security, information security, cryptographic protocols, and applied cryptography.

Jheng-Jia Huang was born in Kaohsiung, Taiwan. He received the M.S. degree in information management from the National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, in 2012. He is currently working toward the Ph.D. degree in computer science and engineering from the National Sun Yat-sen University, Kaohsiung, Taiwan. His research interests include cloud computing and security, social network security and authentication, network and communication security, information security, and applied cryptography.

Shih-Fen Chen was born in Chiayi, Taiwan. She received the M.S. degree in computer science and engineering from the National Sun Yat-sen University, Kaohsiung, Taiwan, in 2015. Her research interests include cloud computing and cloud storage, network and communication security, and applied cryptography.

Hiroaki Kikuchi (M'96) received the B.E., M.E., and Ph.D. degrees from Meiji University, Tokyo, Japan, in 1988, 1990, and 1994, respectively. After working with the Fujitsu Laboratories Ltd., in 1990, he was with the Tokai University from 1994 to 2013. He is currently a Professor in the Department of Frontier Media Science, School of Interdisciplinary Mathematical Sciences, Meiji University. In 1997, he was a Visiting Researcher in the School of Computer Science, Carnegie Mellon University. His research interests include network security, cryptographical protocol, privacy-preserving data mining, and fuzzy logic. He received the Best Paper Award for Young Researcher of Japan Society for Fuzzy Theory and Intelligent Informatics in 1990, the Best Paper Award for Young Researcher of IPSJ National Convention in 1993, the Best Paper Award of Symposium on Cryptography and Information Security in 1996, the IPSJ Research and Development Award in 2003, the Journal of Information Processing Outstanding paper Award in 2010, and the IEEE AINA Best Paper Award in 2013. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan, the Information Processing Society of Japan (IPSJ), the Japan Society for Fuzzy Theory and Systems, and ACM. Since 2013, he has been a Director of IPSJ. He is an IPSJ Fellow.