This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3294725

Privacy-aware and Security-enhanced Efficient Matchmaking Encryption

Jianfei Sun, Guowen Xu, Tianwei Zhang, Xuehuan Yang, Mamoun Alazab, Robert H. Deng, *Fellow, IEEE*

***Abstract*—Data sharing technologies enable users to outsource data and privately share information with arbitrary recipients without geographic barriers. However, existing efforts for secure data sharing are either inflexible, insufficiently-secure or inefficient. In this paper, we invent PS-ME, the first Privacy-aware and Security-enhanced efficient Matchmaking Encryption (ME) for flexible data sharing. To be more specific, we first formulate an identity-based broadcast matchmaking encryption (IB-BME) for one-to-many data sharing, which enables both participants to specify respective access policies to the encrypted data, such that the data can be revealed by multiple recipients in the case that both access policies are satisfied. In IB-BME, a general matchmaking transformation solution realizing one-to-many sharing is initialized. We also formulate the PS-ME with the general matchmaking transformation solution of IB-BME as the underlying approach, which in addition to featuring IB-BME's all desirable properties, enables efficient decryption, identity anonymity and CCA-security, where we address the open problem of ME regarding CCA-security (raised in CRYPTO'2019). Finally, the comprehensively rigorous security proofs indicate the security of the suggested methodologies. The experimental results are also shown to demonstrate their practicability and effectiveness.**

***Index Terms*—Non-interactive, matchmaking encryption, flexible, anonymity, CCA-security.**

I. Introduction

CLOUD service platforms have emerged as the preferred paradigm for individuals or businesses to share and process data from anywhere and anytime, mainly due to the powerful storage and computing capabilities of the cloud [1]–[5]. For example, Microsoft OneDrive is available to all users to share their photos with their geographically dispersed friends. Google Health allows all participants via Google Drive to exchange their personal health information with various healthcare organizations or individuals. To ensure the confidentiality of shared data, identity/attribute-based cryptographic techniques [6]–[9] as the most frequently-exploited solutions have been applied to encrypting sensitive data. In such a data sharing scenario, data senders generally perform data encryption and outsource the encoded data to the cloud for data sharing, such that only the granted recipients can decode-then-access the data.

A. Security, Privacy & Efficiency Concerns

Despite the fact that many data sharing solutions have been formulated, the state-of-the-art efforts still have inadequacies in security, privacy, and efficiency, as stated as follows.

(1) Insufficiency of considerations to assure data confidentiality and authenticity: Outsourced data are conventionally encrypted-then-uploaded to clouds so that only the granted users can access them. Data encryption can indeed preserve data confidentiality, however, it fails to prevent outsourced data from being edited, modified and even forged, leading to data authenticity breaches. One potential solution is to sign-then-encrypt the data with the integrated technologies of digital signature [11] and traditional public key encryption [10]. Nevertheless, this approach is often infeasible since the integrated solution generally has significant computation overheads and its security cannot be guaranteed as precisely as the respective technologies. Another alternative methodology is to directly exploit existing signcryption techniques [12], [13]. Whereas, the majority of known signcryption techniques (*e.g.*, identity/attribute-based signcryption) are computationally inefficient due to the large number of pairing calculation operations involved. Hence, how to *efficiently* provide data confidentiality and authenticity is the prime concern.

(2) Lack of flexible non-interactive secret handshake mechanisms to ensure the respective intended participants: Considering the practical data exchange scenarios, each participant requires to ensure that the parties participating in the data exchange are the ones intended. This is significantly essential since in data-centric networks [14]–[16], it frequently requires sharing data packets with intended recipients for communication. Hence, data to be shared securely is really necessary between two or more parties who are seeking to establish a match or connection based on certain criteria. A straightforward method would be to use the secret handshake (SH) methodology [17]–[19], which is publicly thought of as a solution to secure data exchange. This solution is less practicable since it requires each participant to be constantly online to complete the authentication and interaction. This interactivity probably leads to some privacy leakage from data traffic analysis. As a non-interactive SH version, matchmaking encryption (ME) technology enables senders/recipients to encode/decode the data offline given only the public key of the recipient/sender, thus eliminating the need for real-time interactions, and mitigating or blocking data traffic analysis. However, existing ME solutions [20]–[24] (See Section II for more details) either only target one-to-one inflexible non-interactive data exchange (*i.e.*, identity-based ME) or have

Jianfei Sun, Guowen Xu, Tianwei Zhang and Xuehuan Yang are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore; (email: {jianfei.sun,guowen.xu,tianwei.zhang, xuehuan.yang}@ntu.edu.sg). Guowen Xu is the corresponding author.

Mamoun Alazab is with the College of Engineering, IT and Environment, Charles Darwin University, NT0810, Australia. (email: alazab.m@ieee.org).

Robert. H Deng is with the School of Computing and Information Systems, Singapore Management University, Singapore; (email: robertdeng@smu.edu.sg).

prohibitive computation and storage costs (*i.e.*, attribute-based ME). Consequently, how to *efficiently* design a *flexible* non-interactive SH mechanism that ensures the respective intended participants remains an open question.

(3) Difficulty in preventing the disclosure of identity privacy: Perfectly, the outsourced encrypted data should be as private as possible, *i.e.* data information and recipients' identities should be anonymous to non-granted entities in the system. This is highly indispensable for scenarios with high privacy requirements, such as medical scenarios. Assuming that an attacker can positively identify that the patient's medical data are destined for a specific doctor from encrypted data, then the disease the patient is suffering from could be revealed with an overwhelming probability from the doctor's identity. One of the frequently-utilized methodologies [25] is to separate the identity into two blind parts, so that an adversary is incapable of differentiating the specific identity via bilinear pairing operations, thus enabling the identity anonymity of the recipients. Such a solution spitting an identity into two parts used for ciphertext generation apparently leads to additional calculation and communication costs compared to the solution that the identity as a whole is used for encryption. Another potential method is to use the anonymous technique [26]–[28], such as hidden vector encryption (HVE) or inner product encryption (IPE), to convert an identity into a vector hidden for producing the private key or ciphertext. However, existing HVE and IPE-based anonymous techniques either suffer from inefficiency issues or undergo insecure hazards based on symmetric prime-order group construction. As a consequence, how to *securely and efficiently* achieve identity anonymity is also challenging.

(4) Absence of formidable attack-resistance countermeasures to guarantee stronger security requirements: Generally, the security assurance for outsourced encrypted data with most encryption mechanisms only reaches the semantic security against (passive) chosen-plaintext attacks (CPA), which enables encryption to be secure against eavesdropping only but fails to guarantee secrecy under some active attacks, such as tampering attacks and impersonation attacks, denial-of-service attacks, etc. Withstanding the active attacks more than passive attacks is fundamentally imperative, especially in data sharing scenarios since active attacks seek to locate and destroy the data whereas the passive attacks aim to steal valuable information; besides the network performance is caused more damage by the active attacks than the passive ones. As an effective countermeasure to be immune to most active attacks, the solution is to achieve security against chosen-ciphertext attacks (CCA). However, realizing *CCA security* is not a simple task but an intractable one [29]. The reason is primarily that in CCA there are basically no limitations to regulate the modification attacks the active adversaries launch.

B. Solutions & Technical Challenges

Imperfection of existing solutions: To the best of our knowledge, there are no existing studies that can simultaneously well-address the above challenges. As briefly-illustrated above, the identity-based signature/signcryption (IBS/IBSC) techniques [11]–[13] enable a user to embed his/her identity-based encryption key into a ciphertext, so that the receiver can verify the signature validity with public keys; the identity/attribute-based ME (IB-ME/AB-ME) technologies [20]–[24] allow both participants to designate respective access policies (such as an identity or a set of attributes) to encrypted data, such that only the user satisfying both policies can reveal the encrypted data. Besides, IB-ME/AB-ME [20]–[24] also realize data authenticity via embedding identity/attribute-based encryption into the ciphertext; the HVE and IPE methodologies [26]–[28] enable the transformation from an access policy to an access vector, which is absolutely hidden in the ciphertext to realize identity anonymity; the CCA-secure public-key cryptographic (PKC) technologies, including broadcast encryption [30], identity-based BE (IBBE) [31], identity/attribute-based encryption (IBE/ABE) [32], [33], enable various active attack resistances, thus guaranteeing stronger security requirements. These cryptographic technologies may be exploited to mitigate the above challenges, whereas they are only applicable to solve certain ones.

Concisely, with the property of supporting data unforgeability, although IBS/IBSC can ensure data confidentiality and authenticity, they do not cater to the requirements of (2) to (4); apart from ensuring data authenticity, IB-ME/AB-ME are capable of supporting non-interactive SH via designating bilateral access policies for the respective intended participants. However, they are infeasible for the requirements of (3) and (4). Besides, existing IB-ME works are inflexible as they only support one-to-one data sharing. Although AB-ME solutions enable one-to-many data sharing, they suffer from a serious inefficiency issue, *i.e.*, the computing and storage costs grow linearly with the complexity of access policies; as an effective approach to anonymize identity, HVE/IPE works enable hiding the access vector corresponding to the access policy in the ciphertext, thus achieving identity anonymity, but they do not satisfy the requirements of (1), (2) and (4); while standard CCA-secure PKC solutions reach higher security assurance, they are incapable of fulfilling the functionality demands of (1) to (3).

Potential solutions & technical challenges: Intuitively, the security, privacy and efficiency requirements may be settled by the convergence of the aforementioned technologies. The most natural strategy is to apply IB-ME, HVE/IPE to CCA-secure IBBE or introduce CCA-secure BE, HVE/IPE to IB-ME. However, implementing the technically-seamless convergence of these technologies to construct a privacy-aware and security-enhanced matchmaking encryption (PS-ME) scheme is incredibly intractable, due to the following challenges.

(I) Constructing such an efficient CCA-secure PS-ME is not simple to unite these methodologies together. For the integration of IB-ME, HVE/IPE and CCA-secure IBBE, it is challenging to create identity-based encryption/secret keys that have no effect on the IBBE-based ciphertext's original structure and identity anonymity. This is because the identity formats in the ciphertexts of each respective primitive are different, *i.e.*, the identity format in IB-ME/IBBE is a string while that in HVE/IPE is an identity vector. Besides, most IB-ME, HVE/IPE and CCA-secure IBBE primitives have serious security issues due to their constructions based on

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3294725

symmetric prime-order groups [36], which further makes them unsuitable for integration. Hence, it is essential to transform these insecure primitives into secure ones. (II) Practically, it is also not trivial to convert them based on symmetric prime groups to those based on asymmetric prime order groups, since any transformation of the public parameters used for the whole construction may lead to the failure of security reduction. For incorporating CCA-secure BE, HVE/IPE to IB-ME, it's also facing the same challenges as that in above (I) & (II). Besides, even though the combination is feasible, the constructed scheme is probably CPA-secure since existing IB-ME solutions are CPA-secure and the CCA-secure IB-ME construction has been an open problem.

C. Our Contributions

In this paper, we design an efficient PS-ME, the first-ever privacy-aware and security-enhanced matchmaking encryption for flexible data sharing. The significant novelties are primarily the following innovations: (I) We observe that an identity-based BE can be exploited with IB-ME to construct a standard flexible identity-based broadcast ME (IB-BME). In which, the proposed general transformation solution innovatively overcomes the previous inflexible issue, *i.e.*, an encryption key can be only used for simultaneously encrypting a set of identities as access policies. This desirable innovation for the first time enables the transformation from one-to-one IB-ME to one-to-many IB-BME. (II) With this transformation approach, we also construct a flexible PS-ME scheme, which in addition to preserving all desirable features of IB-BME further realizes CCA-security and identity anonymity. The main contributions of this paper are as follows:

- *Data confidentiality and authenticity*: To ensure the confidentiality and authenticity of the transmitted data, the proposed IB-BME and PS-ME allow a sender to insert a signature related to his/her identity into the ciphertext, such that the data would not be edited, tampered with, and even replaced.
- *One-to-many flexible no-interactive SH*: To guarantee that the parties participating in the data exchange are the ones intended, both IB-BME and PS-ME empower the participants (*i.e.*, both senders and recipients) to self-specify their respective *one-to-many* access policies to encrypted data, such that the data can be revealed if both policies are satisfied by the counterpart.
- *Identity privacy leakage-resistance*: To block identity privacy leakage, our PS-ME employs the IPE-based broadcasting technique to hide all recipients' identities under an access policy, thus realizing identity anonymity. With the PS-ME, any user cannot infer other recipients' identities from the ciphertext.
- *Efficiency and security assurance*: To provide lightweight data access with stronger security assurance, the PS-ME only takes a few pairing calculations (*i.e.*, constant-level) and can achieve CCA-security against active attacks. Our PS-ME is the first scheme to solve the open problem of CCA-secure ME.

In addition, we present strict security proofs to prove the CPA-secure IB-BME, and CCA-secure PS-ME with identity anonymity. It is worth knowing that our PS-ME solves a long-term open problem of ME posed in CRYPTO'2019. The experimental evaluations (*git@github.com:xuehuan-yang/PSME.git*) are also indicated to show the efficiency of PS-ME.

II. Related Work

Matchmaking encryption (ME) as a new encryption form was proposed by Ateniese *et. al* [20], in which both the sender and the recipient are allowed to designate respective access policies the counterpart must hold in order for the cleartext to be recovered. In a ME [20], a sender with his/her attributes $\sigma \in \{0,1\}^*$ encodes the plaintext after creating the receiver's access policy $\mathbb{R}$ of the intended recipient and a recipient with his/her attributes $\rho \in \{0,1\}^*$ is authorized a decryption key $\mathsf{dk}_{\mathbb{S}}$ before decoding the ciphertext from the sender matching the designated policy $\mathbb{S}$. The ciphertext can be correctly decrypted if and only if the match holds (the sender's attributes $\sigma \in \{0,1\}^*$ matches with the recipient's access policy $\mathbb{S}$ of the recipient and vice-versa). In [20], Ateniese *et. al* also instantiated identity-based matchmaking encryption (IB-ME) in the random oracle model via specifying the sender's and recipient's identities instead of general policies, which enables data authenticity assurance by embedding an encryption key in a ciphertext. Following this work, Francati *et al.* [21] invented the first IB-ME scheme without data authenticity in the standard model, and then exploited the non-interactive zero-knowledge (NIZK) proof technique to formulate the first IB-ME construction with data authenticity in the standard model. Recently, Chen *et al.* [22] put forward an IB-ME without other crypto tools under the standard assumptions in the standard model. Although these IB-ME schemes guarantee data privacy and authenticity, they only realize one-to-one data sharing instead of one-to-many data sharing. In other words, if aiming to realize one-to-many data sharing without harming data privacy and authenticity, the sender must encrypt the same message under the distinct public keys of various recipients with IB-ME. Obviously, this will lead to high communication and computation costs as well as the storage of multiple copies of the same data.

To target one-to-many data sharing while preserving the advantages of IB-ME, Xu *et al.* [23] suggested the first matchmaking attribute-based encryption (MABE) with ABE and ME technologies. In which, both the sender and the recipient are permitted to enforce attribute-based access policies to the encrypted data, such that data can be successfully recovered by the recipients if and only if both access policies are satisfied. Following Xu *et al.*'s work, Sun *et al.* [24] slightly modified the security model and put forward an attribute-based bilateral access control scheme for IoT healthcare. While these MABE works feature fine-grained one-to-many access control over the encrypted data, they also confront construction inefficiency issues, *i.e.,* the computation and communication costs of both encryption and decryption grow linearly with the incremental complexity of access policies. Further, all existing one-to-many ME-related schemes fail to achieve efficient decryption

TABLE I: Property-wise Comparisons among Matchmaking Encryption Schemes

Type of scheme	Data Authenticity	One-to-many Bilateral Access Control	Asymmetric Prime-order Groups	Anonymity	High Efficiency	Security
AFN+ [20]	✓	✗	✗	✗	✓	IND-CPA
FGR+ (I/II) [21]	✗/✓	✗	✓	✗	✓	IND-CPA
CLW+ [22]	✓	✗	✓	✓	✓	IND-CPA
XNL+ [23]	✓	✓	✗	✗	✗	IND-CPA
SYT+ [24]	✓	✓	✗	✗	✗	IND-CPA
IB-BME	✓	✓	✓	✗	✗	IND-CPA
PS-ME	✓	✓	✓	✓	✓	IND/AN/WR-CCA

Note: IND-CPA: Indistinguishability under chosen-plaintext attacks; IND-CCA: Indistinguishability under chosen-ciphertext attacks; AN-CCA: Anonymity-based CCA; WR-CCA: weakly robust against CCA. "✓" indicates that the scheme support this functionality; "✗" signifies the scheme fails to have this functionality.

efficiency, anonymity and CCA security. That is to say, to date, no effort has been capable of simultaneously supporting secure one-to-many bilateral access control, data authenticity, user anonymity, high efficiency, and stronger CCA security assurance.

TABLE. I summarizes the characteristic comparisons among existing related ME works. *Data authenticity* ensures data unforgeability. *One-to-many bilateral access control* implies that both participants specify respective access policies to the encrypted data for realizing non-interactive SH, in particular, the sender enables multiple recipients' data sharing by simply encrypting the data once. *Asymmetric prime-order groups* indicate the secure construction compared to symmetric prime-order groups that have serious security issues [36]. *Anonymity* supports stronger user privacy. *High efficiency* means constant decryption computation costs. From TABLE. I, it is straightforward to see that only the works [21], [22] and our frameworks are securely constructed under asymmetric prime-order groups; only the works [22] and our PS-ME achieve identity anonymity; only the works [23], [24] and our solutions achieve one-to-many bilateral access control; all works except our PS-ME cannot realize CCA (*i.e.*, IND-CCA, AN-CCA and WR-CCA) security. To summarize, only our PS-ME simultaneously realizes data authenticity, one-to-many bilateral access control, anonymity, constant decryption efficiency and CCA security.

III. Preliminaries

In this section, some basic knowledge is introduced including hardness assumptions, system architecture, threat model & design objectives.

A. Hardness Assumptions

***Definition** 1 (ADDH):* Let $\mathcal{BG} = (\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, p, e)$ be a Type-III pair with generators $g \in \mathbb{G}_0$, $h \in \mathbb{G}_1$. Given an ADDH instance $(g, g^{\alpha'}, g^{\beta_2}, g^{\tau\beta}, h, h^{\beta}, h^{\tau\beta}, h^{\tau\beta_2}, g^{1/\tau}, \mathcal{Z} = g^{\alpha'\beta_2+\xi})$, where $\alpha', \beta_2, \tau, \beta \in \mathbb{Z}_p$, the goal of the augmented decisional Diffie-Hellman (ADDH) on $\mathbb{G}_0$ is to decide $\mathcal{Z} = g^{\alpha'\beta_2}$ or $\mathcal{Z}$ is a random element of $\mathbb{G}_0$, *i.e.*, $\mathcal{Z} = g^{\alpha'\beta_2+\xi}$, where ξ is a random value of $\mathbb{Z}_p$.

***Definition** 2 (DDH):* Given a DDH instance $(g, h, h^{\alpha'}, h^{\beta_2}, \mathcal{Z} = h^{\alpha'\beta_2+\xi})$, where $\alpha', \beta_2 \in \mathbb{Z}_p$, $g \in \mathbb{G}_0$, $h \in \mathbb{G}_1$, the goal of the decisional Diffie-Hellman (DDH) on $\mathbb{G}_1$ is to judge $\mathcal{Z} = h^{\alpha'\beta_2}$ or $\mathcal{Z}$ is a random element of $\mathbb{G}_1$, *i.e.*, $\mathcal{Z} = h^{\alpha'\beta_2+\xi}$, where ξ is a random value of $\mathbb{Z}_p$.

***Definition** 3 (DBDH):* Given a tuple $(g, g^a, g^b, g^c, h^a, h, h^b, h^c, \mathcal{Z})$, where $a, b, c \in \mathbb{Z}_p$, $g \in \mathbb{G}_0$, $h \in \mathbb{G}_1$, the goal of the decisional bilinear Diffie-Hellman (DBDH) is to determine $\mathcal{Z} = e(g, h)^{abc}$ or $\mathcal{Z}$ is a random element of $\mathbb{G}_T$.

***Definition** 4 (CBDH):* Given a CBDH tuple $(g, g^a, g^b, g^c, h^a, h, h^b, h^c, \mathcal{Z})$, where $a, b, c \in \mathbb{Z}_p$, $g \in \mathbb{G}_0$, $h \in \mathbb{G}_1$, the goal of the computational bilinear Diffie-Hellman (CBDH) is to output $\mathcal{Z} = e(g, h)^{abc}$.

B. System Architecture for Flexible Data Sharing

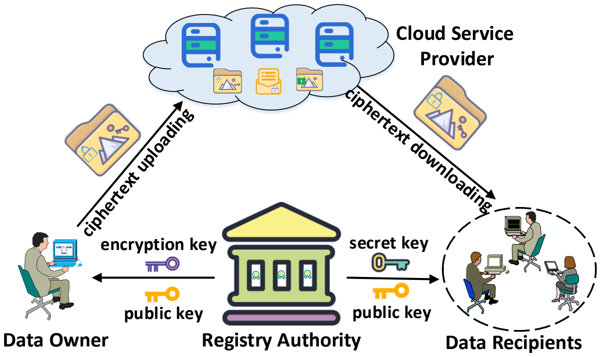


Fig. 1: System architecture of our methodologies

Our system architecture involves four types of entities: registry authority (RA), cloud service provider (CSP), data owner, data recipient. The registry authority working as a key generation center is responsible for generating system public parameters and system master secret keys by running setup (Setup) algorithm. In addition, it produces a decryption key for each data recipient and an encryption key for each data owner by performing decryption key generation (DKGen) and encryption key generation (EKGen) algorithms. For ease of data sharing, a data owner exploits his/her owned encryption key as well as the self-picked access control to encode the data via the encryption (Enc) algorithm and upload the ciphertext to the cloud server for sharing. The cloud service provider offers users infinite cloud storage resources to store encrypted data and responds to the uploading and downloading requests of data owners and recipients. By implementing decryption

(Dec) algorithm, each authorized recipient downloads the intended data ciphertext and verifies-then-decodes the ciphertext with his/her decryption key. (See **Section** IV. A for detailed algorithms).

Remark: There are so many data sharing schemes, for example, the authors [9] proposed a fine-grained hierarchical data sharing (FHDS) scheme, which enables a data owner to encrypt data with his public key, and selectively share encrypted data with users in a hierarchy, thus aiming to solve the inefficiency or inflexibility of data sharing. In [6], the authors for the first time suggested an identity-based encryption transformation (IBET) scheme, which provides a transformation mechanism that converts an IBE ciphertext into an IBBE ciphertext so that a new group of users not specified during the IBE encryption can access the underlying data, thus targeting to address cross-domain data sharing issue. In this paper, we proposed the first privacy-aware and security-enhanced efficient matchmaking encryption (ME) scheme for flexible data sharing, which enables both participants to specify respective access policies to the encrypted data, such that the data can be accessed by multiple recipients in the case that both access policies are satisfied, thus solving the inflexible, insufficiently-secure or inefficient issues of data sharing.

C. Threat Model & Design Objectives

In our PS-BME, we consider four main types of active attacks against our data sharing scenario. To be more specific, (I) any attacker including honest-but-curious CSP and unauthorized users tries to learn the cleartext from a ciphertext without legitimate decryption keys; (II) any malicious attacker who intentionally launches forgery attacks to eavesdrop, forge or replace the raw data plans to undermine the originality of data without valid encryption keys; (III) any adversary regardless of authorized or unauthorized users & CSP attempts to learn other authorized recipients' identities from a ciphertext; (IV) some active attackers who may potentially launch chosen ciphertext attacks (CCA) can modify the transmissive messages. For these real-world existing attacks, the design objectives of our PS-BME are reached as follows:

- *Confidentiality of data.* Only the recipient who owns legitimate decryption keys can recover the encrypted data. In other words, any adversary, including CSP and unauthorized recipients, is inaccessible to the encrypted data if the corresponding decryption keys are incorrect.
- *One-to-many non-interactive SH.* Only the recipient who satisfies both access policies can recover the data. In other words, any adversary cannot access encrypted data if either of the access policy matches fails.
- *Authenticity of data.* Once the ciphertext has been created, any malicious user cannot forge or edit it unless she/he has been granted the legitimate encryption keys of the data encryptor.
- *Identity anonymity.* The access control indicating data recipients' identities is hidden in the ciphertext, any user regardless of whether they are valid recipients or unauthorized users can learn nothing about other recipients' identities from the ciphertext.
- *Active attack resistance.* Although some active adversaries are allowed to implement some modifications of the transmissive data, the security assurance can still be valid unless the CCA security is undermined.

IV. Definitions

This section shows some definitions used for the whole manuscript. In detail, the frameworks of PS-ME are introduced to formalize the definition of our schemes and then the security games are formally defined for the subsequent security proofs.

A. PS-ME Framework

Our PS-ME involves five algorithms, namely, **Setup**, **EKGen**, **SKGen**, **Enc** and **Dec**. To be more specific,

- **Setup**$(\lambda) \rightarrow (\mathsf{pp}, \mathsf{msk})$: The setup algorithm is performed by a trusted registry authority. With the input security parameter λ, it produces the public parameter pp and the master secret key msk.
- **EKGen**$(\mathsf{msk}, \mathsf{id}^*) \rightarrow \mathsf{ek}_{\mathsf{id}^*}$: The encryption key algorithm is also conducted by the trusted registry authority. Based on msk and an identity id^*, it produces an encryption key $\mathsf{ek}_{\mathsf{id}^*}$.
- **DKGen**$(\mathsf{msk}, \mathsf{id}) \rightarrow \mathsf{dk}_{\mathsf{id}}$: The decryption key algorithm is also implemented by the trusted registry authority. With the input msk and an identity id, it returns a decryption key $\mathsf{dk}_{\mathsf{id}}$.
- **Enc**$(\mathsf{pp}, \mathcal{S}, \mathsf{ek}_{\mathsf{id}^*}, m) \rightarrow \mathsf{ct}$: The encryption algorithm is run by a data owner. Given pp, a target identity set $\mathcal{S}$, an encryption key $\mathsf{ek}_{\mathsf{id}^*}$ and the plaintext m, it generates a ciphertext ct.
- **Dec**$(\mathsf{pp}, \mathsf{dk}_{\mathsf{id}_i}, \mathsf{id}^*, \mathsf{ct}) \rightarrow m/\perp$: The decryption algorithm is carried by a data recipient. Based on the public parameter pp, a decryption key $\mathsf{dk}_{\mathsf{id}_i}$, the target identity id^* and the ciphertext ct, it outputs m if the decryption key is valid; otherwise, it outputs $\perp$.

The PS-ME is *sound* if each entity faithfully performs the scheme. Namely, for any ciphertext $\mathsf{ct} \leftarrow$ **Enc**$(\mathsf{pp}, \mathcal{S}, \mathsf{ek}_{\mathsf{id}^*}, m)$ and any decryption key $\mathsf{dk}_{\mathsf{id}} \leftarrow$ **DKGen**$(\mathsf{msk}, \mathsf{id})$, where $\mathsf{ek}_{\mathsf{id}^*} \leftarrow$ **EKGen**$(\mathsf{msk}, \mathsf{id}^*)$, and $(\mathsf{pp}, \mathsf{msk}) \leftarrow$ **Setup**(λ), then the decryption algorithm can always output the plaintext $m \leftarrow$ **Dec**$(\mathsf{pp}, \mathsf{dk}_{\mathsf{id}_i}, \mathsf{id}^*, \mathsf{ct})$.

B. Formal Security Definitions

We define indistinguishability-based chosen ciphertext attacks (IND-CCA), which ensures that even if all clients outside of authorized receivers $\mathcal{S}$ collude, they can learn nothing about the plaintext information. Besides, we consider the anonymity-based CCA (AN-CCA) security definition, which means that any client even if she/he is authorized cannot learn the identities from ciphertext embedding the authorized identity set. We also give the security definition of weakly robust against CCA (WR-CCA), which ensures that if the "invalid" decryption key is utilized, then the plaintext recovery attempts would fail. We finally show the security definition of privacy and authenticity, which ensures the unforgeability for malicious clients.

1) The IND-CCA security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ for our PS-ME is defined as follows:

- **Setup**: $\mathcal{C}$ conducts $(\mathsf{pp}, \mathsf{msk}) \leftarrow$ **Setup**(λ), afterwards issues pp to $\mathcal{A}$ and stores msk in his/her hands.
- **Phase 1**: The following queries are adaptively made by $\mathcal{A}$:
  - Encryption key query: On input id', $\mathcal{C}$ first runs $\mathsf{ek}_{\mathsf{id}'} \leftarrow$ **EKGen**$(\mathsf{msk}, \mathsf{id}')$ and sends $\mathsf{ek}_{\mathsf{id}'}$ to $\mathcal{A}$.
  - Secret key query: On input id, $\mathcal{C}$ first performs $\mathsf{dk}_{\mathsf{id}} \leftarrow$ **DKGen**$(\mathsf{msk}, \mathsf{id})$ and gives $\mathsf{dk}_{\mathsf{id}}$ to $\mathcal{A}$.
  - Decryption query: On input id and ct, $\mathcal{C}$ first implements $m \leftarrow$ **Dec**$(\mathsf{pp}, \mathsf{dk}_{\mathsf{id}}, \mathsf{id}', \mathsf{ct})$ and returns m to $\mathcal{A}$, where $\mathsf{dk}_{\mathsf{id}} \leftarrow$ **DKGen**$(\mathsf{msk}, \mathsf{id})$.
- **Challenge**: Two equal-length plaintexts m_0, m_1 and an identity set $\mathcal{S}^*$ are picked and submitted to $\mathcal{C}$. Here, $\mathcal{A}$ has not made the secret key query on $\mathsf{id} \in \mathcal{S}^*$. Then, $\mathcal{A}$ randomly flips a coin ξ and sends the produced challenge ciphertext **Enc**$(\mathsf{pp}, \mathcal{S}^*, \mathsf{ek}_{\mathsf{id}'}, m_\xi)$.
- **Phase 2**: The queries can be continually made in an adaptive manner as that in **Phase 1**, but the following restrictions are regulated: 1) $\mathcal{A}$ can not issue the secret key query on id, such that $\mathsf{id} \in \mathcal{S}^*$; 2) $\mathcal{A}$ can not issue the decryption query on $(\mathsf{id}, \mathsf{ct}^*)$, where $\mathsf{id} \in \mathcal{S}^*$;
- **Guess**: $\mathcal{A}$ submits a guess $\xi' \in \{0, 1\}$.

***Definition** 5:* If the advantage of $\mathcal{A}$ in winning the IND-CCA game is negligible, then the PS-ME is IND-CCA secure. The $\mathcal{A}$'s advantage is defined as $\mathcal{A}dv^{\mathsf{IND\text{-}CCA}}_{\mathcal{A},\mathsf{PS\text{-}ME}} = |\mathsf{Pr}[\xi' = \xi] - 1/2|$.

2) The AN-CCA security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ for our PS-ME is formalized as follows:

- **Setup**: It is the same as that in the IND-CCA game.
- **Phase 1**: It is also identical to that in the IND-CCA game.
- **Challenge**: $\mathcal{A}$ gives a plaintext m and two various identity sets $\mathcal{S}_0, \mathcal{S}_1$ with equal-length. This phase also requires that $\mathcal{A}$ has not made queries of extraction query on id such that $\mathsf{id} \in \mathcal{S}_0 \cup \mathcal{S}_1 - \mathcal{S}_0 \cap \mathcal{S}_1$. $\mathcal{C}$ then picks a random coin ξ and sends $\mathcal{A}$ the created challenge ciphertext $\mathsf{ct} \leftarrow$ **Enc**$(\mathsf{pp}, \mathcal{S}_\xi, \mathsf{id}', m)$.
- **Phase 2**: The queries can be continually made in an adaptive manner as that in **Phase 1**, but the following restrictions are required: 1) $\mathcal{A}$ can not issue the secret key query on id, where $\mathsf{id} \in \mathcal{S}_0 \cup \mathcal{S}_1 - \mathcal{S}_0 \cap \mathcal{S}_1$; 2) $\mathcal{A}$ can not issue the decryption query on $(\mathsf{id}, \mathsf{ct}^*)$, where $\mathsf{id} \in \mathcal{S}_0 \cup \mathcal{S}_1 - \mathcal{S}_0 \cap \mathcal{S}_1$;
- **Guess**: $\mathcal{A}$ gives a guess $\xi' \in \{0, 1\}$.

***Definition** 6:* If the advantage of $\mathcal{A}$ in winning the AN-CCA game is negligible, then the PS-ME is AN-CCA secure. The $\mathcal{A}$'s advantage is denoted as $\mathcal{A}dv^{\mathsf{AN\text{-}CCA}}_{\mathcal{A},\mathsf{PS\text{-}ME}} = |\mathsf{Pr}[\xi' = \xi] - 1/2|$.

3) The WR-CCA security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ for our PS-ME is described as follows:

- **Setup**: It is the same as that in the IND-CCA game.
- **Query Phase**: It is also identical to the **Phase 1** in the IND-CCA game.
- **Output**: $\mathcal{A}$ gives a plaintext m and two various identity sets $\mathcal{S} = \{\mathsf{id}_1, \dots, \mathsf{id}_t\}$, $\mathcal{C}$ sends $\mathcal{A}$ the generated challenge ciphertext $\mathsf{ct}^* \leftarrow$ **Enc**$(\mathsf{pp}, \mathcal{S}^*, \mathsf{id}', m)$.

If **Dec**$(\mathsf{pp}, \mathsf{dk}_{\mathsf{id}^*}, \mathsf{id}', \mathsf{ct}^*) \neq \perp$, where $\mathsf{id}^* \in \mathcal{S}^*$ and $\mathsf{dk}_{\mathsf{id}^*} \leftarrow$ **DKGen**$(\mathsf{msk}, \mathsf{id}^*)$, then we can call that $\mathcal{A}$ wins this game. Here, it is also needed that the decryption key query on id^* in **Query Phase** had not been queried.

***Definition** 7:* If the advantage of $\mathcal{A}$ in winning the WR-CCA game is negligible, then the PS-ME is WR-CCA secure.

4) The privacy and authenticity security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ for our PS-ME is described as follows:

- **Setup**: $\mathcal{C}$ sends $\mathcal{A}$ the public parameter $\mathsf{pp} \leftarrow$ **Setup**(1^λ). Note that in this phase $\{\mathcal{H}_i\}_{i \in [0,2]}$ are all random oracles used in this game.
- **Query phase**: The following queries can be adaptively issued by $\mathcal{A}$:
  - Encryption key query: With the input id' to the oracle of $\mathcal{H}_1$, $\mathcal{C}$ sends $\mathcal{A}$ the produced encryption key $\mathsf{ek}_{id'} \leftarrow$ **EKGen**$(\mathsf{msk}, \mathsf{id}')$.
  - Secret key query: With the input id to the oracle of $\mathcal{H}_0$, $\mathcal{C}$ sends the produced decryption key $\mathsf{dk}_{id} \leftarrow$ **SKGen**$(\mathsf{msk}, \mathsf{id})$ to $\mathcal{A}$.
- **Forgery**: $\mathcal{A}$ sends $(\mathsf{ct}, \mathsf{id}, \mathsf{id}')$ to $\mathcal{C}$. In response, $\mathcal{C}$ outputs the result of CBDH assumption as the result of $\mathcal{A}$.

***Definition** 8:* If the PS-ME can achieve privacy and authenticity, then we say it is secure.

Remark: Here we omit the CPA-security definitions for IB-BME due to the space limits, and their similarity with CCA-security definitions except that there are no decryption queries in the security games of IB-BME.

V. Concrete Construction of IB-BME

In this section, we first propose an IB-BME and then show the soundness and security proofs to demonstrate its correctness and CPA security.

A. *Identity-based Broadcast ME (IB-BME)*

- **Setup**(λ, ℓ): With the input security parameter λ and the maximum legitimate identity set ℓ, it first picks a bilinear group $\mathcal{BG} = (\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, p, e)$ with three random generators $g, v \in \mathbb{G}_0$ and $h \in \mathbb{G}_1$. Next, it chooses random $(\ell + 1)$-dimensional vectors from $\mathbb{Z}_p$ with $\vec{r_1} = (r_{1,0}, \dots, r_{1,\ell})$ and $\vec{r_2} = (r_{2,0}, \dots, r_{2,\ell})$. It also picks $t_1, t_2, \beta_1, \beta_2, \alpha, \rho \in \mathbb{Z}_p$, $b, \tau \in \mathbb{Z}_p^*$, sets $\vec{r} = \vec{r_1} + b\vec{r_2} = (r_0, \dots, r_\ell)$, $t = t_1 + bt_2$, $\beta = \beta_1 + b\beta_2$ and calculates $R = g^{\vec{r}} = (g^{r_0}, \dots, g^{r_\ell})$, $T = g^t$, $e(g, h)^\beta$. Then, it selects the following hash functions $\mathcal{H}_0 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathbb{G}_0$, $\mathcal{H}_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $\mathcal{H}_3 : \mathbb{G}_T \rightarrow \mathbb{Z}_p$. Finally, it publishes the public parameter $\mathsf{pp} = (\mathcal{BG}, v, v^\rho, g, g^b, R, T, e(g,h)^\beta, h, h^{\vec{r_1}}, h^{\vec{r_2}}, h^{t_1}, h^{t_2}, g^{\tau\beta}, h^{\tau\beta_1}, h^{\tau\beta_2}, h^{1/\tau}, \{\mathcal{H}_i\}_{i \in [0,3]})$ and stores the master secret key $\mathsf{msk} = (h^{\beta_1}, h^{\beta_2}, \alpha, \rho)$.
- **EKGen**$(\mathsf{msk}, \mathsf{id}^*)$: Based on msk and identity id^*, it produces an encryption key $\mathsf{ek}_{\mathsf{id}^*} = \mathcal{H}_1(\mathsf{id}^*)^\alpha$.
- **DKGen**$(\mathsf{msk}, \mathsf{id})$: With the input msk and identity id, it first selects $z \in \mathbb{Z}_p$, random tags $\mathsf{rtag}_1, \dots, \mathsf{rtag}_\ell$ and returns a decryption key $\mathsf{dk}_{\mathsf{id}} =$

$(\mathsf{dk}_1, \mathsf{dk}_2, \mathsf{dk}_3, \mathsf{dk}_4, \mathsf{dk}_5, \mathsf{dk}_6, \{\mathsf{dk}_{7,j}, \mathsf{dk}_{8,j}, \mathsf{rtag}_j\}_{j=1}^{\ell})$, where $\mathsf{dk}_1 = \mathcal{H}_0(\mathrm{id})^{\rho}$,$\mathsf{dk}_2 = \mathcal{H}_0(\mathsf{id})^{\alpha}$, $\mathsf{dk}_3 = \mathcal{H}_0(\mathsf{id})$, $\mathsf{dk}_4 = h^{\beta_1}(h^{t_1})^z$, $\mathsf{dk}_5 = h^{\beta_2}(h^{t_2})^z$, $\mathsf{dk}_6 = h^z$, $\mathsf{dk}_{7,j} = ((h^{t_1})^{\mathsf{rtag}_j} h^{r_{1,j}}/(h^{r_{1,0}})^{(\mathcal{H}_2(\mathsf{id}))^j})^z$, $\mathsf{dk}_{8,j} = ((h^{t_2})^{\mathsf{rtag}_j} h^{r_{2,j}}/(h^{r_{2,0}})^{(\mathcal{H}_2(\mathsf{id}))^j})^z$.

- **Enc**(pp, $\mathcal{S}$, $\mathsf{ek}_{\mathsf{id}^*}$, m): Given pp, a target identity set $\mathcal{S}$ with its length $n \leq \ell$, an encryption key identity $\mathsf{ek}_{\mathsf{id}^*}$ and the plaintext m, it first defines an identity vector $\vec{y} = (y_0, \ldots, y_n, \ldots, y_\ell)$, where y_i is the coefficients from $f(x) = \prod_{\mathsf{id}_j \in \mathcal{S}}(x - \mathcal{H}_2(\mathsf{id}_j)) = \sum_{i=0}^{n-1} y_i x^i + x^n$. Here please note that if $n < \ell$, $y_{n+1} = \ldots = y_\ell = 0$. It next picks $s, d_2, \mathsf{ctag} \in \mathbb{Z}_p$ and computes $C_0 = m \cdot e(g,h)^{\beta s}$, $C_1 = g^s$, $C_2 = g^{bs}$, $C_3 = (T^{\mathsf{ctag}} \prod_{i=0}^{n}(g^{r_i})^{y_i})^{d_2 s}$, $C_4 = v^s$. For each $\mathsf{id}_i \in \mathcal{S}$, it sets $\mathsf{V}_{\mathsf{id}_i} = \mathcal{H}_3(e(\mathcal{H}_0(\mathsf{id}_i), ek_{\mathsf{id}^*} \cdot g^{bs} \cdot v^{\rho s}))$, $g(y) = \prod_{k=1}^{n}(y - \mathsf{V}_{\mathsf{id}_i)} + d_2 = \sum\limits_{k=0}^{n-1} b_k y^k + y^n \mod p$, where $b_0, \ldots, b_n, \ldots, b_\ell$ are the coefficients correspond to y^k. Finally, it generates a ciphertext $\mathsf{ct} = (C_0, C_1, C_2, C_3, C_4, \mathsf{ctag}, b_0, \ldots, b_n)$.
- **Dec**(pp, $\mathcal{S}$, $\mathsf{dk}_{\mathrm{id}_i}$, id^*, ct): Based on the public parameter pp, $\mathcal{S}$, a decryption key $\mathsf{dk}_{\mathrm{id}_i}$, the target identity id^* and the ciphertext $\mathsf{ct} = (C_1, C_2, C_3, b_0, \ldots, b_n)$, it first computes $\mathsf{V}(\mathsf{id}_i) = \mathcal{H}_3(e(\mathsf{dk}_{i,3}, C_2)e(\mathsf{dk}_{i,2}, \mathcal{H}_1(\mathsf{id}^*))e(\mathsf{dk}_{i,1}, C_4)) = \mathcal{H}_3(e(\mathcal{H}_0(\mathsf{id}_i), ek_{\mathsf{id}^*} \cdot g^{bs} \cdot v^{\rho s}))$, $d_2 = g(\mathsf{V}_{\mathsf{id}_i}) = \sum_{j=0}^{n-1} b_j(\mathsf{V}_{\mathsf{id}_i})^j + (\mathsf{V}_{\mathsf{id}_i})^n \mod p$. It next calculates $\mathsf{rtag} = \sum_{i=1}^{\ell} y_i \mathsf{rtag}_i$, if $\mathsf{rtag} = \mathsf{ctag}$, it aborts and outputs $\perp$; otherwise, it computes $\mathsf{A} = (e(C_1, \prod_{j=1}^{\ell} \mathsf{dk}_{7,j}^{y_j})e(C_2, \prod_{j=1}^{\ell} \mathsf{dk}_{8,j}^{y_j})/e(C_3^{1/d_2}, \mathsf{dk}_6))$, $\mathsf{B} = e(C_1, \mathsf{dk}_4) \cdot e(C_2, \mathsf{dk}_5)$ and recovers $m = C_0 \cdot \mathsf{A}^{1/(\mathsf{rtag}-\mathsf{ctag})} \cdot \mathsf{B}^{-1}$.

B. Soundness and Security Proofs of IB-BME

***Theorem** 1:* If a user holds the authorized decryption key, then he/she can perform the successful decryption.

Proof: For a valid ciphertext $\mathsf{ct} = (C_0, C_1, C_2, C_3, C_4, \mathsf{ctag}, b_0, \ldots, b_n)$, an authorized user id_i who has the secret key $\mathsf{dk}_{\mathrm{id}} = (\mathsf{dk}_{\mathrm{id},1}, \mathsf{dk}_{\mathrm{id},2}, \mathsf{dk}_{\mathrm{id},3}, \mathsf{dk}_{\mathrm{id},4}, \mathsf{dk}_{\mathrm{id},5}, \mathsf{dk}_{\mathrm{id},6}, \{\mathsf{dk}_{\mathrm{id},7,j}, \mathsf{dk}_{\mathrm{id},8,j}, \mathsf{rtag}_j\}_{j=1}^{\ell})$ can conduct the following calculations to recover the plaintext m:

1) The authorized user first computes

$$\begin{aligned}\prod_{i=1}^{\ell}(\mathsf{dk}_{7,j})^{y_i} &= \frac{((h^{t_1})^{\sum_{i=1}^{\ell} \mathsf{rtag}_i y_i} h^{\sum_{i=1}^{\ell} r_{1,i} y_i})^z}{((h^{r_{1,0}})^{\sum_{i=1}^{\ell} y_i (\mathcal{H}_2(\mathsf{id})^i)})^z}\\ &= ((h^{t_1 \mathsf{rtag}}) h^{\sum_{i=1}^{\ell} r_{1,i} y_i}/(h^{-r_{1,0} y_0}))^z\\ &= ((h^{t_1 \mathsf{rtag}}) h^{\sum_{i=0}^{\ell} r_{1,i} y_i})^z,\end{aligned}$$

$$\prod_{i=1}^{\ell}(\mathsf{dk}_{8,j})^{y_i} = ((h^{t_2 \mathsf{rtag}}) h^{\sum_{i=0}^{\ell} r_{2,i} y_i})^z,$$

$$\begin{aligned}\mathsf{A} &= (e(C_1, \prod_{j=1}^{\ell} \mathsf{dk}_{7,j}^{y_j})e(C_2, \prod_{j=1}^{\ell} \mathsf{dk}_{8,j}^{y_j})/e(C_3^{1/d_2}, \mathsf{dk}_6)\\ &= e(((h^{t_1 \mathsf{rtag}}) h^{\sum_{i=0}^{\ell} r_{1,i} y_i})^z, g^s) \cdot e(((h^{t_2 \mathsf{rtag}})^z \cdot\\ &\quad (h^{\sum_{i=0}^{\ell} r_{2,i} y_i})^z, g^{bs}) \cdot e(h^z, (T^{\mathsf{ctag}} \prod_{i=0}^{n}(g^{r_i})^{y_i})^{-s})\\ &= e(g,h)^{zst(\mathsf{rtag}\text{-}\mathsf{ctag})},\end{aligned}$$

$$\begin{aligned}\mathsf{B} &= e(C_1, \mathsf{dk}_4) \cdot e(C_2, \mathsf{dk}_5)\\ &= e(g^s, h^{\beta_1}(h^{t_1})^z) \cdot e(h^{\beta_2}(h^{t_2})^z, g^{bs})\\ &= e(g,h)^{s\beta} e(g,h)^{zst}.\end{aligned}$$

2) He/she recovers the plaintext $m = C_0 \cdot \mathsf{A}^{1/(\mathsf{rtag}\text{-}\mathsf{ctag})} \mathsf{B}^{-1}$.

***Theorem** 2:* Assume that ADDH and DDH assumptions hold, then our IB-BME realizes adaptively CPA security.

Before proving the IB-BME security, the semi-functional ciphertexts and secret keys for the normal ciphertexts and secret keys are described in the following. It should be noted that these algorithms replacing the previous algorithms are only used for security proofs, but not in the real construction. This theorem proof can be formally proven via the sequence of hybrid games with the dual system methodology. In the following, we let $(C_1, C_2, C_3, C_4, \mathsf{ctag}, b_0, \ldots, b_n)$ and $e(g,h)^{\beta s} \cdot e(g^{\alpha'}, h^{\beta_1})$ represent the semi-functional headers and secret keys.

- **SF.Enc**(pp, $\mathcal{S}$, $\mathsf{ek}_{\mathsf{id}^*}$, m, $g^{\vec{r_1}}$, g^t): The **Enc** is first performed to produce (C_0', C_1', C_2', C_3'). Then, it selects $\alpha' \in \mathbb{Z}_p$ and sets $C_0 = C_0' e(g^{\alpha'}, h^{\beta_1})$=$m \cdot e(g,h)^{\beta s} \cdot e(g^{\alpha'}, h^{\beta_1})$, $C_1 = C_1' \cdot g^{\alpha'}$, $C_2 = C_2'$, $C_3 = C_3' \cdot g^{\alpha'(\langle \vec{y}, \vec{r} \rangle) + \mathsf{ctag} \cdot t)}$, $C_4 = C_4'$, where $\vec{y} = (y_0, \ldots, y_\ell)$. The resulting ciphertext is $\mathsf{ct} = (C_0, C_1, C_2, C_3, C_4, \mathsf{ctag}, b_0, \ldots, b_n)$.
- **SF.DKGen**(pp,msk, id, $h^{1/b}$): The **DKGen** is conducted to create $\mathsf{dk}_{\mathrm{id}}' = (\mathsf{dk}_1', \mathsf{dk}_2', \mathsf{dk}_3', \mathsf{dk}_4', \mathsf{dk}_5', \mathsf{dk}_6', \{\mathsf{dk}_{7,j}', \mathsf{dk}_{8,j}', \mathsf{rtag}_j\}_{j=1}^{\ell})$. Then, it next chooses a random $\gamma \in \mathbb{Z}_p$ and sets $\mathsf{dk}_1 = \mathsf{dk}_1'$, $\mathsf{dk}_2 = \mathsf{dk}_2'$, $\mathsf{dk}_3 = \mathsf{dk}_3'$, $\mathsf{dk}_4 = \mathsf{dk}_4' \cdot h^{\gamma}$, $\mathsf{dk}_5 = \mathsf{dk}_5'/h^{\gamma/b}$, $\mathsf{dk}_6 = \mathsf{dk}_6'$, $\mathsf{dk}_{7,j} = \mathsf{dk}_{7,j}'$, $\mathsf{dk}_{8,j} = \mathsf{dk}_{8,j}'$. The resulting decryption key $\mathsf{dk}_{\mathrm{id}} = (\mathsf{dk}_1, \mathsf{dk}_2, \mathsf{dk}_3, \mathsf{dk}_4, \mathsf{dk}_5, \mathsf{dk}_6,\ \{\mathsf{dk}_{7,j}, \mathsf{dk}_{8,j}, \mathsf{rtag}_j\}_{j=1}^{\ell})$.

The following descriptions are the sequence of hybrid games between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

- $\mathsf{Game}_{\mathsf{Real}}$: Our IB-BME security game almost follows the adaptive security model of [34].
- Game_0: It is almost identical to $\mathsf{Game}_{\mathsf{Real}}$ except the semi-functional headers and semi-secret keys.
- Game_i: It is almost the same as Game_0 except the first i semi-functional secret keys for $1 \leq i \leq q$, where q denotes the number of secret key queries.
- $\mathsf{Game}_{\mathsf{Final}}$: It is almost identical to Game_Q except that the challenge secret key is a randomness of $\mathbb{G}_T$.

The security proofs of this theorem can be proved via the indistinguishability of each game presented above from the next. Specifically, we first prove the indistinguishability of $\mathsf{Game}_{\mathsf{Real}}$ and Game_0, where $\mathcal{A}$'s advantage is determined by the advantage in solving the DDH hard problem. Then, we demonstrate that Game_{i-1} and Game_i cannot be distinguished from each other, where the $\mathcal{A}$'s advantage is decided by the advantage in solving the DDH assumption. Here, we need to notice that in Game_q, the challenge ciphertext headers and secret keys are all semi-functional, hence any secret keys are not useful for decoding the header parts. Finally, the Game_q and $\mathsf{Game}_{\mathsf{Final}}$ is proved to be indistinguishable under the ADDH assumption.

Lemma 1: Suppose that there is an adversary $\mathcal{A}$ with the overwhelming advantage $\epsilon = |\mathcal{A}dv_{\mathcal{A},\textsf{IB-BME}}^{\textsf{Game}_{\textsf{Real}}} - \mathcal{A}dv_{\mathcal{A},\textsf{IB-BME}}^{\textsf{Game}_0}|$ differentiating $\textsf{Game}_{\textsf{Real}}$ and $\textsf{Game}_0$, then another algorithm $\mathcal{C}_0$ can be created with the same advantage ϵ to successfully solve the DDH assumptions.

Lemma 2: Assume that there is an adversary $\mathcal{A}$ with the non-negligible advantage $\epsilon = |\mathcal{A}dv_{\mathcal{A},\textsf{IB-BME}}^{\textsf{Game}_{i-1}} - \mathcal{A}dv_{\mathcal{A},\textsf{IB-BME}}^{\textsf{Game}_i}|$ distinguishing $\textsf{Game}_i$ and $\textsf{Game}_{i-1}$, where $i \in [1, q]$, then another algorithm $\mathcal{C}_i$ can be invented with the same advantage ϵ to successfully address the DDH assumptions.

Lemma 3: If an adversary $\mathcal{A}$ can distinguish $\textsf{Game}_q$ and $\textsf{Game}_{\textsf{Final}}$ with the non-negligible advantage $\epsilon = |\mathcal{A}dv_{\mathcal{A},\textsf{IB-BME}}^{\textsf{Game}_q} - \mathcal{A}dv_{\mathcal{A},\textsf{IB-BME}}^{\textsf{Game}_{\textsf{Final}}}|$, then another algorithm $\mathcal{C}$ can be built with the same advantage ϵ to break the ADDH assumption.

Proof: We can similarly prove the ***Lemmas 1 & 2*** as the way in [35]. Due to space limitations, we omit the security proofs of these two lemmas. Here, we only show the ***Lemma 3*** proof since it is the most significant part of this theorem. Given an ADDH instance $(g, g^{\alpha'}, g^{\beta_2}, g^{\tau\beta}, h, h^{\beta}, h^{\tau\beta}, h^{\tau\beta_2}, g^{1/\tau}, \mathcal{Z} = g^{\alpha'\beta_2+\xi})$ to $\mathcal{C}$, the goal of $\mathcal{C}$ is to determine $\xi = 0$ or ξ is a randomness of $\mathbb{Z}_p^*$.

- **Setup**: $\mathcal{C}$ randomly chooses two $(\ell+1)$-dimensional vectors $\vec{r_1} = (r_{1,0}, \ldots, r_{1,\ell})$, $\vec{r_2} = (r_{2,0}, \ldots, r_{2,\ell})$, $t_1, t_2, \rho \in \mathbb{Z}_p$, $v \in \mathbb{G}_0$, $b \in \mathbb{Z}_p^*$ and creates the public parameter $\textsf{pp}$: $g = g, g^b, v, v^\rho, R = g^{\vec{r_1}+b\vec{r_2}}, g^{t_1+bt_2}, e(g, h^\beta)$, $h = h, h^{\vec{r_1}}, h^{\vec{r_2}}, h^{t_1}, h^{t_2}, g^{\tau\beta}, h^{\tau\beta_1} = h^{\tau\beta}/(h^{\tau\beta_2})^b, h^{1/\tau}$. Please note that $\beta_1 = \beta - b\beta_2$ is implicitly set.
- **Phases 1 & 2**: After receiving the secret key queries from $\mathcal{A}$ for an identity $\textsf{id} \in \{0,1\}^*$, $\mathcal{C}$ chooses $z, \alpha, \gamma', \textsf{rtag}_1, \ldots, \textsf{rtag}_\ell \in \mathbb{Z}_p$, where $\gamma = \gamma' + b\beta_2$ is also implicitly set and creates the semi-functional key as $\textsf{dk}_1 = \mathcal{H}_0(\textsf{id})^\rho$,$\textsf{dk}_2 = \mathcal{H}_0(\textsf{id})^\alpha$, $\textsf{dk}_3 = \mathcal{H}_0(\textsf{id})$, $\textsf{dk}_4 = h^{\beta_1}(h^{t_1})^z h^{\gamma'}$, $\textsf{dk}_5 = h^{\beta_2}(h^{t_2})^z/h^{\gamma b^{-1}} = (h^{t_2})^z h^{\gamma' b^{-1}}$, $\textsf{dk}_6 = h^z$, $\textsf{dk}_{7,j} = (h^{t_1})^{\textsf{rtag}_j} h^{r_{1,j}}/(h^{r_{1,0}})^{(\mathcal{H}_2(\textsf{id}))^j}$, $\textsf{dk}_{8,j} = (h^{t_2})^{\textsf{rtag}_j} h^{r_{2,j}}/(h^{r_{2,0}})^{(\mathcal{H}_2(\textsf{id}))^j}$. For the encryption key query on identity $\textsf{id}^*$, it produces an encryption key $\textsf{ek}_{\textsf{id}^*} = \mathcal{H}_1(\textsf{id}^*)^\alpha$.
- **Challenge**: $\mathcal{A}$ issues a challenge identity set $\mathcal{S}^* = \textsf{id}_1, \ldots, \textsf{id}_n$ and two equal-length messages m_0, m_1, $\mathcal{C}$ can easily get an access vector $\vec{y} = (y_0, \ldots, y_\ell)$ according to the given challenge identity set and then randomly chooses $s, d_2, \textsf{ctag} \in \mathbb{Z}_p$ and computes $C_0 = m_\xi \cdot e(g, h)^{\beta s} \cdot e(g^{\alpha'}, h^\beta)/e(\mathcal{Z}, h^b)$, $C_1 = g^s \cdot h^{\alpha'}$, $C_2 = g^{bs}$, $C_3 = (T^{\textsf{ctag}} \prod_{i=0}^n (g^{r_i})^{y_i})^{d_2 s} \cdot g^{\alpha'(\langle\vec{y},\vec{r}\rangle)+\textsf{ctag}\cdot t)}$, $C_4 = v^s$. For each $\textsf{id}_i \in \mathcal{S}$, it sets $\textsf{V}_{\textsf{id}_i} = \mathcal{H}_3(e(\mathcal{H}_0(\textsf{id}_i), ek_{\textsf{id}^*} \cdot g^{bs} \cdot v^{\rho s}))$, $g(y) = \prod_{k=1}^n (y - \textsf{V}_{\textsf{id}_i)} + d_2 = \sum\limits_{k=0}^{n} b_k y^k \mod p$, where $b_0, \ldots, b_n, \ldots, b_\ell$ are the coefficients correspond to y^k.
- **Guess**: $\mathcal{A}$ gives a guess ξ' of ξ, $\mathcal{C}$ then returns 0 to guess $\mathcal{Z} = g^{\alpha'\beta_2}$ if $\xi' = \xi$; otherwise, it returns 1 indicating $\mathcal{Z}$ is a random value of $\mathbb{G}_0$. As well, the $\textsf{Game}_q$ is simulated by $\mathcal{C}$ if $\xi = 0$ and $\textsf{Game}_{\textsf{Final}}$ is simulated if ξ is a random value of $\mathbb{Z}_p$. Hence, $\mathcal{A}$'s output can be as the result of $\mathcal{C}$ to distinguish $\mathcal{Z} = g^{\alpha'\beta_2}$, thus determining $\textsf{Game}_q$ and $\textsf{Game}_{\textsf{Final}}$.

VI. PS-ME Construction

In this section, we design an efficient privacy-aware and security-enhanced ME (PS-ME). In our PS-ME, we follow the basic framework of the ME scheme and use the general matchmaking transformation solution shown in the Enc of Section V to realize one-to-many matchmaking. Here, we simply introduce the notations throughout the PS-ME. For two strings a, b, let $[a]_x$ and $[b]^y$ respectively denote the first x bits of a and the last y bits of b. $x||y$ denotes the connection of a with b.

A. Privacy-aware and Security-enhanced ME (PS-ME)

- **Setup**(λ): With the input security parameter λ, it first picks and sets a bilinear group $\mathcal{BG} = (\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, p, e)$, where the bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_1 \to \mathbb{G}_T$ holds and p is the prime order of groups $(\mathbb{G}_0, \mathbb{G}_1)$. Next, it randomly picks a generator $g \in \mathbb{G}_0$, generators $h, u, v, w \in \mathbb{G}_1$, $\alpha, \beta, \rho \in \mathbb{Z}_p$ and calculates $g_1 = g^\rho, h_0 = h^\rho, h_1 = h^\beta$. Then, it selects the following collision-resistant hash functions $\mathcal{H}_0 : \{0,1\}^* \to \mathbb{G}_0$, $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1$, $\mathcal{H}_2 : \mathbb{G}_T \to \mathbb{Z}_p$, $\mathcal{H}_3 : \mathbb{Z}_p^2 \times \mathbb{G}_0 \times \mathbb{G}_1^2 \to \{0,1\}_1^\ell$, $\mathcal{H}_4 : \mathbb{G}_0 \times \mathbb{G}_1^2 \times \{0,1\}^\ell \times \mathbb{Z}_p^{2t} \to \mathbb{Z}_p$. Finally, it publishes the public parameter $\textsf{pp} = (\mathcal{BG}, g, g_1, u, v, w, h, h_0, h_1, \{\mathcal{H}_i\}_{i\in[0,4]})$ and stores the master secret key $\textsf{msk} = (\rho, \alpha)$.
- **EKGen**($\textsf{msk}$, $\textsf{id}^*$): Based on $\textsf{msk}$ and identity $\textsf{id}^*$, it produces an encryption key $\textsf{ek}_{\textsf{id}^*} = \mathcal{H}_1(\textsf{id}^*)^\alpha$.
- **DKGen**($\textsf{msk}$, $\textsf{id}_i$): With the input $\textsf{msk}$ and an identity $\textsf{id}_i$, it returns a decryption key $\textsf{dk}_{\mathrm{id}_i} = (\textsf{dk}_{i,1}, \textsf{dk}_{i,2}, \textsf{dk}_{i,3})$, where $\textsf{dk}_{i,1} = \mathcal{H}_0(\mathrm{id}_i)^\rho, \textsf{dk}_{i,2} = \mathcal{H}_0(\textsf{id}_i)^\alpha, \textsf{dk}_{i,3} = \mathcal{H}_0(\textsf{id}_i)$.
- **Enc**($\textsf{pp}$, $\mathcal{S}$, $\textsf{ek}_{\textsf{id}^*}$, m): Given $\textsf{pp}$, a target identity set $\mathcal{S}$ with its length t, an encryption key identity $\textsf{ek}_{\textsf{id}^*}$ and the plaintext $m \in \{0,1\}^{\ell_1}$, it first picks $s, d_1, d_2, \sigma, \tau \in \mathbb{Z}_p$ and computes $C_0 = h^s$, $C_1 = g^s$, $C_2 = h_1^\tau$. For each $\textsf{id}_i \in \mathcal{S}$, it sets $\textsf{U}_{\textsf{id}_i} = \mathcal{H}_2(e(h_0, \mathcal{H}_0(\textsf{id}_i))^s)$ and $\textsf{V}_{\textsf{id}_i} = \mathcal{H}_2(e(\mathcal{H}_0(\textsf{id}_i), ek_{\textsf{id}^*} \cdot h_1^\tau))$, $f(x) = \prod_{i=1}^t (x - \textsf{U}_{\textsf{id}_i}) + d_1 = \sum\limits_{i=0}^{t-1} a_j x^j + x^t \mod p$ and $g(y) = \prod_{k=1}^t (y - \textsf{V}_{\textsf{id}_i}) + d_2 = \sum\limits_{k=0}^{t-1} b_k y^k + y^t \mod p$, where $a_0, \ldots, a_{t-1}$ and $b_0, \ldots, b_{t-1}$ are the coefficients correspond to x^j and y^k. Next, it sets $C_3 = [\mathcal{H}_3(d_1, d_2, C_1, C_0, C_2)]_{\ell-\ell_1}||(\mathcal{H}_3(d_1, d_2, C_1, C_0, C_2)]^{\ell_1} \oplus m)$, $\varphi = \mathcal{H}_4(C_1, C_0, C_2, C_3, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$ and $C_4 = (u^\varphi v^\sigma w)^s$. Finally, it generates a ciphertext $\textsf{ct} = (\sigma, C_1, C_0, C_2, C_3, C_4, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$.
- **Dec**($\textsf{pp}$, $\textsf{dk}_{\mathrm{id}_i}$, $\textsf{id}^*$, $\textsf{ct}$): Based on the public parameter $\textsf{pp}$, a decryption key $\textsf{dk}_{\mathrm{id}_i}$, the target identity $\textsf{id}^*$ and the ciphertext $\textsf{ct} = (\sigma, C_1, C_0, C_2, C_3, C_4, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$, it first computes $\varphi = \mathcal{H}_4(C_1, C_0, C_2, C_3, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$ and then determines whether $e(C_1, u^\varphi v^\sigma w) = e(g, C_4)$ holds. If not, it returns $\perp$. Otherwise, it computes $\textsf{U}_{\textsf{id}_i} = \mathcal{H}_2(e(C_0, \textsf{dk}_{i,1})) = \mathcal{H}_2(e(C_0, \mathcal{H}_0(\textsf{id}_i)^\rho))$, $d_1 = f(\textsf{U}_{\textsf{id}_i}) = \sum\limits_{j=0}^{t-1} a_j (\textsf{U}_{\textsf{id}_i})^j + (\textsf{U}_{\textsf{id}_i})^t \mod p$ and $\textsf{V}(\textsf{id}_i) = \mathcal{H}_2(e(\textsf{dk}_{i,3}, C_2) e(\textsf{dk}_{i,2}, \mathcal{H}_1(\textsf{id}^*))) =$

$\mathcal{H}_2(e(\mathcal{H}_0(\mathsf{id}_i), ek_{\mathsf{id}^*} \cdot h_1^{\tau}))$, $d_2 = g(\mathsf{V}_{\mathsf{id}_i}) = \sum_{i=0}^{t-1} b_j(\mathsf{V}_{\mathsf{id}_i})^j + (\mathsf{V}_{\mathsf{id}_i})^t \mod p$. If $[C_3]_{\ell-\ell_1} \neq [\mathcal{H}_3(d_1, d_2, C_1, C_0, C_2)]_{\ell-\ell_1}$, it returns $\perp$. Otherwise, it outputs $m = [\mathcal{H}_3(d_1, d_2, C_1, C_0, C_2)]^{\ell_1} \oplus [C_3]^{\ell_1}$.

B. Soundness of PS-ME and Its Security Proofs

***Theorem** 3:* If a user holds an authorized decryption key, then he/she can perform the successful decryption.

Proof: For a valid ciphertext $\mathsf{ct} = (\sigma, d_1, d_2, C_1, C_0, C_2, C_3, C_4, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$ and an authorized user id_i who has the secret key $\mathsf{dk}_{\mathsf{id}_i} = (\mathsf{dk}_{i,1} = \mathcal{H}_0(\mathrm{id}_i)^{\rho}, \mathsf{dk}_{i,2} = \mathcal{H}_0(\mathrm{id}_i)^{\alpha}, \mathsf{dk}_{i,3} = \mathcal{H}_0(\mathsf{id}_i))$ can conduct the following calculations to recover the plaintext m:

1) The authorized user first sets $\varphi = \mathcal{H}_4(C_1, C_0, C_2, C_3, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$ and then passes the verification of the equation $e(C_1, u^{\varphi} v^{\sigma} w) = e(g, C_4)$.
2) He/She next computes $\mathsf{U}_{\mathsf{id}_i} = \mathcal{H}_2(e(C_0,\ \mathsf{dk}_{i,1})) = \mathcal{H}_2(e(C_0, \mathcal{H}_1(\mathsf{id}_i)^{\rho}))$ and obtains $d_1 = f(\mathsf{U}_{\mathsf{id}_i}) = \sum_{i=0}^{t-1} a_j(\mathsf{U}_{\mathsf{id}_i})^j + (\mathsf{U}_{\mathsf{id}_i})^t \mod p$.
3) He/She then calculates
$$\begin{aligned}\mathsf{V}_{\mathsf{id}_i} &= \mathcal{H}_2(e(\mathsf{dk}_{i,3}, C_2)e(\mathsf{dk}_{i,2}, \mathcal{H}_1(\mathsf{id}^*)))\\ &= \mathcal{H}_2(e(\mathcal{H}_0(\mathsf{id}_i), h_1^{\tau})e(\mathcal{H}_0(\mathsf{id}_i)^{\alpha}, \mathcal{H}_1(\mathsf{id}^*)))\\ &= \mathcal{H}_2(e(\mathcal{H}_0(\mathsf{id}_i), \mathcal{H}_1(\mathsf{id}^*)^{\alpha} h_1^{\tau})),\end{aligned}$$
and gets $d_2 = g(\mathsf{V}_{\mathsf{id}_i}) = \sum_{j=0}^{t-1} b_j(\mathsf{V}_{\mathsf{id}_i})^j + (\mathsf{V}_{\mathsf{id}_i})^t \mod p$.
4) If $[C_3]_{\ell-\ell_1} = [\mathcal{H}_3(d_1, d_2, C_1, C_0, C_2)]_{\ell-\ell_1}$, he/she finally recovers the plaintext $m = [\mathcal{H}_3(d_1, d_2, C_1, C_0, C_2)]^{\ell_1} \oplus [C_3]^{\ell_1}$.

***Theorem** 4:* Assume that $\{\mathcal{H}_i\}_{i\in[0,3]}$ are random oracles, then our PS-ME is weakly robust against chosen-ciphertext attacks (WR-CCA).

Proof: Suppose that there exists a WR-CCA adversary $\mathcal{A}$ against the PS-ME, then another algorithm $\mathcal{C}$ can be easily constructed via the $\mathcal{A}$'s assistance to undermine the randomness of $\{\mathcal{H}_i\}_{i\in[0,3]}$ oracle's results.

- **Setup**: $\mathcal{C}$ randomly picks a bilinear group $\mathcal{BG} = (\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, p, e)$ of prime order p, chooses a generator $g \in \mathbb{G}_0$, generators $h, u, v, w \in \mathbb{G}_1$, $\alpha, \beta, \rho \in \mathbb{Z}_p$ and sets $g_1 = g^{\rho}, h_0 = h^{\rho}, h_1 = h^{\beta}$. Next, $\mathcal{A}$ is given the public parameter $\mathsf{pp} = (\mathcal{BG}, g, g_1, u, v, w, h, h_0, h_1, \{\mathcal{H}_i\}_{i\in[0,4]})$, where $\{\mathcal{H}_i\}_{i\in[0,3]}$ are random oracles used by $\mathcal{C}$ and $\mathcal{H}_4$ is a collusion-resistant hash function. $\mathcal{C}$ restores the master secret key $\mathsf{msk} = (\rho, \alpha)$ in its hands.
- **Query phase**: The following queries are adaptively made by $\mathcal{A}$:
  - $\mathcal{H}_0$ query: On input id, $\mathcal{C}$ conducts the following: If a record (id, W, w) has been existed in the $\mathcal{H}_0$-list (it is initialized empty), it returns W; otherwise, it picks $w \in \mathbb{Z}_p$, computes $W = \mathcal{H}_0(\mathsf{id}) = g^w$, adds (id, W, w) into $\mathcal{H}_0$-list and gives W to $\mathcal{A}$.
  - $\mathcal{H}_1$ query: On input id', $\mathcal{C}$ conducts the following: If a record (id', W', w') has been existed in the $\mathcal{H}_1$-list (it is initialized empty), it returns W'; otherwise, it picks $w' \in \mathbb{Z}_p$, computes $W = \mathcal{H}_1(\mathsf{id}') = h^{w'}$, adds (id', W', w') into $\mathcal{H}_1$-list and delivers W' to $\mathcal{A}$.
  - $\mathcal{H}_2$ query: On input X, $\mathcal{C}$ performs the following: If a record (X, x) has been existed in the $\mathcal{H}_2$-list (it is initialized empty), it returns X; otherwise, it picks $x \in \mathbb{Z}_p$, adds (X, x) into $\mathcal{H}_2$-list and returns x to $\mathcal{A}$.
  - $\mathcal{H}_3$ query: On input $(d_1, d_2, C_1, C_0, C_2)$, $\mathcal{C}$ performs the following: If there exists a record $((d_1, d_2, C_1, C_0, C_2), K)$ in the $\mathcal{H}_3$-list (it is initialized empty), it returns K; otherwise, it then selects $K \in \{0,1\}^{\ell}$ and adds $((d_1, d_2, C_1, C_0, C_2), K)$ into $\mathcal{H}_3$-list and sends K to $\mathcal{A}$.
  - Encryption key query: On input id', $\mathcal{C}$ first queries $\mathcal{H}_1$ query on id', assuming (id', W', w') is the corresponding tuple in the $\mathcal{H}_1$-list. Then, it calculates and sets an encryption key $\mathsf{ek} = W' = h^{w'\alpha}$, then sends ek to $\mathcal{A}$.
  - Secret key query: On input id, $\mathcal{C}$ first makes queries of $\mathcal{H}_0$ oracle on id, suppose that (id, W, w) be the tuple in the $\mathcal{H}_0$-list. Then, it sets a decryption key $\mathsf{dk}_1 = W_{\rho} = g^{w\rho}$, $\mathsf{dk}_2 = W_{\alpha} = g^{w\alpha}$ and then returns $\mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2, \mathcal{H}_0(\mathsf{id}))$ to $\mathcal{A}$.
  - Decryption query: On input id and ct, $\mathcal{C}$ could utilize its master secret key $\mathsf{msk} = (\rho, \alpha)$ to respond to any decryption query to $\mathcal{A}$.
- **Output**: $\mathcal{A}$ outputs a message $m \in \{0,1\}^{\ell_1}$ and a collection of receivers $\mathcal{S}^* = \{\mathsf{id}_1^*, \ldots, \mathsf{id}_t^*\}$, $\mathcal{C}$ conducts $\mathsf{ct} \leftarrow \mathbf{Enc}(\mathsf{pp}, \mathcal{S}^*, \mathsf{ek}_{\mathsf{id}'}, m)$ in the following: First pick $s, d_1^*, d_2^*, \sigma^*, \tau \in \mathbb{Z}_p$ and compute $C_0^* = h^s, C_1{}^* = g^s$, $C_2^* = h_1^{\tau}$. Then, for each $\mathsf{id}_i^* \in \mathcal{S}$, set $\mathsf{U}_{\mathsf{id}_i^*} = \mathcal{H}_2(e(h_0, \mathcal{H}_0(\mathsf{id}_i^*))^s)$ and $\mathsf{V}_{\mathsf{id}_i^*} = \mathcal{H}_2(e(\mathcal{H}_0(\mathsf{id}_i^*), ek_1 \cdot h_1^{\tau}))$, $f(x) = \prod_{i=1}^{t}(x - \mathsf{U}_{\mathsf{id}_i^*}) + d_1^* = \sum_{j=0}^{t-1} a_j^* x^j + x^t \mod p$ and $g(y) = \prod_{k=1}^{t}(y - \mathsf{V}_{\mathsf{id}_i^*}) + d_2^* = \sum_{k=0}^{t-1} b_k^* y^k + y^t \mod p$, where $a_0^*, \ldots, a_{t-1}^*$ and $b_0^*, \ldots, b_{t-1}^*$ are the coefficients correspond to x^j and y^k. Next, set $C_3^* = [\mathcal{H}_3(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)]_{\ell-\ell_1} || (\mathcal{H}_3(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)]^{\ell_1} \oplus m)$, $\varphi^* = \mathcal{H}_4(C_1{}^*, C_0^*, C_2^*, C_3^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$, and $C_4^* = (u^{\varphi^*} v^{\sigma^*} w)^s$. Finally, generate a ciphertext $\mathsf{ct} = (\sigma^*, C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$.

Analysis: If the WR-CCA game is won by $\mathcal{A}$, there exists a message $m' \neq \perp$ satisfying $\mathbf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathsf{id}_i^*}, \mathsf{id}', \mathsf{ct}^*) \rightarrow m'$ and $\mathsf{id}^* \notin \mathcal{S}^*$. This means that there indeed exists d_1', d_2' such that $C_4^* = [\mathcal{H}_3(d_1', d_2', C_1{}^*, C_0^*, C_2^*)]_{\ell-\ell_1} || (\mathcal{H}_3(d_1', d_2', C_1{}^*, C_0^*, C_2^*)]^{\ell_1} \oplus m')$, where $\mathsf{U}_{\mathsf{id}^*} = \mathcal{H}_2(e(C_0^*, \mathsf{dk}_{i^*,1})) = \mathcal{H}_2(e(C_0^*, \mathcal{H}_0(\mathsf{id}^*)^{\rho}))$, $d_1' = f(\mathsf{U}_{\mathsf{id}^*}) = \sum_{i=0}^{t-1} a_j^*(\mathsf{U}_{\mathsf{id}^*})^j + (\mathsf{U}_{\mathsf{id}^*})^t \mod p$ and $\mathsf{V}_{\mathsf{id}^*} = \mathcal{H}_2(e(\mathsf{dk}_{i^*,3}, C_2^*)e(\mathsf{dk}_{i^*,2}, \mathcal{H}_1(\mathsf{id}'))) = \mathcal{H}_2(e(\mathcal{H}_0(\mathsf{id}^*), \mathcal{H}_1(\mathsf{id}')^{\alpha} h_1^{\tau}))$, $d_2' = g(\mathsf{V}_{\mathsf{id}^*}) = \sum_{i=0}^{t-1} b_j^*(\mathsf{V}_{\mathsf{id}^*})^j + (\mathsf{V}_{\mathsf{id}^*})^t \mod p$. However, for $\mathsf{id}_i^* \in \mathcal{S}$, $C_3^* = [\mathcal{H}_3(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)]_{\ell-\ell_1} || (\mathcal{H}_3(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)]^{\ell_1} \oplus$

m).

In the WR-CCA game, the $\mathcal{A}$'s advantage in winning this game is negligible. The specific illustrations are as follows:

1) If $d_1' = d_1^*$ and $d_2' = d_2^*$, that is $f(\mathsf{U}_{\mathsf{id}^*}') = f(\mathsf{U}_{\mathsf{id}^*}^*)$ and $g(\mathsf{V}_{\mathsf{id}^*}') = g(\mathsf{V}_{\mathsf{id}^*}^*)$, then we can derive that $\prod_{i=1}^{t}(\mathsf{U}_{\mathsf{id}^*}' - \mathsf{U}_{\mathsf{id}_i^*}^*) = 0$ and $\prod_{k=1}^{t}(\mathsf{V}_{\mathsf{id}^*}' - \mathsf{V}_{\mathsf{id}_k^*}^*) = 0$ due to the fact that $f(x) = \prod_{i=1}^{t}(x - \mathsf{U}_{\mathsf{id}_i}^*) + d_1^*$ and $g(y) = \prod_{i=1}^{t}(y - \mathsf{V}(\mathsf{id}_i^*)) + d_2^*$ for $\mathsf{id}_i^* \in \mathcal{S}^*$. It implies there exists some $\mathsf{U}_{\mathsf{id}_i}^*$ and $\mathsf{V}_{\mathsf{id}_i^*}$ holding ${\mathsf{U}_{\mathsf{id}}'}^* = \mathsf{U}_{\mathsf{id}_i^*}^*$ and ${\mathsf{V}_{\mathsf{id}}'}^* = \mathsf{V}_{\mathsf{id}_k^*}^*$. In other words, $\mathcal{H}_2({X_{\mathsf{id}}'}^*) = \mathcal{H}_2({X_{\mathsf{id}_i^*}}^*)$. Since $\mathcal{H}_2$ is a random oracle, hence $X_{\mathsf{id}^*}' = X_{\mathsf{id}_i^*}^*$. Because $X_{\mathsf{id}^*}' = e(h_0, \mathcal{H}_0(\mathsf{id}^*))^s)$ and $X_{\mathsf{id}_i^*}^* = e(h_0, \mathcal{H}_0(\mathsf{id}^*))^s)$ as well as $X_{\mathsf{id}^*}' = e(\mathcal{H}_0(\mathsf{id}^*), W \cdot h_1^{\tau})$ and $X_{\mathsf{id}_i^*} = e(\mathcal{H}_0(\mathsf{id}_i^*), W \cdot h_1^{\tau})$, it implicitly means $\mathcal{H}_0(\mathsf{id}^*) = \mathcal{H}_0(\mathsf{id}_i^*)$, thus $\mathsf{id}^* = \mathsf{id}_i^*$. However, the fact is contradicted by the assumption $\mathsf{id}^* \notin \mathcal{S}$. Therefore, we can conclude $d_1' = d_1^*$ and $d_2' = d_2^*$ are incorrect.
2) If $d_1' \neq d_1^*$ and $d_2' \neq d_2^*$, since $\mathcal{H}_3$ is also a random oracle, then $\mathcal{H}_3(d_1', d_2', {C_1}^*, C_0^*, C_2^*)]_{\ell-\ell_1} \neq \mathcal{H}_3(d_1^*, d_2^*, {C_1}^*, C_0^*, C_2^*)]_{\ell-\ell_1}$. However, the equation $[\mathcal{H}_3(d_1^*, d_2^*, {C_1}^*, C_0^*, C_2^*)]_{\ell-\ell_1} = [C_3^*]_{\ell-\ell_1}$, then we can get that $[\mathcal{H}_3(d_1', d_2', {C_1}^*, C_0^*, C_2^*)]_{\ell-\ell_1} \neq [C_3^*]_{\ell-\ell_1}$.

Hence, $\mathcal{A}$ can only get the abort symbol $\perp$, which contradicts with $m' \neq \perp$. In this way, the advantage of $\mathcal{A}$ in winning the WR-CCA game is negligible.

***Theorem** 5:* Assume that our PS-ME is WR-CCA secure and DBDH assumption holds, then our PS-ME can achieve the indistinguishability against chosen ciphertext attacks (IND-CCA).

Proof: Suppose that there is an IND-CCA attacker $\mathcal{A}$ that can successfully breach the PS-ME, then another algorithm $\mathcal{C}$ can be easily created via the interaction with $\mathcal{A}$ to address the DBDH assumption or break the WR-CCA security of the PS-ME. Given a tuple $(h^a, h^b, h^c, g^a, g^b, g^c, \mathcal{T})$ to $\mathcal{C}$, the goal of $\mathcal{A}$ is to determine whether $\mathcal{T} = e(g,h)^{abc}$ or $\mathcal{T}$ is a random element of $\mathbb{G}_T$.

- **Setup**: $\mathcal{C}$ sets $g_1 = g^a$, $h_0 = h^a$, $h_1 = h^{\beta}$, $u = h^{bx_1}h^{x_2}$, $v = h^{by_1}h^{y_2}$, $v = h^{bz_1}h^{z_2}$, where $\{x_i, y_i, z_i\}_{i\in\{1,2\}}, \beta \in \mathbb{Z}_p$. Then, $\mathcal{C}$ sends the public parameter $\mathsf{pp} = (\mathcal{BG}, g, g_1, u, v, w, h, h_0, h_1, \{\mathcal{H}_i\}_{i\in[0,4]})$, where $\{\mathcal{H}_i\}_{i\in[0,3]}$ are random oracles mastered by $\mathcal{C}$ and $\mathcal{H}_4$ is a collusion-resistant hash function. The master secret key is implicitly set as $\mathsf{msk} = (a, \beta)$, where a is unknown to $\mathcal{C}$.
- **Phase 1**: The following queries can be adaptively issued by $\mathcal{A}$:
  - $\mathcal{H}_0$ query: On input id, $\mathcal{C}$ does the following: If a record $(\mathsf{id}, Y, y, \zeta)$ has been existed in $\mathcal{H}_0$-list (it is initialized empty), it returns Y; otherwise, it picks $\zeta \in \{0,1\}$, $y \in \mathbb{Z}_p$. If $\zeta = 0$, it computes $Y = \mathcal{H}_0(\mathsf{id}) = g^y$; else it calculates $Y = \mathcal{H}_0(\mathsf{id}) = g^{by}$, and adds $(\mathsf{id}, Y, y, \zeta)$ into $\mathcal{H}_0$-list and sends Y to $\mathcal{A}$.
  - $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ queries: The corresponding query is identical to that in the WR-CCA game.
  - Encryption key query: It is also the same as that in the WR-CCA game.
  - Secret key query: On input id, $\mathcal{C}$ first makes queries of $\mathcal{H}_0$ oracle on id, suppose that $(\mathsf{id}, W, w, \zeta)$ be the tuple in the $\mathcal{H}_0$-list. If $\zeta = 1$, it aborts and returns $\perp$; else it sets a decryption key $\mathsf{dk}_1 = W_{\rho} = g^{aw}, \mathsf{dk}_2 = W_{\alpha} = g^{\alpha w}, \mathsf{dk}_3 = W = g^w$ and then returns $\mathsf{dk} = (\mathsf{dk}_1, \mathsf{dk}_2, \mathsf{dk}_3)$ to $\mathcal{A}$.
  - Decryption query: On input id and ct, where $\mathsf{ct} = (\sigma,\ d_1, d_2, C_1, C_0, C_2, C_3, C_4, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$, $\mathcal{C}$ first make queries of $\mathcal{H}_0$ on id to get $(\mathsf{id}, Y, y, \zeta)$, if $\zeta = 0$, it calculates $\mathsf{dk} = (\mathsf{dk}_1 = g_1^y, \mathsf{dk}_2 = g^{\alpha y}, \mathsf{dk}_3 = g^y)$ and uses it to reply the decryption query; else it implements as follows: compute $\varphi = \mathcal{H}_4(d_1, d_2, C_1, C_0, C_2, C_3, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$ and check whether $e(C_1, u^{\varphi}v^{\sigma}w) = e(g, C_4)$ holds. If this equation does not hold, it means the ciphertext is invalid and outputs $\perp$; else determine whether $x_1\varphi + y_1\sigma + z_1 = 0$ holds; If so, abort and return a random bit; else, proceed to perform the following: Since $C_4 = (u^{\varphi}v^{\sigma}w)^s = (h^{b(x_1\varphi+y_1\sigma+z_1)})^s \cdot (h^{x_2\varphi+y_2\sigma+z_2})^s = C_0^{b(x_1\varphi+y_1\sigma+z_1)}C_0^{(x_2\varphi+y_2\sigma+z_2)}$, it is simple to deduce $C_0^b = (\frac{C_4}{C_0^{x_2\varphi+y_2\sigma+z_2}})^{\frac{1}{x_1\varphi+y_1\sigma+z_1}}$. Hence, $X_{\mathsf{U}_{\mathsf{id}}} = e(Y, h_0)^s = e(g^{by}, h_0)^s = e(C_0^b, h_0)^y$ and $X_{\mathsf{V}_{\mathsf{id}}} = e(Y', h^{w'\alpha} \cdot h_1^{\tau})^s = e((g^{by'}, h^{w'\alpha}h_1^{\tau})^s) = e(C_0^b, h^{w'\alpha}h_1^{\tau})^{y'}$. $\mathcal{C}$ queries $\mathcal{H}_2$ on X_{id} to derive U_{id} and V_{id}, respectively, where $\mathsf{U}_{\mathsf{id}} = \mathcal{H}_2(X_{\mathsf{U}_{\mathsf{id}}})$ and $\mathsf{V}_{\mathsf{id}} = \mathcal{H}_2(X_{\mathsf{V}_{\mathsf{id}}})$. Then, compute $d_1 = f(\mathsf{U}_{\mathsf{id}})$, $d_2 = g(\mathsf{V}_{\mathsf{id}})$ and issue $\mathcal{H}_3$ query on $(d_1, d_2, C_1, C_0, C_2)$ to obtain K, where $K = \mathcal{H}_3(d_1, d_2, C_1, C_0, C_2)$. If $[C_3]_{\ell-\ell_1} \neq [K]_{\ell-\ell_1}$, it returns $\perp$ indicating an invalid ciphertext; otherwise, it outputs $m = [K]^{\ell_1} \oplus [C_3]^{\ell_1}$.
- **Challenge**: Two different equal-length plaintexts m_0, m_1 and the picked identity set $\mathcal{S}^*$ are outputted by $\mathcal{A}$. Here, we need to notice that $\mathcal{A}$ has not queried the decryption key on any id, where $\mathsf{id} \notin \mathcal{S}^*$. For all $\mathsf{id}_i \in \mathcal{S}^*$, $\mathcal{C}$ sends the query of $\mathcal{H}_0$ to acquire $(\mathsf{id}_i, Y_{\mathsf{id}_i}, y_i, \zeta_i)$. If there exists some $\mathsf{id}_i \in \mathcal{S}^*$ and $\zeta_i = 0$, $\mathcal{C}$ aborts; otherwise, for each $\mathsf{id}_i \in \mathcal{S}^*$, let $X_{\mathsf{U}_{\mathsf{id}_i}}^* = \mathcal{T}^{y_i}$ and issue the query of $\mathcal{H}_2$ on $X_{\mathsf{U}_{\mathsf{id}_i}}^*$ to derive $\mathsf{U}_{\mathsf{id}_i}^*$ from $\mathcal{H}_2$-list, where $\mathsf{U}_{\mathsf{id}_i}^* = \mathcal{H}_2(X_{\mathsf{U}_{\mathsf{id}_i}}^*)$. Besides, $\mathcal{C}$ can also derive $\mathsf{V}_{\mathsf{id}_i}^* = e(g,h)^{bcy'(w'\alpha+\beta\tau)}$ from $\mathcal{H}_2$-list, where $\mathsf{V}_{\mathsf{id}}^* = \mathcal{H}_2(X_{\mathsf{V}_{\mathsf{id}}}^*)$. Next, $\mathcal{C}$ randomly picks $d_1^*, d_2^* \in \mathbb{Z}_p$ and computes $f(\mathsf{U}_{\mathsf{id}}^*) = \sum_{j=0}^{t-1} a_j^* x^j + x^t \mod p$ and $g(\mathsf{V}_{\mathsf{id}}^*) = \sum_{k=0}^{t-1} b_k^* x^k + x^t \mod p$, it then outputs $(a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*,)$. Let $C_0^* = h^c, {C_1}^* = g^c, C_2^* = h^{\beta\tau}$, $\mathcal{C}$ issues $\mathcal{H}_3$ query on $(d_1^*, d_2^*, {C_1}^*, C_0^*, C_2^*)$ to get K^*, where $K^* = \mathcal{H}_3(d_1^*, d_2^*, {C_1}^*, C_0^*, C_2^*)$. $\mathcal{C}$ chooses $\xi \in \{0,1\}$ at random, computes $C_3^* = [K^*]_{\ell-\ell_1}||([K^*]^{\ell_1} \oplus m_{\xi})$, $\varphi^* = \mathcal{H}_4({C_1}^*, C_0^*, C_2^*, C_3^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$, $\sigma^* = -\frac{x_1\varphi^*+z_1}{y_1}$. Hence, $\mathcal{C}$ can also calculate $C_4^* = (h^c)^{x_2\varphi^*+y_2\sigma^*+z_2}$ and produce the challenge ciphertext $\mathsf{ct}^* = (\sigma^*, C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$.

This article has been accepted for publication in IEEE Transactions on Information Forensics and Security. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2023.3294725

- **Phase 2**: The following queries can be continually made in an adaptive manner:
  - Encryption key query: On input id', $\mathcal{C}$ does the operations as that in **Phase 1**.
  - Secret key query: On input id, where $\mathsf{id} \notin \mathcal{S}^*$, $\mathcal{C}$ performs them as that in **Phase 1**.
  - Decryption query: On input id and ct, where $\mathsf{ct} = (\sigma, C_0, C_1, C_2, C_3, C_4, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$.
    1) If $\mathsf{ct} \neq \mathsf{ct}^*$, $\mathcal{C}$ judges $\mathcal{H}_4(C_1, C_0, C_2, C_3, C_4, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1}) = \mathcal{H}_4(C_1{}^*, C_0^*, C_2^*, C_3^*, C_4^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$. If yes, it returns $\perp$ and outputs a random bit; otherwise, conducts the same as that in **Phase 1**.
    2) $\mathsf{ct} = \mathsf{ct}^*$ and $\mathsf{id} \in \mathcal{S}^*$, $\mathcal{C}$ returns $\perp$.
    3) $\mathsf{ct} = \mathsf{ct}^*$ and $\mathsf{id} \notin \mathcal{S}^*$, $\mathcal{C}$ returns $\perp$ with a certain advantage since $\mathsf{ct} \leftarrow \mathbf{Enc}(\mathsf{pp}, \mathcal{S}^*, \mathsf{id}', m_\xi)$ and $\mathbf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathsf{id}_i^*}, \mathsf{ek}_{\mathsf{id}'}, \mathsf{ct}^*) \neq \perp$ is negligible for $\mathsf{id} \in \mathcal{S}^*$,
- **Guess**: $\mathcal{A}$ outputs a guess ξ'. If $\xi' = \xi$, it returns 1 implying $\mathcal{T} = e(g, h)^{abc}$; otherwise, it gives 0 indicating $\mathcal{T} = \mathcal{R}$, where $\mathcal{R}$ is a randomness of group $\mathbb{G}_T$.

Analysis: If $\mathcal{T} = e(g, h)^{abc}$, $s^* = c$, the challenge ciphertext queried by $\mathcal{A}$ originates from a distribution, which is the same as that in the construction; if $\mathcal{T} = \mathcal{R}$, where $\mathcal{R}$ is random, the ciphertext $C_3^* = [K^*]_{\ell-\ell_1} || ([K^*]^{\ell_1} \oplus m_\xi)$, where $K^* = \mathcal{H}_3(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)$ is uniformly random. Hence, from the view of $\mathcal{A}$, m_ξ is independent.

***Theorem** 6:* If the PS-ME is IND-CCA secure and DBDH assumption holds, then our PS-ME can achieve receiver anonymity security against chosen ciphertext attacks (RA-CCA).

Proof: Suppose that there exists a RA-CCA adversary $\mathcal{A}$ against the PS-ME, then another algorithm $\mathcal{C}$ can be easily built via the $\mathcal{A}$'s help to address the intractability of DBDH assumption. $\mathcal{C}$ is given a tuple $(h^a, h^b, h^c, g^a, g^b, g^c, \mathcal{T})$, the goal of $\mathcal{A}$ is to decide whether $\mathcal{T} = e(g, h)^{abc}$ or $\mathcal{T}$ is a random element of $\mathbb{G}_T$.

- **Setup**: This phase is identical to that in ***Theorem 5***.
- **Phase 1**: This phase is also the same as that in ***Theorem 5***.
- **Challenge**: $\mathcal{A}$ gives a plaintext m and two various identity sets $\mathcal{S}_0^*, \mathcal{S}_1^*$, where these two identity sets at least have a common identity. Without loss of generality, assuming $\mathcal{S}_0^* = \{\mathsf{id}_0^*, \mathsf{id}_2^*, \ldots, \mathsf{id}_t^*\}$, and $\mathcal{S}_1^* = \{\mathsf{id}_1^*, \mathsf{id}_2^*, \ldots, \mathsf{id}_t^*\}$. This phase also requires that $\mathcal{A}$ has not made queries of extraction query on id such that $\mathsf{id} \in \{\mathsf{id}_0^*, \mathsf{id}_1^*\}$ in **Phase 1**. $\mathcal{C}$ answers as follows: Let $C_0^* = h^c, C_1{}^* = g^c, C_2^* = h^{\beta\tau}$, choose a random bit ϵ, where $\mathcal{S}_\epsilon^* = \{\mathsf{id}_\epsilon^*, \mathsf{id}_2^*, \ldots, \mathsf{id}_t^*,\}$. Send the query of $\mathcal{H}_0$ to obtain $(\mathsf{id}_\epsilon^*, Y_{\mathsf{id}_\epsilon}^*, y_\epsilon, \zeta_\epsilon)$, if $\zeta_\epsilon = 0$, return $\perp$ and abort; else compute $Y_{\mathsf{id}_\epsilon}^* = g^{by_\epsilon}$ and $X_{\mathsf{U}_{\mathsf{id}_\epsilon}}^* = \mathcal{T}^{y_\epsilon}$. Afterwards, issue the query of $\mathcal{H}_2$ on $X_{\mathsf{U}_{\mathsf{id}_\epsilon}}^*$ to derive $\mathsf{U}_{\mathsf{id}_\epsilon}^*$ from $\mathcal{H}_2$-list, where $\mathsf{U}_{\mathsf{id}_\epsilon}^* = \mathcal{H}_2(X_{\mathsf{U}_{\mathsf{id}_\epsilon}}^*)$. Besides, also send the query of $\mathcal{H}_2$ on $X_{\mathsf{V}_{\mathsf{id}_\epsilon}}^* = e(g, h)^{bcy'_\epsilon(w'\alpha+\beta\tau)}$ from $\mathcal{H}_2$-list, where $\mathsf{V}_{\mathsf{id}_\epsilon}^* = \mathcal{H}_2(X_{\mathsf{V}_{\mathsf{id}_\epsilon}}^*)$. For the other case of identity $\mathsf{id}_i \in \mathcal{S}_\epsilon^* \setminus \mathsf{id}_i^*$, send the query of $\mathcal{H}_0$ to obtain $(\mathsf{id}_i, Y_{\mathsf{id}_i}, y_i, \zeta_i)$, if $\zeta_i = 1$, return $\perp$ and abort; else compute $Y_{\mathsf{id}_i} = g^{y_i}$, $X_{\mathsf{U}_{\mathsf{id}_j}}^* = e(g^c, h^a)^{y_i}$ and $X_{\mathsf{V}_{\mathsf{id}}}^* = e(Y', h^{w'\alpha} \cdot h_1^\tau)^s = e((g^{y'}, h^{w'\alpha} h_1^\tau)^s) = e(g^c, h^{w'\alpha} h_1^\tau)^{y'}$. Next, randomly pick $d_1^*, d_2^* \in \mathbb{Z}_p$, compute $f(\mathsf{U}_{\mathsf{id}}^*) = \sum_{j=0}^{t-1} a_j^* x^j + x^t \mod p$ and $g(\mathsf{V}_{\mathsf{id}}^*) = \sum_{k=0}^{t-1} b_k^* x^k + x^t \mod p$, output $(a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*,)$. Then, issue $\mathcal{H}_3$ query on $(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)$ to get K^*, where $K^* = \mathcal{H}_3(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)$. Finally, compute $C_3^* = [K^*]_{\ell-\ell_1} || ([K^*]^{\ell_1} \oplus m)$, $\varphi^* = \mathcal{H}_4(C_1{}^*, C_0^*, C_2^*, C_3^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$, $\sigma^* = -\frac{x_1\varphi^*+z_1}{y_1}$ and $C_4^* = (h^c)^{x_2\varphi^*+y_2\sigma^*+z_2}$. So, the challenge ciphertext is produced as $\mathsf{ct}^* = (\sigma^*, C_0^*, C_1^*, C_2^*, C_3^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$.
- **Phase 2**: The queries can be continually made in an adaptive manner as follows:
  - Encryption key query: On input id', $\mathcal{C}$ does the operations as that in **Phase 1**.
  - Secret key query: On input id, where $\mathsf{id} \notin \{\mathsf{id}_0^*, \mathsf{id}_1^*\}$, $\mathcal{C}$ performs them as that in **Phase 1**.
  - Decryption query: On input id and ct, where $\mathsf{ct} = (\sigma, C_0, C_1, C_2, C_3, C_4, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1})$.
    1) If $\mathsf{ct} \neq \mathsf{ct}^*$, $\mathcal{C}$ judges the equation $\mathcal{H}_4(C_1, C_0, C_2, C_3, a_0, \ldots, a_{t-1}, b_0, \ldots, b_{t-1}) = \mathcal{H}_4(C_1{}^*, C_0^*, C_2^*, C_3^*, a_0^*, \ldots, a_{t-1}^*, b_0^*, \ldots, b_{t-1}^*)$ holds. If yes, it returns $\perp$ and outputs a random bit; otherwise, it conducts the same as that in **Phase 1**.
    2) $\mathsf{ct} = \mathsf{ct}^*$, for $\mathsf{id} \in \{\mathsf{id}_0^*, \mathsf{id}_1^*\}$, $\mathcal{C}$ returns $\perp$. For $\mathsf{id} \in \mathsf{S}_0^* \cap \mathsf{S}_1^*$, $\mathcal{C}$ outputs the plaintext m. For $\mathsf{id} \notin \mathsf{S}_0^* \cup \mathsf{S}_1^*$, $\mathcal{C}$ also outputs $\perp$ with a certain advantage. Since the PS-ME achieves IND-CCA security, namely, $\mathsf{ct}^* \leftarrow \mathbf{Enc}(\mathsf{pp}, \mathcal{S}_\epsilon^*, \mathsf{id}', m)$ and $\mathbf{Dec}(\mathsf{pp}, \mathsf{dk}_{\mathsf{id}}, \mathsf{id}', \mathsf{ct}^*) \neq \perp$ is negligible for $\mathsf{id} \notin \mathsf{S}_0^* \cup \mathsf{S}_1^*$,
- **Guess**: $\mathcal{A}$ outputs a guess ϵ'. If $\epsilon = \epsilon'$, it returns 1 implying $\mathcal{T} = e(g, h)^{abc}$; otherwise, it gives 0 indicating $\mathcal{T} = \mathcal{R}$, where $\mathcal{R}$ is a random element of group $\mathbb{G}_T$.

Analysis: If $\mathcal{T} = e(g, h)^{abc}$, $s^* = c$, the challenge ciphertext queried by $\mathcal{A}$ originates from a distribution, which is the same as that in the construction; if $\mathcal{T} = \mathcal{R}$, where $\mathcal{R}$ is a random value, the ciphertext $C_3^* = [K^*]_{\ell-\ell_1} || ([K^*]^{\ell_1} \oplus m)$, where $K^* = \mathcal{H}_3(d_1^*, d_2^*, C_1{}^*, C_0^*, C_2^*)$ is uniformly random.

***Theorem** 7:* Suppose that the authenticity of our PS-ME can be breached by $\mathcal{A}$ with overwhelming advantage, then it is easy to create an algorithm called challenger $\mathcal{C}$ that enables interaction with $\mathcal{A}$ to address the CBDH problem with a non-negligible advantage ϵ.

***Proof*:** $\mathcal{C}$ is given a CBDH tuple $(g, g^a, g^b, g^c, h^a, h, h^b, h^c, \mathcal{Z})$, the goal of $\mathcal{A}$ is to compute $\mathcal{Z} = e(g, h)^{abc}$. The following is the interaction process between $\mathcal{A}$ and $\mathcal{C}$:

- **Setup**: $\mathcal{C}$ sends $\mathcal{A}$ the public parameter $\mathsf{pp} = (\mathcal{BG}, g_0 = g^a, \{\mathcal{H}_i\}_{i\in[0,2]})$, where $\{\mathcal{H}_i\}_{i\in[0,2]}$ are all random oracles operated by $\mathcal{C}$.
- **Query phase**: The following queries can be adaptively issued by $\mathcal{A}$:
  - $\mathcal{H}_0$ query: If there exists a query id_i in a tuple $(\mathsf{id}_i, W_i, \eta_i, s_i)$, then return W_i; otherwise, produce

TABLE II: Computation cost comparisons of ME related solutions

Scheme	Costs at client side		Costs at registry authority side		
	Enc	**Dec**	**Setup**	**EKGen**	**DKGen**
AFN+ [20]	$2n(e_0+p)$	$3p$	e_0	e_0	$2e_0$
FGR+ [21]	$(3e_0+p)n$	e_1+3p	$2e_0$	e_0	e_0
CLW+ [22]	$17ne_0$	$8e_0+p$	$66e_0+2p$	$24e_0$	$72e_0$
IB-BME	$(l+6)e_0+e_1+lp$	$(2l+1)e_0+e_1+(2l+6)p$	$(3l+12)e_0+e_1$	e_0	$(6l+7)e_0$
PS-BME	$6e_0+2np$	$4p$	$3e_0$	e_0	$2e_0$

TABLE III: Storage cost comparisons of ME related solutions realizing one-to-many data sharing

Scheme	pp storage cost	ek storage cost	dk storage cost	ct storage cost
AFN+ [20]	$2\|\mathbb{G}_0\|$	$\|\mathbb{G}_0\|$	$3\|\mathbb{G}_0\|$	$2n\|\mathbb{G}_0\|+n\|m\|$
FGR+ [21]	$3\|\mathbb{G}_0\|$	$\|\mathbb{Z}_p\|$	$\|\mathbb{G}_1\|+\|\mathbb{Z}_p\|$	$n(\|\mathbb{G}_0\|+2\|\mathbb{G}_T\|+\|\mathbb{Z}_p\|)$
CLW+ [22]	$16\|\mathbb{G}_0\|+2\|\mathbb{G}_T\|$	$8\|\mathbb{G}_0\|$	$18\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$n(8\|\mathbb{G}_1\|+\|\mathbb{G}_T\|)$
IB-BME	$l\|\mathbb{G}_0\|+(2l+9)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$\|\mathbb{G}_0\|$	$(2l+6)\|\mathbb{G}_1\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_1\|+(l+2)\|\mathbb{Z}_p\|$
PS-BME	$2\|\mathbb{G}_0\|$	$\|\mathbb{G}_0\|$	$3\|\mathbb{G}_0\|$	$\|\mathbb{G}_0\|+3\|\mathbb{G}_1\|+(2n+2)\|\mathbb{Z}_p\|+\|m\|$

random $\eta_i \in \mathbb{Z}_p$ and $s_i \in \{0,1\}$, such that the probability of $s_i = 0$ is ξ. If $s_i = 0$, then calculate $W_i = g^{\eta_i}$; otherwise let $W_i = g^{c\eta_i}$. Finally, the list $\mathcal{L}_0$ is added $(\mathsf{id}_i, W_i, \eta_i, s_i)$ and W_i is returned to $\mathcal{A}$.

- $\mathcal{H}_1$ query: If there exists a query id'_i in a tuple $(\mathsf{id}'_i, W_i, \eta_i, s'_i)$, then return W_i; otherwise, produce random $\eta'_i \in \mathbb{Z}_p$ and $s'_i \in \{0,1\}$, such that the probability of $s'_i = 0$ is ξ. If $s'_i = 0$, then calculate $W_i = h^{\eta'_i}$; otherwise let $W_i = h^{a\eta'_i}$. Finally, the list $\mathcal{L}_1$ is added $(\mathsf{id}'_i, W_i, \eta'_i, s'_i)$ and W_i is returned to $\mathcal{A}$.
- $\mathcal{H}_2$ query: The list $\mathcal{L}_2$ is maintained by $\mathcal{C}$ to store the tuples of $(T_i, \hat{h}_i)$. If the query T_i had been queried, then $\mathcal{C}$ returns the $\hat{h}_i$; otherwise, $\mathcal{C}$ randomly chooses $\hat{h}_i \in \mathbb{Z}_p$, adds the new tuple $(T_i, \hat{h}_i)$ into $\mathcal{L}_2$ and outputs $\hat{h}_i$ to $\mathcal{A}$.
- Encryption key query: With the input id' to the oracle of $\mathcal{H}_1$, the result $\mathcal{H}_1(\mathsf{id}') = W_i$ can be obtained from $(\mathsf{id}'_i, W_i, \eta_i, s_i)$ of $\mathcal{L}_1$. If $s_i = 1$, abort and return $\perp$; else, output $\mathsf{ek}_{\mathsf{id}'} = h^{b\eta_i}$.
- Secret key query: With the input id to the oracle of $\mathcal{H}_0$, the result $\mathcal{H}_0(\mathsf{id}) = W_i$ can be obtained from $(\mathsf{id}_i, W_i, \eta_i, s_i)$ of $\mathcal{L}_0$. If $s_i = 1$, abort and return $\perp$; else, output $\mathsf{dk}_{\mathsf{id}} = (g^{a\eta_i}, g^{b\eta_i}, g^{\eta_i})$.

- **Forgery**: $\mathcal{A}$ sends $(c_2, \mathsf{id}, \mathsf{id}')$ to $\mathcal{C}$. In response, $\mathcal{C}$ performs the steps as follows:
  1) Calculate $\mathcal{H}_0(\mathsf{id}) = W$ and $\mathcal{H}_1(\mathsf{id}) = W'$. If the tuples $(\mathsf{id}, W, \eta, s) \in \mathcal{L}_0$ and $(\mathsf{id}', W', \eta', s') \in \mathcal{L}_0$ simultaneously do not have s, s' equal to 1, $\mathcal{C}$ aborts and returns $\perp$; If not, we can implicitly derive that $\mathsf{dk}_{\mathsf{id},2} = g^{cb\eta}$ and $\mathcal{H}_1(\mathsf{id}') = h^{a\eta'}$. So, $\mathcal{H}_2(e(\mathcal{H}_0(\mathsf{id}_i), ek_1 \cdot h_1^{\tau})) = \mathcal{H}_2(e(\mathsf{dk}_{\mathsf{id}_i,2}, \mathcal{H}_1(\mathsf{id}')) \cdot e(\mathsf{dk}_{\mathsf{id}_i,3}, h_1^{\tau}))$, where $e(\mathsf{dk}_{\mathsf{id}_i,2}, \mathcal{H}_1(\mathsf{id}')) = e(g^{cb\eta}, h^{a\eta'})$ and $e(\mathsf{dk}_{\mathsf{id}_i,3}, h_1^{\tau})) = e(g^{\eta}, h^{\tau})$.
  2) From the list $\mathcal{L}_2$, it is easy to obtain $(T_i, \hat{h}_i)$, thus knowing $\mathcal{Z} = (T_i \cdot e(g^{\eta}, h^{\tau})^{-1})^{1/(\eta\eta')}$.

Analysis: Assuming $\mathcal{A}$ makes at most q_0 queries and q_1 queries to oracles: *Encryption key query* and *Secret key query*, the probability of the non-abortion for $\mathcal{A}$ is $\xi^{q_0+q_1}$. Similarly, the probability of $\mathcal{C}$ that does not abort in the phase of **Forgery** is $(1-\xi)^2$. Therefore, the entire probability of not aborting is $\xi^{q_0+q_1}(1-\xi)^2$. If we let $\theta = (q_0+q_1)/(q_0+q_1+2)$ be the probability for getting $s_i = 0$ in the queries of $\mathcal{H}_0$ and $\mathcal{H}_1$, then the entire probability not aborting is $\xi^{q_0+q_1}(1-\xi)^2 \leq 4/(e^2(q_0+q_1+2)^2)$. If the abortion fails, the probability of outputting the correct $\mathcal{Z}$ is $2\epsilon/q_{\mathcal{H}_2}$. Thus, the probability of solving CBDH problem is $8\epsilon/(e^2 q_{\mathcal{H}_2}(q_0+q_1+2)^2)$.

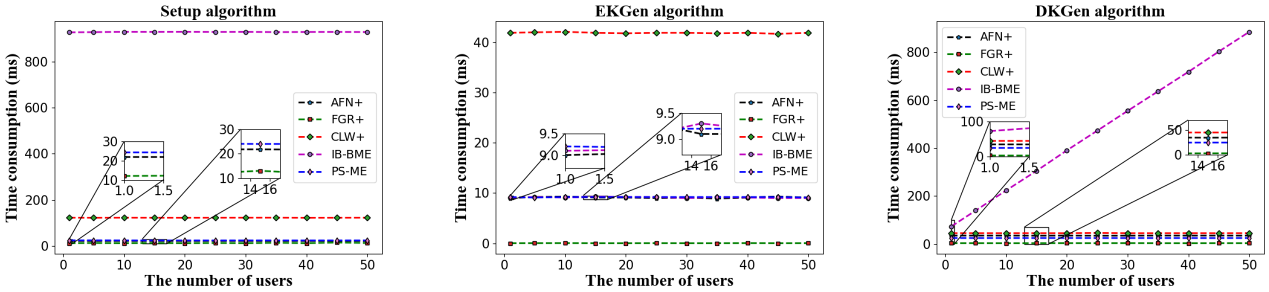
VII. Performance Evaluation

In this section, we first give theoretical analysis via computation & communication (storage) cost comparisons and then evaluate the performance via the experimental simulations to indicate the practicability of our solutions.

A. Theoretical Analysis

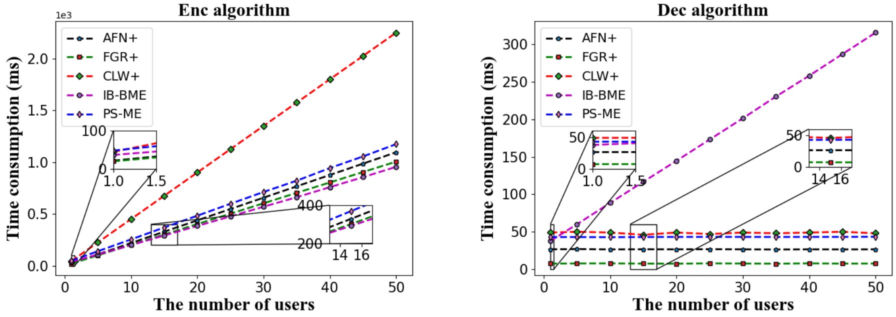
The computation and storage overheads of ME-related works realizing one-to-many data sharing are summarized in TABLEs II & III. In detail, the most time-consuming calculations, such as exponentiation operation and bilinear pairings, are mainly considered in the comparisons. For easy comparisons, we let l, n be the maximum number of recipients supported in the system and the number of recipients to be specified in the access control, respectively. In TABLE II, we let p, e_0 and e_1 be the time to perform one bilinear pairing operation, a single exponentiation computation in $\mathbb{G}_0$, a single exponentiation in $\mathbb{G}_T$, respectively. In TABLE III, we let $|\mathbb{G}_0|$, $|\mathbb{G}_1|$ and $|\mathbb{G}_T|$ be the size of a single group element in $\mathbb{G}_0$, $\mathbb{G}_1$ and $\mathbb{G}_T$. Let $|m|$ denote the length of a string message.

As depicted from TABLE II, it is straightforward to observe that the computational costs of the encryption (**Enc**) phase in each scheme all grow linearly with the incremental number of recipients. It is also easy to conclude that the calculation



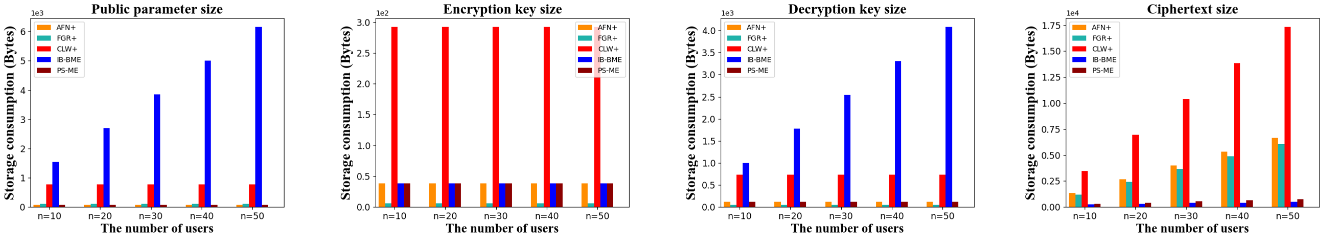
(a) Setup algorithm (b) EKGen algorithm (c) SKGen algorithm

Fig. 2: Running time at the side of registry authority for Setup, EKGen and SKGen algorithms



(a) Enc algorithm (b) Dec algorithm

Fig. 3: Running time at the side of clients for Enc and Dec algorithms



(a) Public parameter (pp) size (b) Encryption key (ek) size (c) Decryption key (dk) size (d) Ciphertext (ct) size

Fig. 4: Storage consumption of pp, ek, dk and ct

costs of the setup (**Setup**), decryption (**Dec**) and secret key (**DKGen**) generation phases in [20]–[22] and PS-BME are constant, while the costs of our IB-BME follow a linear-relationship with the maximum number of system recipients. We can also summarize that the costs of encryption key generation (**EKGen**) phase in all works is also constant. From TABLE II, we can summarize our PS-BME almost has relatively lower calculation costs than other works in terms of the **Enc**, **Dec**, **Setup**, **DKGen** and **EKGen** phases.

As seen from TABLE III, it is not hard to find that the storage costs of storing public parameter (pp) in all works except IB-BME are constant. The storage costs of **EKGen** for producing encryption key (ek) in all works are also constant, and the storage overhead of **Dec** for creating decryption key (dk) in [20]–[22] and PS-BME are constant. We can also observe that the ciphertext storage costs in all works are linearly increasing with the number of (system) recipients. Generally, smaller storage costs of storing dk and ct imply efficient decryption. In our PS-BME, it has relatively lower storage costs of storing dk and ct, thus leading to efficient decryption.

To summarize, our PS-BME enables desirable calculation and storage costs compared to other works. Besides, from TABLE I, it can be concluded that our PS-BME features all the satisfactory properties especially realizing CCA security. In other words, our PS-BME not only achieves desirable efficiency but also ensures stronger security compared to other methodologies.

B. Experimental Analysis

The experimental performance evaluation is implemented with Python 3.6.13 using Charm 0.43, PBC-0.5.14 library, OpenSSL-1.1.1. We also conduct the simulations on a laptop with an Intel Core i9-9900K CPU @ 3.6GHz*16 and 32GB

RAM running the 64-bit Ubuntu 18.04.5 LTS, which can be seen as cloud servers. Besides, a Raspberry Pi 4 Model B device with Broadcom BCM 2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz and 2GB RAM running the Raspbian plays as the role of a mobile user. In our implementation, the 128-bit AES keys are utilized to encode the real data (refer to medical images, https://www.smir.ch/BRATS/Start2015) based on a modified AES algorithm [37] and the encryption algorithm of our IB-BME, PS-ME and other related works are used to encode the AES keys. Please note that when the number of shared users equals 1, each of all the schemes can be initialized as an IB-ME scheme. The experimental source codes could be publicly visited in *git@github.com:xuehuan-yang/PSME.git.*

Fig. 2 exhibits the running time comparisons at the side of the authority for Setup, EKGen & SKGen algorithms. As shown in Fig. 2(a) & Fig. 2(b), we can find that the running time of the Setup & EKGen algorithms in all works is always constant and running Setup & EKGen algorithms in our PS-ME takes moderate computation consumption. In addition, since our IB-BME and PS-ME preserve the original structure of the encryption key as that in AFN+ [20], the time-consumption of the EKGen algorithm in IB-BME and PS-ME is almost the same as that in AFN+ [20]. From Fig. 2(c), it is easy to learn that the DKGen computation cost in IB-BME grows linearly with the number of users (*i.e.*, the length of identity vector), and our PS-ME takes relatively-less computation overhead in performing the DKGen algorithm than CLW+ [22].

Fig. 3 shows the execution time comparisons at the side of the clients for Enc and Dec algorithms. From Fig. 3(a), it is straightforward to summarize that the time to run the Enc algorithm in all works increases linearly with the number of users, and our PS-ME has relatively lower time consumption in performing Enc algorithm than CLW+ [22]. Additionally, our IB-BME has the lowest time consumption compared to other works including CLW+ [22]. As seen from Fig. 3(b), we can observe that the computation cost of all works except IB-BME are almost constant when implementing the decryption while our PS-ME has moderate computation costs among the comparison works.

Fig. 4 presents the storage cost comparisons of conducting Setup, EKGen, DKGen and Enc to produce corresponding pp, ek, dk and ct. From Fig. 4(a), it is easy to conclude that our PS-ME has a relatively lower storage cost for storing pp, the CLW+ and our IB-BME need more storage space to store pp. It can be derived from Fig. 4(b) that our IB-BME & PS-ME almost take the same storage cost as AFN+ and need smaller storage space than CLW+. From Fig. 4(c), it can be observed that our PS-ME has relatively smaller storage costs than the other works except FGR+ [21]. As revealed from Fig. 4(d), we can easily learn that our PS-ME and IB-BME are the two lowest schemes requiring the storage resource for storing ct.

In summary, since our PS-ME has relatively lower costs regardless of computation and storage costs and enables stronger CCA-security assurance, our PS-ME is more appropriate for real-world applications.

VIII. Conclusion

In this paper, we first suggested an identity-based broadcast matchmaking encryption (IB-BME) to solve the inflexible issue of existing ME for data sharing, which enables both participants to specify respective access policies to the encrypted data, such that the data can be revealed by multiple recipients if both access policies hold. In IB-BME, a general matchmaking transformation solution is initialized for one-to-many matchmaking. Then, we designed a privacy-aware and security-enhanced efficient ME (PS-ME) based on the IB-BME's matchmaking transformation methodology, which not only inherits all satisfactory functionalities of the IB-BME, but also achieves recipients' identity anonymity and CCA-security against active attacks. The PS-ME for the first time solved the open problem of how to implement CCA-security left by ME posed in CRYPTO 2019. We also demonstrated that IB-BME and PS-ME are CPA-secure and CCA-secure through comprehensively strict security proofs. The performance was evaluated via experimental simulations to indicate the practicability and effectiveness of our PS-ME.

Acknowledgments

This work was supported in part by the Singapore Ministry of Education (MOE) AcRF Tier 1 RS02/19, and AcRF Tier 1 RG108/19 (S).

References

[1] S. Coutu, I. B. Reshef, A. K. Whitcraft, et al., "Food security: underpin with public and private data sharing", *Nature,* vol. 578, no. 7796, pp. 515-516, 2020.

[2] M. Manulis, C. P. Bridges, et al., "Cyber security in new space: analysis of threats, key enabling technologies and challenges", *International Journal of Information Security,* vol. 20, pp. 287-31, 2021.

[3] J. Sun, J. Tao, H. Zhang, et al., "A Tamper-Resistant Broadcasting Scheme for Secure Communication in Internet of Autonomous Vehicles", *IEEE Transactions on Intelligent Transportation Systems,* DOI: 10.1109/TITS.2023.3265403, 2023

[4] Y. Bao, W. Qiu, X. Cheng, et al., "Fine-grained data sharing with enhanced privacy protection and dynamic users group service for the IoV", *IEEE Transactions on Intelligent Transportation Systems,* DOI: 10.1109/TITS.2022.3187980, 2022.

[5] J. Sun, G. Xu, et al., "Verifiable, Fair and Privacy-Preserving Broadcast Authorization for Flexible Data Sharing in Clouds", *IEEE TIFS,* vol. 18, pp. 683-698, 2022.

[6] H. Deng, Z. Qin, Q. Wu, et al., "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud", *IEEE TIFS,* vol. 15, pp. 3168-3180, 2020.

[7] S. Agrawal, A. Yadav, S. Yamada, "Multi-input attribute-based encryption and predicate encryption", *CRYPTO 2022*, Springer, pp. 590-621, 2022.

[8] I. Kim, W. Susilo, J. Baek, et al., "Harnessing policy authenticity for hidden ciphertext policy attribute-based encryption", *IEEE TDSC,* vol 19, no. 3, pp. 1856-1870, 2020.

[9] H. Deng, Z. Qin, et al., "Achieving fine-grained data sharing for hierarchical organizations in clouds", *IEEE TDSC,* vol. 20, no. 2, pp. 1364-1377, 2022.

[10] R. LaVigne, A. Lincoln, V. V. Williams, "Public-key cryptography in the fine-grained setting", *Crypto 2019* pp. 605-63, 2019.

[11] P. Yi, J. Li, C. Liu, et al., "An efficient identity-based signature scheme with provable security", *Information Sciences (INS),* vol. 576, pp. 790-799, 2021.

[12] Y. Zhao, Y. Wang, Y. Liang, et al., "Identity-Based Broadcast Signcryption Scheme for Vehicular Platoon Communication", *IEEE TII,* DOI: 10.1109/TII.2022.3203724, 2022.

[13] J. Yu, S. Liu, S. Wang, et al., "LH-ABSC: a lightweight hybrid attribute-based signcryption scheme for cloud-fog-assisted IoT", *IEEE IoTJ,* vol. 7, no. 9, pp. 7949-7966, 2020.

[14] J. Sun, G. Xu, T. Zhang, et al., "A practical fog-based privacy-preserving online car-hailing service system", *IEEE TIFS,* vol. 17, pp. 2862-2877, 2022.

[15] S. Mastorakis, A. Mtibaa, "Towards service discovery and invocation in data-centric edge networks", *ICNP 2019*, IEEE, pp. 1-6, 2019.

[16] J. Sun, G. Xu, T. Zhang, et al., "Share your data carefree: An efficient, scalable and privacy-preserving data sharing service in cloud computing", *IEEE Transactions on Cloud Computing,* vol. 11, no. 1, pp. 822-838, 2023.

[17] D. Balfanz, G. Durfee, N. Shankar, et al., "Secret handshakes from pairing-based key agreements", *IEEE S&P,* pp. 180–196, 2003.

[18] S. Arecki, J. Kim, et al., "Beyond secret handshakes: Affiliation-hiding authenticated key exchange", *CT-RSA,* pp. 352–369, 2008.

[19] Z. An, J. Pan, Y. Wen, et al., "Forward-Secure Revocable Secret Handshakes from Lattices", *International Conference on Post-Quantum Cryptography,* Springer, Cham, pp. 453-479, 2022.

[20] G. Ateniese, D. Francati, D. Nunez, et al., "Match me if you can: matchmaking encryption and its applications", *CRYPTO 2019,* vol. 11693, pp. 701–731, 2019.

[21] D. Francati, A. Guidi, L. Russo, et al., "Identity-Based Matchmaking Encryption Without Random Oracles", *INDOCRYPT 2021*, Springer, Cham, pp. 415-435, 2021.

[22] J. Chen, Y. Li, J. Wen, J. Weng, "Identity-Based Matchmaking Encryption from Standard Assumptions", *ASIACRYPT'2022*, https://eprint.iacr.org/2022/1246, 2022.

[23] S. Xu, J. Ning, Y. Li, et al., "Match in my way: Fine-grained bilateral Access Control for Secure Cloud-fog Computing", *IEEE TDSC,* DOI: 10.1109/TDSC.2020.3001557, 2020.

[24] J. Sun, Y. Yuan, M. Tang, et al., "Privacy-preserving Bilateral Fine-grained Access Control for Cloud-enabled Industrial IoT Healthcare", *IEEE TII,* vol. 18, no. 9, pp. 6483 - 6493, 2022.

[25] D. Boneh, X. Boyen, E. J. Goh, "Hierarchical identity based encryption with constant size ciphertext", *EUROCRYPT 2005*, pp. 440-456, 2005.

[26] K. He, J. Weng J, et al., "Anonymous identity-based broadcast encryption with chosen-ciphertext security", *AsiaCCS 2016*, pp. 247-255, 2016.

[27] P. Xu, J. Li, W. Wang, et al., "Anonymous identity-based broadcast encryption with constant decryption complexity and strong security", *AsiaCCS 2016*, pp. 223-233, 2016.

[28] J. Bartusek, B. Carmer, A. Jain, et al., "Public-key function-private hidden vector encryption (and more)", *EUROCRYPT 2019*, Springer, Cham, pp. 489-519, 2019.

[29] F. Kitagawa, T. Matsuda, "CPA-to-CCA transformation for KDM security", *TCC*, Springer, Cham, pp. 118-148, 2019.

[30] H. Wang, Y. Zhang, K. Chen, et al., "Functional broadcast encryption with applications to data sharing for cloud storage", *INS,* vol. 502, pp. 109-124, 2019.

[31] J. Lai, Y. Mu, F. Guo, et al., "Identity-based broadcast encryption for inner products", *The Computer Journal,* vol. 61, no. 8, pp. 1240-1251, 2018.

[32] Y. Wang, J. Pan, Y. Chen, "Fine-grained secure attribute-based encryption", *CRYPTO 2021*, Springer, Cham, pp. 179-20, 2021.

[33] C. P. Mejia, J. V. Medina., "Lattice-based cryptoprocessor for CCA-secure identity-based encryption", *IEEE Transactions on Circuits and Systems I*, vol. 67, no. 7, pp. 2331-2344, 2020.

[34] C. Gentry, B. Waters, "Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts)", *EUROCRYPT 2009*, Springer, LNCS 5479, pp. 171-188, 2009.

[35] S. C. Ramanna, "More Efficient Constructions for Inner-product Encryption", *ACNS*, Springer, Cham, LNCS. 9796, pp. 231-248, 2016.

[36] S. Agrawal, M. Chase, "FAME: fast attribute-based message encryption", *CCS 2017*, pp. 665-682, 2017.

[37] M. Zeghid, M. Machhout, L. Khriji, et al., "A modified AES based algorithm for image encryption", *IJCSE,* vol. 1, no. 1, 70-75, 2007, (https://github.com/JHUISI/charm).

Jianfei Sun received his Ph.D. degree from the University of Electronic Science and Technology of China (UESTC). He is currently a research fellow at the School of Computer Science and Engineering, Nanyang Technological University. His research interests include network security and IoT security. He has published many papers on IEEE TDSC, IEEE TIFS, IEEE TII, IEEE TCC, IEEE TVT, IEEE IoTJ, Inf. Sci, IEEE Systems, etc. His research interests include network security and IoT security.

Guowen Xu is currently a Research Fellow with Nanyang Technological University, Singapore. He received his Ph.D. degree in 2020 from the University of Electronic Science and Technology of China. He has published a wealth of papers in reputable conferences/journals, including ACM CCS, NeurIPS, ECCV, IEEE TIFS, TDSC, ASIACCS, ACSAC, ESORICS, etc. He is the recipient of the Best Paper Award of the 26th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2020), the Best Student Paper Award of the Sichuan Province Computer Federation (SCF 2019), the Student Conference Award of IEEE International Conference on Computer Communications (INFOCOM 2020), and the Distinguished Reviewer of ACM Transactions on the Web. His research interests include applied cryptography and privacy-preserving Deep Learning. His research interests include applied cryptography and privacy-preserving issues in Deep Learning.

Tianwei Zhang is an assistant professor at School of Computer Science and Engineering, at Nanyang Technological University. His research focuses on computer system security. He is particularly interested in security threats and defenses in machine learning systems, autonomous systems, computer architecture and distributed systems. He received his Bachelor's degree at Peking University in 2011, and the Ph.D degree at Princeton University in 2017. He is serving on the editorial boards of IEEE Transactions on Circuits and Systems for Video Technology, ACM Transactions on Sensor Networks.

Xuehuan Yang received his Bachelor's degree from Nanyang Technological University. Now he is a Ph.D in School of Computer Science and Engineering, at Nanyang Technological University. His research focuses on software engineering and secure autonomous vehicle technology.

Mamoun Alazab (Senior Member, IEEE) is currently a full professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. His research is multidisciplinary that focuses on cyber security and digital forensics of computer systems. He works closely with the government, industry and some top scientists. He also served on the editorial boards of many international journals including IEEE Transactions on Computational Social Systems, Journal of Information Security and Journal of Cybersecurity and Privacy, etc.

Robert H. Deng (F'16) is AXA Chair Professor of Cybersecurity, Director of the Secure Mobile Centre, and Deputy Dean for Faculty & Research, School of Computing and Information Systems, Singapore Management University (SMU). His research interests are in the areas of data security and privacy, network security, and applied cryptography. He received the Outstanding University Researcher Award from National University of Singapore, Lee Kuan Yew Fellowship for Research Excellence from SMU, and Asia-Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium. He serves/served on the editorial boards of ACM Transactions on Privacy and Security, IEEE Security & Privacy, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, Journal of Computer Science and Technology, and Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. He is a Fellow of IEEE and Fellow of Academy of Engineering Singapore.