Verifiable, Fair and Privacy-preserving Broadcast Authorization for Flexible Data Sharing in Clouds

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

***Abstract*—The cloud-based data sharing technology with cryptographic primitives enables data owners to outsource data into paradigms and privately share information with arbitrary recipients without geographic barriers. However, we argue that most of existing efforts for outsourced data sharing are either inefficient, inflexible, or incompletely secure due to the following problems: (1) lack of efficient strategies for dynamically designating target ciphertexts to multiple recipients; (2) how to hide the identity of the recipient and (3) how to verify the correctness of outsourced ciphertext transformation without any denial. To the best of our knowledge, no previous work has thoroughly explored the above three issues, motivating us to design such an efficient and comprehensively secure outsourced data sharing mechanism.**

We design VF-PPBA, the first Verifiable, Fair and Privacy-preserving Broadcast Authorization framework for flexible data sharing in clouds. In more detail, we first invent a new primitive, privacy-preserving multi-recipient broadcast proxy re-encryption (PPMR-BPRE), which enables the authorization of a given ciphertext to different recipients with efficient ciphertext transformation, and further guarantees that any malicious adversary deduces nothing about the identity of the recipient. Then, we present VF-PPBA for flexible data sharing with PPMR-BPRE as the underlying structure, which in addition to inheriting all the functionalities of PPMR-BPRE, is capable of supporting the verifiability of the outcome correctness of the outsourced conversion task, and being immune to the malicious accusation if the outsourcing outcome is correctly completed. We formalize the adversarial models and render comprehensively strict security proofs to prove the security of our proposed solutions. Its performance is also validated via experimental simulations to showcase the practicability and effectiveness.

***Index Terms*—Privacy-preserving, malicious accusation resistance, verifiability, fairness and practicability.**

I. Introduction

CLOUD computing has progressively become the dominant paradigm for individuals or businesses to share data, primarily enabled by its powerful storage and computational capabilities [1]–[3]. For instance, cloud service platforms, such as Google Health [4], Microsoft Health Vault [5], are available to all participants for conveniently and efficiently sharing health information with different medical institutions or individuals. To protect the privacy of data outsourced to third-party clouds, many representative works have been proposed to encrypt sensitive information, mainly relying on cryptographic primitives including identity-based encryption [9]–[11] and attribute-based encryption [6], [7], etc. In such a scenario, a data owner first performs data encryption, and then outsources encrypted data to the clouds, so that the data can be accessed by any recipient owning the authorization key.

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

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).

A. Efficiency and Security Concerns

While private data sharing techniques have been studied for decades, we argue that the state-of-the-art efforts still suffer from shortcomings in efficiency and security, summarized as follows.

1) Lack of efficient strategies for dynamically designating ciphertexts to multiple recipients: Outsourced data are often encrypted so that only certain authorized recipients can access them. This raises the question of how to authorize the data to other recipients beyond those already authorized in the system. This is necessary since sharing personal information or medical records with additional recipients for communication or diagnosis is frequently common. A naive approach would be to decrypt-then-re-encrypt a given message, and then outsource it to the cloud to facilitate the access by new recipients. This is obviously inefficient since it involves the data owner to encrypt particular data several times and requires the server to store multiple copies of the target data.

Many representative works have been designed to alleviate the above problems, among them the most advanced works are mainly based on three cryptographic primitives: attribute-based proxy encryption (ABPRE) [33]–[35], inner product proxy re-encryption (IP-PRE) [37]–[39] and multi-recipient broadcast proxy re-encryption (MR-BPRE) [36]. Although the above solutions enable sharing ciphertexts dynamically with multiple recipients, these efforts are either inefficient or insecure. Specifically, ABPRE enables the ciphertext transformation from a set of recipients to another new set of recipients, with specialized encryption/re-encryption methods based on the access policy. This is computationally intensive and the complexity grows linearly with the expressiveness of the access policy (See Section II-C for more details). IP-PRE is also commonly used to realize data sharing for dynamic recipients with the technology of ciphertext conversion. However, beyond the inefficiency issue, it requires an always-on entity to act as a key generation center (KGC) for generating authorization tokens for ciphertext conversion (See Section II-C). This is difficult to implement in practical applications. The MR-BPRE solution as a novel approach is capable of supporting efficiently flexible data sharing with new recipients

beyond those previously authorized. However, the existing MR-BPRE technique also suffers from the same weaknesses as that of ABPRE.

2) Difficulty of hiding the identities of recipients: Ideally, the sharing of outsourced ciphertext data should be absolutely private, *i.e.*, plaintext information and the identities of the data recipients should be blind to other entities in the system. This is necessary especially in medical scenarios. Assuming an adversary can deterministically infer that a certain patient's medical records were sent to a particular doctor, it has an overwhelming probability to infer the disease a certain patient may be suffering from the doctor's identity. Conventional solutions [25]–[27] to preserve the identity privacy are to partition an identity into two blind parts, such that an adversary fails to discern the specific counterpart by bilinear pairings. Apparently, such a solution dividing an identity into two parts used for encoding is inefficient compared to that of the identity as a whole. To alleviate this problem, IP-PRE is commonly used to hide the recipient's identity by customizing the inner product-based proxy re-encryption technique. However, as previously discussed, the IP-PRE implementation is typically resource-intensive and needs additional KGC to create authorization tokens for real-time ciphertext conversion. Therefore, how to design such an efficient data sharing mechanism to perfectly hide the identities of dynamic data recipients is still an open problem.

3) Difficulty of verifying the correctness of outsourced ciphertext transformation without any denial: As described above, to realize that a given ciphertext can be accessed by any recipient, the mainstream method is to perform appropriate ciphertext transformation on the ciphertext, thereby facilitating decryption by new recipients for the new ciphertext. To boost efficiency, a typical method is to migrate complex operations in ciphertext conversion to cloud servers. However, this raises two new security challenges: (a) How to ensure that the untrusted cloud server correctly performs the specified outsourced computing tasks. Since outsourced computing tasks are usually resource-intensive, a "lazy" cloud server may perform only part of the computations, or even return random ciphertexts to the recipient in order to save overhead. (b) How to treat the computation results returned by the server fairly, *i.e.*, to prevent denial from malicious recipients if the server faithfully performs the computations. To the best of our knowledge, the existing efforts rarely tackle the above challenges in the field of outsourced data sharing.

B. Technical Goals and Challenges

1) Technical goals: In this paper, we focus on designing a verifiable, fair and privacy-preserving broadcast authorization (VF-PPBA) scheme for efficiently flexible data sharing in clouds. We aim to accomplish the following goals: (a) The authorization token should be able to be solely-produced and delegated by any granted recipient to cloud servers for realizing flexible ciphertext conversion from a set of initial recipients to another set of new extra recipients. (b) For either original ciphertext or transformed ciphertext, the identity of any recipient should be always anonymized for the prevention of some privacy disclosure of both recipients and sharers. (c) VF-PPBA should enable verifiability and malicious accusation resistance for the outcome of the outsourced ciphertext conversion tasks. In other words, any recipient and any third-party verifier in VF-PPBA should be capable of efficiently verifying the correctness of the outsourced outcome.

2) Challenges: The fundamental challenges to build such a VF-PPBA scheme mainly originate from two aspects: i) how to construct a privacy-preserving multi-recipient broadcast proxy re-encryption (PPMR-BPRE), which simultaneously enables the realization of identity anonymity and efficiently flexible ciphertext conversion. ii) On the basis of the PPMR-BPRE, how to verify the correctness of outsourced computing without any denial.

The first dilemma seems to be naturally settled by the following integration of anonymous IBBE (AIBBE) [25]–[27] and PRE [28]–[32] or MR-BPRE [36], anonymous IBE (AIBE) [15]–[18]. However, reaching this goal efficiently and seamlessly is non-trivial because of the following problems.

(a) *It is tricky to integrate AIBBE and PRE to construct an efficient PPMR-BPRE.* In brief, it is a major challenge for recipients to independently create a proper authorization token embedding a new access control without holding the master secret key. The reason is that an authorization token is essentially a special form of a secret key, which implies that it must need the participation of the master secret keys in the generation of an authorization token. It is also intractable to ensure that the original structure of the AIBBE-based ciphertext would not be destroyed after the authorization token is used to complete the ciphertext conversion. This is mainly due to the fact that the generated authorization token related to the set of recipients' identities commonly features two types of structures of both decryption key and ciphertext simultaneously, which inevitably affects the structure of the original ciphertext associated with a set of previously designated recipients' identities.

(b) *There is no straightforward approach to integrating the MR-BPRE type ciphertext and AIBE type ciphertext*. The reasons stem from the following aspects: the two forms of ciphertexts in these various primitives are separately created but must be merged in a cohesive manner based on the same public keys. However, it is complicated to merge two separated ciphertexts with diverse and low coupling into a new ciphertext in an elegant manner. Furthermore, using distinct types of identities in two primitives leads to the failure of mutual integration. Specifically, the user identity in MR-BPRE as a whole symbol is embedded into a ciphertext while that in AIBE is generally and randomly split into two blind components to impede its recognition via the bilinear map. Even if the identity could be partitioned into two blind pieces to achieve identity anonymity in MR-BPRE, it is challenging to broadcast the ciphertext to multiple recipients using two separated identity parts.

The second conundrum appears to be elegantly resolved by the verifiable outsourced computation (VOC) technique [41]–[43] and zero-knowledge proof (ZKP) technology [44]–[46]. Indeed, the private validation of verifiability could be easily achieved by the VOC technique since the ciphertext conversion delegated to a third party (*e.g.*, cloud server)

is essentially a special instance of outsourced computation. Although the public verification of fairness could be realized by the ZK proof, which not only empowers an untrustworthy third party to produce proof of the computation (without any secret disclosure), but also enables anyone holding verification keys to publicly check the correctness of the third party's output. Regrettably, applying such a technique in a PPMR-BPRE scheme leads to significant calculations related to proof generation, which could become even worse if the PPMR-BPRE scheme is highly complex.

C. Our Contributions

To address the above problems, We design VF-PPBA, the *first-ever* verifiable, fair and privacy-preserving broadcast authorization scheme for efficiently flexible data sharing in clouds. The key novelties of our VF-PPBA are yielded from the following insights: (a) We observe that the inner product encryption (IPE) could be employed for constructing the PPMR-BPRE by combining it with PRE technique. To achieve identity anonymity and flexible ciphertext conversion without KGC, we use the vector generation method to correspondingly transform the user's identity and identity access list into an identity vector and an access vector, such that these vectors can be perfectly hidden in the produced decryption key and ciphertext (*e.g.*, original and converted ciphertexts). We exploit the structure-preserving technique to enable authorized recipients to self-produce an authorization token. (b) To guarantee the verifiability and fairness, the verifiable outsourced computation technique and the commitment technology are exploited. Specifically, the solution for validating the output correctness of the outsourced computations can be realized by adding redundant ciphertext, in which encrypting the plaintext and a random value together and committing the random value to the plaintext are done. The commitment for checking the proof of correctness can be created by adding the redundant part in the authorization token.

The contributions of this paper can be summarized as follows:

- *Secure Flexible Data Sharing*: To support secure flexible data sharing, both the PPMR-BPRE and VF-PPBA schemes permit a sender to share his/her data with a set of recipients while also allowing any initial recipient to forward raw data to another set of recipients for access.
- *Privacy-preserving of Identity*: To provide privacy-preserving of the recipients' identities, both the PPMR-BPRE and VF-PPBA schemes are immune to adversarial users for inferring the identity privacy from the ciphertext regardless of original and transformed ones.
- *Verifiability and Fairness*: To enable verifiability and fairness, the VF-PPBA scheme empowers any recipient to privately check the correctness of the output of the outsourced tasks. Furthermore, it also provides a verification mechanism to publicly verify the reliability of the outsourced outcome to impede the malicious accusation for cloud servers.
- *Security and Efficiency*: To realize various data security requirements, PPMR-BPRE makes a strong security assumption, *i.e.*, the cloud servers are considered as semi-trusted entities. In contrast, the VF-PPBA scheme considers a weaker security assumption to ensure stronger security, *i.e.*, the cloud servers are supposed to be malicious. To achieve higher efficiency, both the PPMR-BPRE and VF-PPBA schemes are built based on the most computationally-friendly IPE solution. Besides, the efficiency of both methods could be further improved by our solution depicted in Section VII. These considerations can make our framework more efficient and practical in real scenarios.

Besides, we rigorously prove the security of both PPMR-BPRE and VF-PPBA. Moreover, the experimental results demonstrate our methodologies relatively outperform existing solutions. The experimental source codes can be publicly found in https://github.com/xuehuan-yang/VFPPBA.

II. Related Works

A. Secure Outsourced Data Protection

As extensively well-known methods, cryptographic encryption approaches have been widely utilized to secure outsourced data stored in clouds. Conventional public-key encryption solutions have been adopted to realize user-centric cloud data management, however, this kind of solution has severe certificate management burdens for all users. As a solution to completely eliminate the burden brought by certificate management, identity-based encryption (IBE) allows any recognizable strings (such as email addresses or other identifiers) to serve as public keys and enables a trusted key generator center to create private keys for all users. Since the first seminal introduction of IBE was introduced by Boneh and Franklin [9], various IBE schemes have been intensively researched for IoT and mobile cloud due to its efficient key management, and lightweight encryption and decryption operations. For example, an IBE was employed by Wei *et al.* [10] to secure data sharing in a cloud computing environment. Karati *et al.* [11] exploited an IBE scheme to realize secure data sharing in a cyber-physical cloud environment. He *et al.* [12] utilized an IBE to establish a cross-domain handshake protocol with symptoms-matching in mobile healthcare social networks. Yang *et al.* [13] used an IBE to invent a secure data possession protocol with compressed cloud storage. While the above IBE-based solutions enable secure data protection in different practical scenarios, they all fail to consider identity privacy protection. To realize identity anonymity, Abdalla *et al.* [14] proposed an anonymous IBE scheme, which splits the user identity randomly into two blind parts, thus hindering its recognition by the bilinear pairing. After that, a number of anonymous IBE (AIBE) schemes [15]–[18] are put forward. While AIBE schemes can elegantly secure outsourced data, they only support one-to-one data sharing instead of one-to-many data sharing.

B. One-to-many Recipient Data Sharing

To securely share the outsourced encrypted data with multiple receivers, one-to-many cryptographic methods have also been extensively investigated. Currently, there are mainly two

technical categories: attribute-based encryption (ABE) [6], [7] and broadcast encryption (BE) [8]. The ABE scheme enables its produced ciphertext to be bound to a specific access structure/attribute set, such that only the user whose attribute set/access structure embedded in a private key matches the access structure/attribute set can decode the ciphertext. Although ABE enables one-to-many fine-grained data sharing, either the ciphertext size or private key size commonly increases linearly with the number of attributes. In general, constant-size ciphertext and private key indicate constant decryption computation overhead. However, the decryption computation overhead in most ABE works is heavy since it linearly scales with the number of attributes, which makes these solutions unaffordable to resource-constraint end-users. As an alternative one-to-many data sharing technique, identity-based broadcast encryption (IBBE) [19] permits a user to designate an identity list to regulate who can access the data. Compared to the ABE technique, IBBE enables more efficient decryption operations. To date, lots of IBBE schemes [20]–[24] have been done concerning functionality, security and efficiency. Whereas, these efforts rarely considered identity privacy-protection. To protect the privacy protection of user identity from being learned, a number of anonymous IBBE (AIBBE) schemes via identity partitioning solutions to impede identity recognition are proposed [25]–[27]. Though AIBBE realizes a secure multi-recipient data sharing, it is incapable of flexibly supporting data sharing with new recipients beyond those initially specified.

C. Flexible Ciphertext Transformation

To address the dilemma of sharing encrypted data with new users beyond those initially designated, the concept of proxy re-encryption (PRE) was invented by Blaze *et al.* [28]. PRE enables a proxy to be authorized to convert Alice's ciphertext into Bob's ciphertext, such that the capability of deciphering the ciphertext can be shifted from Alice to Bob. As a combination technique of IBE and PRE, identity-based PRE (IBPRE) preserves all PRE's merits and additionally solves the cumbersome certificate management problem. To realize the collusion attack resistance, an IBPRE scheme was suggested by Zhang *et al.* [29] to prevent the proxy and authorized users from further sharing the re-encrypted ciphertext. To support a single-recipient ciphertext to be converted into a multi-recipient ciphertext, Xu *et al.* [30] proposed a conditional identity-based broadcast PRE (IB-BPRE) that allows a delegator to conditionally re-encrypt the ciphertext. Ge *et al.* [31] proposed an IB-BPRE scheme, which enables the transformation of a ciphertext for a recipient into a new one for a set of recipients. Very recently, Deng *et al.* [32] also presented an IB-BPRE scheme that can convert a single-receiver's ciphertext into mult-receiver's broadcast ciphertext. Unfortunately, these IB-BPRE schemes neither support the ciphertext transformation from multiple-recipients to a set of new recipients nor realize the correctness checking of the re-encrypted ciphertext executed by the cloud servers.

As a solution to enable dynamical ciphertext conversion, the attribute-based proxy re-encryption scheme (ABPRE) initially proposed by Liang *et al.* [33] could realize the designation of a proxy who re-encrypts a ciphertext related to a certain access policy to another one with a different access policy, where these two access policies define respective access recipients. Following this work, Liang *et al.* [34] also put forward an ABPRE scheme that achieves more expressiveness of access control and efficient data sharing with a specified set of recipients matching the policy. However, this solution has prohibitive decryption computation overhead, and fails to ensure the ciphertext transformation correctness. After that, Ge *et al.* [35] proposed a novel ABPRE scheme. A user associated with a set of attributes is permitted to determine the correctness of transformed ciphertext, thus detecting the malicious behavior of the proxy. Besides, this solution can protect the proxy from malicious accusations if the ciphertext transformation is correctly done. While ABPRE enables the ciphertext conversion from multi-recipients to some new multi-recipients, the existing ABPRE schemes either have computationally-heave decryption costs, or fail to ensure strong privacy-protection, *e.g.,* non-authorized users can learn some attribute privacy from the ciphertext due to the attributes in the cleartext form directly attached to the ciphertext. Very recently, Deng *et al.* [36] introduced IBBE and ABE into PRE to design a multi-recipient broadcast PRE scheme (MR-PRE), which also achieves the ciphertext transformation from a set of recipients to other multi-recipients. However, the decryption cost is heavy and the privacy of recipients' identities is not considered.

To preserve all merits of ABPRE and additionally achieve stronger privacy of identity or attributes, inner product proxy re-encryption (IP-PRE) was initially formulated by Backes *et al.* [37]. However, apart from the construction based on composite orders, one serious drawback in their scheme is that the re-encryption key generation can be only produced by the trusted key generation center (KGC) instead of the delegators themselves, which means that the KGC needs to be online at any time. Clearly, it is impractical for the real-world applications. Following Backes *et al.*'s work, Sepehri *et al.* [38], [39] proposed new IP-PRE schemes based on prime order groups. However, these schemes have the same serious defect as that in [37]. Besides, existing IP-PRE schemes all assume that the proxy is fully trusted to perform the mission given by the delegators, thus incapable of providing the correctness verification of ciphertext conversion.

In this paper, we mainly aim to realize an efficiently flexible data sharing primitive called verifiable, fair and privacy-preserving broadcast authorization (VF-PPBA) scheme based on our PPMR-BPRE, in which a delegator can be allowed to dependently create a re-encryption key (known as an authorization token) to convert the original ciphertext for multi-recipients into a new one for a group of new recipients. Our VF-PPBA also allows any authorized recipient to efficiently access the respective accessible ciphertext without leaking any identity privacy. Moreover, our VF-PPBA enables the detection of whether the proxy honestly performs the ciphertext conversion as well as protecting the proxy (i.e., cloud servers) from malicious accusations if the ciphertext transformation is correctly conducted. In summary, this paper proposes a VF-

PPBA scheme, which is the first to simultaneously realize efficient multi-recipient sharing of encrypted data, identity anonymity, verifiable ciphertext conversion and malicious accusation resistance.

III. Preliminaries and Definitions

This section first introduces the notations, complexity assumptions and message-locked encryption used in our solutions. Afterward, it illustrates the system architecture and describes the threat model and security goals.

A. Notation

The following notations are summarized in TABLE I.

TABLE I: Notations

Notation	Description
$H, \mathcal{H}_1, \mathcal{H}_2$	Hash functions
$\mathbf{x}/\mathbf{y}$	Access vector/Identity vector
$\mathsf{ct}_{\mathbf{x}}$	Ciphertext related to an access vector $\mathbf{x}$
$\mathsf{sk}_{\mathbf{y}}$	Secret key associated with an identity vector $\mathbf{y}$
$\mathsf{ct}_{\mathbf{x}\rightarrow\mathbf{w}}$	Converted ciphertext related to an access vector $\mathbf{w}$
$\mathsf{at}_{\mathbf{y}\rightarrow\mathbf{w}}$	Authorization token with $\mathbf{y}$ converting $\mathsf{ct}_{\mathbf{x}}$ into $\mathsf{ct}_{\mathbf{x}\leftarrow\mathbf{w}}$
π	a proof showing outsourced conversion outcome

B. Complexity Assumptions

Our PPMR-BPRE and VF-PPBA schemes depend on the following complexity assumptions:

***Definition** 1:* (DL assumption [35]) Let g be a generator of multiplicative cyclic group $\mathbb{G}$ with its prime order p. Given a tuple (g, g^{ϱ}) of Discrete Logarithm (DL) assumption, where ϱ is a random element of $\mathbb{Z}_p$, it is difficult for all adversaries $\mathcal{A}$ to derive the integer ϱ.

***Definition** 2:* (DBDH assumption [34]) Given a tuple $(g, g^a, g^c, u, u^a, u^b, \mathcal{D})$ of asymmetric **D**ecisional **B**ilinear **D**iffie-**H**ellman (DBDH) assumption, where g, u are corresponding generators of multiplicative cyclic groups $\mathbb{G}_0$ and $\mathbb{G}_1$, a, b, c are random elements of $\mathbb{Z}_p$. For all adversaries $\mathcal{A}$, it is hard to judge $\mathcal{D} = e(g,u)^{abc}$ or $D = \mathcal{R}_1$, where $\mathcal{R}_1$ is a random element of group $\mathbb{G}_T$.

***Definition** 3:* ($\mathcal{P}$-DBDH assumption [47]) Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be multiplicative cyclic groups with corresponding generators $g \in \mathbb{G}_0, u \in \mathbb{G}_1$. Given a tuple $(g, g^a, g^{ab}, g^c, u, u^a, u^b, \mathcal{D})$ of $\mathcal{P}$-asymmetric **D**ecisional **B**ilinear **D**iffie- **H**ellman ($\mathcal{P}$-DBDH) assumption, where a, b, c are also random elements of $\mathbb{Z}_p$. For all adversaries $\mathcal{A}$, it is intractable to differentiate $\mathcal{D} = g^{abc}$ or $D = \mathcal{R}_2$, where $\mathcal{R}_2$ is a random element of group $\mathbb{G}_0$.

C. Message-locked Encryption (MLE)

There are five algorithms involved in a message-lock encryption (**MLE**) scheme. In detail, the parameter generation algorithm Setup receives a security parameter 1^{λ} and outputs the public parameter $\mathsf{pp} \leftarrow \mathsf{Setup}(1^{\lambda})$; the key generation algorithm KeyGen takes a message m as input and creates a message-derived key $\mathcal{K} \leftarrow \mathsf{KeyGen}(m)$; the encryption algorithm Enc produces a ciphertext $C \leftarrow \mathsf{Enc}(m, \mathcal{K})$ based on the given message m and the message-derived key $\mathcal{K}$; the tag generation algorithm Tag are given the ciphertext C and outputs a tag $T \leftarrow \mathsf{Tag}(C)$; the decryption algorithm Dec returns $m/\perp \leftarrow Dec(C, \mathcal{K})$ based on the input the ciphertext C and message-derived key $\mathcal{K}$, where $\perp$ indicates the invalid symbol. In our manuscript, we let $\mathcal{H}_2(\mathbf{MLE}(m||\mathcal{K}))$ correspond to $\mathcal{H}_2(m||\mathcal{K})$, where $K \leftarrow \mathbf{MLE}.\mathsf{KeyGen}(m)$.

D. System Architecture for Data Sharing

Our VF-PPBA architecture consists of five kinds of entities: a registry authority (RA), an arbitration institution (AI), cloud server provider (CSP), data owners and data recipients shown in Fig. 1. RA is a fully honest entity responsible for initializing the system public parameters and distributing the private key to data owners and data recipients who make the registration requests. CSP owns unlimited storage and calculation resources for users, which supports the computing and storage service in converting the original ciphertexts into re-encrypted ones. AI is also a fully reliable party responsible for responding to fairness determination requests from data recipients and CSP. That is, AI enables the prevention of malicious accusations of CSP. For the CSP, it actually offers correct transformed ciphertexts to recipients but is accused of returning incorrect results. In real-world applications, an institution could buy cloud services (e.g., Amazon cloud, Google cloud, etc.) from CSP, the IT center of an institution works as RA and the law department of an institution plays the role of AI.

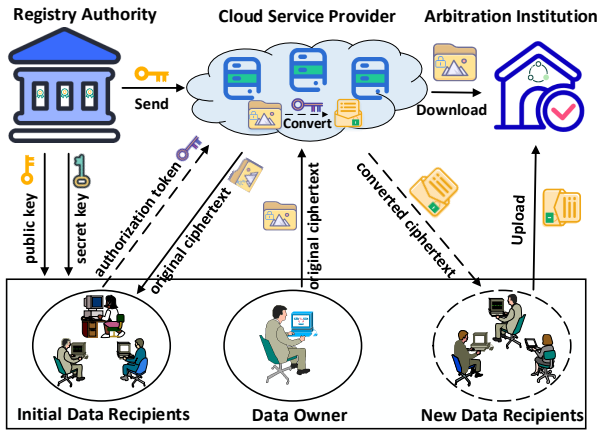


Fig. 1: System Architecture of VF-PPBA

For ease of data sharing, data owners commonly outsource their data to CSP. In detail, a data owner specifies an anonymous access vector corresponding to a set of identities to encode his/her data and then outsources the resulting ciphertext to CSP. The recipient whose identity vector is orthogonal to the designated access vector can decode the ciphertext. If an authorized recipient wants to share the same data with other new recipients, he/she first designates a new access vector indicating the new intended recipients and uses it with his/her private keys to create an authorization token (re-encryption key). With such an authorization token, the original ciphertext

can be transformed into a new one, such that any new recipient whose identity vector is orthogonal to the new access vector can first verify the correctness of converted ciphertexts and then decrypt them When there is a dispute between a recipient and the CSP over whether the converted ciphertext is correct, AI can publicly verify whether the outcome of converted ciphertext is correct and clarify whether the CSP is maliciously accused of returning incorrect results. It is worth noting that the PPMR-BPRE architecture does not contain AI since CSP is assumed to be fully trusted, which always honestly performs the conversion tasks for recipients.

Remark: Here, we recognize that in our VF-PPBE scheme a fully-trusted arbitration institution (AI) is exploited to perform Claim algorithm to keep the fairness for the CSP, which is indeed a strong assumption. The reason leading to this setting mainly originates from the consideration of reducing the computational cost of decryption on the recipients' sides as much as possible, thus making our VF-PPBE more efficient and practical for actual deployments. In fact, such a fully-trusted arbitration institution is not necessarily required in our VF-PPBA, which can be replaced via the deployment of non-interactive zero-knowledge proof (NIZK) on the CSP's side. As we all know, the NIZK technology can realize that the prover (*i.e.*, CSP) could convince the verifier (*i.e.*, recipients) that a statement (*i.e.*, ciphertext transformation operation) is indeed correct without leaking any useful information to the verifier. Although the ciphertext transformation operations faithfully conducted by the CSP can be ensured with this technology, the NIZK deployment on the CSP's side inevitably brings additionally-computational costs of decryption for a recipient, since the recipient requires to perform additional verification of the NIZK proof before decryption. Hence, to achieve the tradeoff between efficiency and security, we use a fully-trusted AI to minimize the computational cost of a recipient's decryption.

E. Threat Model and Security Goals

There are four types of active attacks to be confronted in our data sharing scenarios. First, any adversary including malicious CSP and unauthorized users intends to extract the cleartext from a ciphertext (including the original ciphertext and re-encrypted ciphertext) without valid secret keys. Note that one special case is that unauthorized users collude with malicious CSP to launch collusion attacks for attempting to get a valid decryption key, thus accessing the intended data. Second, any adversary including malicious CSP and unauthorized or authorized users attempts to learn identity privacy from a ciphertext, regardless of whether the adversary has valid secret keys. Third, the malicious CSP would return an incorrect transformed ciphertext without being checked by the recipient. Lastly, a recipient who has successfully recovered the plaintext from a ciphertext may make claims to accuse CSP of returning incorrect results. In summary, the above types of attacks are considered threats to data confidentiality, identity anonymity, transformation verifiability and accusation fairness, respectively. Considering these real existing attacks, the security goals of our works are formalized as follows:

1) *Confidentiality of outsourced data.* If the outsourced encrypted data are stored in CSP, only the authorized user can access them if she/he has valid secret keys. Besides, the data after being re-encrypted can be also accessed by the new permitted recipients. That is to say, any encoded data are unreadable to any adversary having no correct secret keys, including malicious CSP and unauthorized recipients.

2) *Privacy of identity.* If an access vector corresponding to an access list of recipients' identities is embedded into a ciphertext regardless of the original ciphertext and re-encrypted ciphertext, any user cannot learn the identities of other recipients even if he/she has been authorized to decrypt the ciphertext. Also, CSP cannot deduce any identity privacy from the delegated authorization token associated with the identity vector and a new access vector.

3) *Verifiability and fairness.* If a recipient is granted to access the re-encrypted ciphertext, then the correctness of the re-encrypted ciphertext generation done by CSP can be privately checked by himself/herself. Besides, the malicious accusation of CSP returning incorrect results can be publicly determined by a trusted third party (*e.g.*, AI).

F. VF-PPBA Frame and its Security Definitions

In this part, the frame of our VF-PPBA is introduced for concrete constructions. Besides, the security game definitions are presented in **Supplementary Material A** due to the limitations of the space to prove the security of our methodologies.

Our VF-PPBA is formally made up of seven algorithms: **Setup**, **Register**, **Enc**, **Authorize**, **Transform**, **Dec**, **Claim**.

- **Setup**$(1^\lambda) \rightarrow (\mathsf{pp}, \mathsf{msk})$: The system setup algorithm, performed by RA, receives a security parameter 1^λ, and returns the public parameter pp and the master secret key msk.
- **Register**$(\mathsf{pp}, \mathsf{msk}, \mathbf{y}) \rightarrow \mathsf{sk}_\mathbf{y}$: The registration algorithm, also conducted by RA, receives pp, msk and an identity vector $\mathbf{y}$, and outputs a secret key $\mathsf{sk}_\mathbf{y}$.
- **Enc**$(\mathsf{pp}, \mathbf{x}, m) \rightarrow \mathsf{ct}_\mathbf{x}$: The encryption algorithm, implemented by data owner, receives pp, an access vector $\mathbf{x}$ and a plaintext m, and produces a ciphertext ct_x.
- **Authorize**$(\mathsf{pp}, \mathsf{sk}_\mathbf{y}, \mathbf{w}) \rightarrow \mathsf{at}_{\mathbf{x}\rightarrow\mathbf{w}}$: The authorization algorithm, carried out by authorized recipient, receives pp, $\mathsf{sk}_\mathbf{y}$ and a new access vector $\mathbf{w}$, and creates an authorization token $\mathsf{at}_{\mathbf{x}\rightarrow\mathbf{w}}$.
- **Transform**$(\mathsf{pp}, \mathsf{ct}_\mathbf{x}, \mathsf{at}_{\mathbf{x}\rightarrow\mathbf{w}}) \rightarrow \mathsf{ct}_{\mathbf{x}\rightarrow\mathbf{w}}$: The transformation algorithm, executed by cloud servers, receives pp, $\mathsf{ct}_\mathbf{x}$ and $\mathsf{at}_{\mathbf{x}\rightarrow\mathbf{w}}$, and transforms the previous ciphertext ct_x into a new ciphertext $\mathsf{ct}_{\mathbf{x}\rightarrow\mathbf{w}}$.
- **Dec**$(\mathsf{pp}, \mathsf{ct}_\mathbf{x}/\mathsf{ct}_{\mathbf{x}\rightarrow\mathbf{w}}, \mathsf{sk}) \rightarrow m/\perp$: The decryption algorithm, run by data recipient, receives pp, $\mathsf{ct}_\mathbf{x}/\mathsf{ct}_{\mathbf{x}\rightarrow\mathbf{w}}$ and sk. It returns the plaintext m if the inner product of identity vector and access vector is zero. Otherwise, it returns a termination symbol $\perp$.
- **Claim**$(\mathbf{ct}_\mathbf{x}, \mathsf{at}_{\mathbf{y}\rightarrow\mathbf{w}}, \mathsf{ct}_{\mathbf{x}\rightarrow\mathbf{w}}, \pi) \rightarrow \mathit{True}/\mathit{False}$: The claimation algorithm, performed by a trusted AI (public verifier), receives $\mathbf{ct}_\mathbf{x}$, $\mathsf{at}_{\mathbf{y}\rightarrow\mathbf{w}}$, $\mathsf{ct}_{\mathbf{x}\rightarrow\mathbf{w}}$, a proof π, and returns *True* or *False*.

Our VF-PPBA scheme should achieve fairness, that is, if cloud servers return correct conversion results to a recipient,

they should be prevented from being maliciously accused of giving incorrect results. Furthermore, our VF-PPBA scheme should realize soundness and verifiability. Namely, if each algorithm in this framework is honestly run by the corresponding entity, the failure would not occur. In this solution, there are two different types of ciphertexts *i.e.*, $\mathbf{ct_x}$ and $\mathsf{ct_{x\to w}}$. The message m can be recovered from $\mathbf{ct_x}$ and $\mathsf{ct_{x\to w}}$ as follows: For the original ciphertext $\mathsf{ct_x} \leftarrow \mathbf{Enc}(\mathsf{pp}, \mathbf{x}, m)$ and any secret key $\mathsf{sk_y} \leftarrow \mathbf{Register}(\mathsf{pp}, \mathsf{msk}, \mathbf{y})$, if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, then $m \leftarrow \mathbf{Dec}(\mathsf{pp}, \mathsf{ct_x}, \mathsf{sk_y})$; For the transformed ciphertext $\mathsf{ct_{x\to w}} \leftarrow \mathbf{Transform}\ (\mathsf{pp}, \mathsf{ct_x}, \mathsf{at_{x\to w}})$ and $\mathsf{sk_y}$, $\leftarrow \mathbf{Register}(\mathsf{pp}, \mathsf{msk}, \mathbf{y'})$, where $\mathsf{ct_x} \leftarrow \mathbf{Enc}(\mathsf{pp}, \mathbf{x}, m)$ and $\mathsf{at_{x\to w}} \leftarrow \mathbf{Authorize}\ (\mathsf{pp}, \mathsf{sk_y}, \mathbf{w})$, if $\langle \mathbf{y'}, \mathbf{w} \rangle = 0$, then $m \leftarrow \mathbf{Dec}(\mathsf{pp},\ \mathsf{ct_{x\to w}}, \mathsf{sk_y'})$.

IV. Concrete Constructions

In this section, we are the first to invent two various constructions, *i.e.*, PPMR-BPRE and VF-PPBA, from inner product encryption and broadcast encryption for anonymously and dynamically broadcasting targeted ciphertext to multiple receivers in the semi-trusted and untrustworthy cloud environments. The core of our constructions constructing the secret key and broadcasting ciphertext originates from the inclusion relationship between an identity and a subset of identities. For one identity $\mathsf{id} \in \mathbb{Z}_p$, we can easily produce an identity vector $\mathbf{y} = (y_0, \ldots, y_n)$, where $y_i = \mathsf{id}^i \mod p$. For a subset of accessed identities $\mathcal{S} = \mathsf{id}_1, \ldots, \mathsf{id}_n$, we can also create an access vector $\mathsf{x} = (x_0, \ldots, x_n)$, where x_i is the coefficient of $F(z) = \prod_{\mathsf{id}_i \in \mathcal{S}} (z - \mathsf{id}_i) = \sum_{i=0}^{n} x_i z^i$. It is easy to infer the inner product of $\mathbf{x}$ and $\mathbf{y}$ equals 0. In the following concrete constructions, we hide the identity and a subset of identities in **Register & Enc** algorithms by replacing the identity vector and access vector for secret key and ciphertext generation. For the authorization token, it could be created by binding the decryption keys of inner product encryption to recipients' public keys. Specifically, in our two constructions, the secret key $\mathsf{sk_y}$ hiding the identity vector $\mathbf{y}$ is bound with a new access vector $\mathbf{w}$ to realize dynamic authorization for distinct recipients.

A. An Overview

At first glance, the issues shown in the **Introduction** can be easily resolved via dynamic conditional multi-recipient proxy broadcast re-encryption solutions with slight modifications. However, these solutions with slight modifications either actually only solve at most two of the above issues, or are not easy to construct. To the best of our knowledge, current existing dynamical multi-recipient proxy broadcast re-encryption technologies mainly include dynamical attribute-based proxy re-encryption (ABPRE), dynamical identity-based broadcast proxy re-encryption (IB-BPRE, we also call it MR-BPRE in our paper), dynamical inner product proxy re-encryption (IP-PRE), which can realize the strategies for dynamically designating ciphertexts to multiple recipients.

With the dynamic conditional multi-recipient proxy broadcast re-encryption technologies, there are two potential solutions to solve the above issues: *I) It is straightforward to combine ABPRE or IB-BPRE and anonymous technology, commitment technique.* Since ABPRE has been proven to be inefficient due to its computationally-intensive encryption and decryption operations, it is unsuitable for constructing efficient strategies for dynamically designating ciphertexts to multiple recipients even if it can be elegantly combined with anonymous technology and commitment technique. Besides, although IB-BPRE can be slightly modified to realize the fairness of ciphertext transformation verification, it is not easy to realize the privacy-preserving of user identities. This is because most IB-BPRE solutions to preserve the identity privacy are to partition an identity into two blind parts, such that an adversary fails to discern the specific counterpart by bilinear pairings. However, splitting an identity into two pieces and embedding them into the ciphertext will probably lead to the original structure change of the ciphertexts, since an identity commonly features two types of structures of both the decryption key and ciphertext simultaneously. Moreover, the encryption and decryption overhead would be at least twice as high as before. Hence, it is inappropriate to use this solution for preventing identity privacy leakage. Here please note that there is no anonymous IB-BPRE solution that has been designed until now. *II) Applying the commitment technique to IP-PRE directly is another ideal solution.* It is well-known that IP-PRE can be viewed as a high-level primitive of both IB-BPRE and ABPRE since it absolutely features the advantages of both IB-BPRE and ABPRE. Besides, apart from the efficient encryption & decryption operations, it also realizes the anonymity of the identities or attributes, thus preserving the privacy protection of identities and attributes. Regrettably, all existing IP-PRE solutions require an always-on entity acting as a key generation center (KGC) instead of the user themselves to generate authorization tokens for ciphertext conversion, which leads to impractical and inflexible for real-world applications (See Section II. C for more details). Hence, before applying the commitment technique to IP-PRE, an IP-PRE solution ensuring independently-producing authorization tokens is first required to be invented. However, it is not easy to design such an IP-PRE without KGC to produce the authorization token, since the decryption key structure of IP-PRE commonly determines the form of the authorization token.

In this paper, we invent the first-ever PPMR-BPRE, privacy-preserving multi-recipient broadcast proxy re-encryption scheme, which is essentially a kind of IP-PRE solution. In our PPMR-BPRE, the authorization token can be sorely-produced by the user themselves with their own decryption keys via the structure-preserving technique. On the basis of our PPMR-BPRE, we exploit the commitment technique to propose VF-PPBA, the first verifiable, fair and privacy-preserving broadcast authorization scheme.

B. PPMR-BPRE Scheme in *Semi-trusted* Clouds

- **Setup**(1^λ): Based on the security parameter 1^λ, it first sets the bilinear maps $(\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, p, e)$ with its generators $g \in \mathbb{G}_0$, $u, h \in \mathbb{G}_1$, where p is the prime order of $(\mathbb{G}_0, \mathbb{G}_1)$. Next, it chooses a hash function $H : \{0,1\}^* \to$

$\mathbb{G}_1$, randomly samples $\alpha_1, \alpha_2, \ldots, \alpha_n, \gamma, \beta, \theta \in \mathbb{Z}_p$ and performs the following calculations:

$$g_0 = g^{\rho}, g_1 = g^{\theta}, g_{2,1} = g^{\alpha_1}, \ldots, g_{2,n} = g^{\alpha_n},$$
$$u_0 = u^{\beta\gamma}, u_1 = u^{\theta}, u_{2,1} = u^{\alpha_1}, \ldots, u_{2,n} = u^{\alpha_n},$$
$$h_0 = h, h_1 = h^{\theta}, h_{2,1} = h^{\alpha_1}, \ldots, h_{2,n} = u^{\alpha_n}.$$

It finally sets $\mathsf{pp} = (g_0, g_1, g_{2,1}, \ldots, g_{2,n}, h_0, h_1, h_{2,1}, \ldots, h_{2,n}, H, e(g, u^{\beta\gamma}))$ and $\mathsf{msk} = (u^{\beta\gamma}, u, u_1, u_{2,1}, \ldots, u_{2,n})$.

- **Register**$(\mathsf{pp}, \mathsf{msk}, \mathbf{y})$: With the input pp, msk and an attribute vector $\mathbf{y}$, it uniformly samples $r, R \in \mathbb{Z}_p$ and creates the secret key $\mathsf{sk}_{\mathbf{y}} = (\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3, \mathbf{y})$ as follows:

$$\mathsf{sk}_1 = u^{\beta\gamma} \prod_{i=1}^{n} (u_{2,i})^{y_i r} u_1^R, \mathsf{sk}_2 = u^r, \mathsf{sk}_3 = u^R.$$

- **Enc**$(\mathsf{pp}, \mathbf{x}, m)$: Given pp, $\mathbf{x}$ and the message m, it uniformly samples $z \in \mathbb{Z}_p$ and generates a ciphertext $\mathsf{ct}_{\mathbf{x}} = (c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ as follows: it first samples $z \in \mathbb{Z}_p$ and computes $c_0 = m \cdot e(g, u^{\beta\gamma})^z, c_1 = g^z, c_{2,i} = g_0^{x_i z}(g_{2,i})^z, c_3 = g_1^z$. Then, the ciphertext $\mathsf{ct}_{\mathbf{x}} = (c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ is produced.
- **Authorize**$(\mathsf{pp}, \mathsf{sk})$: Taking as input pp and sk, an authorized recipient first picks a new access vector $\mathbf{w} = (w_1, \ldots, w_n)$, randomly selects $t, r', R', q \in \mathbb{Z}_p$ and then computes the authorization token $\mathsf{at}_{\mathbf{y}\to\mathbf{w}} = (d_1, d_{2,i}, \ldots, d_8)$, where $d_1 = g^t, d_{2,i} = g_0^{w_i t}(g_{2,i})^t, d_3 = g_1^t$, $d_4 = \mathsf{sk}_1 \cdot h_1^{R'}$, $d_5 = (\mathsf{sk}_2 \cdot h^{r'})^q$, $d_6 = \mathsf{sk}_3 \cdot h^{R'}$, $d_7 = H(e(g, u^{\beta\gamma})^t) / \prod_{i=1}^{n} (h_{2,i}^{y_i})^{r'}$, $d_8 = (y_1/q, \ldots, y_n/q)$.
- **Transform**$(\mathsf{pp}, \mathsf{ct}, \mathsf{at}_{\mathbf{y}\to\mathbf{w}})$: After getting the authorization token from an authorized recipient, the cloud servers transform the ciphertext that the recipient with the secret key $\mathsf{sk}_{\mathbf{y}}$ can decrypt to another one that a new recipient with the secret key $\mathsf{sk}'_{\mathbf{y}}$ can access. Specifically, given $\mathsf{at}_{\mathbf{y}\to\mathbf{w}} = (d_1, d_{2,i}, \ldots, d_8)$, the cloud servers convert the ciphertext $\mathsf{ct}_{\mathbf{x}} = (c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ into another ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}} = (C_1, C_{2,i}, \ldots, C_6)$, where $C_1 = d_1$, $C_{2,i} = d_{2,i}$, $C_3 = d_3$, $C_4 = c_1$, $C_5 = d_7$ and

$$C_6 = c_0 \cdot e(\prod c_{2,i}^{d_{8,i}}, d_5) \cdot e(c_3, d_6) \cdot e(c_1, d_4)^{-1}$$
$$= c_0 \cdot e(\prod_{i=1}^{n} c_{2,i}^{y_i}, \mathsf{sk}_2 \cdot h^{r'}) \cdot e(c_3, \mathsf{sk}_3 \cdot h_1^{R'}) / e(c_1, \mathsf{sk}_1 \cdot h_1^{R'})$$
$$h^{r'}) \cdot e(c_3, h^{R'}) \cdot e(c_1, h_1^{R'})^{-1}$$
$$= m \cdot e(g, h)^{zr' \sum_{i=1}^{n} (\alpha_i y_i + \rho x_i y_i)} \cdot e(g, u)^{\rho r z \sum_{i=1}^{n} x_i y_i}.$$

- **Dec**$(\mathsf{pp}, \mathsf{ct}_{\mathbf{x}}/\mathsf{ct}_{\mathbf{x}\to\mathbf{w}}, \mathsf{sk})$: With the input pp, $\mathsf{ct}_{\mathbf{x}}/\mathsf{ct}_{\mathbf{x}\to\mathbf{w}}$ and sk, it performs the following operations to decode the encrypted message m. Since there are two various types of ciphertexts in the scheme, *i.e.,* the original ciphertext $\mathsf{ct}_{\mathbf{x}}$ and the transformed ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}}$, the data access to these two kinds of ciphertexts can be described as below:
  - *Case I*: For the original ciphertext $\mathsf{ct}_{\mathbf{x}} = (c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$, the recipient whose identity vector is orthogonal to access vector $\mathbf{x}$ can perform the following calculation to decode the message $m = c_0 \cdot e(c_1, \mathsf{sk}_1)^{-1} \cdot e(c_3, \mathsf{sk}_3) \cdot e(\prod_{i=1}^{n} c_{2,i}^{y_i}, \mathsf{sk}_2)$.
  - *Case II*: For the transformed ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}} = (C_1, C_{2,i}, \ldots, C_6)$, an additionally authorized recipient whose identity is orthogonal to access vector $\mathbf{w}$ can conduct the following calculations to recover the message: Assuming the secret key of the recipient is $\mathsf{sk}_{\mathbf{y}'} = (\mathsf{sk}'_1, \mathsf{sk}'_2, \mathsf{sk}'_3, \mathbf{y}')$, he/she first computes $A = e(C_1, \mathsf{sk}'_1) \cdot e(C_3, \mathsf{sk}'_3)^{-1} \cdot e(\prod_{i=1}^{n} C_{2,i}^{y'_i}, \mathsf{sk}'_2)^{-1}$ and $A' = e(C_5/H(A), C_4)$. Then, he/she recovers the message m by computing $m = C_6 \cdot A'$.

C. VF-PPBA Scheme in *Untrustworthy* Clouds

- **Setup**(1^{λ}): Based on the security parameter 1^{λ}, it first sets the bilinear maps $(\mathbb{G}_0, \mathbb{G}_1, \mathbb{G}_T, p, e)$ with its generators $g \in \mathbb{G}_0$, $u, h \in \mathbb{G}_1$, where p is the prime order of $(\mathbb{G}_0, \mathbb{G}_1)$. Next, it chooses three hash functions $H : \{0,1\}^* \to \mathbb{G}_1$, $\mathcal{H}_1 : \mathbb{G}_T \to \{0,1\}^{2\ell}$, $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{Z}_p$ and a message-lock encryption algorithm **MLE** [35]. It also randomly samples $\alpha_1, \ldots, \alpha_n, \gamma, \beta, \theta \in \mathbb{Z}_p$, $\mu, \sigma \in \mathbb{G}_0$ and performs the following calculations:

$$g_0 = g^{\rho}, g_1 = g^{\theta}, g_{2,1} = g^{\alpha_1}, \ldots, g_{2,n} = g^{\alpha_n},$$
$$u_0 = u^{\beta\gamma}, u_1 = u^{\theta}, u_{2,1} = u^{\alpha_1}, \ldots, u_{2,n} = u^{\alpha_n},$$
$$h_0 = h, h_1 = h^{\theta}, h_{2,1} = h^{\alpha_1}, \ldots, h_{2,n} = u^{\alpha_n}.$$

It finally sets $\mathsf{pp} = (\mu, \sigma, g, g_0, g_1, g_{2,1}, \ldots, g_{2,n}, h_0, h_1, h_{2,1}, \ldots, h_{2,n}, H, \mathcal{H}_1, \mathcal{H}_2, e(g, u^{\beta\gamma}), e(g, h)^{\beta\gamma}, \mathbf{MLE})$ and $\mathsf{msk} = (u^{\beta\gamma}, u, u_1, u_{2,1}, \ldots, u_{2,n})$.
- **Register**$(\mathsf{pp}, \mathsf{msk}, \mathbf{y})$: This algorithm is the same as that in PPMR-BPRE. A secret key $\mathsf{sk}_{\mathbf{y}} = (\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3, \mathbf{y})$ can be produced.
- **Enc**$(\mathsf{pp}, \mathbf{x}, m)$: Given pp, $\mathbf{x}$ and the message m, it uniformly samples $z \in \mathbb{Z}_p$ and generates a ciphertext $\mathsf{ct}_{\mathbf{x}} = (c, c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ as follows: it first chooses $z = \mathcal{H}_2(\mathbf{MLE}(m||\mathcal{K}))$. Next, it conducts the computations: $c_0 = m||\mathcal{K} \oplus \mathcal{H}_1(e(g, u^{\beta\gamma})^z), c_1 = g^z, c_{2,i} = g_0^{x_i z}(g_{2,i})^z, c_3 = g_1^z$, $c = \sigma^{\mathcal{H}_2(m)}\mu^{\mathcal{H}_2(\mathcal{K})}$. Then, the ciphertext $\mathsf{ct}_{\mathbf{x}} = (c, c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ is produced.
- **Authorize**$(\mathsf{pp}, \mathsf{sk})$: Taking as input pp and sk, it picks a new access vector $\mathbf{w} = (w_1, \ldots, w_n)$, $t, r', R', q \in \mathbb{Z}_p, \varphi \in \mathbb{G}_T$ and computes the authorization token $\mathsf{at}_{\mathbf{y}\to\mathbf{w}} = (d_0, d_1, d_{2,i}, \ldots, d_9)$, where $d_0 = \varphi \cdot e(g, u)^{\beta\gamma t}$, $d_1 = g^t, d_{2,i} = g_0^{w_i t}(g_{2,i})^t, d_3 = g_1^t$, $d_4 = \mathsf{sk}_1^{\mathcal{H}_2(\varphi)} \cdot h_1^{R'}$, $d_5 = (\mathsf{sk}_2^{\mathcal{H}_2(\varphi)} \cdot h^{r'})^q$, $d_6 = \mathsf{sk}_3^{\mathcal{H}_2(\varphi)} \cdot h^{R'}$, $d_7 = H(e(g, u^{\beta\gamma})^t) / \prod_{i=1}^{n} (h_{2,i}^{y_i})^{r'}$, $d_8 = (y_1/q, \ldots, y_n/q)$, $d_9 = e(g, h)^{\beta\gamma\mathcal{H}_2(\varphi)}$, $d_{10} = \prod_{i=1}^{n} (g_{2,i}^{y_i})^{r'}$.
- **Transform**$(\mathsf{pp}, \mathsf{ct}, \mathsf{at}_{\mathbf{x}\to\mathbf{w}})$: After getting the authorization token from an authorized recipient, the cloud servers convert the ciphertext. Specifically, given $\mathsf{at}_{\mathbf{x}\to\mathbf{w}} = (d_0, d_1, d_{2,i}, \ldots, d_{10})$, the cloud servers transform the ciphertext $\mathsf{ct}_{\mathbf{x}} = (c, c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ into another ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}} = (C, C_0, C_1, C_{2,i}, \ldots, C_9)$, where $C = c, C_0 = c_0, C_1 = d_1$, $C_{2,i} = d_{2,i}$, $C_3 = d_3$,

$C_4 = c_1$, $C_5 = d_7$, $C_6 = d_9$, $C_7 = d_0$, $C_9 = d_{10}$ and

$$\begin{aligned}C_8 &= e(\prod c_{2,i}^{d_{8,i}}, d_5)^{-1} \cdot e(c_3, d_6)^{-1} \cdot e(c_1, d_4)\\ &= e(\prod_{i=1}^{n} c_{2,i}^{y_i}, \mathsf{sk}_2^{\mathcal{H}_2(\varphi)} \cdot h^{r'})^{-1} \cdot e(c_3, \mathsf{sk}_3^{\mathcal{H}_2(\varphi)} \cdot h_1^{R'})^{-1}\\ &\cdot e(c_1, \mathsf{sk}_1^{\mathcal{H}_2(\varphi)} \cdot h_1^{R'})\\ &= e(\prod_{i=1}^{n} c_{2,i}^{y_i}, \mathsf{sk}_2^{\mathcal{H}_2(\varphi)})^{-1} \cdot e(c_3, \mathsf{sk}_3^{\mathcal{H}_2(\varphi)})^{-1} \cdot e(c_1,\\ &= e(g,h)^{-zr' \sum_{i=1}^{n} (\alpha_i y_i + \rho x_i y_i)} \cdot e(g,u)^{\mathcal{H}_2(\varphi)\beta\gamma z} \cdot e(g,\\ u)^{-\rho r z \mathcal{H}_2(\varphi) \sum_{i=1}^{n} x_i y_i}.\end{aligned}$$

Thus, the ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}}$ is transformed for the identity vector $\mathbf{w}$.

- **Dec**$(\mathsf{pp}, \mathsf{ct}_{\mathbf{x}}/\mathsf{ct}_{\mathbf{x}\to\mathbf{w}}, \mathsf{sk})$: With the input pp, $\mathsf{ct}_{\mathbf{x}}/\mathsf{ct}_{\mathbf{x}\to\mathbf{w}}$ and sk, it performs the following operations to decode the encrypted message m. For data access of $\mathsf{ct}_{\mathbf{x}}$ $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}}$, the descriptions are as follows:
  - *Case I*: For the original ciphertext $\mathsf{ct}_{\mathbf{x}} = (c, c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ or $\mathsf{ct}_{\mathbf{x}} = (c, c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3, c_4)$, the recipient whose identity vector is orthogonal to access vector $\mathbf{x}$ computes $A = e(c_1, \mathsf{sk}_1) \cdot e(c_3, \mathsf{sk}_3)^{-1} \cdot e(\prod_{i=1}^{n} c_{2,i}^{y_i}, \mathsf{sk}_2)^{-1}$ and then derives $m||\mathcal{K} = c_0 \oplus \mathcal{H}_1(A)$. Finally, he/she outputs the message m if $c = \sigma^{\mathcal{H}_2(m)}\mu^{\mathcal{H}_2(\mathcal{K})}$ holds. Otherwise, he/she returns $\perp$.
  - *Case II*: For the transformed ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}} = (C, C_0, C_1, C_{2,i}, \ldots, C_9)$, the new recipient whose identity vector $\mathbf{y}'$ is orthogonal to access vector $\mathbf{w}$ can conduct the following calculations to recover the message: Assuming the secret key of a recipient is $\mathsf{sk}_{\mathbf{y}'} = (\mathsf{sk}'_1, \mathsf{sk}'_2, \mathsf{sk}'_3, \mathbf{y}')$, she/he first computes $A = e(C_1, \mathsf{sk}'_1) \cdot e(C_3, \mathsf{sk}'_3)^{-1} \cdot e(\prod_{i=1}^{n} C_{2,i}^{y'_i}, \mathsf{sk}'_2)^{-1}$, $A' = C_8 \cdot e(H(A)/C_5, C_4)$ and $\varphi = C_7/A$. Then, he/she gets $m||\mathcal{K} = c_0 \oplus H(A'^{1/\mathcal{H}_2(\varphi)})$. Finally, he/she returns the message m if $c = \sigma^{\mathcal{H}_2(m)}\mu^{\mathcal{H}_2(\mathcal{K})}$ satisfies. Otherwise, he/she returns $\perp$. In this step, the recipient will give the proof $\pi = (m, \mathcal{K}, H(A))$ to the trusted AI, where $H(A) = H(e(g, u^{\beta\gamma})^t)$.
- **Claim**$(\mathbf{ct}_{\mathbf{x}}, \mathsf{at}_{\mathbf{y}\to\mathbf{w}}, \mathsf{ct}_{\mathbf{x}\to\mathbf{w}}, \pi)$: Given an original ciphertext $\mathbf{ct}_{\mathbf{x}} = (c, c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3, c_4)$, an authorization token $\mathsf{at}_{\mathbf{y}\to\mathbf{w}} = (d_0, d_1, d_{2,i}, \ldots, d_{10})$, a transformed ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}} = (C, C_0, C_1, C_{2,i}, \ldots, C_9)$, and a proof $\pi = (m, \mathcal{K}, H(A))$, the third trusted verifier (*e.g.,* AI) first checks whether the following equations hold: $C = c, C_0 = c_0, C_1 = d_1$, $C_{2,i} = d_{2,i}$, $C_3 = d_3$, $C_4 = c_4$, $C_5 = d_7$, $C_6 = d_9$, $C_7 = d_0$ and $C_9 = d_{10}$. If one of the above equations fails, then it aborts and returns $\perp$. Otherwise, it computes $z = \mathcal{H}_2(\mathbf{MLE}(m||\mathcal{K}))$ and determines whether $C = c = \sigma^{\mathcal{H}_2(m)}\mu^{\mathcal{H}_2(\mathcal{K})}$, $e(C_5/H(A), g) = e(C_9, h)$, and $C_6^z \neq C_8 \cdot e(C_5/H(A), C_4)) = e(g,u)^{\mathcal{H}_2(\varphi)\beta\gamma z}$. If these two equations hold, then it outputs *True*; otherwise, it returns *False*. Note that it is extremely important to verify the correctness of $H(A)$ of the given proof, because if $H(A)$ fails to pass the verification of the equation $e(C_5/H(A), g) = e(C_9, h)$, then $C_6^z \neq C_8 \cdot e(C_5/H(A), C_4))$ always holds. This is the reason why we need to set the redundant part C_9.

V. Soundness and Security Analysis

In this section, we provide theoretical analysis to prove that our frameworks are sound and realize semantic security, verifiability & fairness.

A. Soundness of PPMR-BPRE and VF-PPBA

***Theorem** 1:* Regardless of whether the ciphertext is the original or the transformed ciphertext, a user, if he/she has the correct secret key, can successfully decrypt the ciphertext.

***Proof** 1:* For a valid original ciphertext $\mathsf{ct}_{\mathbf{x}} = (c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$, a user who has the secret key $\mathsf{sk}_{\mathbf{y}} = (\mathsf{sk}_1, \mathsf{sk}_2, \mathsf{sk}_3, \mathbf{y})$ can perform the following computation:

$$\begin{aligned}m &= c_0 \cdot e(c_1, \mathsf{sk}_1)^{-1} \cdot e(c_3, \mathsf{sk}_3) \cdot e(\prod_{i=1}^{n} c_{2,i}^{y_i}, \mathsf{sk}_2)\\ &= m \cdot e(g, u^{\beta\gamma})^z \cdot e(g^z, u^{\beta\gamma} \prod_{i=1}^{n} (u_{2,i}^{y_i} u^{\theta R}))^{-1}.\\ &e(g^{\theta z}, u^R) \cdot e(\prod_{i=1}^{n} (g_0^{x_i z} g^{\alpha_i z})^{y_i}, u^r)\\ &= m \cdot e(g^z, \prod_{i=1}^{n} u^{\alpha_i y_i r})^{-1} \cdot e(g_0^{\sum_{i=1}^{n} x_i y_i z} g^{\sum_{i=1}^{n} \alpha_i z y_i}, u^r)\\ &= m \cdot e(g,u)^{\rho r z \sum_{i=1}^{n} x_i y_i},\end{aligned}$$

Thus, if identity vector $\mathbf{y}$ is orthogonal to access vector $\mathbf{x}$, then the message m can be successfully recovered.

For a valid transformed ciphertext $\mathsf{ct}_{\mathbf{x}\to\mathbf{w}} = (C_1, C_{2,i}, \ldots, C_6)$, a user who owns his/her secret key $\mathsf{sk}_{\mathbf{y}'}$ can compute

$$\begin{aligned}A &= e(C_1, \mathsf{sk}'_1) \cdot e(C_3, \mathsf{sk}'_3)^{-1} \cdot e(\prod_{i=1}^{n} C_{2,i}^{y'_i}, \mathsf{sk}'_2)^{-1}\\ &= e(g^t, u^{\beta\gamma} \prod_{i=1}^{n} (u_{2,i}^{y'_i}) u^{\theta R'}) \cdot e(g^{\theta t}, u^{R'})^{-1} \cdot e(\prod_{i=1}^{n} (g_0^{w_i t} g^{\alpha_i t})^{y'_i},\\ &u^{r'})^{-1} = e(g,u)^{\beta\gamma t} \cdot e(g^{-\sum_{i=1}^{n} w_i y'_i t}, u^{r'}).\end{aligned}$$

If the identity vector $\mathbf{y}'$ is orthogonal to access vector $\mathbf{w}$, then the user can have $A = e(g,u)^{\beta\gamma t}$. Next, he/she computes $A' = e(C_5/H(A), C_4) = e(\prod_{i=1}^{n} (h_{2,i}^{y_i})^{r'}, g^{-\rho z}) = e(g,h)^{-r'\rho z \sum_{i=1}^{n} \alpha_i y_i}$. Finally, he/she can decode the message by calculating $m = C_6 \cdot A'$. Due to the very similarity and the space limit, we omit the soundness proof of VF-PPBA.

B. Security Proofs for Semantic Security, Verifiability and Fairness

***Theorem** 2:* The selective *identity-hiding* security against chosen plaintext attacks on original ciphertext under DBDH and $\mathcal{P}$-DBDH assumptions in PPMR-BPRE can be realized in the standard model.

***Theorem** 3:* The selective identity-hiding semantic security for converted ciphertext under DBDH and $\mathcal{P}$-DBDH assumptions in PPMR-BPRE can be realized.

***Theorem** 4:* Our VF-PPBA realizes verifiability under the discrete logarithm problem.

***Theorem** 5:* Our VF-PPBA can realize fairness if the discrete logarithm problem holds.

***Theorem** 6:* The selective *identity-hiding* security against chosen plaintext attacks on the original ciphertext of our VF-PPBA under DBDH and $\mathcal{P}$-DBDH assumptions in PPMR-BPRE can be also ensured in the standard model.

***Theorem** 7:* The selective identity-hiding semantic security for converted ciphertext in our VF-PPBA under DBDH and modified $\mathcal{P}$-DBDH assumptions can be also achieved.

Proof: Due to the space limits, here we omit the detailed security proofs. The whole security proofs can be found in **Supplementary Material B**. Specifically, the proofs of ***Theorems 2 & 3***, ***Theorems 4 & 5***, ***Theorems 6 & 7*** are correspondingly shown in **Supplementary Material B. A**, **Supplementary Material B. B**, **Supplementary Material B. C**.

VI. Discussion

In this section, we discuss how to further enhance the efficiency and functionality of both our constructions. Specifically, the functionality of our two frameworks can be also enriched to support fine-grained data sharing based on types of vectors indicating different functionalities. Besides, the efficiency of ciphertext and authorization token generation can be improved.

A. Functionality Enhancements

This part introduces different vector transformation techniques based on the types of user attribute lists and access control. These vectors could be used in our frameworks for secret key and ciphertext generation. Compared to identity and access vectors, these vectors can realize fine-grained access control instead of coarse one. Due to the space limits, the introduction of various attribute-based access controls is omitted here. The readers refer to **Supplementary Material C** for more details.

B. Efficiency Improvements

This part mainly describes the modification of the ciphertext generation algorithm and partial decryption algorithm. Here we omit the descriptions of the modified authorization token algorithm and another part of the decryption algorithm, which are identical to the following algorithms. We also present the security proof to prove the modification captures the same security as that in the previous version.

In detail, without changing the setup and registration algorithms, the encryption algorithm is partitioned into offline encryption and online encryption algorithms:

1) *Offline*: The offline algorithm first randomly samples $z, z_1, \ldots, z_n \in \mathbb{Z}_p$ and computes the offline ciphertext $\mathsf{ct.off} = (m \cdot e(g, u^{\beta\gamma})^z, g^z, g_0^{-z_i}(g_{2,i})^z, g_1^z) = (c_0, c_1, \{c'_{2,i}\}_{i\in[1,n]}, c_3)$. Then, it keeps $(\mathsf{ct.off}, z, z_1, \ldots, z_n)$ for the online encryption.

2) *Online*: With the input $(\mathsf{ct.off}, z, z_1, \ldots, z_n)$ and access vector $\mathbf{x} = (x_1, \ldots, x_n)$, the online algorithm computes $Z_i = z_i + x_i z \mod p$ and returns a final ciphertext $\mathsf{ct.fin} = (\mathsf{ct.off}, Z_1, \ldots, Z_n) = (c_0, c_1, \{c'_{2,i}\}, c_3, \{Z_n\})$.

Correspondingly, the decryption algorithm for deciphering $\mathsf{ct.fin}$ is modified as $m = c_0 \cdot e(c_1, \mathsf{sk}_1)^{-1} \cdot e(c_3, \mathsf{sk}_3) \cdot e(g^{\sum_{i=1}^{n} Z_i y_i} \prod_{i=1}^{n} {c'_{2,i}}^{y_i}, \mathsf{sk}_2) = c_0 \cdot e(c_1, \mathsf{sk}_1)^{-1} \cdot e(c_3, \mathsf{sk}_3) \cdot e(\prod_{i=1}^{n} c_{2,i}^{y_i}, \mathsf{sk}_2)$. *Here, please note that $c_{2,i} = g_0^{x_i z}(g_{2,i})^z$ in original schemes must be changed as $c_{2,i} = g_0^{x_i \rho}(g_{2,i})^z$ for the security proofs, which does not affect the other parts of our constructions.*

1) *Efficiency Analysis*: with the above approach, the encryption calculation efficiency in the online phase can be decreased from $(2n + 4)$ exponentiation calculations to n modular multiplications and the cost is relatively small for decryption due to the fact that the only n modular multiplications and one point multiplication is added in decryption phase.

2) *Security Analysis*: the proofs of the modifications are almost identical to the original proofs except the challenge ciphertext generation phase. Let $\mathsf{ct} = (c_0, c_1, \{c_{2,i}\}_{i\in[1,n]}, c_3)$ on the challenge access vector $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$. The reduction algorithm $\mathcal{B}$ first generates the same challenge ciphertext ct^*, then randomly picks $z'_1, \ldots, z'_n \in \mathbb{Z}_p$ and simulates the offline/online challenge ciphertext $\mathsf{ct.fin}^* = (c_0, c_1, \{c_{2,i}g^{-z'_i}\}, c_3, z'_1, \ldots, z'_n)$. Let $z_i = z'_i - \rho x_i^*$, we get $c_{2,i}g^{-z'_i} = g_0^{x_i^*\rho}(g_{2,i})^z \cdot g^{-z'_i} = (g_{2,i})^z g^{-z_i} = c'_{2,i}$ and $z'_i = z_i + \rho x_i^* = Z_i$.

Hence, $\mathsf{ct.fin}^*$ is a valid offline/online challenge ciphertext. If the security of our modified offline/online schemes could be breached by an adversary, then he/she can breach the security of our original schemes as well.

VII. Performance Evaluation

We give functionality comparisons in this section to demonstrate more capabilities that our PPMR-BPRE and VF-PPBA can provide. Besides, we also show the theoretical and experimental analysis via comparisons to indicate more practicability than the existing related works.

A. Functionality Comparison

In TABLE II, the functionality comparisons among our PPPMR-BPRE, VF-PPBA, and various categories of related works are summarized in terms of anonymity, verifiability, malicious accusation resistance, flexible ciphertext conversion, standard model, malicious cloud environment, high-efficiency, and prime order group-based construction. "✓" indicates that the feature can be implemented while "✗" signifies that the scheme fails to realize this functionality. Anonymity means that no adversary can deduce the recipient's identity or attribute privacy from the original or transformed ciphertext. Verifiability ensures that the proxy servers are performing ciphertext transformation honestly. Fairness signifies that the cloud servers could be immune to malicious accusations if the outsourced outcome of ciphertext conversion is executed correctly. The standard model implies that the security does

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.2022.3226577

TABLE II: Functionality comparisons of related works, PPMR-BPRE and VF-PPBA

Scheme	Anonymity	Verifiability	Fairness	Flexible Ciphertext Conversion	Standard Model	Malicious Server	High-efficiency	Prime Order Construction
LCL+ [33]	✗	✗	✗	✓	✓	✗	✗	✓
BGT+ [37]	✓	✗	✗	✓	✓	✗	✓	✗
ST [38], [39]	✓	✗	✗	✓	✓	✓	✓	✓
LS [34]	✗	✗	✗	✓	✗	✗	✗	✓
GSB+ [35]	✗	✓	✓	✓	✗	✓	✗	✓
HYF [40]	✗	✗	✗	✓	✗	✗	✗	✓
XJW+ [30]	✗	✗	✗	✓	✗	✗	✗	✓
DZQ+ [36]	✗	✗	✗	✓	✗	✗	✓	✓
PPMR-BPRE	✓	✗	✗	✓	✓	✗	✓	✓
VF-PPBA	✓	✓	✓	✓	✓	✓	✓	✓

TABLE III: Computation cost comparisons of our solutions with related schemes

Scheme	**Costs at registry authority side**		**Costs at client side**			**Costs at server side**
	Setup	Register	Encrypt	Authorize	Decrypt	Transform
LCL+ [33]	$(2n+1)e_0$	$(2n+1)e_0$	$(n+2)e_0+e_1+p$	$(2n+1)e_0$	$(n+1)p\|\|e_1+(n+2)p$	$(n+1)p$
BGT+ [37]	$(n+1)e_0+e_1+p$	$(2n+3)e_0$	$(2n+2)e_0+p+e_1$	$(4n+6)e_0+p$	$ne_0+2p\|\|ne_0+3p$	$(4n+6)e_0+p$
ST [38], [39]	$(8n+4)e_0+p$	$13ne_0$	$(12n+2)e_0+e_1+p$	$(25n+2)e_0+e_1+p$	$(4n+2)p\|\|(4n+4)p$	$(4n+2)p$
LS [34]	$(2n+10)e_0+3e_1+3p$	$4ne_0$	$(n+4)e_0+2p+2e_1$	$(4n+6)e_0+2e_1+2p$	$(2n+5)e_0+2p\|\|(3n+6)e_0+2p$	$(3n+4)e_0+e_1+4p$
GSB+ [35]	e_0+e_1+p	$(n+3)e_0$	$(3n+4)e_0+p+e_1$	$(4n+5)e_0+2e_1+p$	$(2n+1)p+ne_1\|\|(2n+1)p+ne_1+3e_0$	$(2n+2)p+ne_1$
HYF [40]	$(2n+3)e_0+2p$	$(2n+3)e_0$	$(3n+3)e_0+2e_1+2p$	$(n+4)e_0+e_1+p$	$2p+(n-1)e_0+e_1\|\|3p+(n-1)e_0+e_1$	$(2n+3)p+(n-1)e_0+e_1$
XJW+ [30]	$(4n+1)e_0+p$	e_0	$(3n+2)e_0+e_1+p$	$(n+3)e_0+e_1+p$	$2p+(n-1)e_0+e_1\|\|3p+(n-1)e_0+e_1$	$p+(n-1)e_0+e_1$
DZQ+ [36]	$(3n+3)e_0+p$	e_0	$(3n+4)e_0+e_1+p$	$(7n+4)e_0+e_1+p$	$2p+(n-1)e_0+e_1\|\|3p+(n-1)e_0+e_1$	$(3n+1)p+2e_0+ne_1$
PPMR-BPRE	$(2n+4)e_0+e_1+p$	$(n+4)e_0$	$(2n+2)e_0+p+e_1$	$(3n+5)e_0+e_1+p$	$3p+ne_0\|\|4p+ne_0$	$3p+ne_0$
VF-PPBA	$(3n+6)e_0+2e_1+2p$	$(n+4)e_0$	$(2n+4)e_0+p+e_1$	$(3n+5)e_0+e_1+p$	$3p+ne_0\|\|4p+(n+2)e_0$	$3p+ne_0$

TABLE IV: Storage cost comparisons of our solutions with related schemes

Scheme	**Costs at client side**			**Costs at server side**	
	pp storage	sk storage	at storage	Original ct storage	Transformed ct storage
LCL+ [33]	$(2n+1)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	(2n+1)$\|\mathbb{G}_0\|$	$(2n+2)\|\mathbb{G}_0\|$	$(n+2)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$
BGT+ [37]	$(n+1)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(n+2)\|\mathbb{G}_0\|$	$(n+5)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$5\|\mathbb{G}_0\|+3\|\mathbb{G}_T\|$
ST [38], [39]	$(8n+4)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(4n+2)\|\mathbb{G}_0\|$	$(8n+4)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(4n+2)\|\mathbb{G}_0\|+3\|\mathbb{G}_T\|$	$(4n+4)\|\mathbb{G}_0\|+3\|\mathbb{G}_T\|$
LS [34]	$(2n+10)\|\mathbb{G}_0\|+3\|\mathbb{G}_T\|$	$3n\|\mathbb{G}_0\|$	$3n\|\mathbb{G}_0\|$	$(n+4)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(2n+6)\|\mathbb{G}_0\|+2\|\mathbb{G}_T\|$
GSB+ [35]	$(n+4)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(n+2)\|\mathbb{G}_0\|$	$(3n+4)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(2n+3)\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$(2n+4)\|\mathbb{G}_0\|+2\|\mathbb{G}_T\|$
HYF [40]	$(3n+5)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$\|\mathbb{G}_0\|$	$(3n+3)\|\mathbb{G}_0\|$	$(2n+3)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$
XJW+ [30]	$(3n+5)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$\|\mathbb{G}_0\|$	$(3n+3)\|\mathbb{G}_0\|$	$(2n+3)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$
DZQ+ [36]	$(3n+5)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$\|\mathbb{G}_0\|$	$(3n+3)\|\mathbb{G}_0\|$	$(2n+3)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$
PPMR-BPRE	$(2n+4)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+n\|\mathbb{Z}_p\|$	$(n+7)\|\mathbb{G}_0\|+n\|\mathbb{Z}_p\|$	$(n+2)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(n+4)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$
VF-PPBA	$(2n+4)\|\mathbb{G}_0\|+2\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+n\|\mathbb{Z}_p\|$	$(n+8)\|\mathbb{G}_0\|+n\|\mathbb{Z}_p\|$	$(n+3)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$	$(n+8)\|\mathbb{G}_0\|+\|\mathbb{G}_T\|$

not require the assistance of hash functions as random oracles to complete the relevant security queries. The term "flexible ciphertext conversion" refers to the process of converting ciphertext from one set of recipients to another set of new recipients. High efficiency here means that the system efficiency is proportional to the number of shared users rather than the number of attributes, due to the fact that the efficiency of attribute-based solutions is typically much higher than that of identity-based broadcast methods.

As revealed from TABLE II, we can readily summarize the following conclusions: the work in LCL+ [33] supports flexible ciphertext conversion in untrustworthy cloud environments. Identity anonymity and flexible ciphertext conversion in semi-trusted clouds are realized in works BGT+ [37] and ST [38], [39]. In the same semi-trusted cloud environments, the solution in LS [34], HYF [40], XJW+ [30] and DZQ+ [36] can enable flexible ciphertext transformation. Some desirable features, such as verifiability, malicious accusation resistance and flexible ciphertext conversion, are offered in GSB+ [35]. The PPMR-BPRE enables identity anonymity, and flexible ciphertext conversion in semi-trusted cloud scenarios. The VF-PPBA used for data sharing in malicious cloud environments

can simultaneously achieve additional advantages than the PPMR-BPRE, such as verifiability and malicious accusation resistance. Except for the construction in BGT+ [37], the rest of other works are constructed based on prime order groups in the standard model.

From TABLE II, we can further conclude that only the works [37]–[39] and our framework realize the anonymity in the standard model. The practical functionalities in malicious cloud environments, such as malicious accusation resistance, can only be achieved in [35] and our VF-PPBA. Based on prime order groups in the standard model, our VF-PPBA is the first work that can simultaneously realize anonymity, verifiability and malicious accusation resistance and high efficiency.

B. Theoretical Analysis

We theoretically analyze and compare the computation and storage cost of our works and other related works in TABLEs III and IV. More specifically, we mainly focus on the most time-consuming calculations in bilinear groups, including exponentiation operation and bilinear pairings). For ease of comparison, let n be the number of attributes or the length of the identity vector. In TABLE III, e_0, e_1 and p denote the execution time of a single exponentiation in $\mathbb{G}_0$, a single exponentiation in $\mathbb{G}_T$ and one bilinear pairing, respectively. In TABLE IV, the size of a single group element in $\mathbb{G}_0$, $\mathbb{G}_1$ and $\mathbb{G}_T$ are correspondingly denoted as $|\mathbb{G}_0|$, $|\mathbb{G}_1|$ and $|\mathbb{G}_T|$. The former "(.)" and the latter refer to the decryption computation cost of the original ciphertext and re-encryption ciphertext in "$(.)\|(.)$" presented in TABLE IV.

As seen from TABLE III, We discover that the calculation costs of the setup algorithm in [30], [33], [34], [36]–[40] and our framework all increase linearly with the number of attributes or the length of the identity vector; the computation costs of generating secret keys for users in [33]–[35], [37]–[40] and our construction are proportional to the number of attributes or identities owing to that a set of attributes or an identity vector is embedded into a secret key; since an access policy or access vector is embedded in the ciphertext or authorization token, the computation costs of encryption and authorization algorithms in all works grow linearly with the number of attributes in the access policy or the access vector length, allowing the data to be accessed by a group of authorized users. Because the authorization key associated with a new access control delegated to the clouds in all works is used for ciphertext conversion, their calculation costs of ciphertext transformation follow linear correlations with the number of attributes or the length of the identity vector. In all works, the computation costs of decrypting an original ciphertext or a transformed ciphertext increase linearly because the secret key associated with the number of attributes or an identity vector is used for decryption regardless of whether the original ciphertext or transformed ciphertext is decrypted.

It is also easy to deduce from TABLE III that the decryption costs in MR-BPRE and VF-PPBA are both independent of the number of pairing computation operations compared to other works. As depicted from TABLE IV, it can be easily derived that the storage costs for storing the public parameter (pp), secret key (sk), authorization token (ak) original ciphertext and transformation ciphertext in all works rise in proportion to the number of attributes, identities, or the vector length involved. We can also summarize that the storage costs for storing sk in our solutions are relatively lower and our storage costs of storing the original or transformed ct, pp and ak are also comparatively lower.

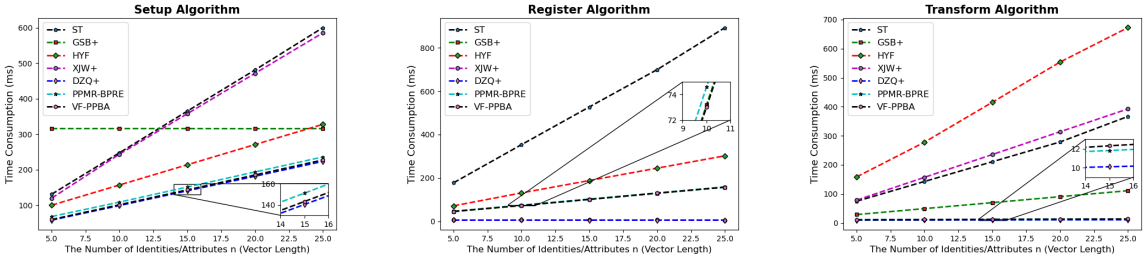
C. Experimental Analysis

In this part, we only compare the performance of the most prominent and state-of-the-art schemes [30], [35], [36], [38], [40], MR-BPRE and VF-PPBA since the chosen comparison works to achieve flexible ciphertext transformation represent the best scheme of IP-PRE [38], ABPRE [35] and MR-BPRE [30], [36], [40], respectively. We implement the experiments of our framework and [35], [36], [38] in Python 3.6.13 using Charm 0.43, PBC-0.5.14 library, OpenSSL-1.1.1 [49] to evaluate the performance superiority of our methodology. We utilize SS512 curve for paring to achieve an acceptable 80-bit security level. The experimental simulations are conducted 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 plays the role of 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 works as the role of a mobile user.

In our implementation, the idea of key encapsulation for backward compatibility is also followed. Here, we utilize 128-bit AES keys to encrypt the real data (refer to medical images) based on a modified AES algorithm [50] and encode the AES keys with the encryption algorithm of our MR-BPRE, VF-PPBA and related works. The medical images tested in our experiments are chosen from the dataset from the Cancer Imaging Archive (https://www.smir.ch/BRATS/Start2015). The detailed experimental source codes can be publicly found in https://github.com/xuehuan-yang/VFPPBA.

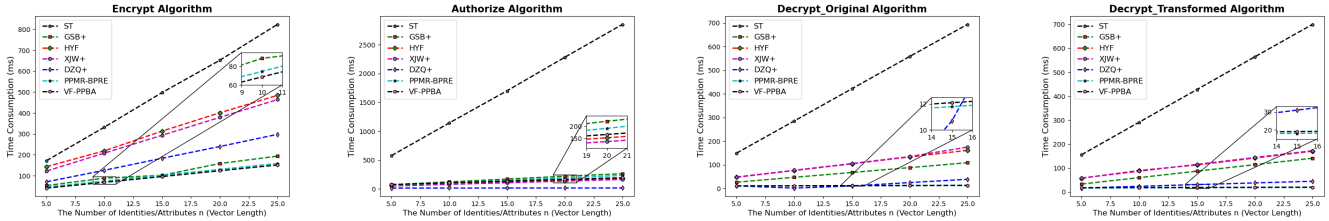
Fig. 2 shows the running time comparisons at the side of TA and CSP for Setup, Register and Transform algorithms. Specifically, Figs. 2(a), 2(b) and 2(c) present the time consumption of performing Setup, Register and Transform algorithms to produce public key, secret key and transformed ciphertext, respectively. As seen from Fig. 2(a), we can learn that other than that in GSB+ [35], the calculation time of Setup algorithm in other schemes including ST [38], HYF [40], XJW+ [30], DZQ+ [36], MR-BPRE and VF-PPBA grow linearly with the number of identities/attributes or the length of access vector. From Figs. 2(b) and 2(c), it can be shown that the time consumption of both the Register and Transform algorithms in all simulated schemes increases with the number of identities/attributes or access vector length to be encrypted. We can also capture that the running time of Setup algorithm in our MR-BPRE and VF-PPBA is slightly higher than that in DZQ+ [36], but much lower than that in HYF [40], XJW+ [30], GSB+ [35] and ST [38]. It is also simple to conclude that the solution in GSB+ [35], XJW+ [30], MR-BPRE and VF-PPBA are much smaller in the running time of Register algorithm than HYF [40], ST [38], but a bit higher than

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.2022.3226577



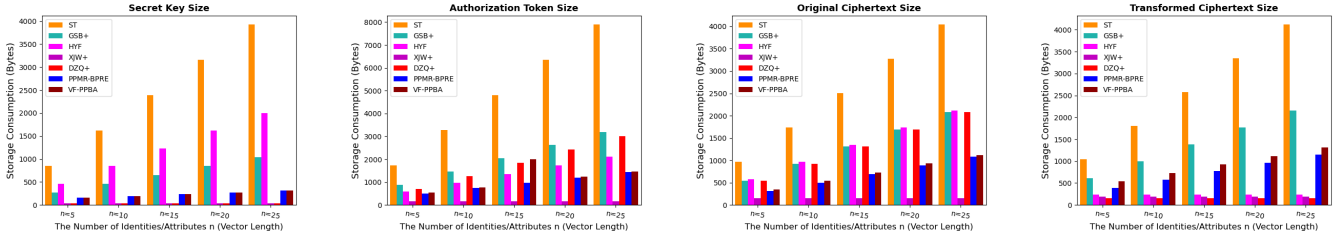
(a) Setup algorithm (b) Register algorithm (c) Transform algorithm

Fig. 2: Running time at the registry authority and the server side for Setup, Register and Transform algorithms



(a) Encrypt algorithm (b) Authorize algorithm (c) Decrypt algorithm (d) Decrypt algorithm

Fig. 3: Running time at the user side for Encrypt, Authorize and Decrypt algorithms



(a) Secret key (sk) size (b) Authorization token (at) size (c) Original ciphertext (ct) size (d) Transformed ciphertext size

Fig. 4: Storage consumption of sk, at, original ct and transformed ct

DZQ+ [36]. The time expense of Register algorithm in GSB+ [35], XJW+ [30] is nearly the same as that in VF-PPBA and somewhat lower than that in MR-BPRE. In addition, it is also simple to observe that the time conducting Transform algorithm of our MR-BPRE and VF-PPBA is marginally higher than DZQ+ [36], but considerably lower than HYF [40], XJW+ [30], GSB+ [35] and ST [38].

Fig. 3 indicates the execution time comparisons at the side of data owner, data recipients for Encrypt, Authorize and Decrypt algorithms. In more detail, Figs. 3(a), 3(b), 3(c) and 3(d) reflect the time cost of conducting Encrypt, Authorize and Decrypt algorithms to correspondingly produce ciphertext, authorization token, and cleartext. From Fig. 3, it is not hard to derive that with the number of identities/attributes or the vector length varying from 5 to 25 with an interval of 5, the time consumption to produce each related parameter in each respective sub-figure increases linearly. It can be also observed from Fig. 3(a) that for the computation efficiency of Encrypt, our MR-BPRE and VF-PPBA are slightly better than GSB+ [35], but more efficient than HYF [40], XJW+ [30], ST [38] and DZQ+ [36]. As well, we can also find from Fig. 3(b) that for the Authorize computation efficiency, our MR-BPRE and VF-PPBA are slightly lower than HYF [40], XJW+ [30], GSB+ [35], but more efficient than ST [38] and DZQ+ [36]. As presented in Figs. 3(c) and 3(d), we easily reveal that when n varies from 5 to 15, the decryption running time decoding the original ciphertext of our MR-BPRE and VF-PPBA is somewhat higher than that of DZQ+ [36], but slightly lower than that of DZQ+ [36] after n ranges from 16 to 25. In addition, we also observe that the time consumption of decrypting original ciphertext in ours is greatly lower than that in HYF [40], XJW+ [30], ST [38] and GSB+ [35]. For the time consumption of running Decryption algorithm deciphering the transformed ciphertext shown in Fig. 3(d), our approaches are slightly lower than DZQ+ [36], and significantly lower than HYF [40], XJW+ [30], ST [38] and GSB+ [35]. The reason leading to the high efficiency of MR-BPRE and VF-PPBA mainly stems from that the number of bilinear map computations involved in ours is apparently much less than that in others. For the execution time comparison of our solutions in transformed ciphertext decryption, the cost of our VF-PPBA is somewhat higher than that of our MR-BPRE

since realizing additional verifiability requires a little time to produce related parameters.

Fig. 4 indicates the storage consumption of sk, at, original ct and transformed ct. As seen from Fig. 4(a), it is readily concluded that when n varies from 5 to 10, the storage cost of storing sk in our MR-BPRE and VF-PPBA is always much lower than that in ST [38], HYF [40] and GSB+ [35] and slightly higher than that in XJW+ [30], DZQ+ [36]. From Figs. 4(b) and 4(c), it can be also seen that the storage consumption of keeping at and original ct in our solutions is the smallest except that the XJW+ [30]. Besides, we can also observe from Fig. 4(d) that our MR-BPRE and VF-PPBA consume lower storage resources in storing transformed ct than ST [38] and GSB+ [35], but require more storage resources than HYF [40], XJW+ [30], DZQ+ [36]. For the storage consumption of at, original and transformed ct, our VF-PPBA is always higher than MR-BPRE since achieving more functionalities such as verifiability and fairness requires more related parameters.

In summary, the smaller size of the secret key and ciphertext commonly means faster time execution, and the lower time consumption of the algorithm implies better performance for the clients or efficiency evaluation. From the above-detailed illustrations, we can learn that the efficiency of performing data encryption and decryption in our MR-BPRE and VF-PPBA outperforms other related solutions, which makes our approaches more practical in real data sharing scenarios.

VIII. Conclusion

In this paper, prior to devising a verifiable, fair and privacy-preserving broadcast authorization (VF-PPBA) scheme for flexible data sharing in untrustworthy cloud environments, we first invented a privacy-preserving multi-recipient broadcast proxy re-encryption scheme (PPMR-BPRE) for flexible data sharing in semi-trusted cloud environments. Our PPMR-BPRE not only allows any initially authorized recipient to realize efficiently dynamical data sharing, but also makes any adversarial users unable to learn authorized recipients' identity privacy from both the original and transformed ciphertexts. Then, we designed a VF-PPBA scheme based on the proposed PPMR-BPRE, which in addition to featuring all functionalities, enables verifiability and malicious accusation resistance for the outcome of the outsourced ciphertext conversion tasks. The extensive rigorous security proofs demonstrated that both PPMR-BPRE and VF-PPBA are secure. Next, we further enhanced our proposals from both efficiency and functionalities. Finally, we studied the performance via experimental simulations to showcase more practicability and effectiveness of our PPMR-BPRE and VF-PPBA compared to the existing prominent and state-of-the-art solutions.

References

[1] K. Cao, S. Hu, Y. Shi, et al., "A survey on edge and edge-cloud computing assisted cyber-physical systems", *IEEE TII,* vol. 17, no. 11, pp. 7806-7819, 2021.

[2] T. Mastelic, A. Oleksiak, H. Claussen, et al., "Cloud computing: Survey on energy efficiency", *ACM Computing Surveys*, 2014, vol. 47, no. 2, pp. 1-36, 2014.

[3] M. Manulis, C. P. Bridges, R. Harrison, et al., "Cyber security in new space", *International Journal of Information Security,* vol. 20, no. 3, pp. 287-311, 2021.

[4] T. Spil, R. Klein, "Personal health records success: why Google Health failed and what does that mean for Microsoft HealthVault?", *IEEE ICSS*, pp. 2818-2827, 2014.

[5] L. Stephanie, R. S. Sharma, "Digital health eco-systems: An epochal review of practice-oriented research", *IJIM,* vol. 53, pp. 102032, 2020.

[6] Z. Wan, J. Liu, R. H Deng, "HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing", *IEEE TIFS,* Vol. 7, No., 2, pp. 743-754, 2012.

[7] J. Lai, R.H. Deng, et al., "Attribute-based encryption with verifiable outsourced decryption", *IEEE TIFS,* Vol. 8, No. 8, pp. 1343-1354, 2013.

[8] D. Boneh, C. Gentry, B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys", *Crypto 2005*, Springer, LNCS 3621, pp. 258-275, 2005.

[9] D. Boneh, M. Franklin, "Identity-based encryption from the weil pairing", *CRYPTO 2001*, Springer, LNCS 2139, pp. 213–229, 2001.

[10] J. Wei, W. Liu, X. Hu, "Secure data sharing in cloud computing using revocable-storage identity-based encryption", *IEEE TCC,* vol. 6, no. 4, pp. 1136-1148, 2018.

[11] R. Karati, R. Amin, S. H. Islam, et al., "Provably secure and lightweight identity-based authenticated data sharing protocol for cyber-physical cloud environment", *IEEE TCC,* vol. 9, no. 1, pp. 318-330, 2021.

[12] D. He, N. Kumar, et. al., "A provably-secure cross-domain handshake scheme with symptoms- matching for mobile healthcare social network," *IEEE TDSC,* vol. 15, no. 4, pp. 633–645, Jul. 2018.

[13] Y. Yang, Y. Chen, F. Chen, et al., "An efficient identity-based provable data possession protocol with compressed cloud storage", *IEEE TIFS,* vol. 17, pp. 1359-1371. 2022.

[14] M. Abdalla, M. Bellare, D. Catalano, et al., "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions", *Crypto 2005,* Springer, Berlin, Heidelberg, LNSC 3621, 205-222, 2005.

[15] X. Boyen, B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)", *Crypto 2006*, Springer, Berlin, Heidelberg, LNCS 4117, 290-307, 2006

[16] J. Han, L. Chen, S. Schneider, et al., "Anonymous single sign-on with proxy re-verification", *IEEE TIFS,* vol. 15, pp. 223-236, 2019.

[17] M. Umar, S.K. Islam, et al., "Provable Secure Identity-Based Anonymous and Privacy-Preserving Inter-Vehicular Authentication Protocol for VANETS Using PUF", *IEEE TVT,* vol. 70, no. 11, pp 12158-12167, 2021.

[18] Y. Li, Q. Cheng, X. Liu, et al., "A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing", *IEEE Systems,* vol. 15, no. 1, pp. 935-946, 2021.

[19] C. Delerablee, "Identity-based broadcast encryption with constant size ciphertexts and private keys", *EUROCRYPTO 2007*, Springer, Berlin, Heidelberg, 200-215, 2007.

[20] A. Ge, P. Wei, "Identity-based broadcast encryption with efficient revocation," *PKC 2019*, Springer, Cham, pp. 405-435, 2019.

[21] L. Liu, Y. Zhang, X. Li, "KeyD: secure key-deduplication with identity-based broadcast encryption," *IEEE TCC,* vol. 9, no. 2, pp. 670-681, 2018.

[22] J. Kim, W. Susilo, M. H. Au, et al., "Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext", *IEEE TIFS,* vol. 10, no. 3, pp. 679-693, 2015.

[23] J. Kim, S. Camtepe, W. Susilo, et al., "Identity-based broadcast encryption with outsourced partial decryption for hybrid security models in edge computing", *AisaCCS 2019*, pp. 55-66, 2019.

[24] J. Wei, X. Chen, J. Wang, et al., "Enabling (end-to-end) encrypted cloud emails with practical forward secrecy", *IEEE TDSC,* 2021.

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

[26] 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.

[27] C. I. Fan, L. Y. Huang, P. H. Ho, "Anonymous multi-receiver identity-based encryption", *IEEE TC,* 2010, vol. 59, no. 9, pp. 1239-1249, 2010.

[28] M. Blaze, G. Bleumer, M. Strauss, "Divertible protocols and atomic proxy cryptography", *EUROCRYPT*, LNCS 1403, pp. 127-144, 1998.

[29] L. Zhang, H. Ma, Z. Liu, and E. Dong, "Security analysis and improvement of a collusion-resistant identity-based proxy re-encryption scheme," *ICBWCCA*. Springer, pp. 839–846, 2016.

[30] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email", *IEEE TC,* vol. 65, no. 1, pp. 66–79, 2016.

[31] C. Ge, Z. Liu, et al., "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds,", *IEEE TDSC,* vol. 18, no. 3, 2021.
[32] H. Deng, Z. Qin, et al.,, "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud", *IEEE TIFS,* vol. 15, pp. 3168–3180, 2020.
[33] X. Liang, Z. Cao, H. Lin, et al., "Attribute based proxy re-encryption with delegating capabilities", *AsiaCCS 2009*, pp. 276-286, 2009
[34] K. Liang, W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage", *IEEE TIFS,* vol. 10, no. 9, pp. 1981-1992, 2015.
[35] C. Ge, W. Susilo, J. Baek, et al., "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds", *IEEE TDSC,* DOI: 10.1109/TDSC.2021.3076580, 2021.
[36] H. Deng, J. Zhang, Z. Qin, et al., "Policy-based Broadcast Access Authorization for Flexible Data Sharing in Clouds", *IEEE TDSC,* DOI: 10.1109/TDSC.2021.3080282, 2021.
[37] M. Backes, M. Gagné, S. Thyagarajan, "Fully secure inner-product proxy re-encryption with constant size ciphertext", *International Workshop on Security in Cloud Computing,* pp. 31-40, 2015.
[38] M. Sepehri, A. Trombetta, "Secure and efficient data sharing with atribute-based proxy re-encryption scheme", *ICARS,* pp. 1-6, 2017.
[39] M. Sepehri, A. Trombetta, "Secure data sharing in cloud using an efficient inner-product proxy re-encryption scheme", *Journal of Cyber Security and Mobility,* pp. 339–378, 2017.
[40] Q. Huang, Y. Yang, J. Fu, "PRECISE: Identity-based private data sharing with conditional proxy re-encryption in online social networks", *FGCS,* vol. 86, pp. 1523-1533, 2018.
[41] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," *IEEE S & P Conference*, IEEE, pp. 1–16, 2011.
[42] B. Qin, R. H. Deng, et al., "Attribute-based encryption with efficient verifiable outsourced decryption", *IEEE TIFS,* vol. 10, no. 7, pp. 1384–1393, 2015.
[43] J. Lai, R. H. Deng, et al.,, "Attribute-based encryption with verifiable outsourced decryption", *IEEE TIFS,* vol. 8, no. 8, pp. 1343–1354, 2013.
[44] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation", *Commun. ACM,* vol. 59, no. 2, pp. 103–112, 2016.
[45] L. Grassi, D. Khovratovich, C. Rechberger, et al., "Poseidon: A New Hash Function for Zero-Knowledge Proof Systems", *USENIX Security 21*, pp. 519-535, 2021.
[46] Z. Wan, Y. Zhou, K. Ren, "ZK-AuthFeed: Protecting Data Feed to Smart Contracts with Authenticated Zero Knowledge Proof", *IEEE TDSC,* DOI: 10.1109/TDSC.2022.3153084, 2022.
[47] J. Sun, H. Xiong, X. Liu, et al., "Lightweight and privacy-aware fine-grained access control for IoT-oriented smart health", *IEEE IoTJ,* vol. 7, no. 7, pp. 6566-6575, 2020.
[48] 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).
[49] J. A. Akinyele, C. Garman, I. Miers, et. al., "Charm: a framework for rapidly prototyping cryptosystems", *Journal of Cryptographic Engineering,* pp. 111–128, 2013.
[50] M. Zeghid, M. Machhout, L. Khriji, et al., "A modified AES based algorithm for image encryption", *IJCSE*, vol. 1, no. 1, pp. 70-75,2007.

Jianfei Sun is currently a research fellow at School of Computer Science and Engineering, Nanyang Technological University. His research interests include network security and IoT security. He has published many paper 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 at the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He has published many paper on ACM CCS, IEEE TDSC, IEEE TIFS, IEEE TII, IEEE TVT, ACM ACSAC, ACM AsiaCCS, etc. His research interests include AI security and privacy-preserving issues in Deep Learning.

Tianwei Zhang is an assistant professor at the 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 his Ph.D. degree at Princeton University in 2017.

Xuehuan Yang received his Bachelor's degree from Nanyang Technological University. Now he is a Ph.D. in the 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 an Associate 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 the 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.