Privacy-Improving Multi-Authority Ciphertext-Policy Attribute-Based Encryption with Internal Fraud Attack Resistance Based on Blockchain

Zhaoqian Zhang, Bei Gong, *Member, IEEE*, Xin Fan, Yilin Yuan, Fan Yang, Weizhi Meng, *Senior Member, IEEE* and Qiang Zhu

Abstract—Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a promising cryptographic mechanism for protecting data confidentiality, and Multi-Authority CP-ABE (MACP-ABE) further widely employed due to its stronger scalability. However, MACP-ABE suffers from privacy issues and internal fraud attacks. To protect user privacy, previous researches adopted the Anonymous Credential System (ACS), which is centralized and reduces the reliability of the scheme. Moreover, they ignore the internal fraud attacks launched through collusion between attribute authorities and malicious insiders, leading to unauthorized data access. In this paper, we propose the first privacy-improving MACP-ABE scheme capable of resisting internal fraud attacks. First, we use smart contracts to perform anonymous and credible identity authentication. We allow users to participate with pseudonyms, ensuring that the traceability of any pseudonym cannot be linked to a specific user. Furthermore, we present a blockchain-based anonymous key distribution protocol, where the key issuing process is recorded and verified by the blockchain. This ensures that malicious insiders and corrupt attribute authorities cannot bypass the blockchain to perform spurious key distribution, safeguarding data confidentiality. We analyze the security of our scheme rigorously and compare it with others comprehensively. The performance comparison shows that the computation and storage overheads of our scheme are affordable.

Index Terms—MACP-ABE, Privacy Preserving, Access Control, Blockchain, Data Sharing

1 Introduction

The public cloud has become a primary method for individuals and organizations to leverage the value of data, owing to its higher performance, greater flexibility, and superior customer support. However, once data is uploaded to the cloud, users lose absolute control over it, putting data confidentiality at serious risk. To protect data confidentiality while improving sharing efficiency, Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [1] was proposed, where both the policy and the private key are associated with attributes. The data owner encrypts data using a policy composed of a set of attributes, and any user whose attributes satisfy the policy can decrypt the ciphertext with their private keys. This "one-to-many" encryption/decryption mode and fine-grained access control have made CP-ABE a significant focus of both scholarly research and practical application since its inception.

In a typical CP-ABE scheme, a single attribute authority is responsible for distributing private keys to users, which can easily become a bottleneck for the entire system. To address this issue, Multi-Authority CP-ABE (MACP-ABE) [2], [3], [4] was proposed, involving multiple attribute authorities for key distribution, thus enhancing the scalability of CP-ABE. Furthermore, MACP-ABE supports a wider range of application scenarios. For example, as shown in Fig.1, data users with the attributes "*Doctor*" from the hospital and "*Researcher*" from the institute can decrypt the ciphertext which encrypted by the access policy "*Hospital.Doctor* AND *Institute.Researcher*". However, due to the independence of the attribute authorities in key distribution, users with the attribute "*Doctor*" and users with the attribute "*Researcher*" could potentially combine their private keys or decryption results to crack the ciphertext, resulting in unauthorized data access. To prevent such situations, MACP-ABE assigns

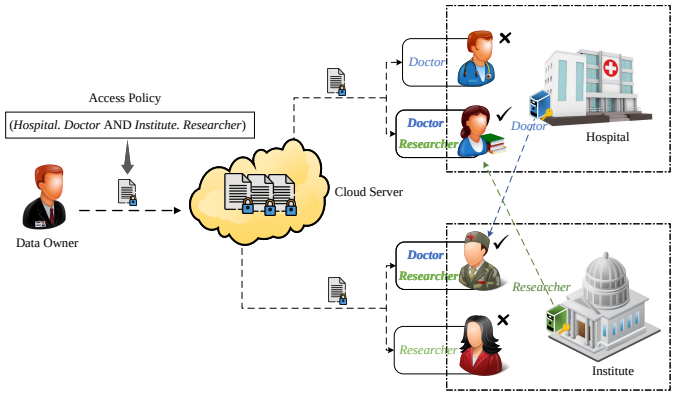


Fig. 1. Multi-Authority CP-ABE.

- *Z.Zhang, X.Fan and Q.Zhu are with the Science and Technology Research Institute, Three Gorges Corporation, 101199, Beijing, China.*
- *B.Gong is with the Beijing Key Laboratory of Trusted Computing, Beijing University of Technology, 100124, Beijing, China. E-mail: gongbei@bjut.edu.cn.*
- *Y.Yuan and F.Yang are with College of Information Engineering, Beijing Institute of Graphic Communication, 102600, Beijing, China.*
- *W. Meng is with the School of Computing and Communications, Lancaster University, United Kingdom. E-mail: w.meng3@lancaster.ac.uk.*

This work was supported by Major Science and Technology Projects in Yunnan Province (202202AD080013) and Natural Science Foundation of Hubei Province (2022CFD025)

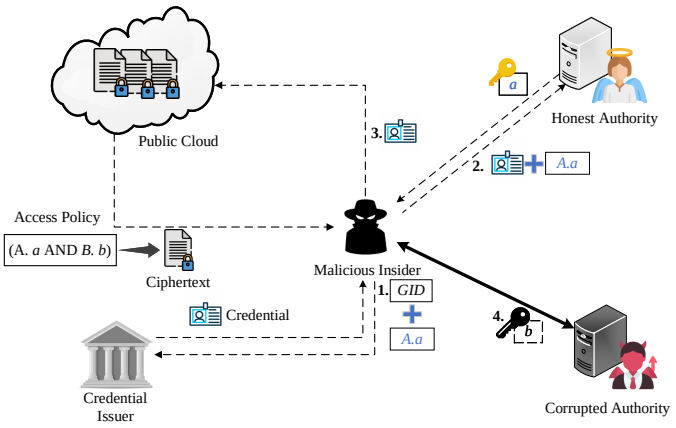


Fig. 2. Internal Fraud Attack.

a Global Identifier (GID) to each user and binds it to the user's private keys. While GID prevents the users collusion attacks, it raises a new privacy concern: the attribute authorities could collusively obtain all of a user's attributes and corresponding private keys through the GID.

Currently, MACP-ABE schemes typically adopt the Anonymous Credential System (ACS) [5] to mitigate the risk of collusion attacks among authorities. In ACS, an anonymous credential is generated by an issuer who verifies the user's GID and attributes. The user then proves his identity by demonstrating possession of this credential when requesting private keys, without disclosing any sensitive information. While ACS reduces the risk of privacy leakage, it reintroduce bottlenecks that limit the whole scheme, resulting in decreased reliability. Additionally, the issuer must be fully trusted, posing challenges in detecting and addressing any potential malfeasance by the issuer.

More seriously, existing MACP-ABE schemes only consider users collusion and authorities collusion, neglecting the possibility of user-authority collusion. Although it is challenging for an external malicious user to conspire with a corrupt attribute authority to access data, a malicious insider with a legal identity can easily do so. Consider the scenario depicted in Fig.2. Even though the malicious insider only has the attribute "a" from the honest authority, he can decrypt the ciphertext encrypted by the access policy $(A.a$ AND $B.b)$ through the following steps. First, he applies an anonymous credential from the issuer using GID and attribute "a". Second, he applies for private keys from the honest authority by proving possession of the anonymous credential. Third, he requests the ciphertext from the public cloud by proving possession of the anonymous credential. Fourth, he colludes with the corrupt authority to obtain the private keys corresponding the attribute "b". Finally, the malicious insider combines the private keys of "a" and "b" to generate illegal private keys and decrypt the ciphertext. We refer to this type of attack as an internal fraud attack, wherein a malicious insider exploits his legitimate credentials to perpetrate fraud and access data.

1.1 Our Contributions

In this paper, we propose a privacy-improving MACP-ABE scheme that is the first to resist the internal fraud attacks. The contributions of this paper are summarized as follows.

1) We present a decentralized anonymous authentication mechanism based on smart contract to achieve trustworthy and privacy-preserving identity authentication. The core of the mechanism consists of two aggregators implemented by smart contracts: the GID aggregator and the attribute aggregator. The GID aggregator contains Pedersen commitments [6] for all users' GIDs, and is used to verify that pseudonyms correspond to the same GID. The attribute aggregator contains Pedersen commitments for all users' attribute sets, and is used to verify the correctness of a user's attribute subset when requesting private keys, preventing the submission of fake attributes. The trace of any user can only be linked to the two aggregators recorded on the blockchain, effectively preventing the authorities collusion attacks.
2) We propose an anonymous key distribution protocol based on blockchain to address the internal fraud attacks. Following the above authentication, user first executes the Non-Interactive Set Membership Proof (NISMP) [7] protocol with the attribute authority to get a "Semi-Finished" key. This process is transparently verified and recorded by the blockchain. Subsequently, we employ a cloud server to impose the "On-Chain" status on the "Semi-Finished" key, after which the final private key is recovered locally by the user. Combined with anonymous authentication, neither the user nor the attribute authority can pass the blockchain's verification with fake attribute commitments. Therefore, a malicious insider cannot obtain the private key with On-Chain status and perform unauthorized access. Moreover, This method of constructing private keys through three parties can prevent the attribute authority and the cloud server from recovering the private keys.
3) We rigorously analyze the scheme's security in theory and conduct a comprehensive comparison with other schemes in terms of functionality and performance. Besides, we locally evaluate the performance of the blockchain platform used in our scheme.

1.2 Organization

The rest of this paper is organized as follows. The related work is introduced in Section 2, and the preliminaries is in Section 3. We present the overview of our scheme in Section 4, and Section 5 contains the detailed designs. The security of the scheme is rigorously proved and detailed analyzed in Section 6. And we make comprehensive comparisons and evaluate the 7. Finally, we present a conclusion in Section 8.

2 Related Work

CP-ABE has been widely adopted for data sharing due to its fine-grained access control capabilities. MACP-ABE extends single attribute authority to multiple attribute authorities, which alleviates the burden on a single authority and broadens the application scope. Resistance to users collusion is

a fundamental security requirement in the attribute-based cryptography, and this challenge is intensified in MACP-ABE due to the independence of authorities. To address this issue, Chase [2] first proposed binding a GID to a user's private key to prevent users collusion. This approach has been maintained in numerous MACP-ABE schemes [8], [9], [10], [11], [12], [13]. However, these schemes overlook the privacy issues that arise from authorities collusion, which is a critical concern that any robust MACP-ABE scheme must address.

The first privacy-preserving MACP-ABE scheme was proposed by Chase et al. [14], which uses the ACS to hide the GID. However, this scheme uses composite-order groups and requires each pair of authorities to perform a two-party key exchange to construct their public and master keys, which is inefficient. Despite this, the introduction of the ACS has become the primary approach to addressing privacy issues in subsequent MACP-ABE schemes. Qian et al. [15] proposed a privacy-preserving MACP-ABE scheme with prime-order groups, but it still requires collaboration between authorities. The scheme presented by Lyu et al. [16] avoids the collaboration, but its access structure is limited to AND-Gate, which is not expressive. Ye et al. [17] proposed a decentralized access control scheme for smart grids, which uses Linear Secret Sharing Scheme (LSSS) access structure and hides the GID through an identity certificate similar to the ACS. Zhang et al. [18] designed a privacy preserving multi-authority access control scheme supporting traceability and revocation. It employs the anonymous identity credential to protect the GID,akin to Chase's scheme [14]. Similar approaches include recently proposed schemes by Zhang et al. [19], Ma et al. [20], Su et al. [21], Ashouri et al. [22], and Liang et al. [23]. Not all privacy-preserving MACP-ABE schemes employ the ACS to prevent GID leakage. For example, Fan et al. [24] and Roy et al [25] hides the GID by constructing a GID-based pseudonym but neglects the verification of it, making it vulnerable to attacks such as the witch attack, posing serious security risks. The above schemes only prevent the GID from being disclosed, requiring users to submit privacy-related attributes when applying for private keys. Therefore, simply hiding the GID is insufficient to fully protect user privacy. Han et al. [7] proposed a privacy-improving MACP-ABE scheme to protect both GID and attributes. In this scheme, the user performs NISMP protocol with the attribute authority to prove that he has correct attributes. However, it suffers from users collusion issues [26]. Subsequently, researchers successively proposed various MACP-ABE schemes that protect both GIDs and attributes during the key generation process, such as [27], [28], [29], [30]. Nevertheless, these schemes still rely on centralized ACS.

Since blockchain is decentralized, some privacy-preserving MACP-ABE schemes have integrated it to secure data. Liu et al. [31] proposed a scheme for supply chain data sharing, utilizing blockchain to store ciphertexts. Although the scheme designs an anonymous key issuing protocol to protect user privacy, it is constructed by composite-order group which is inefficient. Li et al. [32] introduced a blockchain-based privacy-preserving data sharing scheme, which continues to adopt the ACS to protect user identity. Similarly, the access control scheme proposed by Duan et al. [33] for sensitive digital assets also uses the ACS to achieve privacy protection. Zhang et al. [34] proposed a privacy-preserving MACP-ABE scheme with decentralized data storage based on blockchain, which only supports GID hiding.

As of now, blockchain-based privacy-preserving MACP-ABE schemes still follow the ideas of the Chase's scheme [14] and rarely consider protecting both GIDs and attributes.

After a detailed investigation, we found that current privacy-preserving MACP-ABE schemes overly rely on the centralized ACS. More seriously, even if the GID is endorsed by the centralized authority of the system, it is still difficult to resist internal fraud attacks that trigger unauthorized access to data due to the independence of the authorities.

3 PRELIMINARIES

3.1 Access Structure and Linear Secret Sharing Scheme

3.1.1 Access Structure

Let $\{P_1, P_2, P_3, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,P_3,...,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, P_3, ..., P_n\}$, i.e.,$\mathbb{A} \subseteq 2^{\{P_1,P_2,P_3,...,P_n\}} \setminus \{\emptyset\}$ (the access structure and the collection are monotone, respectively).The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ called the unauthorized sets. In our scheme, the role of parties is taken by the attributes, thus the access structure $\mathbb{A}$ (monotone) will contain the authorized sets of attributes.

3.1.2 Linear Secret Sharing Scheme

A LSSS over $\mathbb{Z}_p$ is a secret sharing scheme Π over a set of parties P if the shares for each party form a vector over $\mathbb{Z}_p$ and an $\ell \times n$ matrix M called the share-generating matrix for Π exists. For all $j = 1, 2, ..., \ell$, the j-th row of the matrix M is labeled by a party $\rho(j)$ with $\rho : \{1, 2, ..., \ell\} \to P$ being a function to map a row of the matrix to a party for labeling.

Consider that the column vector $v = (s, v_2, v_3, ..., v_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $v_2, v_3, ..., v_n \in \mathbb{Z}_p$ are randomly chosen, and then Mv is the vector of ℓ shares of the secret s according to Π. The share $(Mv)_j$ belongs to the party $\rho(j)$.

Every LSSS enjoys the linear reconstruction property [4]. Let Π be an LSSS for an access structure $\mathbb{A}$, and $\mathrm{A} \in \mathbb{A}$ be any authorized set, and define $I \subseteq \{1, 2, ..., \ell\}$ as $I = \{j : \rho(j) \in \mathrm{A}\}$. Then the vector $(1, 0, ..., 0)$ is in the span of rows of a matrix M indexed by I, and there are constants $\{\omega_j \in \mathbb{Z}_p\}_{j \in I}$ that for any valid shares λ_j of a secret s according to Π, it holds that $\sum_{j \in I} \omega_j \lambda_j = s$.

3.2 Bilinear Maps and Complexity Assumptions

3.2.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_\mathbb{T}$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and e be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathbb{T}$. The bilinear map e has the following properties:

- Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab} = e(u^b, v^a)$.
- Non-degeneracy: $e(g, g) \neq 1$.

3.2.2 Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption

The definition of decisional q-parallel BDHE problem as follows. Choose a group $\mathbb{G}$ of prime order p according to the security parameter. Randomly choose $a, s, b_1, ..., b_q \in \mathbb{Z}_p$ and g be a generator of $\mathbb{G}$. Given a tuple y=

$$g, g^s, g^a, ..., g^{(a^q)}, , g^{(a^{q+2})}, ..., g^{(a^2 q)}$$
$$\forall 1 \leqslant j \leqslant q \quad g^{s \cdot b_j}, g^{a \cdot b_j}, ..., g^{(a^q / b_j)}, , g^{(a^{q+2}/b_j)}, ..., g^{(a^{2q}/b_j)}$$
$$\forall 1 \leqslant j, k \leqslant q, k \neq j \quad g^{(a \cdot s \cdot b_k / b_j)}, ..., g^{(a^q \cdot s \cdot b_k / b_j)}$$

Algorithm $\mathcal{B}$ distinguishes $e(g,g)^{a^{q+1}s} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$. Define the advantage of $\mathcal{B}$ solving the desision q-parallel BDHE problem under $\mathbb{G}$ and $\mathbb{G}_T$ as $Adv_{\mathcal{A}} =$

$$|\Pr[\mathcal{B}(\overrightarrow{y}, e(g,g)^{a^{q+1}s}) = 0] - \Pr[\mathcal{B}(\overrightarrow{y}, e(g,g)^r) = 0]| \geqslant \epsilon(k)$$

The assumption holds if no polytime algorithm has a non-negligible advantage in solving the desision q-parallel BDHE problem.

3.3 Blockchain and Smart Contract

Blockchain is a distributed ledger based on cryptography. It connects blocks through a chain structure and prevents tampering by cryptographic hashes, digital signatures, and the Merkle tree. A transaction is the smallest unit that carries data in a blockchain system, and there are two kinds of transactions, normal transaction used for interaction between users and contract-invoking transaction used to execute the smart contracts. Once a transaction is packed into a block, and the block is successfully added to the longest chain later, the data in that transaction cannot be modified.

Smart contracts are computer programs deployed on the blockchain network whose codes are verified by blockchain nodes, eliminating the need for a mediator or a third party. Any blockchain node can create and publish smart contracts. Once a smart contract is successfully deployed, its code and execution results are public to all nodes. This ensures more transparent and secured facilitation and performance of the contractual terms. Fig.3 illustrates the invocation process of a smart contract. A blockchain user only needs to publish a contract-invoking transaction, after which miner nodes will execute the smart contract, and the execution result will be recorded in the blockchain.

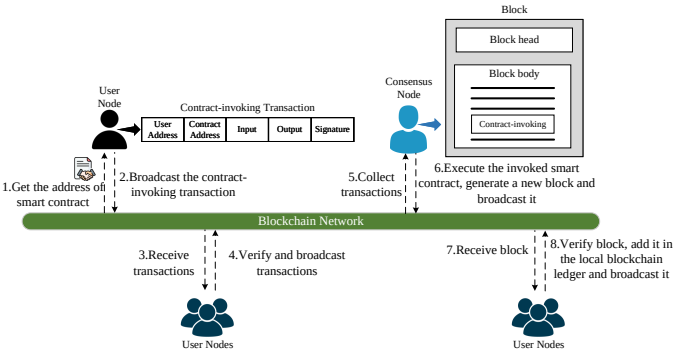


Fig. 3. Process of Invoking Smart Contract.

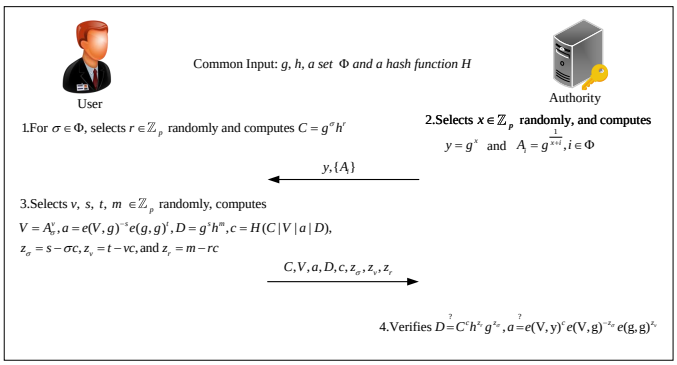


Fig. 4. Non-Interactive Set-Membership Protocol.

3.4 Pedersen Commitment

Pedersen commitment [6] is a cryptographic commitment mechanism, which has the following properties:

- Perfectly hiding: given a commitment C, every value x is equally likely to be the value committed in C, that is receiver cannot learn what is committed in C unless the sender releases it later.
- Computationally binding: assuming the Discrete Logarithm (DL) is hard, the sender cannot open the commitment with another value, that is, the sender cannot convince the receiver that another value x' is committed in C.
- Homomorphic: given two commitment $C_1 = com(a, r_1)$, $C_2 = com(b, r_2)$, where a, b is committed values, r_1, r_2 is random factors, there exists $C_1 \cdot C_2 = com(a+b, r_1+r_2)$.

To ensure the committed value is not disclosed, the sender and the receiver need to perform Zero Knowledge Proof (ZKP) under the Schnorr protocol. We use the notation introduced by literature [35] for various proofs of knowledge of descrete logarithms, for instance, $PoK = \{(\alpha, \beta, \gamma) : y{=}g^\alpha h^\beta \wedge \tilde{y} = \tilde{g}^\alpha \tilde{h}^\gamma\}$ denotes a ZKP of knowledge of integers α, β and γ such that $y{=}g^\alpha h^\beta$ and $\tilde{y} = \tilde{g}^\alpha \tilde{h}^\gamma$ hold on the group $\mathbb{G}{=}\langle g \rangle = \langle h \rangle$ and $\tilde{\mathbb{G}}{=}\langle \tilde{g} \rangle = \langle \tilde{h} \rangle$. In general, Greek letters denote the knowledge being proved, except that the verifier knows all the other parameters. With this representation, a ZKP protocol can concisely represent what is to be proved while hiding all the details.

3.5 Non-Interactive Set-membership Proof

The set-membership proof [36] aims to help the user prove that he holds the correct attributes. In our scheme, to resist internal fraud attack, the interaction process between the user and authority is public to the blockchain so that each blockchain user can verify the correctness of commitments to attributes and private keys, and the cloud server imposes On-Chain status according to the interaction process that has been recorded on the blockchain. To achieve this, we use NISMP protocol, as shown in Fig.4 where $y, \{A_i\}$ are recorded on the blockchain, and H is a cryptographic hash function.

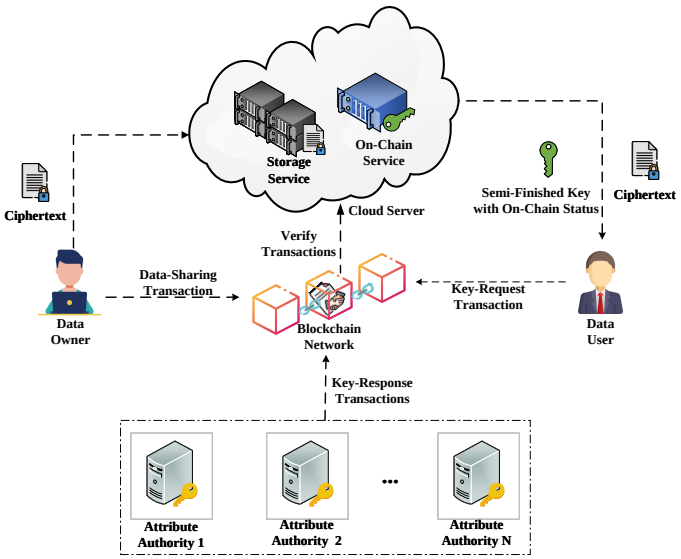


Fig. 5. System Model.

4 Overview of Our Scheme

4.1 System Model

The system model is illustrated as Fig.5. There are five entities: cloud server(CS), data owner(DO), data user(DU), attribute authorities(AAs), and blockchain network(BN).

- **CS** provides both storage service and On-Chain service. The storage service allows users to upload and download the ciphertexts. Any legal user can freely download, but for uploading, CS must check if the Data-Sharing Transaction corresponding to the ciphertext is valid in the BN. The On-Chain service imposes On-Chain status for Semi-Finished keys. Additionally, CS is responsible for setting up global parameters. In our scheme, it is assumed that CS will perform these duties honestly and will not collude with other entities.
- **DO** creates a ciphertext using one or more access structures, each consisting of the attributes from an AA. Before uploading the ciphertext to the CS, DO needs to publish a Data-Sharing Transaction in the BN, which is used to verify the integrity of the data.
- **DU** holds the attributes from one or more AAs, and to maintain anonymity, DU interacts with AAs and CS by pseudonyms. DU performs the anonymous key distribution protocol with AAs through publishing Key-Request Transaction and and gets Semi-Finished key, then he submits it to the CS and gets Semi-Finished key with On-Chain status. Finally, he can recover the final private key locally. If his attribute set satisfies the access structure, he will get the plaintext through decryption.
- **AAs** set up their key pairs and issue the Key-Response Transactions containing the Semi-Finished keys. In our scheme, AAs cannot be fully trusted.
- **BN** verifies the legitimacy of the Key-Request Transactions and Key-Response Transactions, then records the legitimate transactions in the ledger.

4.2 Table of Notations

Table 1 demonstrates the notations in this article.

TABLE 1
Explanation of Symbols

Notation	Description
$\mathbb{G}, \mathbb{G}_T$	Two multiplicative cyclic groups of prime order p
g, h, e	g, h are generators of $\mathbb{G}$,e is bilinear map: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$
κ,PP	Security parameter and public parameter
OPK_i, OMK_i	On-Chain public key, On-Chain master key for AA_i
A_i	Attribute set of AA_i
PK_i, MK_i	Public keys, master keys of AA_i
$\mathcal{M}$	Message for encryption
$\mathbb{A}_i$	Access structure where attributes from AA_i
M, ρ	M is an $\ell \times n$ matrix, ρ maps a row of M to an attribute
CT	Ciphertext
u,S	u is GID of user,S is attribute set of user
$\mathcal{F}$	Attribute authorities involved in the access structures
SK_u^i	Private keys from AA_i for user u
$\mathcal{A}$,$\mathcal{B}$	Adversary and challenger
$\mathcal{H}$	Hash function: $\{0,1\}^* \to \mathbb{Z}_p$
Nym	Pseudonym
ACU, ACT	ACU is the GID aggregator, ACT is the attribute aggregator
T_s, TID_s	Data-Sharing Transaction and its ID
T_{V1}, TID_{V1}	Contract-Invoking Transactions of Pseudonym Verification and its ID
T_{V2}, TID_{V2}	Contract-Invoking Transactions of Attribute Verification and its ID
T_R, TID_R	Key-Request Transaction and its ID
$T_{R'}, TID_{R'}$	Key-Response Transaction and its ID

4.3 Formal Definition

Our scheme contains five algorithms: Global Setup, AA Setup, Encrypt, Key Generation, and Decrypt.

- **Global Setup**$(1^\kappa) \to (PP, OPK_{i\in[1,N]}, OMK_{i\in[1,N]})$.
  CS executes the algorithm to initialize the system. Suppose that the system contains N AAs, and the algorithm takes as input a security parameter κ, and outputs the public parameters PP, the public and master keys (OPK_i, OMK_i) of On-Chain status for $AA_{i\in[1,N]}$.
- **AA Setup**$(PP, OPK_i, A_i) \to (MK_i, PK_i)$.
  $AA_{i\in[1,N]}$ executes the algorithm to initialize the authority. It takes as input public parameter PP, On-Chain status public key OPK_i and attribute set A_i, outputs master keys MK_i and public keys PK_i, where MK_i should be kept secret and PK_i is public in the system.
- **Encrypt**$(PP, \mathcal{M}, \{\mathbb{A}_i, PK_i\}_{i\in\mathcal{F}}, TID_s) \to CT$.
  DO executes the algorithm to create a ciphertext. It takes as input public parameter PP, a message $\mathcal{M}$, a set of access structures $\{\mathbb{A}_i = (M_i, \rho_i)\}_{i\in\mathcal{F}}$, relevant AAs' public keys $\{PK_i\}_{i\in\mathcal{F}}$ and Data-Sharing Transaction ID TID_s, outputs the ciphertext CT. Assuming that $\{\mathbb{A}_i\}_{i\in\mathcal{F}}$ is implicitly included in the ciphertext.
- **Key Generation**$(PP, OMK_i, (PK_i, MK_i), (u, S \cap A_i)) \to SK_u^i$.
  The algorithm is executed by $AA_{i\in[1,N]}$ and DU through the anonymous key distribution protocol. It takes as input public parameter PP, On-Chain status master key OMK_i, AA_i's public keys and master keys PK_i, MK_i, user's GID u and the attribute set $(S \cap A_i)$, it outputs a private key SK_u^i.
- **Decrypt**$(PP, CT, u, \{SK_u^i\}_{i\in\mathcal{F}}) \to \mathcal{M}$.
  DU executes the algorithm to get the message $\mathcal{M}$. It takes as input public parameter PP, ciphertext CT, user's GID u, and private keys $\{SK_u^i\}_{i\in\mathcal{F}}$, outputs $\mathcal{M}$.

4.4 Security Model

The security model for our scheme is defined via the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$. The scheme is secure under the Chosen Plaintext Attack (CPA) if there is no probabilistic polynomial-time adversary has the non-negligible advantage in the following game.

- **Init**. Parameter initialization phase.
  The adversary $\mathcal{A}$ submits a group of corrupt authorities $\mathcal{C} = \{\text{AA}_i\}_{i\in\mathcal{F}'}$ and a set of challenge access structures $\{\mathbb{A}_i = (M_i, \rho_i)\}_{i\in\mathcal{F}^*}$, where $\mathcal{F}^{'}, \mathcal{F}^* \subseteq \{1, 2, ..., N\}$. The restriction on $\mathcal{A}$ is that at least one of $\{\mathbb{A}_i\}_{i\in\mathcal{F}^*}$ is not satisfied by the attribute set provided by $\mathcal{A}$, denoted by $\mathbb{A}^* = (M^*, \rho^*)$. $\mathcal{A}$ sends $\mathcal{C} = \{\text{AA}_i\}_{i\in\mathcal{F}'}$ and $\{\mathbb{A}_i = (M_i, \rho_i)\}_{i\in\mathcal{F}^*}$ to the challenger $\mathcal{B}$.
- **Global Setup**. Global parameters setting phase.
  The challenger $\mathcal{B}$ runs the **Global Setup** algorithm and gives the public parameter PP and On-Chain status public keys $\{OPK_i\}_{i\in[1,N]}$ to $\mathcal{A}$.
- **AA Setup**. Authority parameters setting phase, including two cases.
  Case 1: $\text{AA}_i \in \mathcal{C}$. AA_i is a corrupt AA. $\mathcal{B}$ runs the **AA Setup** algorithm and gives the (PK_i, MK_i) to $\mathcal{A}$.
  Case 2: $\text{AA}_i \notin \mathcal{C}$, which means AA_i is not a corrupt AA. $\mathcal{B}$ runs the **AA Setup** algorithm and gives the PK_i to $\mathcal{A}$.
- **Phase 1**. Private key query phase.
  $\mathcal{A}$ uses a GID u and a set of attributes S to requests the private keys repeatedly, and the $\mathcal{B}$ runs the **Key Generation** algorithm to generate the private keys. There are two cases.
  Case 1: $\text{AA}_i \in \mathcal{C}$. AA_i is a corrupt AA, it is able to collude with $\mathcal{A}$ and gets the illegal private keys. However, the illegal private keys cannot has the On-Chain status.
  Case 2: $\text{AA}_i \notin \mathcal{C}$, which means AA_i is not a corrupt AA. The private keys has the On-Chain status.
- **Challenge**. $\mathcal{A}$ submits two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$. The $\mathcal{B}$ flips a random coin $b \in \{0, 1\}$, and encrypts $\mathcal{M}_b$ into CT^* under $\mathbb{A}_i$. CT^* is given to $\mathcal{A}$.
- **Phase 2**. **Phase 1** is repeated.
- **Guess**. $\mathcal{A}$ outputs a guess of $b^{'}$ of b.

4.5 Security Definition of Anonymous Key Distribution Protocol

Anonymous key distribution protocol needs to satisfy the security properties of Leak-Freeness and Selective-Failure Blindness [7].

- *Leak-free.* If the honest authorities execute the protocol, then the malicious users cannot get any information that they cannot know during the protocol process.
- *Selective-failure.* If the corrupt authorities execute the protocol, they cannot learn any information of DU' GID and attributes and make the protocol selectively fail.

5 Concrete Construction

5.1 Anonymous Authentication Mechanism

DU completes user initialization and authentication through the anonymous authentication mechanism based on smart contract. The identity information of a DU consists of a GID u and a set of attributes. GID can be generated locally and verified by the User Initialization Contract, and is ultimately integrated into the GID aggregator ACU, which contains the GIDs of all DUs. To maintain anonymity, DU requests private keys using different pseudonyms Nym generated from the GID u.

For the attributes, DU needs to construct a Pedersen commitment for the attribute set during initialization, which will be integrated into the attribute aggregator ACT, containing the attribute set of all DUs. Since the attribute is public, it is important to guarantee the uniqueness of the attribute set commitment. We use the attribute, GID u and a random number to construct it. Therefore, the attribute set commitment of each DU is unique, preventing users from requesting private keys by forging attribute commitments. For example, DU U_1's GID is u_1 and he only has the attribute a. The U_1 selects a random number r_1 and construct the attribute set commitment $g^{au_1}h^{r_1}$, which is aggregated in the ACT. There is another DU U_2 whose GID is u_2. Since the U_2 does not have the attribute a, thus he try to create a fake commitment about a. However, ACT does not contains the value au_2, so that the above attack is impossible. If we just set the attribute set commitment as $g^a h^{r_1}$, then it is easy for U_2 to construct $g^a h^{r_2}$ to pass the verification.

5.1.1 User Initialization

DU performs user initialization after the Global Setup is complete. DU first chooses $r_u \in \mathbb{Z}_p$ randomly. For each attribute $a_{i,j} \in S$, $i \in [1, N]$, $j \in A_i$, DU randomly selects $r_{a_{i,j}} \in \mathbb{Z}_p$ and computes the GID $u = \mathcal{H}(r_u|a_{i,j})$ while retaining $r_{a_{i,j}\in S}$. Then, DU constructs the commitment $c_u = g^u h^{r_u}$ for u and the commitment $c_a = g^{\sum_{a_{i,j}\in S} a_{i,j}u} h^{\sum_{a_{i,j}\in S} r_{a_{i,j}}}$ for the attribute set. Next, DU randomly selects $u^{'}, r_u^{'}, r_a, r_a^{'} \in \mathbb{Z}_p$ and computes $c_u^{'} = g^{u^{'}} h^{r_u^{'}}$, $c_a^{'} = g^{r_a} h^{r_a^{'}}$. Let $\overline{c} = \mathcal{H}(c_u|c_u^{'}|c_a|c_a^{'})$, compute $z_1 = u^{'} - \overline{c}u$, $z_2 = r_u^{'} - \overline{c}r_u$, $z_3 = r_a - \overline{c}\sum_{a_{i,j}\in S} a_{i,j}u$, $z_4 = r_a^{'} - \overline{c}\sum_{a_{i,j}\in S} r_{a_{i,j}}$. Finally, DU calls the User Initialization Contract with $c_u, c_a, \overline{c}, z_1, z_2, z_3, z_4$ as inputs to complete the user initialization, as shown in Algorithm 1.

5.1.2 Pseudonym Generation and GID Verification

DU randomly selects $r^{'}, u^{'}, r_1, r_2, r_3, r_3^{'}, r_4, r_4^{'} \in \mathbb{Z}_p$, and computes the pseudonyms $Nym = h^u g^{r^{'}}$. Let $c_1 = g^{u^{'}} h^{r_1}$, $c_2 = h^{u^{'}} g^{r_2}$, $c_3 = g^{r_3^{'}} h^{r_4^{'}}$ while keeping $r^{'}$. The DU then obtains the latest GID aggregator ACU from the blockchain ledger, denoted as ACU_u, and computes $\overline{c} = \mathcal{H}(Nym|c_1|c_2|c_3)$, $w = \frac{ACU_u}{c_u}$, $c_w = w g^{r_3} h^{r_4}$. $z_0 = u^{'} - \overline{c}u$, $z_1 = r_1 - \overline{c}r_u$, $z_2 = r_2 - \overline{c}r^{'}$, $z_3 = r_3^{'} - \overline{c}r_3$, $z_4 = r_4^{'} - \overline{c}r_4$. Finally, DU calls the Pseudonym Verification Contract with $PP, Nym, ACU_u, z_0, z_1, z_2, z_3, z_4, c_1, c_2, c_3, c_w, \overline{c}$ as inputs to verify that he has a legitimate GID, as shown in Algorithm 2.

Algorithm 1 User Initialization Contract

Input: $c_u, c_a, \overline{c}, z_1, z_2, z_3, z_4$
Output: ACU,ACA or $\perp$
Get the latest GID aggregator ACU and attribute aggregator ACT from the blockchain ledger
if c_u, c_a have been used **then**
  Return $\perp$
else
  $c_u^{''} = g^{z_1} h^{z_2} \cdot c_u^{\overline{c}}$
  $c_a = g^{z_3} h^{z_4} \cdot c_{q_i}^{\overline{c}}$
  if $\mathcal{H}(c_u|c_u^{''}|c_a|c_a) == \overline{c}$ **then**
    $ACU = ACU \cdot c_u$
    $ACT = ACT \cdot c_a$
  else
    Return $\perp$
  end if
end if

Algorithm 2 Pseudonym Verification Contract

Input: $PP, Nym, ACU_u, z_0, z_1, z_2, z_3, z_4, c_1, c_2, c_3, c_w, \overline{c}$
Output: $True$ or $\perp$
if Nym has been used **then**
  Return $\perp$
else
  Get the latest GID aggregator ACU, denoted as ACU_n
  $c_w^{'} = c_w \cdot (ACU_n / ACU_u)$
  if $\overline{c} == \mathcal{H}(Nym|c_1|c_2|c_3), (ACU_n)^{\overline{c}} \cdot c_3 \cdot g^{z_0} h^{z_1} == (c_w^{'})^{\overline{c}} \cdot c_1 \cdot g^{z_3} h^{z_4}, Nym^{\overline{c}} \cdot h^{z_0} g^{z_2} == c_2$ **then**
    Return $True$
  else
    Return $\perp$
  end if
end if

5.1.3 Attribute Verification

DU randomly selects $\sigma, \sigma^{'}, r_1, r_2, r_3, r_3^{'}, r_4, r_4^{'} \in \mathbb{Z}_p$, and computes $c_{a_i} = g^{\sum_{a_{i,j} \in (S \cap A_i)} a_{i,j} u} h^{\sum_{a_{i,j} \in (S \cap A_i)} r_{a_{i,j}}}$, $c_1 = g^{\sigma^{'}} h^{r_1}$, $c_2 = g^{\sigma^{'}} h^{r_2}$, $c_3 = g^{r_3^{'}} h^{r_4^{'}}$, while keeping $\sigma, \sigma^{'}$. For each attribute $a_{i,j} \in (S \cap A_i)$, DU computes $c_{a_{i,j}} = g^{a_{i,j} u} h^{\sigma}$, $c_{a_i}^{'} = \prod_{a_{i,j} \in (S \cap A_i)} c_{a_{i,j}}$. Next, DU obtains the latest attribute aggregator ACT from the blockchain ledger, denoted as ACT_u, and computes $\overline{c} = \mathcal{H}(c_{a_i}^{'}|c_1|c_2|c_3)$, $w = \frac{ACT_u}{c_{a_i}}$, $c_w = w g^{r_3} h^{r_4}$, $z_0 = \sigma^{'} - \overline{c} \sum_{a_{i,j} \in (S \cap A_i)} a_{i,j} u$, $z_1 = r_1 - \overline{c} \sum_{a_{i,j} \in (S \cap A_i)} r_{a_{i,j}}$, $z_2 = r_2 - \overline{c}\sigma$, $z_3 = r_3^{'} - \overline{c} r_3$, $z_4 = r_4^{'} - \overline{c} r_4$. Finally, DU calls the Attribute Verification Contract with $PP, ACT_u, z_0, z_1, z_2, z_3, z_4, c_1, c_2, c_3, c_w, \overline{c}$ and $c_{a_{i,j} \in (S \cap A_i)}$ as the inputs to validate the legitimacy of the attributes as shown in the procedure in Algorithm 3.

Algorithm 3 Attribute Verification Contract

Input: $PP, ACT_u, z_0, z_1, z_2, z_3, z_4, c_1, c_2, c_3, c_w, \overline{c}, c_{a_{i,j} \in (S \cap A_i)}$
Output: $\{True, c_{a_{i,j} \in (S \cap A_i)}\}$ or $\perp$
if $c_{a_i}^{'}$ has been used **then**
  Return $\perp$
else
  Get the latest attribute aggregator ACT, denoted as ACT_n
  $c_w^{'} = c_w \cdot (ACT_n / ACT_u)$
  $c_{a_i} = \prod_{a_{i,j} \in (S \cap A_i)} c_{a_{i,j}}$
  if $\overline{c} == \mathcal{H}(c_{a_i}^{'}|c_1|c_2|c_3), (ACT_n)^{\overline{c}} \cdot c_3 \cdot g^{z_0} h^{z_1} == (c_w^{'})^{\overline{c}} \cdot c_1 \cdot g^{z_3} h^{z_4}, (c_{a_i}^{'})^{\overline{c}} \cdot g^{z_0} h^{z_2} == c_2$ **then**
    Return $True, c_{a_{i,j} \in (S \cap A_i)}$
  else
    Return $\perp$
  end if
end if

5.2 Algorithm Construction

This section describes the detailed construction of each algorithm in the scheme.

- **Global Setup**$(1^{\kappa}) \rightarrow (PP, OPK_{i \in [1,N]}, OMK_{i \in [1,N]})$.

  Suppose that the system contains N attribute authorities $\{\mathrm{AA}_1, \mathrm{AA}_2, ..., \mathrm{AA}_N\}$, each of them manages a set of attributes $A_i = \{a_{i,1}, ..., a_{i,q_i}\}$ for private keys distribution. CS first generates the public parameter PP via the security parameter κ as:

  $$PP = \{e, p, g, h, \mathbb{G}, \mathbb{G}_T, \mathcal{H}\}.$$

  For each $\mathrm{AA}_{i \in [1,N]}$, CS randomly selects $\Delta_i \in \mathbb{Z}_p$ and computes $OPK_i^1 = e(g,g)^{\Delta_i}$, $OPK_i^2 = h^{\Delta_i}$. Let the On-Chain key pair be $OPK_i = \{OPK_i^1, OPK_i^2\}, OMK_i = \Delta_i$. CS publishes $PP, \{OPK_1, ..., OPK_N\}$, and keeps $\{OMK_1, ..., OMK_N\}$ locally.
- **AA Setup**$(PP, OPK_i, A_i) \rightarrow (MK_i, PK_i)$.
  AA_i first randomly selects $\alpha_i, \beta_i, x_i, y_i, \gamma_i \in \mathbb{Z}_p$, and computes $I_i = (OPK_i^1)^{\alpha_i}$, $B_i = (OPK_i^2)^{\beta_i}$, $X_i = g^{x_i}$, $Y_i = g^{y_i}$, $\Gamma_i^1 = g^{\gamma_i}$, $\Gamma_i^2 = h^{\gamma_i}$. For the attribute $a_{i,j} \in A_i$, $i \in [1, N]$, $j \in [1, q_i]$, AA_i randomly selects $z_{i,j} \in \mathbb{Z}_p$ and sets $Z_{i,j} = g^{z_{i,j}}$, $V_{i,j} = h^{z_{i,j}} g^{\frac{1}{\gamma_i + a_{i,j}}}$. Finally, the public key of AA_i is:

  $$PK_i = \{I_i, B_i, X_i, Y_i, \Gamma_i^1, \Gamma_i^2, \forall a_{i,j} \in A_i, Z_{i,j}, V_{i,j}\}.$$

  The master key of AA_i is:

  $$MK_i = \{\alpha_i, \beta_i, x_i, y_i, \gamma_i, \forall a_{i,j} \in A_i, z_{i,j}\}.$$
- **Encrypt**$(PP, \mathcal{M}, \{\mathbb{A}_i, PK_i\}_{i \in \mathcal{F}}, TID_s) \rightarrow CT$.
  For each access structure $\mathbb{A}_i = (M_i, \rho_i)$, DO first constructs a random vector $v_i = (s_i, v_{i,2}, v_{i,3}, ..., v_{i,n}) \in \mathbb{Z}_p^n$ to share the secret s_i. Then DO computes $C_i^{'} = g^{s_i}$, $E_i = g^{y_i s_i}$. For each attribute in $\mathbb{A}_i$, DO computes $\lambda_{i,k} = M_{i,k} v_i^T$, $k \in [1, \ell_i]$, where $M_{i,k}$ is the kth row vector of the LSSS matrix M_i. Next, DO randomly selects $r_{i,k} \in \mathbb{Z}_p$ and sets $C_{i,k} = g^{x_i \lambda_{i,k}} (Z_{\rho_i(k)})^{-r_{i,k}}$, $D_{i,k} = g^{r_{i,k}}$. Subsequently, DO computes $C = \mathcal{M} e(g,g)^{\sum_{i \in \mathcal{F}} (\alpha_i \Delta_i s_i)}$, $H = \prod_{i \in \mathcal{F}} h^{\Delta_i \beta_i s_i}$. Finally, the ciphertext is:

  $$CT = \{C, H, \forall i \in \mathcal{F}, C_i^{'}, E_i, k \in [1, \ell_i], C_{i,k}, D_{i,k}\}.$$

Before DO uploads CT to the CS, he must generate a Data-Sharing Transaction T_s containing the message digest $\mathcal{H}$ of $\mathcal{M}$. After this transaction is valid, DO submits CT and the transaction ID TID_s to CS, who checks the validity of the transaction based on TID_s to decide whether to receive the ciphertext or not.

- **Key Generation**(PP, OMK_i, (PK_i, MK_i), (u, $S \cap A_i$)) $\rightarrow SK_u^i$.
In key generation, DU and AAs perform the anonymous key distribution protocol through the BN to obtain the final private key. The process is described in the next section. If the protocol is successfully performed, DU will obtain the private key:

$$SK_u^i = \{K_i = (g^{\alpha_i} g^{x_i t_{u,i}} h^{\beta_i u} h^{\frac{y_i}{\beta_i + u}})^{\Delta_i}, L_i = (g^{t_{u,i}})^{\Delta_i}, R_i = (h^{\frac{1}{\beta_i + u}})^{\Delta_i}, \forall a_{i,j} \in (S \cap A_i), F_{i,j} = (g^{z_{i,j} t_{u,i}})^{\Delta_i}\}.$$

- **Decrypt**(PP, CT, u, $\{SK_u^i\}_{i \in \mathcal{F}}$) $\rightarrow \mathcal{M}$.
Assuming that the attributes set of DU satisfies all the access structures of a ciphertext CT, then for each access structure $\{\{\mathbb{A}_i = (M_i, \rho_i)\}_{i \in \mathcal{F}}$, DU can find a set of constants $\{\omega_{i,k} \in \mathbb{Z}_p\}_{k \in \mathcal{I}_i}$ satisfying $\sum_{k \in \mathcal{I}_i} \lambda_{i,k} \omega_{i,k} = s_i$, where $\mathcal{I}_i \subset \{1, 2, ..., \ell_i\}$ is the authorized set, and defined as $\mathcal{I}_i = \{k : \rho_i(k) \in S\}$. Then, DU performs the decryption as the Equation (1).

$$\mathcal{M} = C \cdot e(H, g^u) \cdot \frac{\prod_{i \in \mathcal{F}} \prod_{k \in \mathcal{I}_i} (e(C_{i,k}, L_i) e(D_{i,k}, F_{\rho_i(k)}))^{\omega_{i,k}} \prod_{i \in \mathcal{F}} e(E_i, R_i)}{\prod_{i \in \mathcal{F}} e(C_i^{'}, K_i)} \tag{1}$$

5.3 Anonymous Key Distribution Protocol

The anonymous key distribution protocol includes two phases, "AA-DU" phase and "CS-DU" phase, shown as Fig.6. In the protocol, DU uses the pseudonym Nym to interact with the other two entities, while Nym is also used for the computation of the final private key.

5.3.1 *"AA-DU" Phase*

In this phase, DU will get the Semi-Finished keys from AAs.

DU first issues two Contract-Invoking Transactions T_{V1} and T_{V2} to invoke the Pseudonym Verification Contract and Attribute Verification Contract to complete the identity authentication. Once the above transactions are valid, DU then constructs a private key request R, containing the commitments of a set of attributes.

Upon receiving a Key-Request Transaction T_R containing R, AA first checks the validity of it. If T_R is not queryable, AA refuses to execute the protocol. Otherwise, it verifies DU's attributes through the NISMP protocol. The detailed process is described below.

- **DU side**: First, DU randomly chooses $\rho_1, \rho_1^{'}, \mu_1, \mu_1^{'}, \mu_2, \mu_2^{'}, r_{a_{i,j}}, \vartheta, \vartheta^{'} \in \mathbb{Z}_p$ to construct the commitments of ρ_1, μ_1, μ_2, that is, $P_1 = h^{\rho_1}, P_1^{'} = h^{\rho_1^{'}}$, $Q = g^{x_i \mu_1} g^{\mu_2}$, $Q^{'} = g^{x_i \mu_1^{'}} g^{\mu_2^{'}}$. For each attribute $a_{i,j} \in (S \cap A_i)$, DU computes $\Theta_{i,j}^{1^{'}} = g^{r_{a_{i,j}}} h^{\sigma^{'}}, \Theta_{i,j}^2 = V_{i,j}^{\vartheta}, \Theta_{i,j}^3 = Z_{i,j}{}^{\vartheta}, \Theta_{i,j}^4 = e(h, \Theta_{i,j}^3)^{r_{a_{i,j}}} e(g, \Theta_{i,j}^2)^{-r_{a_{i,j}}} e(g,g)^{\vartheta^{'}}$. DU sets the private key request R as:

$$R = \{P_1, P_1^{'}, Q, Q^{'}, \forall a_{i,j} \in (S \cap A_i), \Theta_{i,j}^{1^{'}}, \Theta_{i,j}^2, \Theta_{i,j}^3, \Theta_{i,j}^4\}.$$

Next, DU computes $\widetilde{c} = \mathcal{H}(R)$, $z_1 = \rho_1^{'} - \widetilde{c}\rho_1$, $z_2 = \mu_1^{'} - \widetilde{c}\mu_1$, $z_3 = \mu_2^{'} - \widetilde{c}\mu_2$, $z_4 = r_{a_{i,j}} - \widetilde{c} a_{i,j} u$, $z_5 = \vartheta^{'} - \widetilde{c}\vartheta$, $z_6 = \sigma^{'} - \widetilde{c}\sigma$. Let $z = \{z_1, z_2, z_3, z_4, z_5, z_6\}$, the Key-Request Transaction T_R is:

$$T_R = \{Nym, TID_{V1}, TID_{V2}, R, \widetilde{c}, z\}.$$

In the above transaction, TID_{V1} and TID_{V2} are the IDs of T_{V1} and T_{V2}. After the release of T_R, consensus nodes in the BN will check the outputs of T_{V1} and T_{V2}. If the outputs are $\perp$, T_R will be rejected from recording in the blockchain ledger. Otherwise, it will be written into a new block.

- **AA side**: Before responding to DU's request, AA_i first needs to check the validity of T_R. If the transaction does not exist, AA_i will refuse to execute the protocol. Otherwise, it extracts $c_{a_{i,j} \in (S \cap A_i)}$ from T_{v2}, and sets $\Theta_{i,j}^1 = c_{a_{i,j}}$. The verification is as the Equation (2).

$$P_1^{'} \stackrel{?}{=} h^{z_1} P_1^{\widetilde{c}}, Q^{'} \stackrel{?}{=} g^{z_3} X_i^{z_2} Q^{\widetilde{c}}, \Theta_{i,j}^{1^{'}} \stackrel{?}{=} (\Theta_{i,j}^1)^{\widetilde{c}} g^{z_4} h^{z_6},$$
$$\Theta_{i,j}^4 \stackrel{?}{=} (\frac{e(\Gamma_i^1, \Theta_{i,j}^2)}{e(\Gamma_i^2, \Theta_{i,j}^3)})^{\widetilde{c}} \cdot e(g, \Theta_{i,j}^2)^{-z_4} \cdot e(h, \Theta_{i,j}^3)^{z_4} \cdot e(g,g)^{z_5}. \tag{2}$$

If any of the above equations does not hold, AA_i exits the protocol, otherwise generates a Semi-Finished key for DU. Firstly, AA_i randomly selects $\rho_2, e_{u,i} \in \mathbb{Z}_p$, and computes $\eta = \rho_1 \rho_2 (\beta_i + u), d_1 = \frac{1}{\eta}, d_2 = \frac{y_i}{\eta}$, $P_2 = h^{\rho_2}, \widetilde{K}_i = g^{\alpha_i} Q^{e_{u,i}} Nym^{\beta_i} P_1^{\rho_2 d_2}, \widetilde{L}_i = g^{e_{u,i}}, \widetilde{R}_i = P_2^{d_1}$. For each attribute $a_{i,j} \in (S \cap A_i)$, computes $\widetilde{F}_{i,j} = (\Theta_{i,j}^3)^{e_{u,i}}$, where $\rho_1(\beta_i + u)$ can be calculated by a generic two-party protocol [37] or Paillier homomorphic encryption [38]. Then, AA_i randomly selects $\alpha_i^{'}, e_{u,i}^{'}, \beta_i^{'}, \rho_2^{'}, d_1^{'}, d_2^{'} \in \mathbb{Z}_p$, and computes $P_2^{'} = h^{\rho_2^{'}}, \widetilde{K}_i^{'} = g^{\alpha_i^{'}} Q^{e_{u,i}^{'}} Nym^{\beta_i^{'}} P_1^{\rho_2^{'} d_2^{'}}, \widetilde{L}_i^{'} = g^{e_{u,i}^{'}}, \widetilde{R}_i^{'} = P_2^{d_1^{'}}$. For the attribute $a_{i,j} \in (S \cap A_i)$, calculates $\widetilde{F}_{i,j}^{'} = (\Theta_{i,j}^3)^{e_{u,i}^{'}}$. Finally, the private key response $R^{'}$ is:

$$R^{'} = \{P_2, P_2^{'}, \widetilde{K}_i, \widetilde{K}_i^{'}, \widetilde{L}_i, \widetilde{L}_i^{'}, \widetilde{R}_i, \widetilde{R}_i^{'}, a_{i,j} \in (S \cap A_i), \widetilde{F}_{i,j}, \widetilde{F}_{i,j}^{'}\}.$$

AA_i computes $\widetilde{c} = \mathcal{H}(R^{'})$, $z_1 = \rho_2^{'} - \widetilde{c}\rho_2, z_2 = d_1^{'} - \widetilde{c} d_1, z_3 = e_{u,i}^{'} - \widetilde{c} e_{u,i}, z_4 = \alpha_i^{'} - \widetilde{c}\alpha_i, z_5 = \beta_i^{'} - \widetilde{c}\beta_i, z_6 = d_2^{'} - \widetilde{c}\rho_2 d_2$. Let $z = \{z_1, z_2, z_3, z_4, z_5, z_6\}$, the Key-Response Transaction $T_{R^{'}}$ is:

$$T_{R^{'}} = \{TID_R, R^{'}, \widetilde{c}, z\}.$$

In the above transaction, TID_R is the ID of the Key-Request Transaction T_R. After the release of $T_{R^{'}}$,

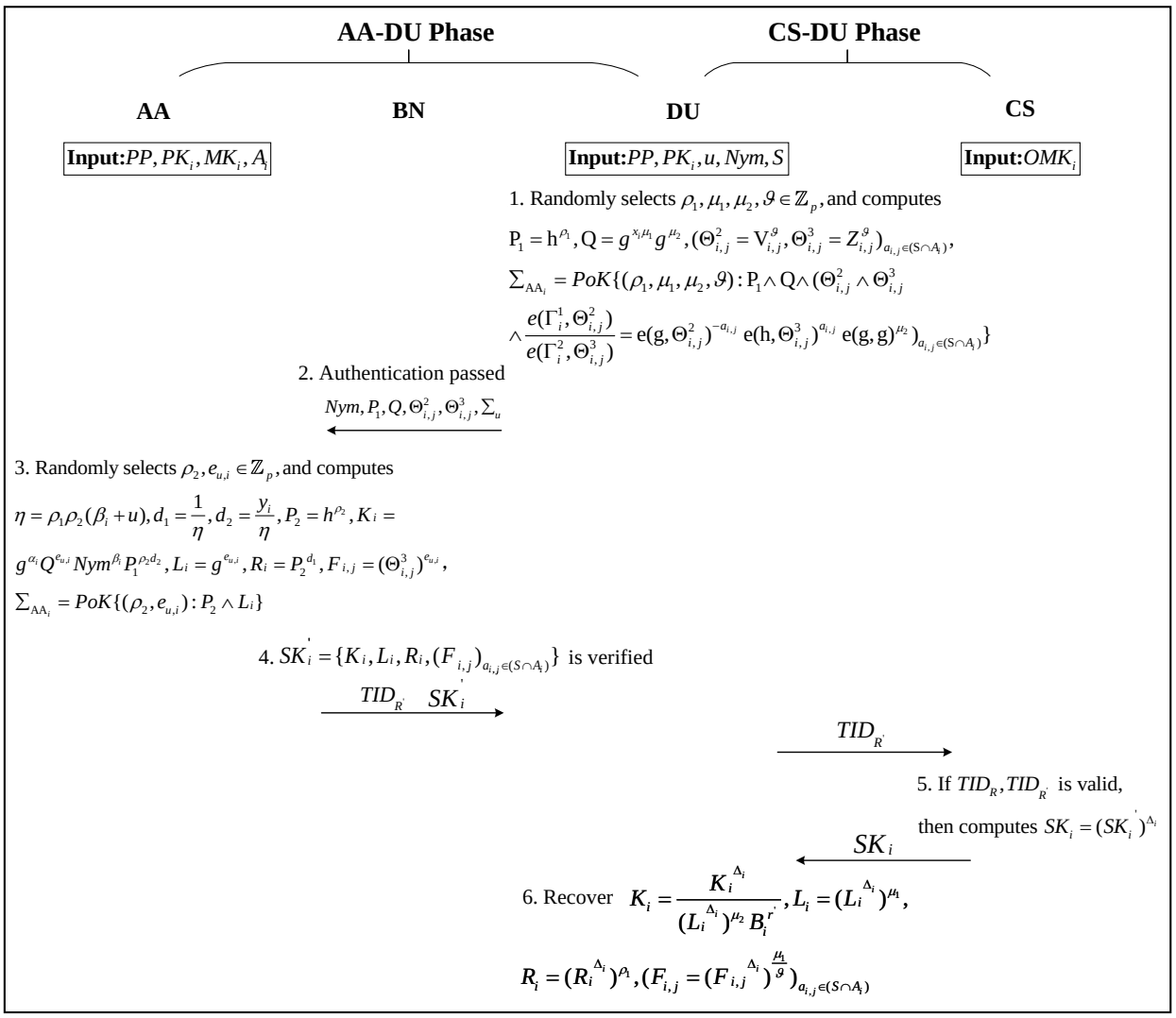


Fig. 6. Anonymous Key Distribution Protocol

consensus nodes in BN will extract $Nym, P_1, Q, \Theta^3_{i,j}$ from T_R, and perform the following verification.

$$P_2^{'} \stackrel{?}{=} (P_2)^{\widetilde{c}} \cdot h^{z_1}, \widetilde{K}_i^{'} \stackrel{?}{=} \widetilde{K}_i^{\widetilde{c}} \cdot g^{z_4} Q^{z_3} Nym^{z_5} P_1^{z_6},$$
$$\widetilde{L}_i^{'} \stackrel{?}{=} \widetilde{L}_i^{\widetilde{c}} \cdot g^{z_3}, \widetilde{R}_i^{'} \stackrel{?}{=} R_i^{\widetilde{c}} \cdot P_2^{z_2}, \widetilde{F}_{i,j}^{'} \stackrel{?}{=} \widetilde{F}_{i,j}^{\widetilde{c}} \cdot (\Theta^3_{i,j})^{z_3}. \quad (3)$$

If any of the above equations does not hold, consensus nodes will refuse to pack $T_{R'}$ in the new block. Otherwise, $T_{R'}$ will be recorded into the blockchain ledger, and DU will get the Semi-Finished key as:

$$\widetilde{SK_u^i}^{'} = \{\widetilde{K}_i, \widetilde{L}_i, \widetilde{R}_i, a_{i,j} \in (S \cap A_i), \widetilde{F}_{i,j}\}.$$

5.3.2 "CS-DU" Phase

In this phase, CS will impose the On-Chain status to $\widetilde{SK_u^i}^{'}$. Since the transactions T_R and $T_{R'}$ are one-to-one on the blockchain, thus DU only needs to submit $TID_{R'}$ to CS, who can easily query T_R from $T_{R'}$ and verify the validity of both transactions.

- **CS side**: CS extracts $\widetilde{SK_u^i}^{'}$ from $T_{R'}$, and computes $\widetilde{SK_u^i} = \{\widetilde{K}_i^{\Delta_i}, \widetilde{L}_i^{\Delta_i}, \widetilde{R}_i^{\Delta_i}, a_{i,j} \in (S \cap A_i), \widetilde{F}_{i,j}^{\Delta_i}\}$, which will be send to DU.
- **DU side**: After receiving the above information, DU calculates:

$$K_i = \frac{\widetilde{K}_i^{\Delta_i}}{(\widetilde{L}_i^{\Delta_i})^{\mu_2} (B_i)^{r'}}, L_i = (\widetilde{L}_i^{\Delta_i})^{\mu_1}, R_i = (\widetilde{R}_i^{\Delta_i})^{\rho_1},$$
$$a_{i,j} \in (S \cap A_i), F_{i,j} = (\widetilde{F}_{i,j}^{\Delta_i})^{\frac{\mu_1}{\vartheta}}. \quad (4)$$

Let $t_{u,i} = \mu_1 e_{u,i}$, and the final private key SK_u^i is:

$$SK_u^i = \{K_i = (g^{\alpha_i} g^{x_i t_{u,i}} h^{\beta_i u} h^{\frac{y_i}{\beta_i + u}})^{\Delta_i}, L_i = (g^{t_{u,i}})^{\Delta_i},$$
$$R_i = (h^{\frac{1}{\beta_i + u}})^{\Delta_i}, \forall a_{i,j} \in (S \cap A_i), F_{i,j} = (g^{z_{i,j} t_{u,i}})^{\Delta_i}\}.$$

5.4 Correctness Analysis

The correctness of decryption is proved by the following equations. First, we prove the numerator of the Equation (1), shown as Equation (5) and Equation (6)

$$\prod_{k\in\mathcal{I}_i}(e(C_{i,k},L_i)e(D_{i,k},F_{\rho_i(k)}))^{\omega_{i,k}}$$
$$=\prod_{k\in\mathcal{I}_i}(e(g^{x_i\lambda_{i,k}}g^{-z_{\rho_i(k)}r_{i,k}},g^{t_{u,i}\Delta_i})e(g^{r_{i,k}},g^{z_{\rho_i(k)}t_{u,i}\Delta_i}))^{\omega_{i,k}}$$
$$=\prod_{k\in\mathcal{I}_i}e(g,g)^{x_it_{u,i}\Delta_i\lambda_{i,k}\omega_{i,k}} \quad (5)$$
$$=e(g,g)^{x_it_{u,i}\Delta_is_i}.$$

$$e(E_i,R_i)=e(g^{y_is_i},h^{\frac{\Delta_i}{\beta_i+u}})=e(g,h)^{\frac{y_is_i\Delta_i}{\beta_i+u}}. \quad (6)$$

Next, we prove the denominator of the Equation (1), shown as Equation (7).

$$e(C_i^{'},K_i)=e(g^{s_i},g^{\alpha_i\Delta_i}g^{x_it_{u,i}\Delta_i}h^{\beta_iu\Delta_i}h^{\frac{y_i\Delta_i}{\beta_i+u}})=$$
$$e(g,g)^{s_i\alpha_i\Delta_i}e(g,g)^{s_ix_it_{u,i}\Delta_i}e(g,h)^{s_i\beta_iu\Delta_i}e(g,h)^{\frac{s_iy_i\Delta_i}{\beta_i+u}}. \quad (7)$$

Finally, we can recover the message $\mathcal{M}$ through the above equations, shown as Equation (8).

$$\mathcal{M}=\frac{C\cdot e(H,g^u)}{\prod_{i\in\mathcal{F}}e(g,g)^{s_i\alpha_i\Delta_i}e(g,h)^{s_i\beta_i\Delta_iu}}$$
$$=\frac{\mathcal{M}\prod_{i\in\mathcal{F}}e(g,g)^{\alpha_i\Delta_is_i}\cdot e(g,g)^{\Delta_i\beta_is_iu}}{\prod_{i\in\mathcal{F}}e(g,g)^{s_i\alpha_i\Delta_i}e(g,h)^{s_i\beta_i\Delta_iu}}. \quad (8)$$

6 Security Analysis

6.1 Security Analysis of Confidentiality

Theorem 1. *Assuming that the Decisional q-parallel BDHE assumption holds, then our scheme is IND-CPA secure, that is, there is no probabilistic polynomial-time adversary can selectively break our scheme with a set of challenge access structures.*

Proof. Suppose that a probabilistic polynomial-time adversary $\mathcal{A}$ is able to break our scheme by a non-negligible advantage ϵ, then there exists a challenger $\mathcal{B}$ that can break the Decisional q-parallel BDHE problem by the advantage $\frac{\epsilon}{2}$.

$\mathcal{B}$ first selects the group $\mathbb{G}$ and its generators g, the group $\mathbb{G}_\mathbb{T}$ and the bilinear map $e:\mathbb{G}\times\mathbb{G}=\mathbb{G}_\mathbb{T}$. Then $\mathcal{B}$ randomly selects $u\in\{0,1\}$. If $u=0$, $\mathcal{B}$ takes as input y,T where $T=e(g,g)^{a^{q+1}s}$. If $u=1$, $\mathcal{B}$ takes as input y,T where T is a random element in $\mathbb{G}_\mathbb{T}$. $\mathcal{B}$ guesses whether $T=e(g,g)^{a^{q+1}s}$ or a random element in $\mathbb{G}_\mathbb{T}$ by performing the following game with the adversary $\mathcal{A}$.

- ***Init***: $\mathcal{A}$ submits a group of corrupt authorities $\mathcal{C}=\{\mathrm{AA}_i\}_{i\in\mathcal{F}'}$ and a set of challenge access structures $\{\mathbb{A}_i=(M_i,\rho_i)\}_{i\in\mathcal{F}^*}$, where $\mathcal{F}^{'},\mathcal{F}^*\subseteq\{1,2,...,N\}$. There is at least one honest AA* in $\mathcal{F}^*$, and the corresponding challenge access structure is $\mathbb{A}^*=(M^*,\rho^*)$, which cannot be satisfied by $\mathcal{A}$'s attribute set. In $\mathbb{A}^*=(M^*,\rho^*)$, M^* is an $\ell^*\times n^*$ LSSS matrix, where $\ell^*,n^*\leqslant q$.
- ***Global Setup***: $\mathcal{B}$ runs the **Global Setup**. $\mathcal{B}$ first randomly selects $\tau\in\mathbb{Z}_p$, and computes $h=g^\tau$. The public parameter $PP=\{e,p,g,h,\mathbb{G},\mathbb{G}_T,\mathcal{H}\}$. For $\mathrm{AA}_{i\in[1,N]}$, $\mathcal{B}$ randomly chooses $\Delta_i\in\mathbb{Z}_p$, and sets On-Chain public key $OPK_i=\{OPK_i^1=e(g,g)^{\Delta_i},OPK_i^2=h^{\Delta_i}\}$. Finally, $\mathcal{B}$ sends $PP,\{OPK_i\}_{i\in[1,N]}$ to $\mathcal{A}$.
- ***AA Setup***: $\mathcal{B}$ runs **AA Setup**.
Case 1: $\mathrm{AA}_i\in\mathcal{C}$, AA_i is a corrupt AA.
$\mathcal{B}$ randomly selects $\alpha_i,\beta_i,x_i,y_i,\gamma_i\in\mathbb{Z}_p$, and computes $I_i=e(g,g)^{\Delta_i\alpha_i}$, $B_i=h^{\Delta_i\beta_i}$, $X_i=g^{x_i}$, $Y_i=g^{y_i}$, $\Gamma_i^1=g^{\gamma_i}$, $\Gamma_i^2=g^{\tau\gamma_i}$. For $a_{i,j}\in A_i$, $j\in[1,q_i]$, $\mathcal{B}$ selects a random $z_{i,j}\in\mathbb{Z}_p$, and sets $Z_{i,j}=g^{z_{i,j}}$, $V_{i,j}=g^{\tau z_{i,j}}g^{\frac{1}{\gamma_i+a_{i,j}}}$. Finally, the public key PK_i of AA_i is:
$$PK_i=\{I_i,B_i,X_i,Y_i,\Gamma_i^1,\Gamma_i^2,a_{i,j}\in A_i,Z_{i,j},V_{i,j}\}.$$
The master key MK_i is :
$$MK_i=\{\alpha_i,\beta_i,x_i,y_i,\gamma_i,a_{i,j}\in A_i,z_{i,j}\}.$$
In this case, $\mathcal{B}$ sends (MK_i,PK_i) to $\mathcal{A}$.
Case 2: $\mathrm{AA}_i\notin\mathcal{C}$, which means AA_i is not a corrupt AA.
1) $\mathrm{AA}_i\neq\mathrm{AA}^*$. $\mathcal{B}$ randomly selects $\alpha_i,\beta_i,x_i,y_i,\gamma_i\in\mathbb{Z}_p$, and computes $I_i=e(g,g)^{\Delta_i\alpha_i}$, $B_i=h^{\Delta_i\beta_i}$, $X_i=g^{x_i}$, $Y_i=g^{y_i}$, $\Gamma_i^1=g^{\gamma_i}$, $\Gamma_i^2=g^{\tau\gamma_i}$. For $a_{i,j}\in A_i$, $\mathcal{B}$ chooses a random $z_{i,j}\in\mathbb{Z}_p$ and sets $Z_{i,j}=g^{z_{i,j}}$, $V_{i,j}=g^{\tau z_{i,j}}g^{\frac{1}{(\gamma_i+a_{i,j})}}$. Finally, the public key PK_i of the AA_i is:
$$PK_i=\{I_i,B_i,X_i,Y_i,\Gamma_i^1,\Gamma_i^2,a_{i,j}\in A_i,Z_{i,j},V_{i,j}\}.$$
The master key MK_i is:
$$MK_i=\{\alpha_i,\beta_i,x_i,y_i,\gamma_i,a_{i,j}\in A_i,z_{i,j}\}.$$
In this situation, $\mathcal{B}$ just sends PK_i to $\mathcal{A}$.
2) $\mathrm{AA}_i=\mathrm{AA}^*$. The AA_i is the honest AA*. $\mathcal{B}$ randomly chooses $\alpha^{'},\beta^*,x^*,y^*,\gamma^*\in\mathbb{Z}_p$, and sets $\alpha^*=\alpha^{'}\Delta^*+a^{q+1}-\sum_{i\in\mathcal{F}^*}\alpha_i\Delta_i$. After that, $\mathcal{B}$ computes $I^*=e(g,g)^{\alpha^{'}}$, $B^*=h^{\Delta^*\beta^*}$, $X^*=g^a$, $Y^*=g^{y^*}$, $\Gamma_*^1=g^{\gamma^*}$, $\Gamma_*^2=g^{\tau\gamma^*}$.
Let ∂ be the set of index $k\in[1,\ell^*]$, which makes $\rho^*(k)=a_{*,j}$ hold. For the attribute $a_{*,j}\in\partial$, $\mathcal{B}$ first selects a random $\widetilde{z}_{*,j}\in\mathbb{Z}_p$, and sets:
$$z_{*,j}=\widetilde{z}_{*,j}+\sum_{\iota=1}^{n^*}\frac{a^\iota M_{k,\iota}^*}{b_k}.$$
Then, $\mathcal{B}$ computes:
$$Z_{*,j}=g^{\widetilde{z}_{*,j}}\prod_{k\in\partial}\prod_{\iota=1}^{n^*}g^{\frac{a^\iota M^*_{k,\iota}}{b_k}},$$
$$V_{*,j}=g^{\tau\widetilde{z}_{*,j}}\prod_{k\in\partial}\prod_{\iota=1}^{n^*}g^{\tau\frac{a^\iota M^*_{k,\iota}}{b_k}}g^{\frac{1}{\gamma^*+a_{*,j}}}.$$
If $\partial=\emptyset$, $\mathcal{B}$ randomly selects $z_{*,j}\in\mathbb{Z}_p$, and computes $Z_{*,j}=g^{z_{*,j}}$, $V_{*,j}=g^{\tau z_{*,j}}g^{\frac{1}{\gamma_*+a_{*,j}}}$. Finally, the public key of the AA* is:
$$PK^*=\{I^*,B^*,X^*,Y^*,\Gamma_*^1,\Gamma_*^2,a_{*,j}\in A^*,Z_{*,j},V_{*,j}\}.$$
The master key MK^* is:
$$MK^*=\{\alpha^*,\beta^*,a,y^*,\gamma^*,a_{*,j}\in A^*,z_{*,j}\}.$$

In this situation, $\mathcal{B}$ just sends PK^* to $\mathcal{A}$.

- ***Phase 1***: $\mathcal{A}$ submits GID u and the attribute set S to the AA_i to request the private key, and the S cannot satisfy the access structure $\mathbb{A}^*$ of the AA^*.
Case 1: $\mathrm{AA}_i \in \mathcal{C}$, which means AA_i is a corrupt AA. The AA_i is able to collude with $\mathcal{A}$ to issue the fake private key. However, the fake private key lacks the On-Chain status since it cannot be verified by the BN. Therefore, $\mathcal{B}$ chooses random $t_{u,i}, \Delta_i^* \in \mathbb{Z}_p$, and computes:

$$K_i = (g^{\alpha_i} g^{x_i t_{u,i}} h^{\beta_i u} h^{\frac{y_i}{\beta_i+u}})^{\Delta_i^*}, L_i = g^{t_{u,i}\Delta_i^*}, R_i = h^{\frac{\Delta_i^*}{\beta_i+u}}$$
$$\forall a_{i,j} \in (S \cap A_i), F_{i,j} = g^{z_{i,j} t_{u,i} \Delta_i^*}.$$

Finally, $\mathcal{B}$ sends $SK_u^i = \{K_i, L_i, R_i, \forall a_{i,j} \in (S \cap A_i), F_{i,j}\}$ to $\mathcal{A}$.
Case 2: $\mathrm{AA}_i \notin \mathcal{C}$, which means AA_i is not a corrupt AA.
1) $\mathrm{AA}_i \neq \mathrm{AA}^*$. $\mathcal{B}$ randomly selects $t_{u,i} \in \mathbb{Z}_p$ and computes:

$$K_i = (g^{\alpha_i} g^{x_i t_{u,i}} h^{\beta_i u} h^{\frac{y_i}{\beta_i+u}})^{\Delta_i}, L_i = g^{t_{u,i}\Delta_i}, R_i = h^{\frac{\Delta_i}{\beta_i+u}}$$
$$\forall a_{i,j} \in (S \cap A_i), F_{i,j} = g^{z_{i,j} t_{u,i} \Delta_i}.$$

Finally, $\mathcal{B}$ sends $SK_u^i = \{K_i, L_i, R_i, \forall a_{i,j} \in (S \cap A_i), F_{i,j}\}$ to $\mathcal{A}$.
2) $\mathrm{AA}_i = \mathrm{AA}^*$. $\mathcal{B}$ randomly selects $r \in \mathbb{Z}_p$, and construct a vector $\omega = (\omega_1, \omega_2, ..., \omega_{n^*})$ where $\omega_1 = -1$. For the index k of $\rho(k) \in S$, there exists $\omega \cdot M_k^* = 0$. $\mathcal{B}$ sets t as:

$$t = r + \omega_1 a^q + \omega_2 a^{q-1} + ... + \omega_{n^*} a^{q-n^*+1}.$$

Then, $\mathcal{B}$ computes:

$$L^* = g^{t\Delta^*} = g^{r\Delta^*} \prod_{\iota=1}^{n^*} g^{\omega_\iota a^{q-\iota+1}\Delta^*},$$
$$R^* = h^{\frac{\Delta^*}{\beta^*+u}}, K^* = g^{\alpha^*} (g^{at} h^{u\beta^*} h^{\frac{y^*}{\beta^*+u}})^{\Delta^*}.$$

If there is no k to make $\rho^*(k) = a_{*,j}$ hold, then compute $F_{*,j} = (L^*)^{z_{*,j}} = {Z_{*,j}}^{t\Delta_i}$. Otherwise computes:

$$F_{*,j} = (L^*)^{z_{*,j}}$$
$$= (L^*)^{\widetilde{z}_{*,j}} \prod_{\iota=1}^{n^*} (g^{\frac{ra^\iota}{b_k}} \prod_{j=1, j\neq\iota}^{n^*} (g^{\frac{a^{q-j+\iota+1}}{b_k}})^{\omega_\iota})^{M_{k,\iota}^* \Delta^*}.$$

Finally, $\mathcal{B}$ sends $SK_u^* = \{K^*, L^*, R^*, \forall a_{*,j} \in (S \cap A^*), F_{*,j}\}$ to $\mathcal{A}$.

- ***Challenge***: $\mathcal{B}$ performs **Encryption**. $\mathcal{A}$ gives two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$ to $\mathcal{B}$. $\mathcal{B}$ flips a coin $b \in \{0, 1\}$, and creates the ciphertext as follows.
1) $\mathrm{AA}_i \notin \mathcal{C}$ and $\mathrm{AA}_i \neq \mathrm{AA}^*$. $\mathcal{B}$ selects a random $s_i \in \mathbb{Z}_p$ and computes $C_i^{'} = g_{}^{s} g^{-s_i}$, $E_i = g^{y_i(s-s_i)}$. Then, $\mathcal{B}$ randomly chooses $y_{i,2}^{'}, ..., y_{i,n_i}^{'} \in \mathbb{Z}_p$ and constructs the vector $v_i = (s - s_i, y_{i,2}^{'}, ..., y_{i,n_i}^{'})$ to share the secret $-s_i$. Next, $\mathcal{B}$ randomly selects $r_{i,1}, ..., r_{i,\ell_i} \in \mathbb{Z}_p$, and computes:

$$C_{i,k} = g^{x_i(s-s_i)M_{k,1}^i} \prod_{\iota=2}^{n_i} g^{y_{i,\iota} M_{k,\iota}^i} Z_{\rho_i(k)}^{-r_{i,k}},$$
$$D_{i,k} = g^{r_{i,k}}.$$

2) $\mathrm{AA}_i = \mathrm{AA}^*$. $\mathcal{B}$ first computes $C_*^{'} = g^s$, $E_* = g^{sy^*}$, $H_* = g^{s\beta^*}$. Then, $\mathcal{B}$ randomly selects $y_2^{'}, ..., y_{n^*}^{'} \in \mathbb{Z}_p$ and constructs the vector $v^* = (s, sa + y_2^{'}, ..., sa^{n^*-1} + y_{n^*}^{'})$ to share the secret s. Next, $\mathcal{B}$ randomly chooses $r_1, ..., r_{\ell^*} \in \mathbb{Z}_p$. For $j \in [1, n^*]$, let R_j be the set of $j \neq k$ to make $\rho^*(\iota) = \rho^*(k)$ hold. $\mathcal{B}$ computes:

$$C^* = \mathcal{M}_b T \cdot e(g^s, g^{\alpha^{'}\Delta^*}) \prod_{i\in\mathcal{F}^*, i\neq *} e(g,g)^{-\alpha_i s_i \Delta_i},$$
$$H^* = g^{s\Delta^*\beta^*} \prod_{i\in\mathcal{F}^*, i\neq *} h^{\Delta_i\beta_i(s-s_i)},$$
$$C_{*,k} = Z_{\rho^*(k)}^{-r_k} (\prod_{\iota=2}^{n^*} (g^a)^{y_\iota^{'} M_{k,\iota}^*}) (g^{b_k s})^{-z_{\rho^*(k)}} (\prod_{j\in R_j} \prod_{\iota=1}^{n^*} (g^{\frac{a^\iota s b_k}{b_j}})^{M_{j,\iota}^*}),$$
$$D_{*,k} = g^{-r_k} g^{-sb_k}.$$

Finally, the ciphertext is:

$$CT^* = \{C^*, H^*, \forall i \in \mathcal{F}^*, (C_i^{'}, E_i, \forall k \in [1, \ell_i], (C_{i,k}, D_{i,k}))\}.$$

- ***Phase 2***: **Phase 1** is repeated.
- ***Guess***: $\mathcal{A}$ outputs a guess $b^{'} \in \{0, 1\}$ of b. If $b^{'} = b$, $\mathcal{B}$ outputs $u^{'} = 0$, and guesses that $T = e(g,g)^{a^{q+1}s}$, which means CT^* is a legal ciphertext of $\mathcal{M}_b$. If $b^{'} \neq b$, $\mathcal{B}$ outputs $u^{'} = 1$, and guesses that T is a random group element in $\mathbb{G}_\mathbb{T}$.

If the $\mathcal{A}$ is able to break our scheme with a non-negligible advantage ϵ, then when $u = 1$, $\mathcal{M}_b$ is perfectly hidden from $\mathcal{A}$, and he cannot obtain any valid information about b. At this point, $\Pr[b^{'} \neq b | u = 1] = \frac{1}{2}$, and the $\mathcal{B}$ guesses $u^{'} = 1$ with the maximum advantage $\Pr[u^{'} = u | u = 1] = \frac{1}{2}$. When $u = 0$, the $\mathcal{A}$ can get the correct ciphertext, and the advantage is ϵ. At this point, $\Pr[b^{'} = b | u = 0] = \frac{1}{2} + \epsilon$, and the $\mathcal{B}$ guesses $u^{'} = 0$ with the minimum advantage $\Pr[u^{'} = u | u = 0] = \frac{1}{2} + \epsilon$. In summary, the advantage of the $\mathcal{B}$ is:

$$Adv_\mathcal{B} = |\frac{1}{2}(\Pr[u^{'} = u | u = 0] + \Pr[u^{'} = u | u = 1]) - \frac{1}{2}| = \frac{\epsilon}{2}. \quad (9)$$

Since ϵ is non-negligible, thus $\frac{\epsilon}{2}$ is likewise non-negligible, thus the $\mathcal{B}$ is able to solve Decisional q-parallel BDHE problem with a non-negligible advantage. However, it is against the security assumption. Therefore, there is no probabilistic polynomial-time adversary $\mathcal{A}$ that can defeat our scheme with a non-negligible advantage, that is our scheme is IND-CPA secure. ■

6.2 Security Analysis of Anonymous Key Distribution Protocol

Theorem 2. *The anonymous key distribution protocol is Leak-Free and Selective-Failure Blind.*

The proof for the above two properties is similar to literatures [7], [39], we omit them here. ■

6.3 Security Analysis of Internal Fraud Attack

In multi-authority scenario, a corrupt AA_i and a malicious insider can collude to perform the internal fraud attack: AA_i uses the master key MK_i to generate an illegal private

key for DU to achieve unauthorized access, as implicitly described in ***Phase 1*** of the "Security Analysis of Confidentiality". In the following, we will explain in detail how our scheme resists internal fraud attack.

Suppose that AA_1 is corrupt and AA_2 is honest. In our scheme, a malicious insider $\mathcal{A}$ cannot collude with AA_1 to get the correct private key. There are two cases as follows.

Case 1: AA_1 runs **AA Setup** correctly.

- ***System Setup***: Assume that the **Global Setup** is executed correctly, and the public parameters is:

$$PP = \{e, p, g, h, \mathbb{G}, \mathbb{G}_T, \mathcal{H}\}.$$

AA_1's On-Chain public key is $OPK_1^1 = e(g,g)^{\Delta_1}$, $OPK_1^2 = h^{\Delta_1}$, the attribute set is $A_1 = \{a_{1,1}, ..., a_{1,n}\}$, and the public and master keys are:

$$PK_1 = \{I_1, B_1, X_1, Y_1, \Gamma_1^1, \Gamma_1^2, a_{1,j} \in A_1, Z_{1,j}, V_{1,j}\},$$
$$MK_1 = \{\alpha_1, \beta_1, x_1, y_1, \gamma_1, a_{1,j} \in A_1, z_{1,j}\}.$$

AA_2's On-Chain public key is $OPK_2^1 = e(g,g)^{\Delta_2}$, $OPK_2^2 = h^{\Delta_2}$, the attribute set is $A_2 = \{a_{2,1}, ..., a_{2,n}\}$, and the public and master keys are:

$$PK_2 = \{I_2, B_2, X_2, Y_2, \Gamma_2^1, \Gamma_2^2, a_{2,j} \in A_2, Z_{2,j}, V_{2,j}\},$$
$$MK_2 = \{\alpha_2, \beta_2, x_2, y_2, \gamma_2, a_{2,j} \in A_2, z_{2,j}\}.$$

- ***Encryption***: The DO encrypts the message $\mathcal{M}$ using the access structure of AA_1 and AA_2, and the ciphertext is:

$$CT = \{C = \mathcal{M}e(g,g)^{\sum_{i\in\{1,2\}}(\alpha_i \Delta_i s_i)},$$
$$H, i \in \{1,2\}, C_i^{'}, E_i, k \in [1, \ell_i], C_{i,k}, D_{i,k}\}.$$

- ***Key Generation***: There exists a malicious internal user $\mathcal{A}$ who has he legitimate GID u and the attributes from AA_2 in the ciphertext, but does not have the attributes from AA_1. Firstly, $\mathcal{A}$ runs the anonymous key distribution protocol with AA_2 to get the Semi-Finished key $\widetilde{SK_u^2}^{'}$. After CS imposes the On-Chain state, the final private key from AA_2 is:

$$SK_u^2 = \{K_2 = g^{\alpha_2\Delta_2} g^{x_2 t_{u,2}\Delta_2} h^{\beta_2 u \Delta_2} h^{\frac{y_2\Delta_2}{\beta_2+u}}, L_2 = g^{t_{u,2}\Delta_2},$$
$$R_2 = h^{\frac{\Delta_2}{\beta_2+u}}, \forall a_{2,j} \in (S \cap A_2), F_{2,j} = g^{z_{2,j} t_{u,2}\Delta_2}\}.$$

Since $\mathcal{A}$ does not have the attributes from AA_1, thus he tries to collude with AA_1 to get the illegal private key:

$$(SK_u^1)^{'} = \{K_1 = g^{\alpha_1} g^{x_1 t_{u,1}} h^{\beta_1 u} h^{\frac{y_1}{\beta_1+u}}, L_1 = g^{t_{u,1}},$$
$$R_1 = h^{\frac{1}{\beta_1+u}}, \forall a_{1,j} \in (S \cap A_1), F_{1,j} = g^{z_{1,j} t_{u,1}}\}.$$

However, $\mathcal{A}$ can only get the Semi-Finished key from AA_1. Finally, the private key is $SK_u = \{(SK_u^1)^{'}, SK_u^2\}$.

- ***Decryption***: $\mathcal{A}$ performs **Decryption**, and get the following result:

$$Result = \frac{\mathcal{M}e(g,g)^{\alpha_1\Delta_1 s_1} e(g,h)^{u\Delta_1\beta_1 s_1}}{e(g,g)^{s_1\alpha_1} e(g,h)^{s_1\beta_1 u}} \quad (10)$$

From Equation (10), it can be seen that $\mathcal{M}$ can be recovered if $\mathcal{A}$ obtains $OMK_1 = \Delta_1$. However, under the DL assumption, $\mathcal{A}$ cannot get $OMK_1 = \Delta_1$ from the known parameters in polynomial time, as this is computationally hard for $\mathcal{A}$.

Suppose that $\mathcal{A}$ tries to construct a Key-Request Transaction T_R containing the fake attribute commitments, but it is difficult to be verified by the blockchain, and T_R fails to be packed into the blockchain. Suppose that $\mathcal{A}$ constructs a Key-Request Transaction T_R containing the correct attribute commitments, and provides the fake attribute commitments to AA_1 trying to obtain the relevant private keys. However, AA_1 cannot generate a Key-Response Transaction $T_{R'}$ containing the fake attribute commitments to pass the verification of blockchain.

Therefore, neither the malicious insider nor the corrupt the AA can use the fake attribute commitments to bypass the blockchain, so that $\mathcal{A}$ cannot obtain a private key with On-Chain status to enable unauthorized access.

Case 2: AA_1 runs **AA Setup** incorrectly.

AA_1 attempts to publish the public key that does not have the On-Chain status, that is:

$$PK_1 = \{I_1 = e(g,g)^{\alpha_1}, B_1 = h^{\beta_1}, X_1, Y_1,$$
$$\Gamma_1^1, \Gamma_1^2, a_{1,j} \in A_1, Z_{1,j}, V_{1,j}\}.$$

However, using PK_1 in encryption means that the data is at risk of leakage, which is useless for DO. ■

6.4 Security Analysis of User collusion

Suppose there are AA_1, AA_2, and u_1, u_2, u_3 in the system, u_1 and u_2 have the same attribute set and get their private keys $SK_{u_1}^1$ and $SK_{u_2}^1$ from the AA_1 respectively, u_3 obtain the private keys $SK_{u_3}^2$ from AA_2.

$$SK_{u_1}^1 = \{K_1 = g^{\alpha_1\Delta_1} g^{x_1 t_{u_1,1}\Delta_1} h^{\beta_1 u_1 \Delta_1} h^{\frac{y_1\Delta_1}{\beta_1+u_1}}, L_1 = g^{t_{u_1,1}\Delta_1},$$
$$R_1 = h^{\frac{\Delta_1}{\beta_1+u_1}}, \forall a_{1,j} \in (S \cap A_1), F_{1,j} = g^{z_{1,j} t_{u_1,1}\Delta_1}\},$$
$$SK_{u_2}^1 = \{K_1 = g^{\alpha_1\Delta_1} g^{x_1 t_{u_2,1}\Delta_1} h^{\beta_1 u_2 \Delta_1} h^{\frac{y_1\Delta_1}{\beta_1+u_2}}, L_1 = g^{t_{u_2,1}\Delta_1},$$
$$R_1 = h^{\frac{\Delta_1}{\beta_1+u_2}}, \forall a_{1,j} \in (S \cap A_1), F_{1,j} = g^{z_{1,j} t_{u_2,1}\Delta_1}\},$$
$$SK_{u_3}^2 = \{K_2 = g^{\alpha_2\Delta_2} g^{x_2 t_{u_3,2}\Delta_2} h^{\beta_2 u_3 \Delta_2} h^{\frac{y_2\Delta_2}{\beta_2+u_3}}, L_2 = g^{t_{u_3,2}\Delta_2},$$
$$R_2 = h^{\frac{\Delta_2}{\beta_2+u_3}}, \forall a_{2,j} \in (S \cap A_2), F_{2,j} = g^{z_{2,j} t_{u_3,1}\Delta_2}\}.$$

User collusion consists of two cases.

Case 1: Collusion to generate illegal private keys.

Since there is a linear relationship between the different components in the private keys, users with the same attribute set can collude to generate private keys corresponding to the same attribute set for another user.

In our scheme, user's private keys do not have any linear relationship, if DU u_1 and DU u_2 try to generate illegal private key $SK_{\widetilde{u}}^1$ for $\widetilde{u}$, where

$$SK_{\widetilde{u}}^1 = \{K_1 = g^{\alpha_1\Delta_1} g^{x_1 t_{\bar{u},1}\Delta_1} h^{\beta_1 \widetilde{u} \Delta_1} h^{\frac{y_1\Delta_1}{\beta_1+\widetilde{u}}}, L_1 = g^{t_{\bar{u},1}\Delta_1},$$
$$R_1 = h^{\frac{\Delta_1}{\beta_1+\widetilde{u}}}, \forall a_{1,j} \in (S \cap A_1), F_{1,j} = g^{z_{1,j} t_{\bar{u},1}\Delta_1}\}.$$

TABLE 2
Comparison of Privacy-Preserving MACP-ABE Schemes

Scheme	Access Structure	Order Group	GID Hiding	Attribute Hiding	Decentralized Authentication	Internal Fraud Resistance
Ye [17]	LSSS	Prime	✓	×	×	×
Ma [20]	LSSS	Prime	✓	×	×	×
Han [7]	LSSS	Prime	✓	✓	×	×
Hu [27]	LSSS	Prime	✓	✓	×	×
Zhang [19]	LSSS	Prime	✓	×	×	×
Zhang [28]	LSSS	Prime	✓	✓	×	×
Liu [31]	LSSS	Composite	✓	×	×	×
Li [32]	LSSS	Prime	✓	×	×	×
Duan [33]	LSSS	Prime	✓	×	×	×
Ours	LSSS	Prime	✓	✓	✓	✓

they must reconstruct $h^{\frac{\Delta_1}{\beta_1+u}}$ or $h^{\frac{y_1\Delta_1}{\beta_1+u}}$. However, it is impossible to achieve this only by $u_1, u_2, SK^1_{u_1}, SK^1_{u_2}$ under the DL assumption.

Case 2: Collusion to recover the plaintext.

Users who cannot satisfy the access policy on their own will attempt to recover plaintext by combining their decryption results. Assume a message is encrypted by $(\mathbb{A}_1 \wedge \mathbb{A}_2)$, and the ciphertext is:

$$CT = \{C = \mathcal{M}e(g,g)^{(\alpha_1\Delta_1 s_1+\alpha_2\Delta_2 s_2)}, H = h^{(\beta_1\Delta_1 s_1+\beta_2\Delta_2 s_2)}, \\ i \in \{1,2\}, C_i^{'}, E_i, k \in [1,\ell_i], C_{i,k}, D_{i,k}\}.$$

Let u_1 satisfy $\mathbb{A}_1$ and u_3 satisfy $\mathbb{A}_2$. u_1 and u_3 execute **Decrypt** algorithm, and combine their calculation results into $Result = \mathcal{M} \cdot e(g,h)^{(u_1\beta_2\Delta_2 s_2+u_3\beta_1\Delta_1 s_1)}$. To recover $\mathcal{M}$, they must get $e(g,h)^{(u_1\beta_2\Delta_2 s_2+u_3\beta_1\Delta_1 s_1)}$. However, this is difficult to achieve with only u_1, u_3, $h^{\beta_1\Delta_1}$, $h^{\beta_2\Delta_2}$, and CT under the DL assumption. ■

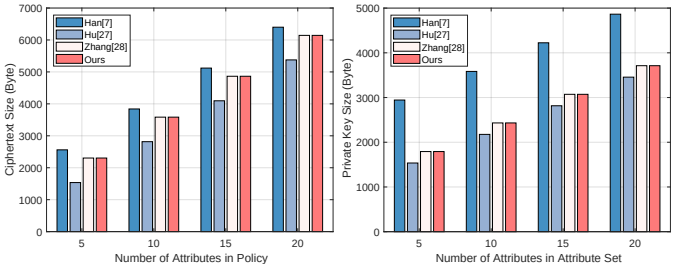
7 Performance Analysis

In this section, we provide a theoretical and experimental analysis of our scheme with the related privacy-preserving MACP-ABE schemes. Table 2 shows the features comparison among Ye [17], Ma [20],Han [7],Zhang [19],Zhang [28],Liu [31],Li [32] and Duan [33] in terms of Access Structure, Order Group, GID Hiding, Attribute Hiding, Decentralized and Internal Fraud Resistance.

From the table, we can easily observe that most schemes adopt the LSSS access structure and Prime order groups to construct the scheme, which is expressive to access policy and better performance compared to Composite order groups. In terms of privacy, although all schemes achieve GID hiding, only schemes Han [7], Hu [27], Zhang [28] and ours further achieve attribute hiding. In key generation of other schemes, the AA can explicitly learn the user's attributes, leading to privacy leakage. Furthermore, only our scheme achieves decentralized authentication and internal fraud resistance.

7.1 Theoretical and Experimental Analysis

Since schemes Han [7], Hu [27], Zhang [28] and ours simultaneously achieve protection for both GID and attributes, thus we conduct a further theoretical and experimental analysis of these schemes. The experiments are run on a desktop computer (Intel(R) Core(TM) i5-4590 CPU@ 3.30GHz with 4GB RAM) based on Ubuntu 18.04 LTS and the Java paring-based cryptography library 2.0.0 [40]. We adopt the Type



(a) Ciphertext Size (b) Private Key Size

Fig. 7. Comparison of Storage

A pairing constructed on the curve $y^2 = x^3 + x$ over the field $\mathbb{F}_q$ for some prime $q = 3 \mod 4$. For each policy, the attributes come from 3 AAs, that is $\mathcal{F} = 3$, and we repeat the experiment 20 times and adopt the average values.

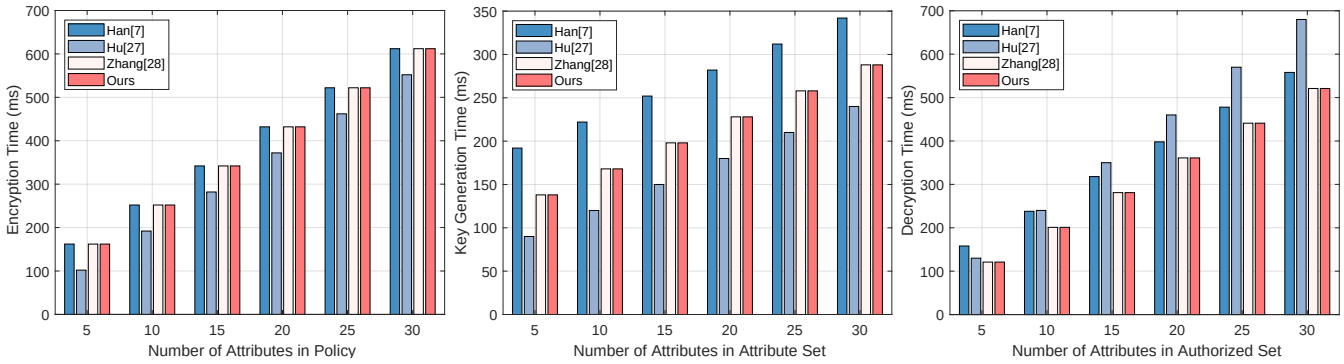
Table 3 presents a comparison of storage and algorithm computation across the the four schemes. In terms of storage, as shown in Fig.7, We can see that our scheme and Zhang [28] perform similarly, both outperforming Han [7]. This is because Han [7] contains redundant components in ciphertext and private key, resulting in higher storage cost for both. Hu [27] performs best in both characteristics, mainly because it only sets a final secret and does not set sub-secrets for each access structure. This setting also enables the scheme to have a lower computational overhead in encryption and key generation than the other schemes, shown as Fig.8(a) and Fig.8(b). However, this results in the intermediate results of the access structure during decryption being unable to be added together and offset against the final secret, which significantly increases the decryption cost, as shown in Fig.8(c). In contrast, our scheme performs best in decryption, similar to Zhang [28]. From the comparison of storage and algorithm computation above, we observe that our scheme and Zhang [28] perform very close. However, Zhang [28] relies too much on the centralized ACS and the scheme has poor reliability, while our scheme effectively reduces the centralization risk and improves the reliability through the decentralized anonymous authentication mechanism.

Furthermore, since the above schemes, except for Hu [27], utilize the NISMP protocol for anonymous private key distribution, we theoretically analysis the performance of anonymous key distribution among the three schemes in Table 4, as shown in Fig.9. It is evident that the communica-

TABLE 3
Comparison of Storage and Algorithm Computation

Scheme	Ciphertext	Private Key	Encryption	Key Generation	Decryption
Han [7]	$\lvert\mathbb{G}_\mathbb{T}\rvert + (3\lvert\mathcal{F}\rvert + 2\sum_{i\in\mathcal{F}}\ell_i)\lvert\mathbb{G}\rvert$	$(6\lvert\mathcal{F}\rvert + \lvert S\rvert)\lvert\mathbb{G}\rvert$	$\lvert\mathcal{F}\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + (3\lvert\mathcal{F}\rvert + 3\sum_{i\in\mathcal{F}}\ell_i)T_{E_\mathbb{G}}$	$(9\lvert\mathcal{F}\rvert + \lvert S\rvert)T_{E_\mathbb{G}}$	$(4\lvert\mathcal{F}\rvert + \sum_{i\in\mathcal{F}}2\ell_i)T_P + (\lvert\mathcal{F}\rvert + \sum_{i\in\mathcal{F}}\ell_i)T_{E_\mathbb{G}}$
Hu [27]	$\lvert\mathbb{G}_\mathbb{T}\rvert + (1 + 2\sum_{i\in\mathcal{F}}\ell_i)\lvert\mathbb{G}\rvert$	$(2\lvert\mathcal{F}\rvert + 1 + \lvert S\rvert)\lvert\mathbb{G}\rvert$	$T_{E_{\mathbb{G}_\mathbb{T}}} + (1 + 3\sum_{i\in\mathcal{F}}\ell_i)T_{E_\mathbb{G}}$	$(3\lvert\mathcal{F}\rvert + 1 + \lvert S\rvert)T_{E_\mathbb{G}}$	$(\lvert\mathcal{F}\rvert + 1 + \sum_{i\in\mathcal{F}}2\ell_i)T_P + \sum_{i\in\mathcal{F}}2\ell_i T_{E_\mathbb{G}}$
Zhang [28]	$\lvert\mathbb{G}_\mathbb{T}\rvert + (2\lvert\mathcal{F}\rvert + 1 + 2\sum_{i\in\mathcal{F}}\ell_i)\lvert\mathbb{G}\rvert$	$(3\lvert\mathcal{F}\rvert + \lvert S\rvert)\lvert\mathbb{G}\rvert$	$\lvert\mathcal{F}\rvert T E_{\mathbb{G}_\mathbb{T}} + (3\lvert\mathcal{F}\rvert + 3\sum_{i\in\mathcal{F}}\ell_i)T_{E_\mathbb{G}}$	$(6\lvert\mathcal{F}\rvert + \lvert S\rvert)T_{E_\mathbb{G}}$	$(2\lvert\mathcal{F}\rvert + 1 + \sum_{i\in\mathcal{F}}2\ell_i)T_P + (\sum_{i\in\mathcal{F}}\ell_i + 1)T_{E_\mathbb{G}}$
Ours	$\lvert\mathbb{G}_\mathbb{T}\rvert + (2\lvert\mathcal{F}\rvert + 1 + 2\sum_{i\in\mathcal{F}}\ell_i)\lvert\mathbb{G}\rvert$	$(3\lvert\mathcal{F}\rvert + \lvert S\rvert)\lvert\mathbb{G}\rvert$	$\lvert\mathcal{F}\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + (3\lvert\mathcal{F}\rvert + 3\sum_{i\in\mathcal{F}}\ell_i)T_{E_\mathbb{G}}$	$(6\lvert\mathcal{F}\rvert + \lvert S\rvert)T_{E_\mathbb{G}}$	$(2\lvert\mathcal{F}\rvert + 1 + \sum_{i\in\mathcal{F}}2\ell_i)T_P + (\sum_{i\in\mathcal{F}}\ell_i + 1)T_{E_\mathbb{G}}$

$|\mathbb{G}|$, $|\mathbb{G}_\mathbb{T}|$: The length of element in $\mathbb{G}$ and $\mathbb{G}_\mathbb{T}$, $\mathcal{F}$: The number of the AA corresponding to the access structure or the private key, ℓ_i: The number of attribute in the ith access structure, $|S|$: The size of user attribute set, $T_{E_\mathbb{G}}$: The exponential operation in group $\mathbb{G}$, $T_{E_{\mathbb{G}_\mathbb{T}}}$: The exponential operation in group $\mathbb{G}_\mathbb{T}$, T_P: The bilinear pairing operation.



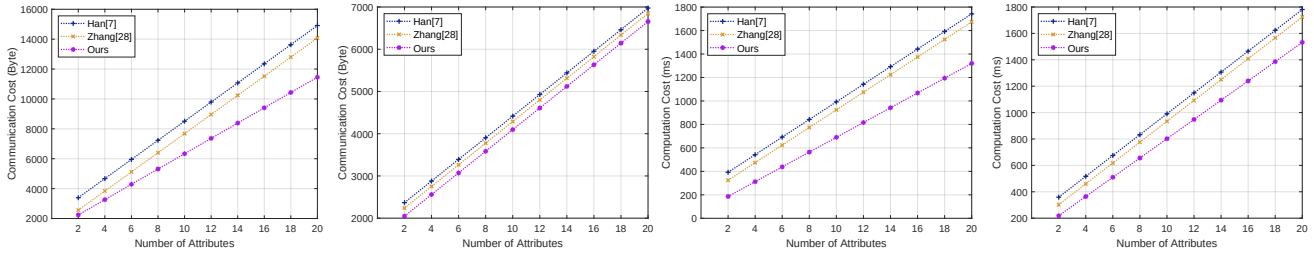
(a) Encryption (b) Key Generation (c) Decryption

Fig. 8. Comparison of Algorithms

TABLE 4
Comparison of Anonymous Key Distribution

Schemes	Communication Cost		Computation Cost	
	DU-AA	AA-DU	DU	AA
Han [7]	$9\lvert\mathbb{Z}_p\rvert + (12 + 4\lvert S\cap A_i\rvert)\lvert\mathbb{G}\rvert + \lvert S\cap A_i\rvert\lvert\mathbb{G}_\mathbb{T}\rvert$	$5\lvert\mathbb{Z}_p\rvert + (12 + 2\lvert S\cap A_i\rvert)\lvert\mathbb{G}\rvert$	$(37 + 7\lvert S\cap A_i\rvert)T_{E_\mathbb{G}} + 3\lvert S\cap A_i\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + (4 + 3\lvert S\cap A_i\rvert)T_P$	$(31 + 5\lvert S\cap A_i\rvert)T_{E_\mathbb{G}} + 4\lvert S\cap A_i\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + (3 + 5\lvert S\cap A_i\rvert)T_P$
Zhang [28]	$8\lvert\mathbb{Z}_p\rvert + (6 + 4\lvert S\cap A_i\rvert)\lvert\mathbb{G}\rvert + \lvert S\cap A_i\rvert\lvert\mathbb{G}_\mathbb{T}\rvert$	$7\lvert\mathbb{Z}_p\rvert + (10 + 2\lvert S\cap A_i\rvert)\lvert\mathbb{G}\rvert$	$(29 + 7\lvert S\cap A_i\rvert)T_{E_\mathbb{G}} + 3\lvert S\cap A_i\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + 3\lvert S\cap A_i\rvert T_P$	$(24 + 5\lvert S\cap A_i\rvert)T_{E_\mathbb{G}} + 4\lvert S\cap A_i\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + 5\lvert S\cap A_i\rvert T_P$
Ours	$9\lvert\mathbb{Z}_p\rvert + (5 + 3\lvert S\cap A_i\rvert)\lvert\mathbb{G}\rvert + \lvert S\cap A_i\rvert\lvert\mathbb{G}_\mathbb{T}\rvert$	$8\lvert\mathbb{Z}_p\rvert + (8 + 2\lvert S\cap A_i\rvert)\lvert\mathbb{G}\rvert$	$(10 + 5\lvert S\cap A_i\rvert)T_{E_\mathbb{G}} + 3\lvert S\cap A_i\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + 3\lvert S\cap A_i\rvert T_P$	$(12 + 4\lvert S\cap A_i\rvert)T_{E_\mathbb{G}} + 4\lvert S\cap A_i\rvert T_{E_{\mathbb{G}_\mathbb{T}}} + 5\lvert S\cap A_i\rvert T_P$

$|\mathbb{G}|$, $|\mathbb{G}_\mathbb{T}|$, $|\mathbb{Z}_p|$: The length of element in $\mathbb{G}$, $\mathbb{G}_\mathbb{T}$ and $\mathbb{Z}_p$, $|S\cap A_i|$: The number of attribute in $S\cap A_i$, $T_{E_\mathbb{G}}$: The exponential operation in group $\mathbb{G}$, $T_{E_{\mathbb{G}_\mathbb{T}}}$: The exponential operation in group $\mathbb{G}_\mathbb{T}$, T_P: The bilinear pairing operation.



(a) Communication Cost(DU-AA) (b) Communication Cost(AA-DU) (c) Computation Cost (DU) (d) Computation Cost (AA)

Fig. 9. Comparison of Anonymous Key Distribution

tion and computation costs of all schemes increase linearly with the number of attributes. However, our scheme demonstrates the best performance in both the key request phase (Fig. 9(a) and Fig. 9(c)) and the key response phase (Fig. 9(b) and Fig. 9(d)). This superior performance is primarily due to the use of blockchain technology in our scheme to reliably execute the authentication process. As a result, the DU and the AA only need to generate the request and response parameters and validate the transactions.

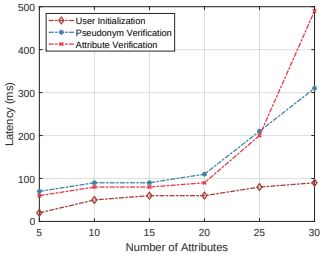


Fig. 10. Smart Contract in Anonymous Authentication

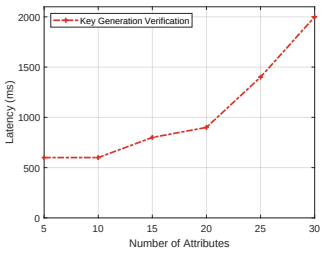


Fig. 11. Verification of Key Generation

7.2 Blockchain Performance

Our scheme employs the blockchain to verify the identity of the DU and the process of key generation. Therefore, we evaluate the blockchain performance, especially the smart contract. We chose Hyperleder Fabric v2.2 as the blockchain platform with Raft consensus mechanism [41] and tested blockchain performance using Hyperledger Caliper v0.5.0 [1], a performance benchmark tool capable of measuring different blockchain platforms.

We evaluate the latency of User Initialization, Pseudonym Verification, Attribute Verification and Key Generation Verification (We achieved this by a smart contract) from contract invocation to contract confirmation (writing to the blockchain ledger), shown in Fig.10 and Fig.11. As can be seen from the figures, the latency of User Initialization is at a relatively steady level, while the latency of other contracts shows a more evident linear increase with the growth of the number of attributes. When the number of attributes is less, the latency grows slowly, and when the number of attributes is higher, the latency grows faster. Fortunately, it is tolerable for a stand-alone test environment, and the performance can be better when a blockchain with higher throughput and optimized consensus mechanism is employed.

8 Conclusion

This paper introduces a privacy-improving MACP-ABE with internal fraud attack resistance based on blockchain. First, we achieved decentralized anonymous authentication mechanism based on smart contract, significantly enhancing system reliability. Second, we designed a blockchain-based anonymous key distribution protocol to resist internal fraud attack. The private keys of users are generated by three distinct parties, ensuring that neither the AA nor CS can access any information about the private keys. We thoroughly analyzed the security of the scheme and compared it in detail with others, and the results showed that our scheme has better performance and provides stronger security. We will focus on the traitor tracking issues in multi-authority scenarios in the future.

Acknowledgments

This work was supported by Major Science and Technology Projects in Yunnan Province (202202AD080013).

1. Hyperledger caliper.https://www.hyperledger.org/projects/caliper.

References

[1] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *International Workshop on Public Key Cryptography*. Springer, 2011, pp. 53–70.
[2] M. Chase, "Multi-authority attribute based encryption," in *Theory of cryptography conference*. Springer, 2007, pp. 515–534.
[3] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *Information Sciences*, vol. 180, no. 13, pp. 2618–2632, 2010.
[4] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2011, pp. 568–588.
[5] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2001, pp. 93–118.
[6] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual international cryptology conference*. Springer, 1991, pp. 129–140.
[7] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. A. Au, "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," *IEEE transactions on information forensics and security*, vol. 10, no. 3, pp. 665–678, 2014.
[8] C. Ruan, C. Hu, R. Zhao, Z. Liu, H. Huang, and J. Yu, "A policy-hiding attribute-based access control scheme in decentralized trust management," *IEEE Internet of Things Journal*, 2023.
[9] S. Zhang, Y. Wang, E. Luo, Q. Liu, K. Gu, and G. Wang, "A traceable and revocable decentralized multi-authority privacy protection scheme for social metaverse," *Journal of Systems Architecture*, vol. 140, p. 102899, 2023.
[10] B. Gong, C. Guo, C. Guo, C. Guo, Y. Sun, M. Waqas, and S. Chen, "Slim: A secure and lightweight multi-authority attribute-based signcryption scheme for iot," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1299–1312, 2024.
[11] Z. Deng, J. Chen, S. Yao, and P. Li, "Ppadma-abe: a novel privacy-preserving and auditable attribute-based encryption under dynamic multi-authority setting," *International Journal of Applied Cryptography*, vol. 4, no. 3-4, pp. 176–194, 2024.
[12] J. Du, G. Dong, J. Ning, Z. Xu, and R. Yang, "Blockchain-based and multi-authority hierarchical access control data sharing scheme," *Computers and Electrical Engineering*, vol. 119, p. 109547, 2024.
[13] P. Duan, Z. Ma, H. Gao, T. Tian, and Y. Zhang, "Multi-authority attribute-based encryption scheme with access delegation for cross blockchain data sharing," *IEEE Transactions on Information Forensics and Security*, 2025.
[14] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security*, 2009, pp. 121–130.
[15] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *International Journal of Information Security*, vol. 14, no. 6, pp. 487–497, 2015.
[16] M. Lyu, X. Li, and H. Li, "Efficient, verifiable and privacy preserving decentralized attribute-based encryption for mobile cloud computing," in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2017, pp. 195–204.
[17] Y. Ye, L. Zhang, W. You, and Y. Mu, "Secure decentralized access control policy for data sharing in smart grid," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6.
[18] L. Zhang, C. Zhao, Q. Wu, Y. Mu, and F. Rezaeibagha, "A traceable and revocable multi-authority access control scheme with privacy preserving for mhealth," *Journal of Systems Architecture*, vol. 130, p. 102654, 2022.
[19] L. Zhang, Y. Ye, and Y. Mu, "Multiauthority access control with anonymous authentication for personal health record," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 156–167, 2020.
[20] R. Ma and L. Zhang, "Spmac: Secure and privacy-preserving multi-authority access control for fog-enabled iot cloud storage," *Journal of Systems Architecture*, vol. 142, p. 102951, 2023.
[21] Y. Su, X. Zhang, J. Qin, and J. Ma, "Efficient and flexible multi-authority attribute-based authentication for iot devices," *IEEE Internet of Things Journal*, 2023.

[22] M. Ashouri-Talouki, N. Kahani, and M. Barati, "Privacy-preserving attribute-based access control with non-monotonic access structure," in *2023 7th Cyber Security in Networking Conference (CSNet)*. IEEE, 2023, pp. 32–38.
[23] X. Liang, Y. Liu, and J. Ning, "An access control scheme with privacy-preserving authentication and flexible revocation for smart healthcare," *IEEE Journal of Biomedical and Health Informatics*, 2024.
[24] H. Fan, Q. Li, J. Xiong, R. Li, W. Chen, and H. Huang, "Decentralized access control for privacy-preserving cloud-based personal health record with verifiable policy update," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 887–16 901, 2024.
[25] S. Roy, J. Agrawal, A. Kumar, and U. P. Rao, "Mh-abe: multi-authority and hierarchical attribute based encryption scheme for secure electronic health record sharing," *Cluster Computing*, vol. 27, no. 5, pp. 6013–6038, 2024.
[26] M. Wang, Z. Zhang, and C. Chen, "Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1237–1245, 2016.
[27] S. Hu, J. Li, and Y. Zhang, "Improving security and privacy-preserving in multi-authorities ciphertext-policy attribute-based encryption," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 12, no. 10, pp. 5100–5119, 2018.
[28] L. Zhang, X. Gao, L. Kang, P. Liang, and Y. Mu, "Distributed ciphertext-policy attribute-based encryption with enhanced collusion resilience and privacy preservation," *IEEE Systems Journal*, 2021.
[29] L. Wang, H. Zhong, J. Cui, J. Zhang, L. Wei, I. Bolodurina, and D. He, "Privacy-preserving and secure distributed data sharing scheme for vanets," *IEEE Transactions on Mobile Computing*, 2024.
[30] H. Nasiraee and M. Ashouri-Talouki, "Privacy-preserving distributed data access control for cloudiot," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2476–2487, 2022.
[31] C. Liu, F. Xiang, and Z. Sun, "Multiauthority attribute-based access control for supply chain information sharing in blockchain," *Security and Communication Networks*, vol. 2022, pp. 1–18, 2022.
[32] B. Li, J. Yang, Y. Wang, X. Huang, J. Ren, and L. Wang, "A blockchain-based privacy-preserving data sharing scheme with security-enhanced access control," in *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2023, pp. 825–830.
[33] D. Pengfei, M. Zhaofeng, Z. Yuqing, W. Jingyu, and L. Shoushan, "Blockchain-enabled privacy protection and access control scheme towards sensitive digital assets management," *China Communications*, 2024.
[34] J. Zhang and A. Datta, "Blockchain-enabled data governance for privacy-preserved sharing of confidential data," *PeerJ Computer Science*, vol. 10, p. e2581, 2024.
[35] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Annual International Cryptology Conference*. Springer, 1997, pp. 410–424.
[36] J. Camenisch, R. Chaabouni *et al.*, "Efficient protocols for set membership and range proofs," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2008, pp. 234–252.
[37] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham, "Randomizable proofs and delegatable anonymous credentials," in *Annual International Cryptology Conference*. Springer, 2009, pp. 108–125.
[38] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.
[39] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.
[40] A. De Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *2011 IEEE symposium on computers and communications (ISCC)*. IEEE, 2011, pp. 850–855.
[41] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 172–181, 2019.

Zhaoqian Zhang received the B.S. degree from Yanshan University in 2016 and the Ph.D. degree from Beijing University of Technology in 2023. He is currently an engineer of the Science and Technology Research Institute, Three Gorges Corporation, Beijing, China. His research interests include public key cryptography, blockchain and trusted computing.

Bei Gong received his B.S. degree from Shandong University in 2005 and his Ph.D. degree from Beijing University of Technology in 2012. He has contributed to in six national invention patents and one monograph textbook. In the past five years, he has published more than 30 papers in the first-class SCI/EI journal and other well-known international journals and top international conferences in relevant research fields. His research interests include trusted computing, Internet of Things security, the mobile Internet of Things, and mobile edge computing. He has presided over 8 national projects, such as projects of the National Natural Science Foundation, and 6 provincial and ministerial projects, such as the general science and technology program of the Beijing Municipal Education Commission.

Xin Fan received the B.S. degree in communication engineer from Sichuan University, China, in 2000, the M.E. degree in signal and information processing from Peking University, China, in 2004, and the Dr.-Ing. degree in computer engineer from Berlin Humboldt University, Berlin, Germany, in 2013. He was with the Shanghai Jade Technologies, China, as senior ASIC engineer from 2004 to 2008, with the IHP, Germany, as researcher from 2008 to 2015, with the imec-NL, Netherlands, as senior researcher from 2015 to 2017, with the Institute of Integrated Digital Systems, RWTH Aachen University, Germany, as research leader from 2017 to 2022, and with the Beijing Bitmain Ltd. Co. as IC design expert in 2023. Dr. Fan join the China Three Gorges Corporation, China, in 2024, and at present as laboratory and group manager he is responsible for innovative IC design for industrial and cryptography applications. Dr. Fan has published more than 50 peer-reviewed journal articles and conference papers. He was a recipient of the IEEE International Conference on Computer Design (ICCD) Best Paper Award in 2009 and the IEEE International Symposium on Quality Electronic Design (ISQED) Best Paper Award in 2017.

Yilin Yuan received the Ph.D. degree from Beijing University of Technology in 2022. She is engaged in science research and education work with the College of Information Engineering, Beijing Institute of Graphic Communication, Beijing, China. Her research interests include cloud computing, cloud security and blockchain.

Yang Fan is currently pursuing for the Master's degree in the College of Information Engineering, Beijing Institute of Graphic Communication, Beijing, China. He is currently engaging in cloud computing research work. His research interests include cloud computing, blockchain and network security.

Weizhi Meng is a Full Professor in the School of Computing and Communications, Lancaster University, United Kingdom, and an adjunct faculty in the Department of Applied Mathematics and Computer Science, Technical University of Denmark, Denmark. He obtained his Ph.D. degree in Computer Science from the City University of Hong Kong. He was a recipient of the Hong Kong Institution of Engineers (HKIE) Outstanding Paper Award for Young Engineers/Researchers in both 2014 and 2017. He also received the IEEE ComSoc Best Young Researcher Award for Europe, Middle East, & Africa Region (EMEA) in 2020. His primary research interests are blockchain technology, cyber security and artificial intelligence in security including intrusion detection, blockchain applications, smartphone security, biometric authentication, and IoT security. He is a senior member of IEEE.security.

Qiang Zhu received the B.S. degree and the Master's degree from Zhengzhou University in 2002 and 2005, received the Ph.D. degree from Peking University in 2009. He is currently the Director of the Science and Technology Research Institute, Three Gorges Corporation, Beijing, China. His research interests include Industrial security, Space-Air-Ground Integrated Network.