TrustAccess: A Trustworthy Secure Ciphertext-Policy and Attribute Hiding Access Control Scheme Based on Blockchain

Sheng Gao, Guirong Piao, Jianming Zhu, Xindi Ma, and Jianfeng Ma

***Abstract*—Ciphertext-policy attribute-based encryption (CP-ABE) is widely used in fine-grained access control to achieve the secure data sharing. However, most of the existing CP-ABE access control schemes involve intermediary entities, which might suffer from a high trust-building cost, single point of failure and so on. Due to the decentralization and transparency of blockchain, some blockchain-based access control schemes are proposed to address these problems, but bring new challenges, such as the privacy leakage of access policy or attribute. In this paper, we propose a new trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain, named TrustAccess, to achieve trustworthy access while guaranteeing the privacy of policy and attribute. For one thing, to make the existing hidden policy CP-ABE more efficient and scalable for blockchain, we propose an optimized hidden policy CP-ABE, named OHP-CP-ABE, to ensure policy privacy while satisfying the large universe access requirement. For another thing, we use the multiplicative homomorphic ElGamal cryptosystem to ensure the attribute privacy during authorization validation. Finally, we theoretically prove the security of our TrustAccess from the aspects of blockchain operations and OHP-CP-ABE. Comprehensive comparisons and extensive experiments are conducted to demonstrate the advantages of our TrustAccess.**

***Index Terms*—Blockchain, access control, CP-ABE, security, privacy.**

I. Introduction

In RECENT years, attribute-based encryption (ABE) [1] is considered to be an advanced cryptographic tool for providing fine-grained access control over encrypted data, which has been widely used in various scenarios [2]–[4]. Generally, existing schemes based on ABE can be roughly divided into two variants, namely key-policy ABE (KP-ABE) [5] and ciphertext-policy ABE (CP-ABE) [6]. In KP-ABE, secret keys are generated based on access policies that determine which ciphertexts can be decrypted and ciphertexts are associated with attribute sets. On the contrary, in CP-ABE, secret keys are generated over attribute sets and ciphertexts are associated with access policies. Overall, CP-ABE allows data owners to define their own access policies, which is considered to be more suitable for access control than KP-ABE in decentralized autonomous systems.

In most existing CP-ABE schemes [2], [3], [7], to provide a unified and convenient data access control while reducing the cost in terms of configuration, deployment and management, a data owner usually delegates the encrypted data under an access policy to an intermediary entity such as a cloud server, and a data user can get a secret key from an authority using a set of attributes to decrypt the encrypted data from the cloud server. Apparently, the trustability of data access control is strongly dependent on the intermediary entities, such as the cloud server and the authority [8]–[10]. However, they are assumed to be semi-trusted or trusted respectively, which is not practical in reality and might suffer from the issues such as a high trust-building cost, single point of failure and so on.

To alleviate these problems, a promising blockchain technology with advantages of decentralization, trust machine, disintermediation, transparency and immutability has been proposed to enhance the trustability of data access control [4], [11]–[15]. More specifically, to achieve a trustworthy access control, most of the existing work concentrates on deploying access control policies on a blockchain [11]–[13], [15], and some others proposed to use a blockchain to store encrypted secret keys [4] or a set of attributes [14]. However, due to the transparency of blockchain, it would pose a great threat to the disclosure of privacy by directly sending an access policy or a user's attribute set to the blockchain [7], [16], [17]. Particularly in CP-ABE schemes, one can infer some private attributes from the access policies that are embedded into ciphertexts [18]–[20]. Therefore, to achieve a trustworthy secure access control scheme based on blockchain, it is necessary to design a CP-ABE scheme with hidden access policies and attributes.

Policy Hiding: Most of the existing work focused on CP-ABE schemes with hidden access policies takes two forms, namely fully hidden [21] and partially hidden [22], where the former refers to that none of the attributes can be revealed with the access

Manuscript received September 2, 2019; revised November 29, 2019; accepted December 31, 2019. Date of publication January 16, 2020; date of current version June 18, 2020. This work was supported in part by the Nature Key Research and Development Program of China under Grant 2017YFB1400700, in part by the National Natural Science Foundation of China under Grants 61602537, U1509214, and 61902290, in part by the Beijing Social Science Foundation under Grant 16XCC023, in part by the Science and Technology Development Center of Ministry of Education under Grant 2019J02022, in part by the Central University of Finance and Economics program for the Youth Talent Support Plan under Grant QYP1808, in part by the China Postdoctoral Science Foundation Funded Project under Grant 2018M640962, in part by the National Natural Science Foundation of Shaanxi Province under Grant 2019JM-109 and in part by the Fundamental Research Funds for the Central Universities under Grant JB191508. The review of this article was coordinated by Prof. H. Li. *(Corresponding author: Sheng Gao.)*

S. Gao, G. Piao, and J. Zhu are with the School of Information, Central University of Finance and Economics, Beijing 100081, China (e-mail: sgao@cufe.edu.cn; piao_gr@163.com; zjm@cufe.edu.cn).

X. Ma and J. Ma are with the School of Cyber Engineering, Xidian University, Xi'an 710071, China (e-mail: xdma@xidian.edu.cn; jfma@mail.xidian.edu.cn).

Digital Object Identifier 10.1109/TVT.2020.2967099

policies and the latter means hiding sensitive attribute values revealed in the access policies and keeping only the attribute names. Note that fully hidden access policies can guarantee a better privacy, while partially hidden access policies can provide a better efficiency [7]. Generally, a CP-ABE scheme with fully hidden access policies can be achieved by Inner-product Predicate Encryption (IPE) [21]. Lai *et al.* [17] proposed a ciphertext-policy hiding CP-ABE scheme, and prove that it is fully secure. However, it is appropriate for a small universe. That is, the key size increases linearly as the number of attributes increases. As a result, it is hard to define the system boundary of key sizes, which is unsuitable for blockchain systems with high scalability requirements.

Attribute Hiding: In existing schemes [7], [16], [23], [24], a data user should provide an intermediary entity such as an authority with a set of attributes for getting the secret key. However, the presence of corrupted intermediary entity might cause a great threat to the attribute privacy. Note that the attribute privacy refers to the privacy of an attribute set possessed by the data user. Michalevsky and Joye [16] proposed the first attribute-hiding multi-authority ABE scheme based on decentralized IPE. However, it can only hide the overall set of attributes while still revealing the attributes that an authority is controlling. Thus, it is very urgent to achieve a full attribute privacy protection while getting the secret key in CP-ABE based on blockchain.

In this paper, we propose a trustworthy secure ciphertext-policy and attribute hiding access control scheme, named TrustAccess, to address the issues mentioned above such as the high trust-building cost, single point of failure, trustability of data access control, small universe and privacy leakage. To the best of our knowledge, this is the first practical blockchain-based access control scheme with guaranteeing the privacy of both access policies and attributes.

Overall, our main contributions are summarized as follows.

- We propose a new trustworthy secure access control system based on blockchain, named TrustAccess, which can address the high trust-building cost, security and privacy issues caused by the involvement of an intermediary entity. In our TrustAccess, by sending the ciphertext policies to a blockchain via transactions, we can achieve a distributed and trustworthy access control management.
- We propose a ciphertext-policy and attribute hiding scheme to protect the privacy during the access control in our TrustAccess. For one thing, we optimize the scheme of Lai *et al.* [17] by designing a large universe and hidden policy CP-ABE for blockchain, named OHP-CP-ABE, which can encrypt the data and address privacy leakage of access policies on account of the transparency of blockchain. For another thing, we use the multiplicative homomorphic ElGamal cryptosystem to provide a full attribute privacy protection while getting the secret key in CP-ABE based on blockchain.
- We theoretically prove the security of TrustAccess in terms of blockchain operations and OHP-CP-ABE. Comparisons of related schemes and comprehensive experiments are conducted to demonstrate the advantages of our TrustAccess.

The remainder of this paper is organized as follows. We introduce the related work in Section II, and the preliminaries in Section III. We present an overview of our proposed TrustAccess in Section IV. More design details of our TrustAccess are presented in Section V. We analyze the security of our TrustAccess in Section VI and make comparisons and performance evaluation of our TrustAccess in Section VII. Finally, we make a conclusion in Section VIII.

II. Related Work

Access control is an important method for privacy protection [25]–[28]. In this section, we would introduce some related work in the filed of blockchain-based access control.

A. Transaction-Based Access Control Schemes

Recent studies have taken advantages of blockchain to establish transaction-based access control systems for secure data sharing. Li *et al.* [12] proposed a secure system for IoT by using blockchain and certificateless cryptography. Access policies determined by the data owner are represented by an access control list on the blockchain, and any data user that intends to access data has to check if its identity belongs to the access control list by a transaction. Dorri *et al.* [13] proposed a local and private blockchain for access control scheme among smart home devices. Specifically, the access policies are stored in the block header while the access operations, device additions and deletions are recorded in the block body. Ding *et al.* [14] proposed a novel attribute-based access control scheme, where the attributes of each IoT device are recorded in transactions and managed by multiple attribute authorities. The data access is allowed when the attributes recorded on the blockchain are enough to match the access policy. Di Francesco Maesa *et al.* [15] achieved an access control system based on Bitcoin platform, which can manage access policies and realize the transfer of permissions between users through transactions. Wu *et al.* [23] proposed a traceable attribute-based encryption scheme, which sends the hidden policies achieved by attribute bloom filter to the blockchain via transactions. Ouaddah *et al.* [29] proposed a distributed access control framework named FairAccess based on blockchain within their prior proposed reference model [30]. Zyskind *et al.* [31] proposed a decentralized system for protecting mobile application data, in which access permissions are also placed on the blockchain through transactions. The data is encrypted by the shared key and routed to a distributed hashtable for off-blockchain key-value storage. Wang *et al.* [4] proposed a fine-grained access control framework for decentralized storage system based on IPFS, Ethereum blockchain and ABE. However, there are still some security challenges, such as the privacy leakage of access policies, a poor level of intelligence and scalability.

B. Smart Contract-Based Access Control Schemes

Smart contract is a computerized transaction protocol, which specifies transactions rules and can be self-verifying and self-executing in the blockchain [32]. It has been introduced into

TABLE I
NOTATIONS IN THIS PAPER

Notations	Descriptions
$\mathbb{G}, \mathbb{G}_T$	Two cyclic multiplicative groups
Att_i	An attribute
$S = \{Att_1, Att_2, \cdots, Att_n\}$	An attribute set
$\mathbb{A} = \{W_1, W_2, \cdots, W_n\}$	An access policy
$\mathcal{T}$	A tree-structured access policy
λ	A security parameter
PK	A public key in OHP-CP-ABE
MSK	A master secret key in OHP-CP-ABE
$SK_{\vec{v}}$	A secret key in OHP-CP-ABE
m	A message (data sharing by data owner)
CT	A ciphertext
$\vec{x}$	A vector generated by an access policy
$\vec{v}$	A vector generated by an attribute set
BPK	A public key registered in blockchain
BSK	A private key registered in blockchain
$E(\cdot)$	ElGamal encryption algorithm

access control management to improve the level of intelligent control. Lin *et al.* [33] proposed a blockchain-based security mutual authorization system to solve the security problems in the centralized management of industrial equipments. They achieve the access control by combining permission data hash table and smart contract. Xiao *et al.* [34] proposed the PrivacyGuard system for controlling access to personal privacy data, where an off-chain executed smart contract is used to specify access conditions and access records are stored in a distributed ledger. Azaria *et al.* [35] proposed the MedRec, which uses smart contract on the blockchain to achieve permission management for secure medical data access. To improve the security of MedRec, Dagher *et al.* [36] proposed a blockchain-based framework named Ancile for secure medical records sharing, which also uses smart contract to heighten access control. Alphand *et al.* [37] proposed a blockchain architecture for secure IoT, in which smart contracts are used to authorize access permissions and the user obtains the key through a trusted server. Novo [38], [39] proposed an access control architecture based on blockchain for IoT, in which the miners authorize devices based on the smart contract written by the manager according to the access rules. Xu *et al.* [40] proposed the BlendCAC, a capability delegation mechanism for permission propagation, which realizes the registration, propagation and revocation of access authorization by using smart contract. Zhang *et al.* [41] proposed three types of contracts, namely access control contract, judge contract and register contract, to improve the level of trust and intelligent access control. However, these schemes mainly focus on improving the level of automated access and reducing the management cost, while doing insufficiently to meet the fine-grained, scalability and privacy access requirements.

III. PRELIMINARIES

In this section, we review some basic knowledge associated with our TrustAccess. For clarity, we first present some notations used in this paper in Table I.

A. Blockchain

Blockchain is a distributed shared ledger, which integrates with various techniques discussed below to build trust relationship among peers that distrust each other without the involvement of a trusted authority [42], [43]. Compared with a public blockchain, a consortium blockchain is formed by a set of preselected nodes, wherein access control is one of the main methods to achieve privacy protection.

- Data structure: Each data block is made up of a block header and a block body, where the block header contains a cryptographic hash to previous block for traceability and immutability and the block body contains the details of transactions [42], [44]. To guarantee the blockchain with tamper-proof, integrity and authenticity, cryptographic hashes and digital signatures as two primary cryptographic primitives are used [45].
- Consensus protocol: It defines a common rule among nodes in blockchain to generate a new block. A great deal of research on blockchain consensus has been done, which can be classified into probabilistic consensus and deterministic consensus [46]–[49]. Compared with probabilistic consensus, deterministic consensus algorithms are mostly BFT-based and its variants, which can achieve much higher throught and are widely used in consortium blockchains.
- Smart contract: The concept is firstly described as a computerized transaction protocol by Nick Szabo [50]. The emergence of blockchain provides a secure running environment for smart contract, and it is firstly integrated into Ethereum by Vitalik Buterin [51]. Smart contracts negotiated by all nodes are broadcast to the blockchain by transactions for consensus, where those contract terms can be self-verifying and self-executing once one or more predefined conditions are triggered. A blockchain has built-in smart contract that implement its transaction logics and any distributed application can invoke smart contract via contract address in the form of transactions.

B. Bilinear Groups of Composite Order

Definition 1 (Composite Order Bilinear Groups [17], [52]): Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic multiplicative groups of order $N = pqr$, where p, q, r are distinct primes. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map with the properties:

- Bilinearity: $\forall u, v \in \mathbb{G}, a, b \in Z_N$, we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.
- Non-degeneracy: $\exists g \in \mathbb{G}$ such that $\hat{e}(g, g)$ has order N in $\mathbb{G}_T$.
- Computability: $\forall u, v \in \mathbb{G}$, $\hat{e}(u, v)$ can be efficiently computed.

Further, let $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r$ denote the subgroups of $\mathbb{G}$ with order p, q, r, respectively. Therefore $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r$, and $\forall h_p \in \mathbb{G}_p, h_q \in \mathbb{G}_q$, we have $\hat{e}(h_p, h_q) = 1$. The same is true for applying $\hat{e}$ to elements in distinct subgroups.

C. Attribute and Access Policy

We associate each attribute Att_i with a unique element in Z_N, which can be achieved by a collision-resistant hash function H from $\{0, 1\}^*$ to Z_N. Then we assume that there are n attribute categories.

Definition 2 (Attribute set and Access policy [17]): Let the $n \times l$ matrix $M = (V_1, \ldots, V_i, \ldots, V_n)$ be the possible attributes, where $V_i = (k_{i,1}, \ldots, k_{i,j}, \ldots, k_{i,l})$ and $k_{i,j} \in Z_N$. In other words, V_i is the set of all possible values

of the i-th category attribute. So the user's attribute set $S = \{Att_1, Att_2, \ldots, Att_n\}$ can be represented as $S = \{k_{1,j_1}, k_{2,j_2}, \ldots, k_{i,j_i}, \ldots, k_{n,j_n}\}$, $j_i \in \{1, 2, \ldots, l\}$. There is an access policy $\mathbb{A} = \{W_1, \ldots, W_i, \ldots, W_n\}$, where $W_i \subseteq V_i$. The attribute set S satisfies the access policy $\mathbb{A}$ if and only if $k_{i,j_i} \in W_i$, $1 \leq i \leq n$.

Definition 3 (Vector [17], [21]): Both the access policy and the attribute set can be represented by a vector of $(d+1)^t$ elements, where t represents the categories of attributes and d represents the number of attribute values. In TrustAccess, we represent the access policy $\mathcal{T}$ as a tree structure, where each non-leaf node of the tree represents a threshold gate and each leaf node is described by an attribute.

The predicate $\mathbf{OR}_{I_1,I_2}$ can be encoded as the polynomial

$$p(x_1, x_2) = (x_1 - I_1) \cdot (x_2 - I_2),$$

$\mathbf{AND}_{I_1,I_2}$ can be encoded as the polynomial

$$p(x_1, x_2) = (x_1 - I_1) + r(x_2 - I_2),\ r \in Z_N.$$

Now we give an example of the vector representation. Suppose an access policy with $t = 3$, $d = 1$, we randomly select $r \in Z_N$ and indicate the hash attribute by $I_{Att_i} = H(category : Att_i)$. The access policy is as follows.

$$\mathcal{T} = (company : A\,\mathbf{OR}\,position : PM)\,\mathbf{AND}\,(detail : U/G).$$

It can be represented by the polynomial

$$\begin{aligned} p(x_1, x_2, x_3) &= (x_1 - I_A) \cdot (x_2 - I_{PM}) + r(x_3 - I_{U/G}) \\ &= 0 \cdot x_1x_2x_3 + 1 \cdot x_1x_2 + 0 \cdot x_1x_3 \\ &\quad + 0 \cdot x_2x_3 + (-I_{PM})x_1 + (-I_A)x_2 \\ &\quad + rx_3 + (I_AI_{PM} - rI_{U/G}). \end{aligned}$$

So the $\mathcal{T}$ can be represented by an 8-element vector

$$\vec{x} = (0, 1, 0, 0, -I_{PM}, -I_A, r, I_AI_{PM} - rI_{U/G}).$$

On the other hand, a user has an attribute set

$$S = \{company : A,\ position : QA,\ detail : U/G).$$

It can be represented by an 8-element vector

$$\begin{aligned} \vec{v} = (&I_AI_{QA}I_{U/G}, I_AI_{QA}, I_AI_{U/G}, I_{QA}I_{U/G}, \\ &I_A, I_{QA}, I_{U/G}, 1). \end{aligned}$$

Obviously, the attribute set S satisfies the access policy $\mathcal{T}$ and $\vec{x} \cdot \vec{v} = 0$. In the above process, the $\mathcal{T}$ cannot be derived from the vector $\vec{x}$, but the $\vec{x} \cdot \vec{v}$ can be used to determine whether the access policy is satisfied.

D. ElGamal Cryptosystem

ElGamal cryptosystem [53] is a public key encryption system with the multiplicative homomorphic property, which is described as follows.

- Key generation: Let $\mathbb{G}_1$ be a cyclic group of large prime order q_1. It selects a generator g_1 of $\mathbb{G}_1$ and a random integer x $(1 \leq x \leq q_1 - 2)$, and then computes $h = g_1^x mod\ q_1$. The public key is $(\mathbb{G}_1, q_1, g_1, h)$ and the secret key is x.
- Encryption: To encrypt a message $m \in \mathbb{G}_1$, it selects a random integer r and then computes the ciphertext $E(m) = (c_1, c_2) = (g_1^r mod\ q_1, mh^r mod\ q_1)$.
- Decryption: To decrypt the ciphertext $E(m) = (c_1, c_2)$, the message $m = c_2 \cdot c_1^{-x} mod\ q_1$.
  In fact, it is homomorphic with respect to multiplication. That is,

$$\begin{aligned} E(m_1) \times E(m_2) &= (g_1^{r_1}, m_1h^{r_1}) \times (g_1^{r_2}, m_2h^{r_2}) \\ &= (g_1^{r_1+r_2}, m_1 \times m_2h^{r_1+r_2}) \\ &= E(m_1 \times m_2). \end{aligned}$$

E. Review of Lai et al.'s [17] Scheme

The scheme of Lai *et al.* [17] is mainly composed of four algorithms, namely *Setup*, *Encrypt*, *KeyGen* and *Decrypt*.

- `Setup(`1^λ`)`: In this algorithm, it takes an implicit security parameter λ as input, and outputs a public key PK and a master secret key MSK.
- `Encrypt(`PK`, `m`, `$\mathbb{A}$`)`: The encryption algorithm takes as inputs the public key PK, a message m, and an access policy $\mathbb{A}$ over the attributes. It outputs a ciphertext CT.
- `KeyGen(`PK`, `MSK`, `S`)`: The key generation algorithm takes as inputs the public key PK, the master secret key MSK and an attribute set S. It outputs a secret key SK_S that can decrypt the CT.
- `Decrypt(`PK`, `CT`, `SK_S`)`: The decryption algorithm takes as inputs the public key PK, the ciphertext CT and the secret key SK_S. It outputs the message m if the attribute set S satisfies the access policy $\mathbb{A}$.

IV. System Overview

In this section, we first introduce the system architecture of TrustAccess and then outline the OHP-CP-ABE. Finally, we present our security model and design goals.

A. System Architecture of TrustAccess

We introduce the system architecture of our TrustAccess depicted by Fig. 1, which involves three entities, namely data owner (DO), data user (DU) and consortium blockchain (CB).

- **DO**: A DO is responsible for making access policies, and encrypting the data using OHP-CP-ABE. To achieve a trustworthy secure access control while improving system efficiency, the DO sends the ciphertext address to the CB via a storage transaction $Tx_{storage}$.
- **DU**: A DU should get the secret key for decrypting the ciphertext indexed by the address. To protect the attribute privacy, a DU locally matches the set of attributes with the hidden policy and encrypts it by ElGamal cryptosystem. If the matching is successful, it would generate a **Proof** for access without privacy leakage, and then broadcast it to the other nodes in the CB for consensus validation. Note that the matching and validation process are done by a smart contract, which can improve the level of intelligent control. Once verified, the DO generates an access transaction

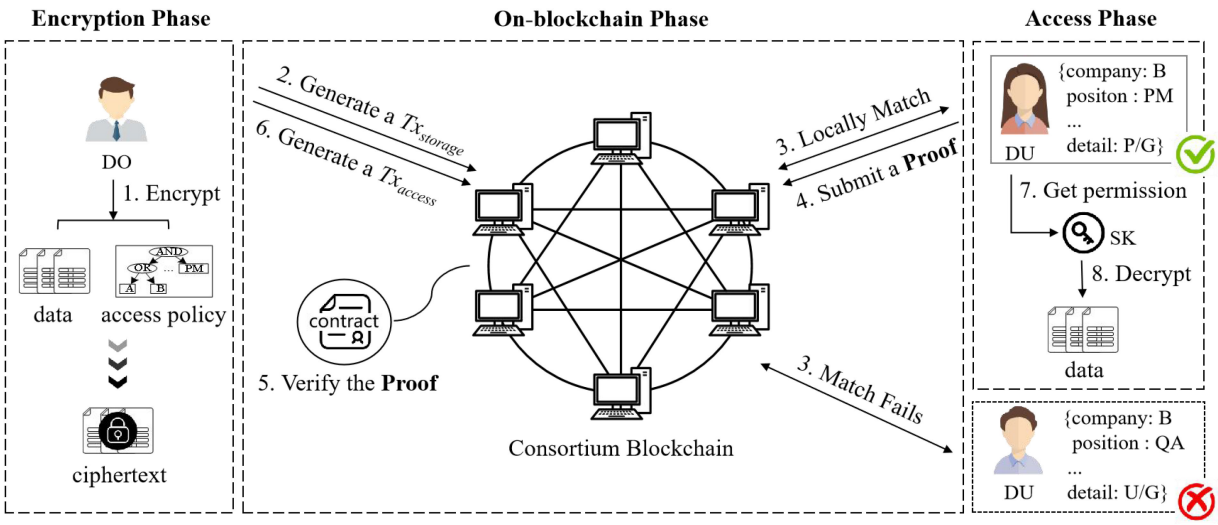

Fig. 1. System architecture of TrustAccess.

Tx_{access} to the CB for consensus validation. Once it is authorized, the DU can obtain the secret key from the DO to decrypt the ciphertext indexed by the address. Furthermore, a malicious DU might obtain the secret key by forging a **Proof**, but it cannot decrypt the ciphertext because its attribute set cannot satisfy the access policy made by the DO.

- **CB**: A CB is a distributed platform to record $Tx_{storage}$ and Tx_{access}, which is formed by DOs and DUs.

In our TrustAccess, only DOs and DUs in a CB are involved in the access control management as described above. The computational processes in terms of encryption, decryption and matching are done locally without the involvement of an intermediary entity in our TrustAccess. It can prove that our TrustAccess can not only achieve a distributed and trustworthy access control, but also protect the privacy of access policies and attributes.

B. Overview of the OHP-CP-ABE Scheme

In this section, we introduce the OHP-CP-ABE, including *Setup*, *Enc*, *KeyGen*, and *Dec* algorithms.

- $\texttt{Setup}(1^\lambda) \rightarrow (PK, MSK)$: The setup algorithm takes an implicit security parameter λ as input, and outputs a public key PK and a master secret key MSK.
- $\texttt{Enc}(PK, m, \mathcal{T}) \rightarrow CT$: The encryption algorithm takes as inputs the public key PK, a message m, and an access policy $\mathcal{T}$ over the attributes. It outputs a ciphertext CT of m with respect to the access policy $\mathcal{T}$. Note that only the user whose attribute set satisfies the access policy $\mathcal{T}$ can decrypt the ciphertext CT. To achieve policy hiding, the $\vec{x}$ represented by the access policy vectorization is embedded into the ciphertext.
- $\texttt{KeyGen}(PK, MSK) \rightarrow SK_{\vec{v}}$: The key generation algorithm takes as inputs the public key PK and the master secret key MSK. It outputs a secret key $SK_{\vec{v}}$ that can decrypt the ciphertext CT.
- $\texttt{Dec}(PK, CT, S, SK_{\vec{v}}) \rightarrow m \text{ or } \perp$: The decryption algorithm takes as inputs the public key PK, the ciphertext CT, an attribute set S and the secret key $SK_{\vec{v}}$. If the attribute set S satisfies the access policy $\mathcal{T}$, that is $\vec{x} \cdot \vec{v} = 0$, it outputs the message m. Otherwise, it outputs the terminated symbol $\perp$.

C. Security Model

In this paper, we use the consortium blockchain and consider that each node generally follows our protocol, but tries to find out as much secret data as possible [54]. The public key of each node can be easily obtained when necessary, and the communication channel between the users is assumed to be secure, which can be guaranteed by security protocols such as SSL/TLS. Next, we define a security model through an interactive game between an adversary and a challenger. Based on [7], [17], [21], the security game is described as follows.

- **Setup**: The challenger runs the setup algorithm *Setup* to generate a public key PK and a master secret key MSK. The MSK is saved by itself and the PK is sent to the adversary.
- **Phase 1**: The adversary adaptively issues queries to the challenger for the secret key. Afterwards, the challenger runs key generation algorithm *KeyGen* and returns the secret key $SK_{\vec{v}}$ to the adversary.
- **Challenge**: The adversary submits two equal length messages m_1, m_2 and two restricted access policies $\mathcal{T}_1, \mathcal{T}_2$ that cannot be satisfied by any of the queried attribute sets. The challenger selects a random bit $y \in \{0, 1\}$, computes $CT^* \leftarrow Enc(PK, m_y, \mathcal{T}_y)$, and returns CT^* to the adversary as its challenge ciphertext.
- **Phase 2**: The adversary continues to probe the challenger for the secret key with the added restriction that none of attribute sets satisfies the $\mathcal{T}_1$ and $\mathcal{T}_2$.
- **Guess**: The adversary outputs its guess $y' \in \{0, 1\}$ for y and wins the game if $y = y'$.

In this game, the advantage of the adversary is defined as the $P = |Pr[y = y'] - \frac{1}{2}|$, where the probability is occupied by the random bits used by the challenger and the adversary.

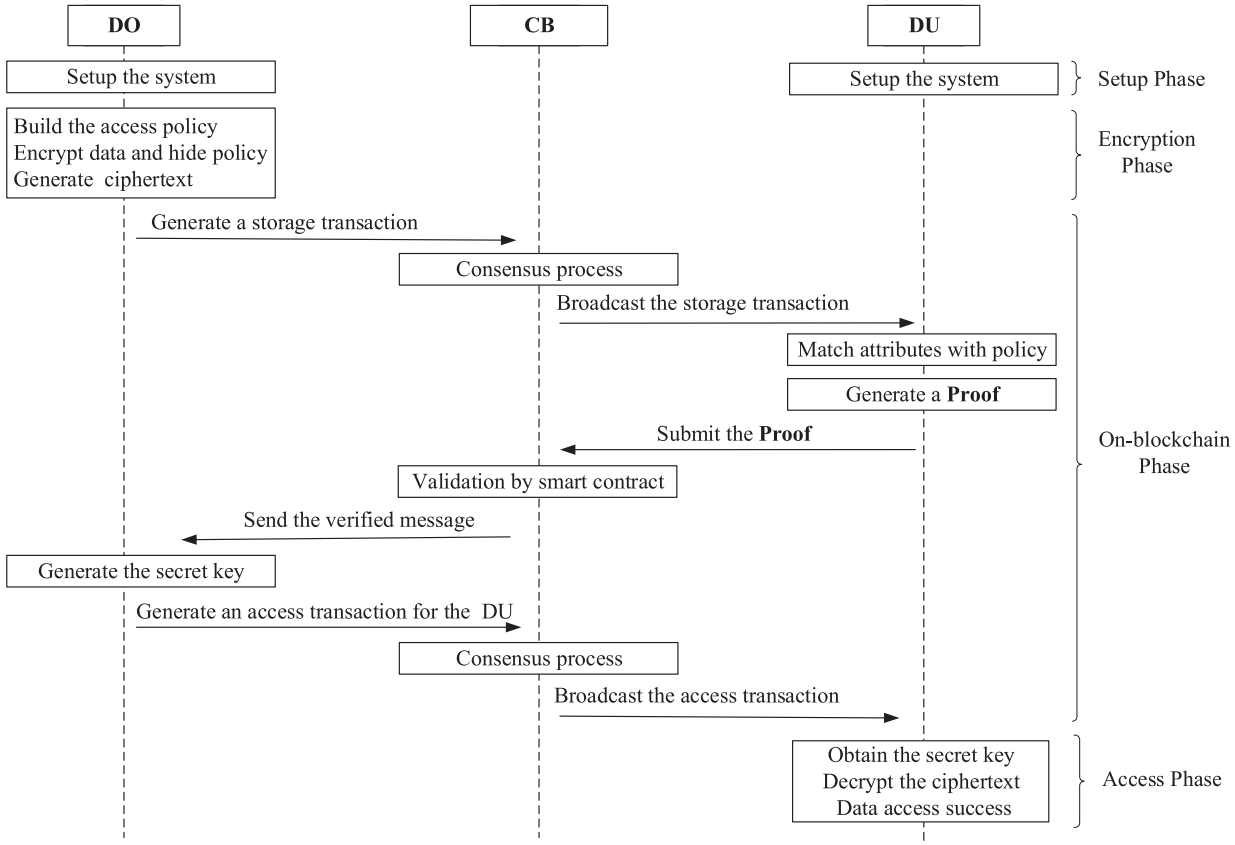

Fig. 2. The overview procedure of TrustAccess.

Definition 4: The OHP-CP-ABE scheme achieves fully secure if all polynomial time adversaries have at most a negligible advantage P in the security game.

D. Design Goals

The overall goal of our TrustAccess is to achieve a trustworthy secure ciphertext-policy and attribute hiding access control management, which can be decomposed into the following respects concretely: (1) It should guarantee the data confidentiality owned by the DO and achieve a fine-grained and trustworthy data access control. (2) The access policy should be hidden and only those eligible DUs can access the data without revealing any personal information. (3) It should satisfy the practical access requirements on effectiveness and scalability.

V. The Design of TrustAccess

In this section, we first present the overview procedure of our TrustAccess and then present the construction in details.

A. The Overview Procedure of TrustAccess

Generally, the overview procedure of our TrustAccess can be simply divided into four phases depicted by Fig. 2, namely setup phase, encryption phase, on-blockchain phase, access phase.

- *Setup Phase:* This phase initializes the parameters in the system, such as composite order bilinear groups and ElGamal cryptosystem.
- *Encryption Phase:* A DO specifies an access policy based on a set of attributes and encrypts the data by the OHP-CP-ABE scheme for locally secure storage.
- *On-blockchain Phase:* A DO sends the ciphertext address with the hidden access policy to the blockchain via a storage transaction. Any DU that wants to access data owned by the DO should locally match the hidden access policy $\vec{x}$ with a set of attributes $\vec{v}$. If successful, the DU would generate a **Proof** by ElGamal cryptosystem, broadcast it to the other nodes in the CB and trigger the smart contract that has been deployed in each node in the CB to verify the validity of the **Proof**. Once agreed upon, the DO would generate a secret key and send an access transaction to the CB. It is worth noting that the access transaction is used to record the access control history of the DO, which can be used for traceability and accountability.
- *Access Phase:* The DU can get the secret key once the consensus process on access transaction is achieved. Note that the secret key is locally stored in the DO, which can not be accessed by the CB. It would be distributed automatically without any human intervention via a secure communication channel. The DU can use the secret key to decrypt the ciphertext indexed by the address.

Now we give a brief overview of the **Proof**, more details would be presented in Section V-B. The **Proof** is an identification to indicate a successful matching between a hidden access policy and a set of attributes, which can be regarded as the permission to get the secret key. As we discussed, the hidden policy has been stored and can be accessed by any node in the CB. In order to protect a DU's attributes privacy while getting the secret key to achieve a trustworthy access, the DU would locally match the hidden access policy with a set of attributes, that is to check if $\vec{x} \cdot \vec{v} = 0$. If true, a **Proof** would be generated by the ElGamal cryptosystem. Afterwards, it would be sent to the other nodes in the CB for validation. By using the multiplicative homomorphic property of ElGamal encryption, each node would locally check $E(\vec{x}) \cdot E(\vec{v}) = E(0)$ to judge the validity of the **Proof**. Obviously, because the attributes are encrypted and each verification process is performed locally, the privacy of any DU's attributes can be protected.

Furthermore, we use the Practical Byzantine Fault Tolerance (PBFT) [55] in our TrustAccess to reach a consensus on the storage transaction and access transaction. PBFT is a state machine replication algorithm [55], which is usually used in consortium blockchains. Due to the limited space, more details on PBFT are not discussed in this paper. In addition, in order to address the storage limitation of blockchain, we store ciphertext locally and send the ciphertext address and integrity check code to the CB to ensure the locally stored data integrity, while improving the system scalability.

B. Construction

1) Setup Phase: Our TrustAccess takes a security parameter λ as input and runs the group generator $\boldsymbol{\mathcal{G}}(1^{\lambda})$ to get $(p, q, r, \mathbb{G}, \mathbb{G}_T, \hat{e})$, where $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r$, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = pqr$. Then the DO performs the setup algorithm as follows.

- `Setup(`1^{λ}`)`: It picks generators g_p, g_r of $\mathbb{G}_p, \mathbb{G}_r$, respectively, and then randomly chooses $a, \omega \in Z_N$. It also chooses $R_0, R_1 \in \mathbb{G}_r$ uniformly at random. The public key is

$$PK = (A_0 = g_p \cdot R_0,\ A_1 = g_p^a \cdot R_1,\ g_r,$$
$$Y = \hat{e}(g_p, g_p)^{\omega}).$$

And the master secret key is

$$MSK = (g_p, a, \omega).$$

2) Encryption Phase: A DO specifies an access policy $\mathcal{T}$, which is represented as a matrix with $n \times l$. To guarantee the access policy privacy, the DO first vectorizes $\mathcal{T}$ and encrypts a message m, which is shown as follows.

- `Enc(PK, m,` $\mathcal{T}$`)`: Let $\mathcal{T} = (W_1, \ldots, W_i, \ldots, W_n)$ with $W_i \subseteq V_i$, where $V_i = (k_{i,1}, \ldots, k_{i,j}, \ldots, k_{i,l})$. The vector associated with the $\mathcal{T}$ is $\vec{x}$. The algorithm randomly chooses $s, s_{i,j} \in Z_N$ and $R'_0, R'_{i,j} \in \mathbb{G}_r$, and then computes $\tilde{C} = m \cdot Y^s$, $C_0 = A_0^s \cdot R'_0$, where $m \in \mathbb{G}_T$. It also computes

$$C_{i,j} = \begin{cases} A_1^s \cdot R'_{i,j}, & \text{if } k_{i,j} \in W_i; \\ A_1^{s_{i,j}} \cdot R'_{i,j}, & \text{otherwise.} \end{cases}$$

where $1 \le i \le n$ and $1 \le j \le l$. Finally, the generated ciphertext is

$$CT = (\vec{x},\ \tilde{C},\ C_0,\ \{C_{i,j}\}_{1 \le i \le n,\ 1 \le j \le l}).$$

Algorithm 1: Generate a Storage Transaction.

Input:
The identifier S of the storage transaction;
The storage address $storeAddress$ of the ciphertext;
The Ciphertext CT generated by the encryption algorithm;
The private key BSK_{DO} of the DO;

Output:
The storage transaction $Tx_{storage}$;

/* Compute the message digest for CT */
$checkCode = H(CT)$;
/* Compute the message digest for transaction */
$MD = H(S, storeAddress, checkCode)$;
/* Encrypt the message digest with the BSK_{DO} */
$sign = Sign_{BSK_{DO}}(MD)$;
/* Generate the storage transaction */
$Tx_{storage} = \{S, storeAddress, checkCode, sign\}$;
return $Tx_{storage}$;

3) On-blockchain Phase: To make a trustworthy secure access control management, the DO will send the ciphertext address with the hidden access policy to the CB via a storage transaction, that is,

$$Tx_{storage} = \{S,\ storeAddress,\ checkCode,\ sign\},$$

where S is used to identify storage transaction, $storeAddress$ denotes the ciphertext storage location (e.g., the hash pointer of the ciphertext), which is used to index the ciphertext CT, $checkCode$ is the integrity check code of ciphertext, $sign$ is the digital signature generated by the DO's private key registered in the CB. Algorithm 1 illustrates the entire process of a storage transaction generation. The $checkCode$ is generated by hashing ciphertext, thus any DU can verify the integrity of shared ciphertext through $checkCode$. The $sign$ is used to prove that the transaction is indeed sent by the DO.

After the $Tx_{storage}$ is generated, it is broadcast to the other nodes in the CB for verification. That is to verify the validity of the transaction through the $sign$ and the ciphertext through the $checkCode$. The details are shown in Algorithm 2.

The validity of the $Tx_{storage}$ is verified by comparing the message digest MD' generated by the received $Tx_{storage}$ with the message digest MD generated by decrypting the $sign$ using the public key BPK_{DO} of the DO. If they are equal, the $Tx_{storage}$ is considered to be valid. Note that in order to reduce the storage cost, the DO only sends the storage address of ciphertext to the CB. To ensure the integrity of ciphertext stored

Algorithm 2: Verify a Storage Transaction.

Input:
$Tx = \{S, storeAddress, checkCode, sign\}$;
The public key BPK_{DO} of the DO registered in the CB;

Output:
The validation of Tx and CT;

/* Compute the message digest for the transaction */
$MD' = H(S, storeAddress, checkCode)$;
/* Verify the $sign$ with the public key */
$MD = Compute_{BPK_{DO}}(sign)$;
if $MD' = MD$ **then**
  Obtain the CT according to the $storeAddress$;
  /* Compute the message digest for CT */
  $checkCode' = H(CT)$;
  if $checkCode' = checkCode$ **then**
    return *True*;
return *False*;

locally, we should further check whether the message digest $checkCode'$ of the ciphertext related to the $storeAddress$ is equal to the message digest $checkCode$ in the $Tx_{storage}$. Once verified, it would be packaged into a block for consensus by PBFT [55].

To get the secret key from the DO and ensure the privacy of attributes, a DU would locally match $\vec{x}$ from the CB with the $\vec{v}$ that are generated locally based on a set of attributes. Once $\vec{x} \cdot \vec{v} = 0$, it means that the matching is successful, and the DU would generate a **Proof** to trigger the smart contract for validation. Specially,

$$\mathbf{Proof} = \{BPK_{DU}, storeAddress, E(\vec{v}), sign\},$$

where BPK_{DU} represents the DU's public key registered in the CB as its identity, $storeAddress$ denotes the ciphertext storage location that the DU wants to access, $E(\vec{v}) = (E(v_1), E(v_2), \ldots, E(v_n))$ is the result by using ElGamal cryptosystem to encrypt each element in the $\vec{v}$ and $sign$ is the DU's digital signature of the **Proof**.

After verifying that the signature of the **Proof** is valid, an access transaction can be generated for the DU that satisfies the hidden access policy defined by the DO. Besides, to ensure that any DU with the valid signing **Proof** has the access right indeed, it will broadcast the **Proof** to the other nodes in the CB and trigger each smart contract to validate if the DU has the access right. The validation procedure is shown in Algorithm 3.

To be more specific, it takes as inputs the $E(\vec{v})$ in the **Proof**, the hidden policy and parameters required for ElGamal encryption and then outputs the verification result. Each node in the CB invokes the smart contract to locally perform multiplicative homomorphic ElGamal cryptosystem to encrypt each element in the $\vec{x}$ generated by an access policy. Note that the verification process would not reveal the secret key of the DU and the privacy of the DU's attributes can be protected. Because of the

Algorithm 3: Verify the Access Right.

Input:
The $E(\vec{v}) = (E(v_1), E(v_2), \cdots, E(v_n))$ in the **Proof**;
The hidden policy $\vec{x} = (x_1, x_2, \cdots, x_n)$ associated with the DO's ciphertext;
The public key $(\mathbb{G}_1, q_1, g_1, h)$ of the DU in ElGamal;
The random number r;

Output:
The verified result of **Proof**;

/* The ElGamal cryptosystem encrypts each element */
for $1 \le j \le n$ **do**
  $E(x_j) = (g_1^r mod\ q_1, x_j h^r mod\ q_1)$;
/* Generate the encrypted access policy */
$E(\vec{x}) = (E(x_1), E(x_2), \cdots, E(x_n))$;
/* Multiply the encrypted vectors */
$result = E(v_1) \cdot E(x_1) + E(v_2) \cdot E(x_2) + \cdots + E(v_n) \cdot E(x_n)$;
/*Verify the result and get the access right */
if $result = E(0)$ **then**
  return *True*;
return *False*;

Algorithm 4: Generate an Access Transaction.

Input:
The identifier A of the access transaction;
Proof = $\{BPK_{DU}, storeAddress, E(\vec{v}), sign\}$;

Output: The access transaction Tx_{access};

if *Proof is valid* **then**
  Get the current time $time$;
  /* Compute the message digest for transaction */
  $MD = H(A, BPK_{DU}, storeAddress, time)$;
  /* The DO signs the transaction */
  $sign = Sign_{BSK_{DO}}(MD)$;
  $Tx_{access} = \{A, BPK_{DU}, storeAddress, time, sign\}$;
  return Tx_{access};
return $\perp$;

multiplicative homomorphism, apparently $E(\vec{x}) \cdot E(\vec{v}) = E(0)$ if and only if $\vec{x} \cdot \vec{v} = 0$. Therefore, it can enable any node in the CB to verify whether a set of attributes matches the hidden access policy to ensure the validity of the **Proof**, while guaranteeing the privacy of the DU's attributes.

After that, the DO generates an access transaction for a valid **Proof**, which is described by Algorithm 4. The smart contract verifies access right and generates the access transaction

$$Tx_{access} = \{A, BPK_{DU}, storeAddress, time, sign\},$$

where A is used to identify access transaction, $time$ denotes the Tx_{access} generated time and $sign$ is used to prove that the Tx_{access} is indeed sent by the DO. Note that the Tx_{access} publisher needs to be consistent with the owner of the $storeAddress$ within the Tx_{access}.

4) Access Phase: In this phase, the DU authorized by the CB can obtain the secret key $SK_{\vec{v}}$, which is generated by the DO that runs the key generation algorithm *KeyGen*, and then the DU locally performs decryption algorithm *Dec* to obtain the data. Therefore, the access process can be described as follows.

- `KeyGen(PK, MSK)`: The DO randomly selects $t_i \in Z_N$ uniformly for $i = 1, 2, \ldots, n$, and sets $t = \Sigma_{i=1}^{n} t_i$. Then it computes

$$D_0 = g_p^{\omega - t},\ D_i = g_p^{t_i/a}.$$

Finally, the secret key is

$$SK_{\vec{v}} = (D_0,\ \{D_i\}_{1 \le i \le n}).$$

- `Dec(PK, CT, S, SK`$_{\vec{v}}$`)`: The set of attributes of the DU can be denoted as $S = \{k_{1,j_1}, k_{2,j_2}, \ldots, k_{i,j_i}, \ldots, k_{n,j_n}\},\ \ j_i \in \{1, 2, \ldots, l\}$. Note that the ciphertext can be correctly decrypted if and only if the k_{i,j_i} satisfies the access policy made by the DO. The decryption process can be computed as

$$\begin{aligned}
&\frac{\tilde{C}}{\hat{e}(C_0, D_0) \cdot \prod_{i=1}^{n} \hat{e}(C_{i,j_i}, D_i)} \\
&= \frac{m \cdot Y^s}{\hat{e}(A_0^s \cdot R_0', g_p^{\omega - t}) \cdot \prod_{i=1}^{n} \hat{e}(A_1^s \cdot R_{i,j_i}', g_p^{t_i/a})} \\
&= \frac{m \cdot \hat{e}(g_p, g_p)^{\omega s}}{\hat{e}(g_p^s, g_p^{\omega - t}) \cdot \prod_{i=1}^{n} \hat{e}((g_p^a)^s, g_p^{t_i/a})} \\
&= \frac{m \cdot \hat{e}(g_p, g_p)^{\omega s}}{\hat{e}(g_p^s, g_p^{\omega - t}) \cdot \prod_{i=1}^{n} \hat{e}(g_p^s, g_p^{t_i})} \\
&= \frac{m \cdot \hat{e}(g_p, g_p)^{\omega s}}{\hat{e}(g_p^s, g_p^{\omega})} = m.
\end{aligned}$$

VI. Security Analysis

In this section, we analyze the security of TrustAccess from the two aspects of blockchain operations and the OHP-CP-ABE.

A. Security Analysis of Blockchain Operations

There are some security requirements that need to be addressed, namely trustability, privacy, integrity, traceability.

1) Trustability: To achieve a trustworthy secure data access control, most of the existing work requires an intermediary entity, which would cause a high trust-building cost, single point of failure, privacy leakage and so on. In this paper, our TrustAccess takes advantage of the blockchain with the characteristics of distribution, disintermediation, transparency and immutability to address these issues. By sending the generated ciphertext address with the hidden access policy to the blockchain, we can guarantee the trustability of access control management without the involvement of any intermediary entity. In the phase of access control authorization, the **Proof** of each DU is verified by a pre-defined smart contract that achieves trusted authorization with less human intervention. Therefore, the blockchain platform serving data sharing and access control will be trustworthy.

2) Privacy: Most of the existing work poses a great threat to the disclosure of privacy by directly storing the access policies or attribute sets on the blockchain. To protect these privacy, the necessary hiding processing needs to be carried. In terms of access policies privacy, our OHP-CP-ABE can hide access policies through vector representations. This scheme can not only realize the fine-grained access control, but also store the data in an encrypted form such that eavesdroppers cannot obtain the sensitive data. In terms of the attribute privacy leakage that a DU faces, we use ElGamal cryptosystem to encrypt the attributes of the DU during permission validation. Thus, the privacy protection can be realized in the transparent and distributed environment of blockchain.

3) Integrity: To increase the scalability of the system, we store the ciphertext address in the blockchain. However, the ciphertext associated with the ciphertext address may be changed by the DO. To ensure that DUs can obtain the integrated data, we include the integrity check code of the ciphertext in the storage transaction, which can allow the ciphertext to be verified at any time to ensure data integrity.

4) Traceability: Our TrustAccess can track and verify the access control information in the blockchain. Any authorization is recorded as an immutable access transaction, so a DO will know which DUs have accessed the locally stored data and any DU cannot deny the access operation. In addition, storage transactions can also make it impossible for DOs to deny not providing their data. The DO can detect malicious attempts by the DUs through verification.

B. Security Analysis of the OHP-CP-ABE

The OHP-CP-ABE we improved based on Lai *et al.*'s scheme [17] can ensure the secure access control off the blockchain. Our security proof is based on four assumptions, among which Assumption 1,2 and 3 are used in [7], [17], [56], [57], and Assumption 4 is used in [7], [17], [56], [58]. Note that Assumption 1 and 3 are the same as that used in [17]. The specific assumptions are described as follows.

Assumption 1: Let $\boldsymbol{\mathcal{G}}$, the group generator, be defined as the following distribution.

$$\begin{aligned}
&(N = pqr,\ \mathbb{G},\ \mathbb{G}_T,\ \hat{e}) \leftarrow \boldsymbol{\mathcal{G}}(1^{\lambda}),\ g_p \xleftarrow{R} \mathbb{G}_p, \\
&g_r \xleftarrow{R} \mathbb{G}_r,\ D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_r), \\
&T_1 \xleftarrow{R} \mathbb{G}_p \times \mathbb{G}_q,\ T_2 \xleftarrow{R} \mathbb{G}_p.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is defined as

$$Adv1_{\mathcal{A}} = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 5: $\boldsymbol{\mathcal{G}}$ satisfies Assumption 1, if $Adv1_{\mathcal{A}}$ is negligible for any probability polynomial time algorithm $\mathcal{A}$.

Assumption 2: Let $\boldsymbol{\mathcal{G}}$, the group generator, be defined as the following distribution.

$$\begin{aligned}
&(N = pqr,\ \mathbb{G},\ \mathbb{G}_T,\ \hat{e}) \leftarrow \boldsymbol{\mathcal{G}}(1^{\lambda}), \\
&g_p, X_1, Y_1 \xleftarrow{R} \mathbb{G}_p,\ X_2, Y_2 \xleftarrow{R} \mathbb{G}_q,\ g_r \xleftarrow{R} \mathbb{G}_r, \\
&D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, X_1X_2, Y_1Y_2, g_r), \\
&T_1 \xleftarrow{R} \mathbb{G}_p \times \mathbb{G}_q, T_2 \xleftarrow{R} \mathbb{G}_p.
\end{aligned}$$

TABLE II
COMPARISONS OF BLOCKCHAIN-BASED ACCESS CONTROL SCHEMES

Scheme	Encrypted Storage	Fine Granularity	Attribute Privacy	Policy Privacy	No intermediary entity involved
[13]	×	×	×	×	✓
[11], [14], [29]	×	✓	×	×	✓
[37]	✓	×	×	×	×
[23]	✓	✓	×	✓	×
Our	✓	✓	✓	✓	✓

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is defined as

$$Adv2_{\mathcal{A}} = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 6: $\boldsymbol{\mathcal{G}}$ satisfies Assumption 2, if $Adv2_{\mathcal{A}}$ is negligible for any probability polynomial time algorithm $\mathcal{A}$.

Assumption 3: Let $\boldsymbol{\mathcal{G}}$, the group generator, be defined as the following distribution.

$$(N = pqr,\ \mathbb{G},\ \mathbb{G}_T,\ \hat{e}) \leftarrow \boldsymbol{\mathcal{G}}(1^\lambda),$$

$$\omega, s \in Z_N,\ g_p, Z_1 \xleftarrow{R} \mathbb{G}_p,\ X_2, Y_2, Z_2 \xleftarrow{R} \mathbb{G}_q,\ g_r \xleftarrow{R} \mathbb{G}_r,$$

$$D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p, g_p^\omega X_2, g_p^s Y_2, Z_1 Z_2, g_r),$$

$$T_1 \xleftarrow{R} \hat{e}(g_p, g_p)^{\omega s}, T_2 \xleftarrow{R} \mathbb{G}_T.$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is defined as

$$Adv3_{\mathcal{A}} = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 7: $\boldsymbol{\mathcal{G}}$ satisfies Assumption 3, if $Adv3_{\mathcal{A}}$ is negligible for any probability polynomial time algorithm $\mathcal{A}$.

Assumption 4: Let $\boldsymbol{\mathcal{G}}$, the group generator, be defined as the following distribution.

$$(N = pqr,\ \mathbb{G},\ \mathbb{G}_T,\ \hat{e}) \leftarrow \boldsymbol{\mathcal{G}}(1^\lambda),$$

$$a \in Z_N,\ g_p \xleftarrow{R} \mathbb{G}_p,\ g_q, Q_1, Q \xleftarrow{R} \mathbb{G}_q,$$

$$g_r, R_0, R_1, R \xleftarrow{R} \mathbb{G}_r,$$

$$D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g_p R_0, g_p^a R_1, g_p Q_1, g_q, g_r),$$

$$T_1 = g_p^a Q R,\ T_2 \xleftarrow{R} \mathbb{G}_T.$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is defined as

$$Adv4_{\mathcal{A}} = |Pr[\mathcal{A}(D, T_1) = 1] - Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 8: $\boldsymbol{\mathcal{G}}$ satisfies Assumption 4, if $Adv4_{\mathcal{A}}$ is negligible for any probability polynomial time algorithm $\mathcal{A}$.

Theorem 1: If Assumptions 1, 2, 3 and 4 hold, then the proposed OHP-CP-ABE is fully secure.

Proof: To prove the security of OHP-CP-ABE, we first introduce two additional structures, namely *Semi-functional Ciphertext* and *Semi-functional Key*. And then we prove the security by using a hybrid argument over a sequence of games. Finally, the lemmas on which the proof depends are given. More details about the security proof can be found in Appendix. ■

VII. COMPARISONS AND PERFORMANCE ANALYSIS

In this section, we first make comprehensive comparisons of our TrustAccess with the related work, and then conduct extensive experiments to demonstrate the efficiency.

A. Comprehensive Comparisons of Our TrustAccess

We compare TrustAccess with related blockchain-based access control schemes, which is presented in Table II. We analyze the method mainly from the following aspects, namely data encrypted storage, fine-grained access control, privacy of access policies and users' attributes, and whether an intermediary entity is needed. The access control scheme proposed by Dorri *et al.* [13] and Alphand *et al.* [37] can only achieve coarse-grained access control. Some work [11], [13], [14], [29] sends access policies or attributes to a blockchain for a trustworthy access control management. However, they do not take the security issues caused by transparency into account. Wu *et al.* [23] and Alphand *et al.* [37] proposed to encrypt and store the data off the blockchain. Both of them depend on an intermediary entity to distribute the keys, which results in some privacy concerns with the disclosure of a DU's attributes. Overall, these blockchain-based access control schemes cannot protect the privacy of access policies and attributes at the same time. We propose the TrustAccess based on blockchain to achieve a trustworthy secure ciphertext-policy and attribute hiding access control management. By storing the data encrypted via the OHP-CP-ABE locally and sending the ciphertext address with a hidden access policy to the blockchain, we can guarantee the trustability of access control with the characteristics of blockchain. In terms of privacy, we propose the OHP-CP-ABE scheme to prevent the privacy leakage of access policy, and use the ElGamal encryption to guarantee the DU's attribute privacy.

Furthermore, we compare our OHP-CP-ABE with some related CP-ABE schemes, which is shown in Table III. Bethencourt *et al.* [6] proposed the first CP-ABE schemes, which do not take the policy hiding into consideration. To protect the privacy of access policy, Lai *et al.* [56] proposed a partially hidden policy CP-ABE scheme in a small universe construction. Zhang *et al.* [7] also proposed a partially hidden policy CP-ABE scheme, while extending the scheme proposed by Lai *et al.* [56] to a large universe construction. These schemes support Linear Secret Sharing Schemes (LSSS) realizable access policies. To achieve a better privacy, Phuong *et al.* [59] proposed a hidden policy CP-ABE scheme based on the Viète's formula, but it is not fully secure. Lai *et al.* [17] proposed a fully secure ciphertext-policy hiding CP-ABE scheme, which is combined with IPE scheme to realize the partially hidden policy. However, it is a small universe construction. Generally, these schemes either

TABLE III
COMPARISONS OF CP-ABE SCHEMES

Scheme	Hidden policy	Expressiveness	Large Universe	Fully Secure	Blockchain
[6]	×	LSSS	×	✓	×
[56]	✓	LSSS	×	✓	×
[7]	✓	LSSS	✓	✓	×
[59]	✓	AND-gates	×	×	×
[17]	✓	AND-gates	×	✓	×
Our	✓	AND-gates	✓	✓	✓

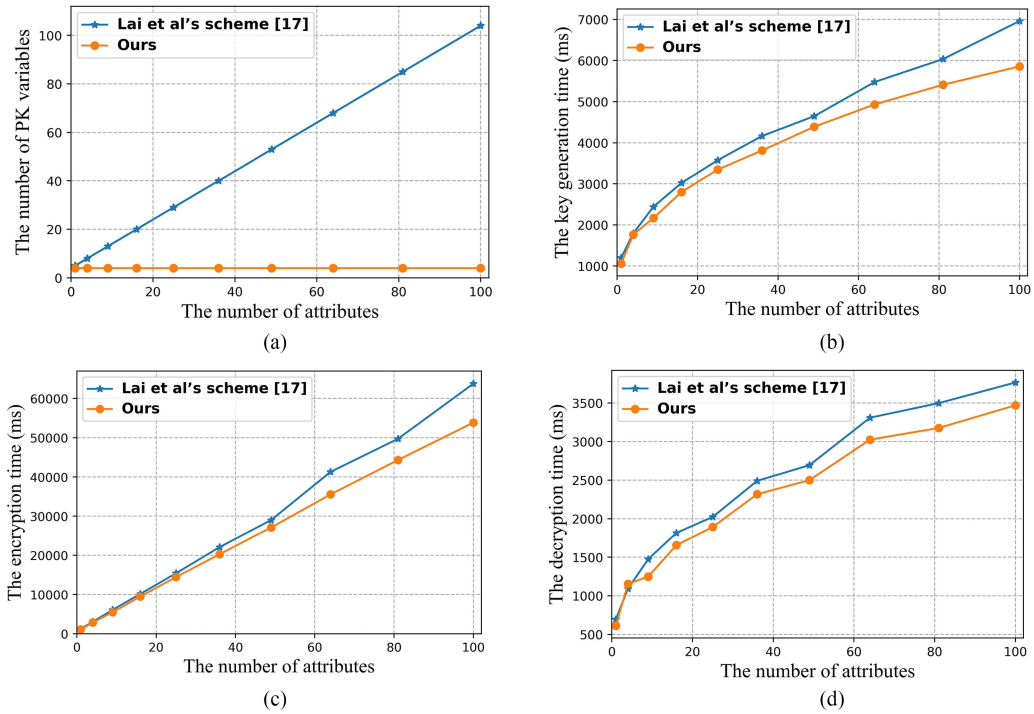

Fig. 3. Performance analysis and comparisons. (a) Public key. (b) Key generation algorithm. (c) Encryption algorithm. (d) Decryption algorithm.

achieve a large universe construction or achieve full security. In our OHP-CP-ABE scheme, we improved the fully secure scheme proposed by Lai *et al.* [17] to achieve a large universe construction and support hidden access policy. In addition, our TrustAccess uses the blockchain to improve the trustability of access control management.

B. Performance Evaluation

In this subsection, we conduct real-world experiments to evaluate the performance of our proposed scheme, and make performance analysis and comparisons.

1) Experimental Setup: The experiments are implemented on a laptop (with 1.60 GHz Intel (R) Core (TM) i5-8250 U CPU and 8 GB RAM memory) based on Windows 10 and the Java Pairing Based Cryptography Library 2.0.0 [60]. We use Type A1 pairing, which is constructed on the curve $y^2 = x^3 + x$ over the field F_N with the composite N being the universe size.

2) Performance Analysis: As we discussed, the efficiency of our TrustAccess is mainly affected by the operations on the blockchain and OHP-CP-ABE. For the efficiency of the operations on the blockchain, in our TrustAccess, we store the ciphertext address instead of the ciphertext to the blockchain, which can improve the system scalability. The performance of operations on the blockchain mainly depends on the consensus algorithm PBFT, which is not the focus in this paper. Here, we mainly concentrate on the efficiency of our OHP-CP-ABE. Compared with the scheme proposed by Lai *et al.* [17], our OHP-CP-ABE can achieve a large universe construction, which is more suitable for blockchain-based systems. The performance analysis and comparisons are depicted by Fig. 3, where Fig. 3(a) denotes the result that we achieve a large universe in the setup phase. Meanwhile, we compare key generation algorithm, encryption algorithm and decryption algorithm with the scheme of Lai *et al.* [17], and experimental results are shown in Fig. 3(b), 3(c) and 3(d) respectively. Note that all the experimental results we get are the average values by repeating running for 20 times. As illustrated, the key generation time, the encryption time and the decryption time increase with the number of attributes grows, where the x-axis is the number of attributes in the system and the y-axis is the running time of the algorithm. In general, the results show that the total time increases as the growth of the number of attributes. That is because in these algorithms, the number of variable operations is greatly affected by the number of attributes. The time that our TrustAccess spends is relatively shorter than the scheme proposed by Lai

et al. [17]. That is because we optimize the key generation process during the setup phase, which achieves a constant key size. Therefore, the product computations of the random variables are less than the scheme proposed by Lai *et al.* [17], which can achieve a more efficient access management.

VIII. Conclusion

The involvement of an intermediary entity would suffer from a high trust-building cost, single point of failure, privacy leakage and so on. In this paper, we propose the TrustAccess based on blockchain, which can achieve a distributed and trustworthy access control management. To address the privacy issues of access policy and user attribute in the TrustAccess, we propose the OHP-CP-ABE to support a large universe and hidden access policy. In addition, we propose to use ElGamal cryptosystem to protect a user's full attribute privacy for trustworthy access. Theoretical analysis and experimental results indicated that TrustAccess is more trustworthy, secure, private and scalable than existing schemes. As for the future research work, we will consider to optimize and evaluate the operation efficiency of TrustAccess on the blockchain.

Appendix
Security Proof of the OHP-CP-ABE

In this appendix, we detail the proof of the security for OHP-CP-ABE.

Proof: According to dual system encryption methodology [57], we need to define two structures that will not be used in the real system but will be used in our proof, namely *semi-functional ciphertext* and *semi-functional key*.

Semi-functional Ciphertext: First, the normal ciphertext generated by the encryption algorithm *Enc* is

$$CT' = (\vec{x},\ \tilde{C}',\ C_0',\ \{C_{i,j}'\}_{1\le i\le n, 1\le j\le l}).$$

Then, let g_q be a generator of the subgroup $\mathbb{G}_q$. We randomly choose exponents $x_0, x_{i,j} \in Z_N$, where $1 \le i \le n, 1 \le j \le l$. Finally, the semi-functional ciphertext is

$$CT = (\vec{x},\ \tilde{C} = \tilde{C}',\ C_0 = C_0' \cdot g_q^{x_0},$$
$$\{C_{i,j} = C_{i,j}' \cdot g_q^{x_{i,j}}\}_{1\le i\le n, 1\le j\le l}).$$

Semi-functional Key: First, the normal key generated by the key generation algorithm *KeyGen* is

$$SK_{\vec{v}}' = (D_0',\ \{D_i'\}_{1\le i\le n}).$$

Then, we randomly choose exponents $y_0, y_i \in Z_N, 1 \le i \le n$.

Finally, the semi-functional key is

$$SK_{\vec{v}} = (D_0 = D_0' \cdot g_q^{y_0},\ \{D_i = D_i' \cdot g_q^{y_i}\}_{1\le i\le n}).$$

We prove the security by a hybrid argument using a sequence of games as below.

- $\mathbf{Game_0}$: This game is a real security game, in which the ciphertext and the key are normal.
- $\mathbf{Game_1}$: In this game, the key is normal but the ciphertext is semi-functional. It is also $\mathbf{Game_{2,0}}$.
- $\mathbf{Game_{2,k}}$: In this game, the ciphertext is semi-functional, the first k keys are semi-functional and the rest of keys are normal. Suppose ζ is the number of keys queried by the adversary, and then $1 \le k \le \zeta$.
- $\mathbf{Game_3}$: In this game, the ciphertext and the key are semi-functional. The ciphertext is a semi-functional encryption of a random message, different from the message provided by the adversary.
- $\mathbf{Game_4}$: This game is the same as $\mathbf{Game_3}$, except that the ciphertext is independent from $\mathcal{T}_1$ and $\mathcal{T}_2$ provided by the adversary. It is obvious that in the $\mathbf{Game_4}$, no adversary can have an advantage greater than 0.

The proof requires the following lemma.

Lemma 1: Suppose that $\mathcal{G}$ satisfies Assumption 1. Then $\mathbf{Game_0}$ and $\mathbf{Game_1}$ are indistinguishable.

Proof: Suppose there is an algorithm $\mathcal{A}$ that distinguishes $\mathbf{Game}_0$ and $\mathbf{Game}_1$, then we will set up an algorithm $\mathcal{B}$ that has non-negligible advantage to break Assumption 1. $\mathcal{B}$ obtains g_p, g_r, T and simulates $\mathbf{Game}_0$ or $\mathbf{Game}_1$ with $\mathcal{A}$.

$\mathcal{B}$ randomly chooses $a,\ a_1,\ a_2,\ \omega \in Z_N$. It then sets $R_0 = g_r^{a_1},\ R_1 = g_r^{a_2},\ A_0 = g_p \cdot R_0,\ A_1 = g_p^a \cdot R_1$, and sends $\mathcal{A}$ the public key

$$PK = (A_0, A_1, g_r, Y = \hat{e}(g_p, g_p)^{\omega}).$$

$\mathcal{B}$ can run the key generation algorithm to generate the normal key in response to $\mathcal{A}$'s key request. $\mathcal{A}$ sends $\mathcal{B}$ two messages of the equal length m_1, m_2 and two access policies $\mathcal{T}_1, \mathcal{T}_2$. $\mathcal{B}$ chooses $\beta = \{0, 1\}$ randomly and does the following.

1) $\mathcal{B}$ randomly chooses $s \in Z_N$ and $R_0' \in \mathbb{G}_r$. It also chooses $s_{i,j} \in Z_N$ and $R_{i,j}' \in \mathbb{G}_r$ at random.
2) $\mathcal{B}$ computes $\tilde{C} = m_\beta \cdot \hat{e}(g_p^{\omega}, T),\ C_0 = T \cdot R_0^s \cdot R_0'$ and

$$C_{i,j} = \begin{cases} T^a \cdot R_1^s \cdot R_{i,j}', & \text{if } v_{i,j} \in W_i; \\ T^{a \cdot r_{i,j}} \cdot R_1^{s_{i,j}} \cdot R_{i,j}', & \text{otherwise.} \end{cases}$$

   where $1 \le i \le n$ and $1 \le j \le l,\ r_{i,j} = s_{i,j}/s$.
3) $\mathcal{B}$ sets the challenge ciphertext as $CT = (\vec{x}, \tilde{C}, C_0, \{C_{i,j}\}_{1\le i\le n, 1\le j\le l})$ and sends to $\mathcal{A}$.

If $T \xleftarrow{R} \mathbb{G}_p \times \mathbb{G}_q$, let $T = g_p^s g_q^{x_0}$, then $\tilde{C} = m_\beta \cdot \hat{e}(g_p, g_p)^{\omega s}, C_0 = g_p^s g_q^{x_0} \cdot R_0^s \cdot R_0'$ and

$$C_{i,j} = \begin{cases} g_p^{as} g_q^{x_{i,j}} \cdot R_1^s \cdot R_{i,j}', & \text{if } v_{i,j} \in W_i; \\ g_p^{as_{i,j}} g_q^{x_{i,j}} \cdot R_1^{s_{i,j}} \cdot R_{i,j}', & \text{otherwise.} \end{cases}$$

where $x_{i,j} = ax_0$ when $v_{i,j} \in W_i$, otherwise, $x_{i,j} = ax_0 r_{i,j}$, and $1 \le i \le n,\ 1 \le j \le l$. Hence, the ciphertext is semi-functional and $\mathcal{B}$ simulates $\mathbf{Game_1}$. If $T \xleftarrow{R} \mathbb{G}_p$, it is a normal ciphertext and $\mathcal{B}$ simulates $\mathbf{Game_0}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. ■

Lemma 2: Suppose that $\mathcal{G}$ satisfies Assumption 2. Then $\mathbf{Game_{2,k-1}}$ and $\mathbf{Game_{2,k}}$ are indistinguishable.

Proof: Suppose there is an algorithm $\mathcal{A}$ that distinguishes $\mathbf{Game_{2,k-1}}$ and $\mathbf{Game_{2,k}}$, then we will set up an algorithm $\mathcal{B}$ that has non-negligible advantage to break Assumption 2. $\mathcal{B}$

obtains $g_p,\ X_1X_2,\ Y_1Y_2,\ g_r,\ T$ and simulates $\mathbf{Game_{2,k-1}}$ and $\mathbf{Game_{2,k}}$ with $\mathcal{A}$.

$\mathcal{B}$ randomly chooses $a,\ a_1,\ a_2,\ \omega \in Z_N$. It then sets $R_0 = g_r^{a_1},\ R_1 = g_r^{a_2},\ A_0 = g_p \cdot R_0,\ A_1 = g_p^a \cdot R_1$, and sends $\mathcal{A}$ the public key

$$PK = (A_0, A_1, g_r, Y = \hat{e}(g_p, g_p)^{\omega}).$$

$\mathcal{B}$ knows the master secret key $MSK = (g_p,\ a,\ \omega)$.

Next, we explain how $\mathcal{B}$ answers the j-th key query.

- For $j < k$, $\mathcal{B}$ creates a semi-functional key by randomly choosing $t_i \in Z_N$ uniformly for $i = 1, 2, \ldots, n$, and sets $t = \Sigma_{i=1}^n t_i$. Then it computes

  $$D_0 = (Y_1Y_2)^{\omega - t},\ D_i = (Y_1Y_2)^{t_i/a}.$$

  Note that this is a distributed semi-functional key.
- For $j > k$, $\mathcal{B}$ can run the key generation algorithm *KeyGen* based on the master secret key MSK to generate the normal key.
- For $j = k$, $\mathcal{B}$ creates a semi-functional key by randomly choosing $t_i \in Z_N$ uniformly for $i = 1, 2, \ldots, n$, and sets $t = \Sigma_{i=1}^n t_i$. Then it computes

  $$D_0 = T^{\omega - t},\ D_i = T^{t_i/a}.$$

  If $T \xleftarrow{R} \mathbb{G}_p \times \mathbb{G}_q$, let $T = g_p g_q^c, c \in Z_N$, then

  $$D_0 = g_p^{\omega - t} g_q^{y_0},\ D_i = g_p^{t_i/a} g_q^{y_i}.$$

  where $y_0 = c(\omega - t),\ \ y_i = ct_i/a$. This is a semi-functional key. It is a normal key, if $T \xleftarrow{R} \mathbb{G}_p$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two messages of the equal length m_1, m_2 and two access policies $\mathcal{T}_1, \mathcal{T}_2$. $\mathcal{B}$ chooses $\beta = \{0, 1\}$ randomly and does the following.

1) $\mathcal{B}$ randomly chooses $s \in Z_N$ and $R_0' \in \mathbb{G}_r$. It also chooses $s_{i,j} \in Z_N$ and $R_{i,j}' \in \mathbb{G}_r$ at random.
2) $\mathcal{B}$ computes $\tilde{C} = m_\beta \cdot \hat{e}(g_p^{\omega}, X_1X_2),\ C_0 = X_1X_2 \cdot R_0^s \cdot R_0'$ and

$$C_{i,j} = \begin{cases} (X_1X_2)^a \cdot R_1^s \cdot R_{i,j}', \text{ if } v_{i,j} \in W_i; \\ (X_1X_2)^{a \cdot r_{i,j}} \cdot R_1^{s_{i,j}} \cdot R_{i,j}', \text{ otherwise.} \end{cases}$$

   where $1 \le i \le n$ and $1 \le j \le l,\ r_{i,j} = s_{i,j}/s$.
3) $\mathcal{B}$ sets the challenge ciphertext as $CT = (\vec{x}, \tilde{C}, C_0, \{C_{i,j}\}_{1 \le i \le n, 1 \le j \le l})$ and sends to $\mathcal{A}$.

Let $X_1X_2 = g_p^s g_q^{x_0}$, then $\tilde{C} = m_\beta \cdot \hat{e}(g_p, g_p)^{\omega s}, C_0 = g_p^s g_q^{x_0} \cdot R_0^s \cdot R_0'$ and

$$C_{i,j} = \begin{cases} g_p^{as} g_q^{x_{i,j}} \cdot R_1^s \cdot R_{i,j}', \text{ if } v_{i,j} \in W_i; \\ g_p^{as_{i,j}} g_q^{x_{i,j}} \cdot R_1^{s_{i,j}} \cdot R_{i,j}', \text{ otherwise.} \end{cases}$$

where $x_{i,j} = ax_0$ when $v_{i,j} \in W_i$, otherwise, $x_{i,j} = ax_0 r_{i,j}$, and $1 \le i \le n,\ 1 \le j \le l$. Hence, if $T \xleftarrow{R} \mathbb{G}_p$ then $\mathcal{B}$ simulates $\mathbf{Game_{2,k-1}}$. If $T \xleftarrow{R} \mathbb{G}_p \times \mathbb{G}_q$, $\mathcal{B}$ simulates $\mathbf{Game_{2,k}}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. ■

Lemma 3: Suppose that $\mathcal{G}$ satisfies Assumption 3. Then $\mathbf{Game_{2,\zeta}}$ and $\mathbf{Game_3}$ are indistinguishable.

Proof: Suppose there is an algorithm $\mathcal{A}$ that distinguishes $\mathbf{Game_{2,\zeta}}$ and $\mathbf{Game_3}$, then we will set up an algorithm $\mathcal{B}$ that has non-negligible advantage to break Assumption 3. $\mathcal{B}$ obtains $g_p,\ g_p^{\omega} X_2,\ g_p^s Y_2,\ Z_1Z_2,\ g_r,\ T$ and simulates $\mathbf{Game_{2,\zeta}}$ and $\mathbf{Game_3}$ with $\mathcal{A}$.

$\mathcal{B}$ randomly chooses $a,\ a_1,\ a_2,\ \omega \in Z_N$. It then sets $R_0 = g_r^{a_1},\ R_1 = g_r^{a_2},\ A_0 = g_p \cdot R_0,\ A_1 = g_p^a \cdot R_1$, and sends $\mathcal{A}$ the public key

$$PK = (A_0, A_1, g_r, Y = \hat{e}(g_p, g_p^{\omega} X_2) = \hat{e}(g_p, g_p)^{\omega}).$$

$\mathcal{B}$ creates a semi-functional key by randomly choosing $t_i \in Z_N$ uniformly for $i = 1, 2, \ldots, n$, and sets $t = \Sigma_{i=1}^n t_i$. Then it computes

$$D_0 = (Z_1Z_2)^{\omega - t},\ D_i = (Z_1Z_2)^{t_i/a}.$$

Note that this is a distributed semi-functional key.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two messages of the equal length m_1, m_2 and two access policies $\mathcal{T}_1, \mathcal{T}_2$. $\mathcal{B}$ chooses $\beta = \{0, 1\}$ randomly and does the following.

1) $\mathcal{B}$ randomly chooses $s \in Z_N$ and $R_0' \in \mathbb{G}_r$. It also chooses $s_{i,j} \in Z_N$ and $R_{i,j}' \in \mathbb{G}_r$ at random.
2) $\mathcal{B}$ computes $\tilde{C} = m_\beta \cdot T,\ C_0 = g_p^s Y_2 \cdot R_0^s \cdot R_0'$ and

$$C_{i,j} = \begin{cases} (g_p^s Y_2)^a \cdot R_1^s \cdot R_{i,j}', \text{ if } v_{i,j} \in W_i; \\ (g_p^s Y_2)^{a \cdot r_{i,j}} \cdot R_1^{s_{i,j}} \cdot R_{i,j}', \text{ otherwise.} \end{cases}$$

   where $1 \le i \le n$ and $1 \le j \le l,\ r_{i,j} = s_{i,j}/s$.
3) $\mathcal{B}$ sets the challenge ciphertext as $CT = (\vec{x}, \tilde{C}, C_0, \{C_{i,j}\}_{1 \le i \le n, 1 \le j \le l})$ and sends to $\mathcal{A}$.

Let $g_p^s Y_2 = g_p^s g_q^{x_0}$, then $\tilde{C} = m_\beta \cdot T,\ C_0 = g_p^s g_q^{x_0} \cdot R_0^s \cdot R_0'$ and

$$C_{i,j} = \begin{cases} g_p^{as} g_q^{x_{i,j}} \cdot R_1^s \cdot R_{i,j}', \text{ if } v_{i,j} \in W_i; \\ g_p^{as_{i,j}} g_q^{x_{i,j}} \cdot R_1^{s_{i,j}} \cdot R_{i,j}', \text{ otherwise.} \end{cases}$$

where $x_{i,j} = ax_0$ when $v_{i,j} \in W_i$, otherwise, $x_{i,j} = ax_0 r_{i,j}$, and $1 \le i \le n,\ 1 \le j \le l$.

If $T \xleftarrow{R} \hat{e}(g_p, g_p)^{\omega s}$, this is a properly distributed semi-functional encryption of m_β and $\mathcal{B}$ simulates $\mathbf{Game_{2,\zeta}}$. If $T \xleftarrow{R} \mathbb{G}_T$, this is a properly distributed semi-functional encryption of a random message in $\mathbf{G}_T$ and $\mathcal{B}$ simulates $\mathbf{Game_3}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. ■

Lemma 4: Suppose that $\mathcal{G}$ satisfies Assumption 4. Then $\mathbf{Game_3}$ and $\mathbf{Game_4}$ are indistinguishable.

Proof: Suppose there is an algorithm $\mathcal{A}$ that distinguishes $\mathbf{Game_3}$ and $\mathbf{Game_4}$, then we will set up an algorithm $\mathcal{B}$ that has non-negligible advantage to break Assumption 4. $\mathcal{B}$ obtains $g_pR_0, g_p^a R_1, g_pQ_1, g_p, g_r, T$ and simulates $\mathbf{Game_3}$ and $\mathbf{Game_4}$ with $\mathcal{A}$.

$\mathcal{B}$ sets $A_0 = g_pR_0,\ A_1 = g_p^a R_1$, and sends $\mathcal{A}$ the public key

$$PK = (A_0, A_1, g_r, Y = \hat{e}(g_p, g_p)^{\omega}).$$

$\mathcal{B}$ creates a semi-functional key by randomly choosing $t_i \in Z_N$ uniformly for $i = 1, 2, \ldots, n$, and sets $t = \Sigma_{i=1}^n t_i$. Then it computes $D_0 = (g_pQ_1)^{\omega - t},\ D_i = (g_pQ_1)^{t_i/a}$. If let $Q_1 =$

$g_q^c, c \in Z_N$, then

$$D_0 = g_p^{\omega - t} g_q^{y_0},\ D_i = g_p^{t_i/a} g_q^{y_i},$$

where $y_0 = c(\omega - t),\ y_i = ct_i/a$. Note that this is a semi-functional key.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two messages of the equal length m_1, m_2 and two access policies $\mathcal{T}_1, \mathcal{T}_2$. $\mathcal{B}$ chooses $\beta = \{0, 1\}$ randomly and does the following.

1) $\mathcal{B}$ randomly chooses $s, \tilde{x}_{i,j} \in Z_N$ and $R'_0 \in \mathbb{G}_r$. It also chooses $s_{i,j} \in Z_N$ and $R'_{i,j} \in \mathbb{G}_r$ at random.
2) $\mathcal{B}$ computes $\tilde{C} \xleftarrow{R} \mathbb{G}_T,\ C_0 = g_p^s g_q^{x_0} \cdot R_0^s \cdot R'_0$ and

$$C_{i,j} = \begin{cases} T^s \cdot R'_{i,j} \cdot g_q^{\tilde{x}_{i,j}}, \text{ if } v_{i,j} \in W_i; \\ T^{s_{i,j}} \cdot R'_{i,j} \cdot g_q^{\tilde{x}_{i,j}}, \text{ otherwise.} \end{cases}$$

   where $1 \le i \le n$ and $1 \le j \le l$.
3) $\mathcal{B}$ sets the challenge ciphertext as $CT = (\vec{x}, \tilde{C}, C_0, \{C_{i,j}\}_{1 \le i \le n, 1 \le j \le l})$ and sends to $\mathcal{A}$.

Let $T = g_p^a QR$, then $Q = g_q^{c'}, c' \in Z_N$, we have $\tilde{C} \xleftarrow{R} \mathbb{G}_T,\ C_0 = g_p^s g_q^{x_0} \cdot R_0^s \cdot R'_0$ and

$$C_{i,j} = \begin{cases} g_p^{as} g_q^{x_{i,j}} \cdot R^s \cdot R'_{i,j}, \text{ if } v_{i,j} \in W_i; \\ g_p^{as_{i,j}} g_q^{x_{i,j}} \cdot R^{s_{i,j}} \cdot R'_{i,j}, \text{ otherwise.} \end{cases}$$

where $x_{i,j} = c's + \tilde{x}_{i,j}$ when $v_{i,j} \in W_i$, otherwise, $x_{i,j} = c's_{i,j} + \tilde{x}_{i,j}$ and $1 \le i \le n,\ 1 \le j \le l$. If $T = g_p^a QR$, then $\mathcal{B}$ simulates $\mathbf{Game_3}$. If $T \xleftarrow{R} \mathbb{G}_T$, then $\mathcal{B}$ simulates $\mathbf{Game_4}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. ■

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2005, pp. 457–473.

[2] H. Li, Y. Yang, Y. Dai, S. Yu, and Y. Xiang, "Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data," *IEEE Trans. Cloud Comput.*, vol. 9, no. 2, pp. 484–494, Apr.–Jun. 2020.

[3] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 97–109, Jan.-Mar. 2018.

[4] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, 2007, pp. 321–334.

[7] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[8] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight fine-grained search over encrypted data in fog computing," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 772–785, Sep./Oct. 1, 2019.

[9] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 4, pp. 870–885, Apr. 2019.

[10] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in internet of things with privacy preserving: Challenges, solutions, and opportunities," *IEEE Netw.*, vol. 32, no. 6, pp. 144–151, Nov./Dec. 2018.

[11] D. Di Francesco Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable access control systems," *Comput. Secur.*, vol. 84, pp. 93–119, 2019.

[12] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale internet of things data storage and protection," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 762–771, 1 Sep./Oct. 2019.

[13] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2017, pp. 618–623.

[14] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for IoT," *IEEE Access*, vol. 7, pp. 38431–38441, 2019.

[15] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proc. IFIP Int. Conf. Distrib. Appl. Interoperable Syst.*, 2017, pp. 206–220.

[16] Y. Michalevsky and M. Joye, "Decentralized policy-hiding ABE with receiver privacy," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2018, pp. 548–567. doi. 10.1007/978-3-319-98989-1_27.

[17] J. Lai, R. H. Deng, and Y. Li, "Fully secure cipertext-policy hiding CP-ABE," in *Proc. Int. Conf. Inf. Secur. Pract. Experience*, 2011, pp. 24–39.

[18] H. Wang, J. Ning, X. Huang, G. Wei, G. S. Poh, and X. Liu, "Secure fine-grained encrypted keyword search for e-healthcare cloud," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2019.2916569.

[19] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *Proc. Int. Conf. Provable Secur.*, 2016, pp. 19–38.

[20] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, and X. Shen, "An efficient and fine-grained big data access control scheme with privacy-preserving policy," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 563–571, Apr. 2017.

[21] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2008, pp. 146–162.

[22] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. Appl. Cryptography Netw. Secur.*, 2008, pp. 111–129.

[23] A. Wu, Y. Zhang, X. Zheng, R. Guo, Q. Zhao, and D. Zheng, "Efficient and privacy-preserving traceable attribute-based encryption in blockchain," *Ann. Telecommun.*, vol. 74, no. 7–8, pp. 401–411, 2019.

[24] S. Xu, Y. Li, R. Deng, Y. Zhang, X. Luo, and X. Liu, "Lightweight and expressive fine-grained access control for healthcare Internet-of-Things," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2019.2936481.

[25] G. Xu, H. Li, H. Ren, K. Yang, and R. H. Deng, "Data security issues in deep learning: Attacks, countermeasures and opportunities," *IEEE Commun. Mag.*, vol. 57, no. 11, pp. 116–122, Nov. 2019.

[26] W. Jiang, H. Li, G. Xu, M. Wen, G. Dong, and X. Lin, "PTAS: Privacy-preserving thin-client authentication scheme in blockchain-based PKI," *Future Gener. Comput. Syst.*, vol. 96, no. 7, pp. 185–195, Jul. 2019.

[27] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2019.2945367.

[28] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 911–926, 2020.

[29] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "FairAccess: A new blockchain-based access control framework for the Internet of Things," *Secur. Commun. Netw.*, vol. 9, pp. 5943–5964, 2016.

[30] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. Ait Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Comput. Netw.*, vol. 112, pp. 237–262, 2017.

[31] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, 2015, pp. 180–184.

[32] K. Delmolino, M. Arnett, A. Kosba, A. Miller, and E. Shi, "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2016, pp. 79–94.

[33] C. Lin, D. He, X. Huang, K. R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," *J. Netw. Comput. Appl.*, vol. 116, pp. 42–52, 2018.

[34] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "Enforcing private data usage control with blockchain and attested off-chain contract execution," *CoRR, abs/1904.07275*, 2019. [Online]. Available: https://arxiv.org/abs/1904.07275

[35] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data*, 2016, pp. 25–30.

[36] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, 2018.
[37] O. Alphand *et al.*, "IoTChain: A blockchain security architecture for the internet of things," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2018, pp. 1–6.
[38] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 1184–1195, Apr. 2018.
[39] O. Novo, "Scalable access management in IoT using blockchain: A performance evaluation," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4694–4701, Jun. 2019.
[40] R. Xu, Y. Chen, E. Blasch, and G. Chen, "BlendCAC: A blockchain-enabled decentralized capability-based access control for IoTs," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Physical Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2018, pp. 1027–1034.
[41] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2019.
[42] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 1, 2018.
[43] Y. Yuan and F. Wang, "Blockchain and cryptocurrencies: Model, techniques, and applications," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 48, no. 9, pp. 1421–1428, Sep. 2018.
[44] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[45] L. Wang, X. Shen, J. Li, J. Shao, and Y. Yang, "Cryptographic primitives in blockchains," *J. Netw. Comput. Appl.*, vol. 127, pp. 43–58, 2019.
[46] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Gener. Comput. Syst.*, vol. 107, pp. 760–769, Jun. 2020.
[47] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-Work vs. BFT replication," in *Proc. Int. Workshop Open Problems Netw. Secur.*, 2016, pp. 112–125.
[48] S. Gao, T. Yu, J. Zhu, and W. Cai, "T-PBFT: An eigentrust-based practical byzantine fault tolerance consensus algorithm," *China Commun.*, vol. 16, pp. 111–123, 2019.
[49] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *CoRR abs/1904.04098*, 2019. [Online]. Available: https://arxiv.org/abs/1904.04098
[50] N. Szabo, "Smart Contracts," 1994. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html
[51] V. Buterin, "A next-generation smart contract and decentralized application platform," Tech. Rep., 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper
[52] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. Theory Cryptography Conf.*, Amsterdam, The Netherlands, 2007, pp. 535–554.
[53] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[54] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3854–3865, Apr. 2019.
[55] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, 1999, pp. 173–186.
[56] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proc. 7th ACM Symp. Inf., Comput. Commun. Secur.*, 2012, pp. 18–19.
[57] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2010, pp. 62–91.
[58] A. De Caro, V. Iovino, and G. Persiano, "Fully secure anonymous hibe and secret-key anonymous IBE with short ciphertexts," in *Proc. Int. Conf. Pairing-Based Cryptography*, 2010, pp. 347–366.
[59] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 1, pp. 35–45, Jan. 2016.
[60] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun.*, 2011, pp. 850–855.

Sheng Gao received the B.S. degree in information and computation science from Xi'an University of Posts and Telecommunications, Xian, China, in 2009, and the Ph.D. degree in computer science and technology from Xidian University, Xi'an, China, in 2014. He is currently an Associate Professor with the School of Information, Central University of Finance and Economics, Beijing, China. He has authored or coauthored more than 30 research papers in refereed international journals and conferences. His current research interests include data security, privacy computing, and blockchain technology.

Guirong Piao received the B.S. degree in software engineering from Beijing Union University, Beijing, China, in 2018. She is currently working toward the master's degree in software engineering with Central University of Finance and Economics, Beijing, China. Her current research interests include information security and blockchain.

Jianming Zhu received the M.S. degree in computer application from Taiyuan University of Technology, Taiyuan, China, in 1998, and the Ph.D. degree in computer application technology from Xidian University, Xi'an, China, in 2004. He is currently a Professor with the School of Information, Central University of Finance and Economics, Beijing, China. He has authored or coauthored 5 books and more than 100 research papers in refereed international journals and conferences. His research interests include financial information security, wireless network security, and blockchain.

Xindi Ma received the B.S. degree from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2013 and the Ph.D. degree in computer science from Xidian University, Xi'an, China, in 2018. He is currently a Postdoctor with the School of Cyber Engineering, Xidian University. His current research interests include privacy computing, recommender system and machine learning with focus on security and privacy issues.

Jianfeng Ma received the B.S. degree in mathematics from Shaanxi Normal University, Xi'an, China, in 1985, and the M.S. and Ph.D. degrees in computer software and communications engineering from Xidian University, Xi'an, China, in 1988 and 1995, respectively. He is currently a Professor and Ph.D. supervisor with the School of Cyber Engineering, Xidian University. From 1999 to 2001, he was a Research Fellow with Nanyang Technological University of Singapore. He has authored or coauthored more than 400 papers in refereed international journals and conferences. His current research interests include cryptography, wireless and mobile security, data security, and privacy protection.