IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 29, 2020, accepted September 12, 2020, date of publication September 18, 2020,
date of current version September 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025060

A Blockchain-Assisted Verifiable Outsourced Attribute-Based Signcryption Scheme for EHRs Sharing in the Cloud

XIAODONG YANG[1], (Member, IEEE), TING LI[1], WANTING XI[1], AIJIA CHEN[1], AND CAIFEN WANG[1,2]

[1]College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
[2]College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China

Corresponding author: Xiaodong Yang (y200888@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662069 and Grant 61562077, in part by the China Postdoctoral Science Foundation under Grant 2017M610817, in part by the Science and Technology Project of Lanzhou City of China under Grant 2013-4-22, and in part by the Foundation for Excellent Young Teachers by Northwest Normal University under Grant NWNU-LKQN-14-7.

ABSTRACT The sharing of electronic health records (EHRs) has shown great advantages in the accurate treatment of patients and the development of medical institutions. However, it is easy to cause some security problems in the process of medical data sharing. Generally, after a patient's EHRs are generated by different medical institutions, they are outsourced to the cloud server (CS) by the authorized medical institutions for storage, which causes the patient to lose control of EHRs. Moreover, malicious medical institutions and semi-trusted cloud servers may collude to tamper with EHRs to seek benefits, which threatens the integrity of EHRs. Therefore, we propose a blockchain-assisted verifiable outsourced attribute-based signcryption scheme (BVOABSC) which realizes the secure sharing of EHRs in a multi-authority cloud storge environment. Firstly, we use the attribute-based signcryption algorithm to realize the confidentiality and unforgeability of the EHRs and protect the privacy of the signer. Secondly, it greatly reduces the computational burden of users by using verifiable outsourcing computation mechanism. Most of the designcryption calculation is performed by the cloud server, and the correctness of the generated partial designcryption ciphertext is verified by users. Furthermore, we use blockchain technology to protect outsourced EHRs from tampering by illegal users. Specifically, each operation on outsourced EHRs is stored as a transaction on the public blockchain, which ensures that EHRs cannot be modified. At the same time, the auditor can verify the integrity of the outsourced EHRs by checking the corresponding transactions. In addition, the smart contract created by the patient can solve the problems in cloud storage, such as tampering EHRs and returning incorrect results. Finally, security analysis and performance evaluation show that the proposed BVOABSC scheme satisfies stronger security and higher efficiency than similar schemes.

INDEX TERMS Attribute-based signcryption, blockchain, cloud storage, electronic health records, verifiable outsourced calculation.

I. INTRODUCTION

With the rapid development of the Internet and the introduction of our country's new medical reform policies, the number of electronic health records (EHRs) has increased dramatically. Taking the patients' health throughout life as the core, EHRs realize the dynamic collection of multi-channel information and meet the information resources required by patients for self-care. EHRs have great application value in the fields of hospital development, clinical services, clinical research, and patient health. For example, a variety of standardized templates and auxiliary tools provided by EHRs can free doctors from the heavy medical record writing work and assistant them to focus on the diagnosis and treatment of patients [1].

A large number of EHRs are generated after the patients undergo medical examinations, which are basically kept in separate hospitals. As a result, EHRs face a problem

The associate editor coordinating the review of this manuscript and approving it for publication was SK Hafizul Islam.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

called information island [2]. It takes a lot of resources and time to transfer EHRs among the databases of different medical institutions when patients and doctors need to use medical data. Fortunately, EHRs sharing among medical institutions can solve these problems. It can provide more historical reference materials for doctors' decision-making and improve the correct rate of diagnosis of the patient's diseases. However, EHRs sharing has many problems. First, the EHRs model used by various medical institutions is quite different and the EHRs format is not unified. Second, users need to verify their identity and audit their access rights to access EHRs, which leads to a long access cycle. Third, the huge EHRs system involves users' personal privacy, leading to the problems such as low storage security, data leakage, and data tampering. Therefore, how to solve the problems above is a research hotspot in the medical industry.

With the rise of cloud storage, EHRs have been significantly developed in recent years. In the cloud storage model, cheap computing and huge capacity attract more and more users to outsource EHRs to cloud servers to save local storage and maintenance costs. To improve the capability sharing of EHRs, cloud storage technology is used to build a regional medical information sharing platform, which integrates different medical institutions' systems comprehensively. Zhang *et al.* [3] firstly elaborated on the security requirements of an electronic medical record system based on cloud computing, and proposed a cloud-based system to achieve patient-centric medical services. An attribute-based medical data sharing system was introduced by [4], which achieves flexible access control for users to medical data stored in the cloud. Hua *et al.* [5] put forward that the encrypted medical data should be outsourced and stored in the cloud, which provides accurate medical services for patients and protects their privacy. Biswas *et al.* [6] considered that medical data are stored on a three-tier medical cloud, which facilitates users to access EHRs while ensuring the integrity of EHRs. A medical service model in the cloud (m-health) was constructed by scheme [7], which is an architecture based on distributed events and includes interoperable services with CCR standards. Khan *et al.* [8] pointed out the problems of multi-party data sharing in the cloud environment, and constructed a scheme that allows data owners to store their data safely in an untrusted cloud environment. These schemes above use the cloud to store and share medical data, which improves the efficiency of storage, retrieval, and sharing to a certain extent. However, these schemes also have the common problem, which is highly dependent on cloud service providers. If some targeted attacks are carried out on cloud service providers, data leakage is likely to occur [9]. For example, the vast majority of network devices are directly allowed to be accessed on the public network under the drive of profits, which leads to hackers to steal sensitive medical data through technical means and conduct illegal transactions to make huge profits. In addition, none of the schemes above consider that cloud servers will collude with doctors to tamper with the outsourced EHRs. If this behavior occurs, it is difficult to detect.

Blockchain technology has the characteristics of decentralization, immutability, and traceability, which can solve the problems of tampering, forgery, and leakage of EHRs. In addition, it can promote data sharing in the production practice of the medical and health field, and protect personal privacy and data security. Bera *et al.* designed a blockchain-envisioned secure data delivery and collection scheme to provide in-depth challenges [10]. Jiang *et al.* proposed a health information exchange platform based on blockchain. Offline storage and online verification are used to process data, so that the patient's privacy information is protected [11]. Azaria *et al.* [12] proposed a blockchain-based record management application for processing EHRs, solving the fragmentation of EHRs and the privacy protection of patients, and providing participants with data authorization, data auditing and data sharing. These schemes use blockchain technology to perfectly solve the problems of the concentration of medical data storage in the cloud storage model, and satisfy the need to complete authorization checks and data verification through a third party. Meanwhile, the security and privacy of medical data are increased, and the efficiency of data sharing is improved. However, all transactions must be disclosed to the nodes in the blockchain to reach a consensus in the blockchain network, which will leak the information in the transaction. How to protect the privacy of transaction information has become an important topic to promote the development of blockchain technology.

In order to solve the problem of privacy leakage of transaction information on the blockchain, scholars at home and abroad proposed that cryptography knowledge can be used to protect data on the blockchain. Essentially, EHRs can be regarded as a high-value privacy asset, and the use of blockchain-related cryptography technology can realize real-time supervision of EHRs authorization process. Peterson *et al.* [13] proposed a blockchain-based method to share medical data. This scheme requires all participants to share these data in a pre-agreed structure, which improves the efficiency of medical data utilization. However, it does not provide a general access control strategy, which may lead to the leakage of medical data. Dagher *et al.* [14] proposed an Ancile framework that uses the Ethereum platform to transfer the ownership and control of EHRs to the data owner, develops and utilizes a variety of smart contracts and uses proxy re-encryption technology to further protect the privacy of EHRs. However, the computational overhead is relatively large, and the system efficiency is low. Guo *et al.* [15] designed a system of the EHRs based on the blockchain, which ensures that EHRs cannot be tampered with or forged, and protects the privacy of patients simultaneously. Roehrs *et al.* [16] established a patient-centric medical architecture model by using blockchain, which integrates medical data distributed in different medical institutions into one view, and stores the data in the blockchain. However, limited by the storage space of nodes and the

network, the blockchain cannot store a large amount of EMRs. Aste *et al.* [17] believed that the blockchain system has shortcomings such as high energy consumption, slow business processing, and difficulty in unified management. These disadvantages have become obstacles to the development of the blockchain system in actual production.

In response to the limited storage capacity of the blockchain, Zyskind [18] combined blockchain and cloud storage technology to build a private data management platform that can ensure that the participants can still control their own data after the data are uploaded. Through the use of blockchain to control visitors' access without trusting a third party, this platform provides participants with functions such as storing and sharing data. Xia *et al.* [19] designed a blockchain-based data sharing framework to solve the problem of sharing medical data among large medical databases in a trustless environment. However, the communication overhead of users sharing EHRs is relatively high. Esposito *et al.* [20] pointed out that medical data involves the privacy of patients, and blockchain can be used to protect the privacy of patients. Liu *et al.* [21] raised a medical data sharing scheme, which not only realizes the safe sharing of medical data, but also protects the identity privacy of users. This scheme stores encrypted medical data in the cloud, and then writes the index value of the data in the cloud to the blockchain. A cloud-assisted electronic health system based on blockchain was presented by [9]. Storing EHRs in blockchain transactions ensures that EHRs will not be changed and realizes the safe sharing of EHRs. However, storing a large number of EHRs in the block will reduce the efficiency of the system. Therefore, how to store and share EHRs safely and efficiently are challenges we face now.

A. OUR CONTRIBUTIONS

- We combine the attribute-based signcryption (ABSC) algorithm and blockchain technology to design a secure EHRs sharing scheme called BVOABSC. Our scheme can ensure the confidentiality, correctness and unforgeability of EHRs without relying on any trusted entities.
- Our scheme can implement flexible access control to EHRs stored in the cloud, which facilitates CS to verify the users' identity while protecting their privacy. In addition, the proposed scheme can provide anonymous authentication of the source of EHRs due to the characteristics of ABS, which ensures the EHRs are uploaded by authorized users and protects the privacy of signers.
- The proposed scheme ensures that the size of the ciphertext is constant, which means its size has nothing to do with the number of attributes. Hence, our scheme reduces bandwidth utilization and storage overhead, and it is suitable for EHRs sharing environments.
- Our scheme outsources most of the decryption operations to CS, which can reduce the users' computational burden. Moreover, the partial ciphertext generated by CS allows the user to verify its correctness, which means that our scheme satisfies the requirement of verifiability.
- Our scheme can resist malicious doctors and CS colluding to tamper with outsourced EHRs. Even if malicious doctors collude with cloud servers, their computing power cannot bifurcate the blockchain since it is immutable. Besides, the EHRs of each patient are usually generated by multiple doctors, so our scheme adds a time stamp to the EHRs generated by each doctor. Therefore, the safety of EHRs can be guaranteed.
- The proposed scheme is proved to have strong security in the standard model. Compared with similar schemes, it has higher computing performance and lower communication overhead.

B. ORGANIZATION

The rest of this article is organized as follows. In Section II, we review the related work. Section III introduces some preliminaries, such as bilinear maps, computational Diffie-Hellman assumption, the augmented multi-sequence of exponents computational Diffe-Hellman problem, and aggregation algorithm. Sections IV describes the system architecture, the structure of blockchain, security model, the construction of BVOABSC, and correctness. Afterwards, we analyze the security and estimate the performance of the proposed scheme in Sections V and VI, respectively. Finally, we summarize the conclusion in Section VII.

II. RELATED WORK

This section mainly focuses on the EHRs protection scheme based on cryptography, which can realize the safe storage and sharing of EHRs. In the face of massive EHRs, such as personal information, medical records, and drug records, using cloud servers to manage and store these information can free up local storage space and improve management efficiency. However, the security of the patient's sensitive data and the privacy of the users' identity are not protected [22], [23]. Therefore, it is necessary to not only consider reasonable, safe and effective access control, but also protect the privacy of users. The attribute-based encryption (ABE) scheme [24], [25] provide secure access control for EHRs, and attribute-based signature (ABS) realizes the privacy protection of the signer's identity. Nevertheless, both ABE and ABS only separately provide guarantees for the confidentiality or unforgeability of messages. The ABSC cryptographic mechanism combines the characteristics of ABE and ABS, which can simultaneously realize data confidentiality and unforgeability. Applying the ABSC scheme to the EHRs system has a stronger security. Further expansion of outsourcing functions on the basis of ABSC can reduce the amount of local calculations for users and greatly improve the practicality of attribute-based cryptosystems. However, there is a possibility for EHRs to be tampered with in this case. The birth of blockchain technology provides a decentralized and trusted platform. The non-tamperable and traceable characteristics of the blockchain can better solve the problems of EHRs management, sharing, transaction and

auditing. The above discussion specifically elaborates from three aspects: attribute-based signcryption, attribute-based outsourcing signcryption, and the application of blockchain in EHRs.

A. ATTRIBUTE BASED SIGNCRYPTION

The idea of ABE was first proposed by Sahai and Waters [26], which is a one-to-many encryption mechanism. In this scheme, attributes are used to identify the user's identity information, and the users can encrypt the plaintext message according to a certain access control strategy to achieve fine-grained data access control. As a new encryption mechanism, [27] can solve the shortcomings of the identity-based encryption scheme of a single communication mode and the system taking up a lot of system resources. Applying the ABE scheme to the EHRs system effectively realizes the access control to the EHRs and ensures the confidentiality of the EHRs. Liang *et al.* [28] proposed an attribute-oriented authentication scheme that can help users establish social relationships and share health information with other trusted users. Lu *et al.* [29] introduced a user-centric access control scheme, and allowed medical users to decide who can participate in the calculation to assist in the processing of EHRs. Liu *et al.* [30] presented an online/offline ABE scheme in which the data owners of EHR performed most of the encryption calculations in the offline encryption phase. When the encryption party knows the access policy and EHRs in the online encryption phase, the owner can quickly integrate the information to generate the final ciphertext.

Attribute-Based Signature (ABS) was derived from the fuzzy identity signature scheme proposed by Yang *et al.* [31]. Maji *et al.* [32] proposed the primitives of attribute-based signatures for the first time and constructed an ABS scheme that supports effective privacy protection and resists collusion attacks. Subsequently, domestic and foreign scholars have done a lot of researches on attribute-based signature, and proposed some applications of ABS in terms of function extension and security improvement. Shahandashti and Safavi-Naini [33] raised a threshold attribute-based signature scheme that can be used for anonymous authentication, and proved its unforgeability based on the computational Diffie-Hellman assumption. Okamoto and Takashima [34] considered the limited number of attributes given to signers by a single trusted authority, and proposed a multi-authority ABS scheme. Tang *et al.* [35] put forward an efficient authentication scheme for EHRs that implements fine-grained access control in a cloud computing environment. Liu *et al.* [36] proposed an online/offline attribute-based signature scheme, which realizes data integrity and the privacy protection of signer identity with low local computing cost.

Zheng [37] first introduced the concept of signcryption. The design core of the signcryption scheme is to realize both encryption and signature in an effective step. A reasonable signcryption scheme can achieve a higher level of security. Inspired by [37], Gagné *et al.* [38] proposed an ABSC scheme by using a threshold access strategy, in which users must determine their access structure before the system establishment. Mandal *et al.* presented a new three-factor signcryption-based user access control scheme in [39]. Hu *et al.* [40] constructed a new secure fuzzy ABSC scheme. This scheme can perform data encryption, digital signature and access control on the patient's medical information in the body area network. Based on the linear secret sharing access structure, Rao and Dutta [41] raised an efficient KP-ABSC scheme with a constant ciphertext length. Liu *et al.* [42] proposed a CP-ABSC scheme, which can realize safe data sharing in the personal health record system, solve the problem of fine-grained access control, and prove the safety of the scheme. However, Rao [43] pointed out some errors and problems in the scheme [42], which cannot resist selected ciphertext attacks, and cannot satisfy the public verifiability.

B. OUTSOURCING ATTRIBUTE BASED SIGNCRYPTION

The goal of secure outsourcing computing technology is to blind the user's sensitive information by some means, so that the service party can only access the blind information and bear the user's computing overhead when the original data cannot be explored. Although there are many research results on attribute-based cryptosystems, most of the schemes have not yet been put into practice. The reason is that a large number of expensive computing operations in attribute-based cryptosystems are considered to be the biggest obstacle. If the computational overhead can be properly reduced, the practicability of the attribute-based cryptosystem can be greatly improved. Green *et al.* [44] used outsourcing technology in the decryption process, transferring a large number of bilinear pairing operations to the outsourced computing party for execution. Zhang *et al.* [45] presented a blockchain-based PDP scheme in cloud computing and an outsourcing computing protocol suitable for fog computing, which can illustrate the application of BCPay. In order to achieve overall security and fair payment for outsourcing services without relying on any third party, scheme [46] introduced the blockchain-based fair payment framework BPay for outsourcing services in cloud computing. Deng *et al.* [47] introduced a verifiable outsourcing ABSC scheme that enables users to verify the correctness of the generated partial ciphertext. Liu *et al.* [48] presented a KP-ABSC scheme, which entrusts a trusted server to complete outsourcing and decryption. However, since the user shares the secret key with the server, the server can obtain part of the decrypted ciphertext. A multi-authorized attribute-based signcryption scheme was proposed by [49], which protects users' attribute privacy while outsourcing to the cloud server for partial decryption. However, this scheme still has high computational overhead. The attribute-based signcryption scheme based on the ciphertext strategy can be applied to the EHRs sharing system [50]. A large number of decryption calculations are outsourced to the cloud server, which relieves the users' calculation pressure and has high practicability.

C. APPLICATION of BLOCKCHAIN in EHRs

Blockchain is a decentralized, non-tamperable, and credible distributed ledger that provides a safe, stable, transparent, and auditable way of recording transactions and information interaction. We can solve the problems of access control and data authorization in traditional medical data protection model by using blockchain technology. Ariel Ekblaw *et al.* proposed the Medrec prototype [51], which uses Ethereum to record medical data and allows patients to have control over personal medical data. The work records of medical researchers on the public chain will be given back to the corresponding digital currency to encourage medical researchers to continue to use the platform. Wang and Song [52] designed an attribute-based/identity-based combined encryption and signature (C-AB/IB-ES) scheme to ensure the data integrity and immutability of EHRs. However, this scheme incurs a lot of computational overhead for users. On the basis of [52], Yang *et al.* [53] introduced an attribute-based outsourcing decryption mechanism, which greatly reduces the users' computational overhead. In addition, the use of ABE and ABS improves the computational efficiency of the EHRs system. Scheme [54] combines the ABSC with blockchain technology to realize the secure sharing of cloud data. Compared with the "encryption then signature" adopted by Yang *et al.* [53], there are higher savings in computing overhead and communication costs. However, the computational cost of data encryption and authentication still needs to be improved.

In this article, we design a blockchain-based verifiable outsourcing attribute signcryption scheme to meet the confidentiality and unforgeability of the EHRs stored in the cloud, and reduce the user's computational overhead for decryption. Moreover, the transaction information stored on the blockchain ensures that the EHRs cannot be tampered with, and the smart contract ensures that the user and the cloud server honestly execute the agreement.

III. PRELIMINARIES

In this section, we review the notations and definitions related to the proposed scheme.

A. BILINEAR MAPS

Define a pair of multiplicative groups (G, G_T) with prime order p, where $e : G \times G \rightarrow G_T$ is an effective computable map and g, g_1 are generators of G. For any $(g, g_1) \in G \times G$, we can get $e(g^a, {g_1}^b) = e(g, g_1)^{ab}$ for $a, b \in Z_p$ and $e(g, g_1) \neq 1$ for $g, g_1 \neq 1$ [55].

B. COMPLEXITY ASSUMPTION

1) COMPUTATIONAL DIFFIE-HELLMAN ASSUMPTION

Computational Diffie-Hellman (CDH) Assumption means that, given the input (g, g^a, g^b) for $a, b \overset{R}{\leftarrow} Z_p$, and then calculate g^{ab}.

Definition 1 (CDH Assumption): For the adversary $\mathcal{A}_1$ of arbitrary probability polynomial time, the probability of solving the CDH problem is negligible, which is formally expressed as $\Pr[\mathcal{A}(g, g^a, g^b) = g^{ab} | a, b \overset{R}{\leftarrow} Z_p{}^*] = negl(\lambda)$ [56].

2) THE AUGMENTED MULTI-SEQUENCE OF EXPONE-NTS COMPUTATIONAL DIFFE-HELLMAN PROBLEM

The $(\tilde{l}, \tilde{m}, \tilde{t})$ augmented multi-sequence of exponents computational Diffe-Hellman $((\tilde{l}, \tilde{m}, \tilde{t}) - aMSE - CDH)$ problem is to calculate $Y = e(g_0, h_0)^{k \cdot f(\gamma)}$.

Definition 2: Set a vector $\{x_1, \ldots, x_{\tilde{l}+\tilde{m}}\}^T$ whose elements are pairwise different in Z_p. Define the polynomial $f(X) = \prod_{i=1}^{\tilde{l}} (X + x_i)$ and $g(X) = \prod_{\tilde{l}+1}^{\tilde{l}+\tilde{m}} (X + x_i)$, and set some values as follows:

$$\begin{cases} g_0, {g_0}^{\gamma}, \cdots, {g_0}^{\gamma^{\tilde{l}+\tilde{t}-2}}, {g_0}^{k \cdot \gamma \cdot f(\gamma)} \\ {g_0}^{\omega\gamma}, \cdots, {g_0}^{\omega\gamma^{\tilde{l}+\tilde{t}-2}} \\ {g_0}^{\alpha}, {g_0}^{\alpha\gamma} \cdots, {g_0}^{\alpha\gamma^{\tilde{l}+\tilde{t}}} \\ h_0, {h_0}^{\gamma}, \cdots, {h_0}^{\gamma^{\tilde{m}-2}} \\ {h_0}^{\omega}, {h_0}^{\omega\gamma} \cdots, {h_0}^{\omega\gamma^{\tilde{m}-1}} \\ {h_0}^{\alpha}, {h_0}^{\alpha\gamma} \cdots, {h_0}^{\alpha\gamma^{2(\tilde{m}-\tilde{t})+3}}. \end{cases}$$

where $k, \alpha, \gamma, \omega$ are random elements selected from Z_p, and g_0 and h_0 are generators of G [57].

Finally, output a bit b for $b \in \{0, 1\}$. If $b = 1$, the problem can be solved correctly when $Y = e(g_0, h_0)^{k \cdot f(\gamma)}$; otherwise, Y is a random value.

C. AGGREGATION ALGORITHM

The aggregation algorithm Aggreg was introduced in [58]. Let a list of values $\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1 \leq i \leq n}$, where $r, \gamma \in Z_p{}^*$ and x_i are different. Given $P_{0,m} = g^{\frac{r}{\gamma+x_m}}$, then we can calculate $P_{i,m} = (\frac{P_{i-1,i}}{P_{i-1,m}})^{\frac{1}{x_m - x_i}}$ for $i = \{1, \cdots, n-1\}$ and $m = \{i+1, \cdots, n\}$, where $m \in \{1, \cdots, n\}$. Finally, we can get $P_{i,m} = g^{\frac{r}{(\gamma+x_m)\prod_{k=1}^{i}\gamma+x_k}}$, where $1 \leq i \leq m \leq n$. Therefore, we can calculate $Aggreg(\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1 \leq i \leq n}) = g^{\frac{r}{\prod_{i=1}^{n}(\gamma+x_i)}}$.

IV. THE PROPOSED BVOABSC

The system architecture, the structure of the blockchain and security model are presented in this section.

A. SYSTEM ARCHITECTURE

The system model diagram of our scheme is shown in Fig.1, it contains seven entities, which are Attribute Authorties (AAs), Data Owner (DO), Medical Data Providers (MDP), Medical Data Requester (MDR), Blockchain, Cloud Server(CS) and Auditor.

- **AAs.** Attribute Authorties (AAs) contain various different organizations, such as hospitals, medical insurance organizations and medical research institutions. Based on the attributes submitted by users, each authority is mainly responsible for distributing corresponding keys for them. AAs are not full trusted by the other entities

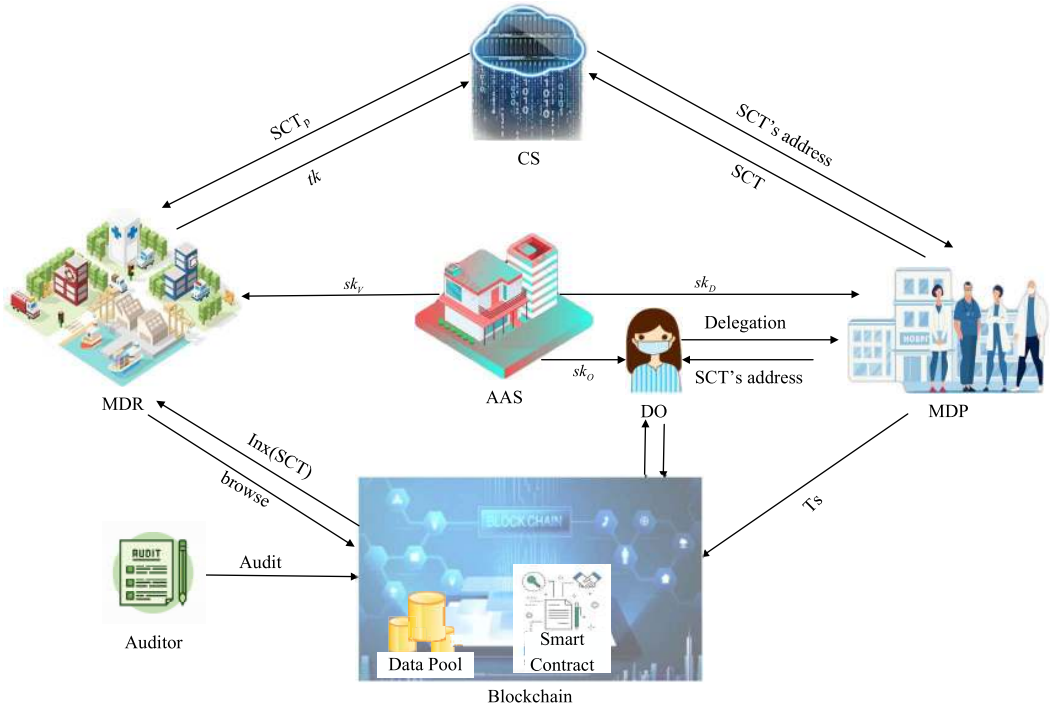

FIGURE 1. System model.

in our system since they may be corrupted and reveal the secret key information from the users. If no polynomial time adversary can sign or decrypt ciphertexts through mutual cooperation without authorization, collusion attacks between users and cloud servers can be prevented.

- **DO.** The patient is the owner of the EHRs, which means the source of the EHRs. First, the patient goes to the hospital to register and provides information about his/her symptoms. After receiving the initial diagnosis information from the hospital (including the information of the assigned doctor, diagnosis time and location, etc.), the patient describes the symptoms in detail to the designated doctor, and sends a letter of authorization to entrust the doctor to diagnose and treat at the specified time. Then, the patient formulates an access control strategy, which enables the users whose attributes meet the conditions to access the EHRs. The access control strategy is sent to the doctor, who is authorized to signcrypt and upload the generated EHRs. Finally, the patient creates a smart contract and uploads it to the blockchain.
- **MDP.** Medical Data Providers (MDP) include hospitals and doctors. After diagnosis and treatment, the doctor signscrypts the generated EHRs according to the access control strategy formulated by the patient, and uploads the ciphertext to the cloud server. After receiving the storage address returned by the cloud server, the doctor sends the address to the patient. At the same time, the doctor creates a transaction, including the storage address of the ciphertext, account information, signature, authorization letter, current time and other information. Finally, the transaction is uploaded to the blockchain by the doctor.
- **Blockchain.**The type of blockchain is the Ethereum blockchain. It is mainly responsible for collecting transaction information, recording all user access requests and access activities, avoiding outsourced EHRs from being illegally modified and ensuring the security of transactions. In addition, the blockchain stores smart contracts created by the patient, ensuring that EHRs are safely shared among different users. Since the blockchain is public, transaction information and smart contracts can be browsed and accessed by all users.
- **CS.** Cloud Server(CS) is honest but curious. It is mainly responsible for storing the ciphertext of EHRs uploaded by doctors. The validity of the ciphertext can be verified. If the ciphertext is invalid, CS can refuse to store it. At the same time, CS can verify the identity of the doctor, whether it is authorized by the patient. Moreover, CS can partially decrypt the outsourced EHRs, which reduces the computational burden of the MDR. The correctness of the partially decrypted ciphertext generated by CS can be checked.
- **MDR.** In order to access the EHRs, the users' decryption attributes should meet the encryption strategy formulated by the patient. First, the MDR browse the address index of EHRs on the smart contract. Then, they request access to the EHRs to the CS by submitting the

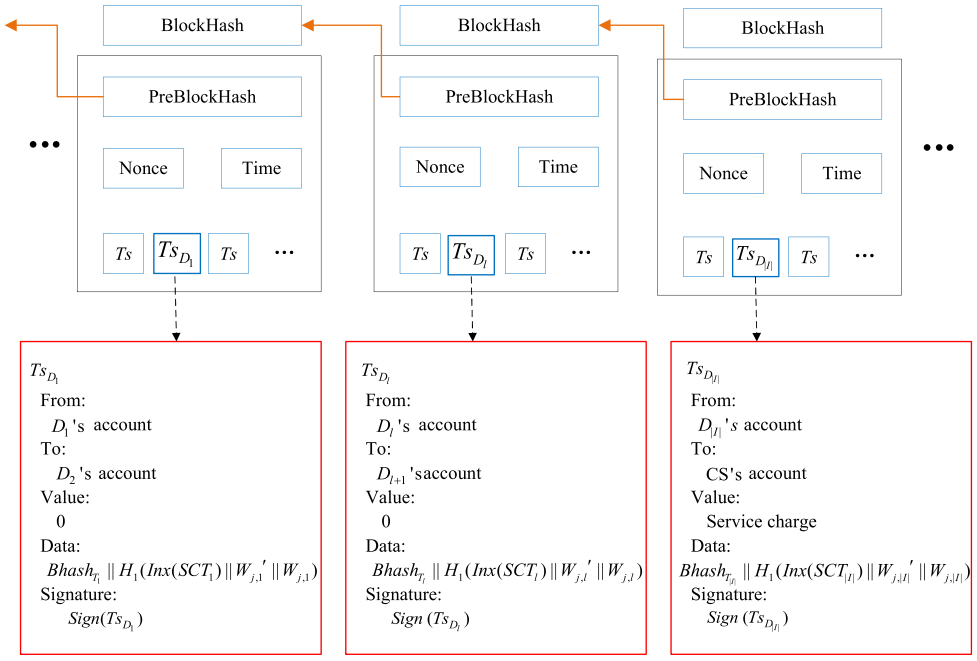

FIGURE 2. Transaction structure.

decryption attribute, the ciphertext address index and the transformed key. If the verification is passed, CS will partially decrypt the corresponding ciphertext of EHRs and send the generated partial ciphertext to the MDR. Finally, the MDR can recover the EHRs by using their private key.

- **Auditor.** The auditor can ensure the integrity and correctness of the outsourced EHRs by verifying the transaction number, transaction time and transaction information.

The specific operation of our BVOABSC scheme is shown in Fig.1, and described as follows.

1) According to the attributes submitted by users, each authority AA_j sends the corresponding key to them. Thus, the patient, MDR and doctors can respectively receive the attribute private key $sk_{O,j}$, $sk_{V,j}$ and $sk_{D,j}$.
2) The patient sends a letter of authorization (W_j', W_j) to the doctor, and entrusts the doctor for diagnosis and treatment at a specified time. Then, the patient authorizes the doctor to signscrypt the generated EHRs by formulating the encryption strategy $(t, R_{e,j})$ and the signature strategy $(t, R_{s,j})$.
3) After diagnosis and treatment, the doctor generates the patient's EHRs. Then, the EHRs are signcrypted according to the access control strategy formulated by the patient.
4) The doctor uploads the ciphertext SCT to CS for storage. Then CS needs verify the validity of the SCT. If the verification fails, the CS refuses to store SCT; otherwise, the CS accepts the SCT and sends the storage address of SCT to the doctor. Afterwards, the doctor sends the address to the patient. Finally, the doctor creates a transaction Ts and uploads it to the block.
5) The patient creates a smart contract and uploads it to the blockchain.
6) The MDR access the ciphertext index $Inx(SCT)$ on the smart contract as required, and then request the CS to access the EHRs by submitting the decryption attributes, $Inx(SCT)$ and the transformed key tpk. If the verification is passed, the CS partially decrypts the ciphertext and returns the generated partial ciphertext SCT^P to the MDR. Finally, MDR decrypt SCT^P and recover the EHRs by using the secret key tsk.

B. THE STRUCTURE OF THE BLOCKCHAIN

The blockchain structure of our scheme is composed of the hash value of the block, the hash value of the previous block, the Nonce value, the timestamp and the transaction, as shown in Figure 2. The following specifically introduces the composition of transaction and the design of smart contract.

1) TRANSACTION STRUCTURE

As shown in Fig.2, we can know that the transaction on EHRs consists of $Bhash_T$, $H_1(Inx(SCT)$, (W_j', W_j) and $Sign(Ts)$, where $Bhash_T$ is the block hash value newly added to the blockchain based on time T, $H_1(Inx(SCT)$ refers to the hash value index of the signcryption ciphertext SCT, (W_j', W_j) represents the authorization letter designated by the patient, and $Sign(Ts)$ represents the signature of the transaction generated on the current block. The doctors have a requirement

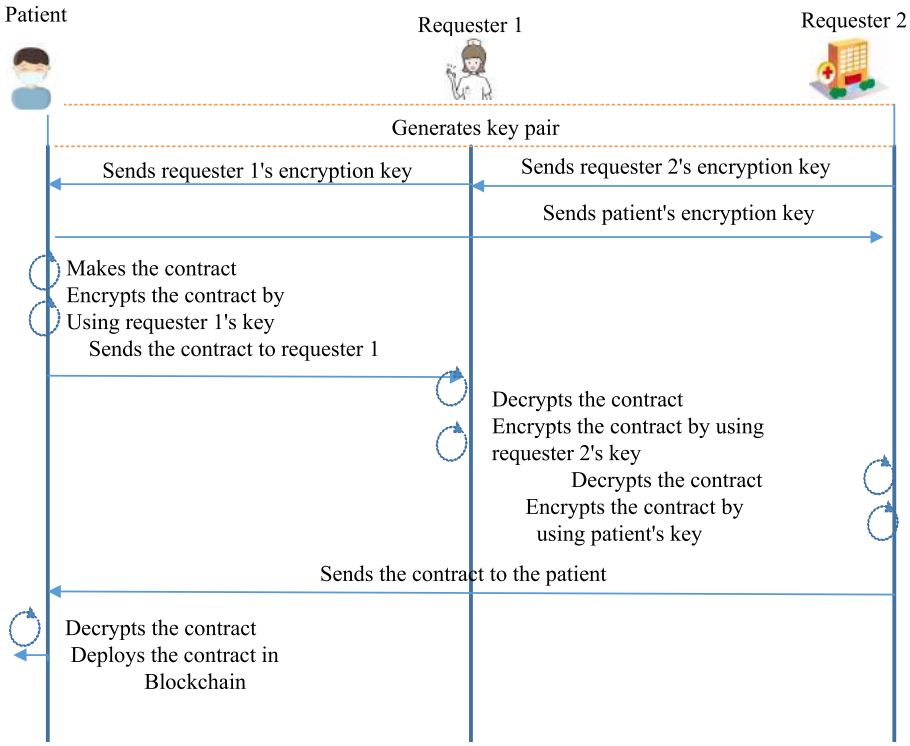

FIGURE 3. Generation of smart contracts.

of creating a new transaction when they generate an EHR for the patient. Storing transactions in the Ethereum ensures that it cannot be tampered with. Of course, doctors need to pay service fees for storing transactions in Ethereum. Therefore, we create an account for each doctor and CS in the system.

2) SMART CONTRACT

The smart contract is the key of the blockchain, and is an event-driven computer program deployed in the blockchain [59]. Our scheme applies it to Ethereum, which promotes the reliable execution of transactions without the involvement of a third party, and ensures that all transactions are traceable and irreversible. In addition, the proposed scheme uses smart contracts to securely share EHRs between patients and MDR.

First, the patient formulates a smart contract and sets its execution conditions. Then, the smart contract is encrypted according to the user's key, and the encrypted smart contract is broadcast to the blockchain. When the requester 1 accesses medical-related information, the smart contract will be triggered to execute. The smart contract verifies whether it is a legitimate user based on the attributes of the requester 1. If the verification fails, the access fails. Otherwise, the requester 1 is allowed to decrypt the contract to obtain corresponding information. If the information in the smart contract is correct, the requester 1 uses the key of the requester 2 to encrypt the contract and broadcast it to the blockchain. Similar to the execution process of the requester 1, the requester 2 encrypts the contract with the patient's key and sends it to the patient after verifying the content of the contract. Then, the patient decrypts the contract to check the correctness of the contract, and returns the verification result to the requester 2. Similarly, the process of other requesters such as doctors and medical institutions accessing the contract is consistent with the above description. We can conclude that only users who meet the access conditions can view the contents of the contract. The process of creating a smart contract and reaching a consensus with requesters is clearly shown in Fig.3.

C. SECURITY MODEL

The BVOABSC scheme fulfills the requirements of confidentiality, unforgeability, privacy, non-tamperability and timeliness of the EHRs.

1) CONFIDENTIALITY

If there is no adversary $\mathcal{A}$ to win the game with a non-negligible advantage in the polynomial time algorithm (PPT), the BVOABSC scheme is indistinguishable from non-adaptively selected ciphertext attacks. A formal definition is given in the following interactive game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$.

Initialization: The adversary $\mathcal{A}$ first selects an encryption strategy $(t, R_{e,j}{}^*)$, where $j \in N^*$, N^* is a set of the authorities and $(t, R_{e,j}{}^*)$ is specified by the authorized institution $AA_j{}^*$. Then, $\mathcal{A}$ sends $(t, R_{e,j}{}^*)$ to $\mathcal{C}$.

Setup: The challenger $\mathcal{C}$ runs *GlobalSetup* to generate public parameters *GP* and sends it to the adversary $\mathcal{A}$.

AuthoritySetup: The challenger $\mathcal{C}$ runs *AuthoritySetup* and sends the generated public key PK to adversary $\mathcal{A}$.

Query Phase 1: $\mathcal{C}$ creates an empty list Tab. $\mathcal{A}$ can request the following queries at several times.

- **Secret Key Query O_{sk}.** $\mathcal{A}$ requests to inquire the set of signcryption attributes related to the threshold, where $|A_{\mathcal{A}}{}^* \cap R_{e,j}{}^*| < t$. $\mathcal{C}$ generates a private key $sk_{\mathcal{A}}$ by running *SecretGen* and forwards it to $\mathcal{A}$.
- **Transformation Key Query O_{tk}.** $\mathcal{A}$ asks for the transformed key $tk_{\mathcal{A}}$. Then, $\mathcal{C}$ searches for $(A_{\mathcal{A}}{}^*, sk_{\mathcal{A}}, tk_{\mathcal{A}})$ in the Tab. If it exists, it will be returned $tk_{\mathcal{A}}$ to $\mathcal{A}$, otherwise, the transformation key generated by running *TransformationKeyGen* and be sent to $\mathcal{A}$.
- **Signcryption Query O_{SC}.** $\mathcal{A}$ submits a message m, an encryption strategy $(t, R_{e,j}{}^*)$ and a signature strategy $(t, R_{s,j}{}^*)$. $\mathcal{C}$ executes the **Signcryption** algorithm and sends the generated signcryption ciphertext SCT to $\mathcal{A}$.
- **Decsigncryption Query O_{DS}.** After receiving SCT, threshold value t and the attribute set $A_{\mathcal{A}}{}^*$ submitted by $\mathcal{A}$, $\mathcal{C}$ executes the **Decsigncryption** algorithm and sends m to $\mathcal{A}$.

Challenge: $\mathcal{A}$ chooses two messages of equal length m_0 and m_1, the encryption strategy $(t, R_{e,j}{}^*)$ and signature strategy $(t, R_{s,j}{}^*)$, and then sends them to $\mathcal{C}$. $\mathcal{C}$ randomly selects a bit $b \in \{0, 1\}^*$, and returns SCT^* to $\mathcal{A}$ as the challenge ciphertext by running the **Signcryption** algorithm.

Query Phase 2: Similar to the **Phase 1**, except that $\mathcal{A}$ cannot ask the ciphertext that has been challenged.

Guess: $\mathcal{A}$ tries to guess which message m_b corresponds to the ciphertext SCT^*, where $b \in \{0, 1\}$. $\mathcal{A}$ outputs a guessed bit b' on b, and $\mathcal{A}$ wins the game if $b' = b$.

The advantage of $\mathcal{A}$ in this game is defined as $Adv_{\mathrm{A}}^{IND-CCA2} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 1: If the probability of any polynomial adversary winning the above game is negligible, then the BVOABSC scheme is indistinguishable under the Chosen Ciphertext Attack (CCA2) and satisfies the characteristics of confidentiality.

2) UNFORGEABILITY

If there is no adversary to win the game with a non-negligible advantage in the polynomial time algorithm (PPT), the BVOABSC scheme is unforgeable under the selective message attack (EUF-CMA). The interactive game between $\mathcal{C}$ and $\mathcal{A}$ is defined as follows.

The **Initialization** and **setup** are consistent with those description in confidentiality.

Query Phase 1: $\mathcal{C}$ creates an empty Tab, and $\mathcal{A}$ can initiate the following inquiry multiple times.

- **Secret Key Query O_{sk}.** $\mathcal{A}$ queries the signcryption attribute set $A_{\mathcal{A}}{}^*$ and threshold value t, which satisfy $|A_{\mathcal{A}}{}^* \cap R_{e,j}{}^*| < t$ and $|A_{\mathcal{A}}{}^* \cap R_{s,j}{}^*| < t$. Then, $\mathcal{C}$ runs *SecretGen* and sends $sk_{\mathcal{A}}$ to $\mathcal{A}$.
- **Signcryption Query O_{SC}.** $\mathcal{A}$ submits a message m, $(t, R_{e,j}{}^*)$ and $(t, R_{s,j}{}^*)$. $\mathcal{C}$ runs the **Signcryption** algorithm and returns SCT to $\mathcal{A}$.
- **Forgery.** $\mathcal{A}$ sends SCT^*, the encryption policy $(t^*, R_{e,j}{}^*)$ and signature policy $(t^*, R_{s,j}{}^*)$ to $\mathcal{C}$ (i.e; $(t^*, R_{e,j}{}^*)$ and s$(t^*, R_{s,j}{}^*)$ have not been asked, and $t^* < t$). $\mathcal{C}$ executes the **Decsigncryption** algorithm to get the message m^*. If SCT^* is correct and m^* has never been asked before, $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ to win the game is $Adv_{\mathcal{A}}^{Priv} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 2: If the probability of any adversary winning the above game within the polynomial time algorithm is negligible, the BVOABSC scheme is unforgeable under the selective message attack (EUF-CMA).

3) PRIVACY

The adversary $\mathcal{A}$ cannot determine the corresponding signcryptor through the signcryption message, which protects the identity privacy of the signcryptor. We introduce the safe game between $\mathcal{A}$ and $\mathcal{C}$ as below.

setup: After receiving a set of attributes $A_{\mathcal{A}}{}^*$ sent by $\mathcal{A}$, $\mathcal{C}$ runs the Setup algorithm. Then, adversary $\mathcal{A}$ can receive the generated public parameters GP.

Challenge: $\mathcal{A}$ selects the encryption strategy $(t, R_{e,j}{}^*)$ and the signature strategy $(t, R_{s,j}{}^*)$ for $1 \leqslant t \leqslant |R_{s,j}{}^*| = |R_{e,j}{}^*|$, two attribute sets $A_{\mathcal{A},1}{}^*$ and $A_{\mathcal{A},2}{}^*$ for $|A_{\mathcal{C},1}{}^* \cap R_{s,j}{}^*| = |A_{\mathcal{C},1}{}^* \cap R_{e,j}{}^*| = |A_{\mathcal{C},2}{}^* \cap R_{s,j}{}^*| = |A_{\mathcal{C},2}{}^* \cap R_{2,j}{}^*| = t$, and a message m. Later, $((t, R_{e,j}{}^*), (t, R_{s,j}{}^*), A_{\mathcal{A},1}{}^*, A_{\mathcal{A},2}{}^*, m)$ is sent to $\mathcal{C}$. The challenger $\mathcal{C}$ selects a random bit $b \in \{0, 1\}^*$ and runs the *SecretGen* to generate $sk_{\mathcal{C}}$. Finally, C runs the **Signcryption** algorithm to generated SCT_b for $\mathcal{A}$.

Guess: $\mathcal{A}$ tries to guess which message m_b corresponds to the ciphertext SCT^*, where $b \in \{0, 1\}$. Then, $\mathcal{A}$ outputs one bit b'. If $b' = b$, $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ in the above game is $Adv_{\mathrm{A}}^{Priv} = |\Pr[b' = b] - \frac{1}{2}|$.

Definition 3: If there is no adversary who wins the above safe game with a non-negligible probability within the polynomial time algorithm, the BVOABSC scheme satisfies computational privacy.

4) VERIFIABILITY

The honest MDR can verify the correctness of the partial ciphertext obtained from CS. We have defined the following safe games between $\mathcal{A}$ and $\mathcal{C}$.

The adversary $\mathcal{A}$ generates an encryption strategy $(t, R_{e,j}{}^*)$ and a signature strategy $(t, R_{s,j}{}^*)$ as a challenge access strategy during the **initialization** and **setup** phase, and requests the public key, which is similar to the description in confidentiality.

Query Phase 1: $\mathcal{C}$ creates an empty Tab and $\mathcal{A}$ can request the following queries repeatedly.

- **Secret Key Query O_{sk}.** $\mathcal{A}$ asks the threshold value t and signcryption attribute set $A_{\mathcal{A}}{}^*$, which satisfies $|A_{\mathrm{A}}{}^* \cap R_{e,j}{}^*| < t$ and $|A_{\mathcal{A}}{}^* \cap R_{s,j}{}^*| < t$. Hereafter, $\mathcal{C}$ generates a private key $sk_{\mathcal{A}}$ by running *SecretGen* and then transmits it to $\mathcal{A}$.
- **Transformation Key Query O_{tk}.** $\mathcal{A}$ inquires about the transformed key $tk_{\mathcal{A}}$. C first finds in the Tab with

$(A_{\mathcal{A}}{}^*, sk_{\mathcal{A}}, tk_{\mathcal{A}})$. If it exists, $\mathcal{C}$ returns $tk_{\mathcal{A}}$ to $\mathcal{A}$, otherwise, it runs the *TransformationKeyGen* and returns the transformation key $tk_{\mathcal{A}}$ to $\mathcal{A}$.

- **Signcryption Query O_{SC}.** $\mathcal{A}$ requests the signcryptiom ciphertext of the message m, which is related to the encryption strategy $(t, R_{e,j}{}^*)$ and the signature strategy $(t, R_{s,j}{}^*)$. Subsequently, $\mathcal{C}$ executes the **Signcryption** algorithm to get the ciphertext SCT, and forwards it to $\mathcal{A}$.
- **Decsigncryption Query O_{DS}.** $\mathcal{A}$ requests the designcryption of SCT related to the threshold value t and the attribute set $A_{\mathcal{A}}{}^*$. $\mathcal{C}$ runs the **Decsigncryption** algorithm, and sends m to $\mathcal{A}$.

Challenge: $\mathcal{A}$ picks a challenge message m^* and forwards it to $\mathcal{C}$. Then, $\mathcal{C}$ generates a challenge signcryption ciphertext m^* by running the **Signcryption** algorithm and returns it to $\mathcal{A}$.

Query Phase 2: Similar to **Phase** 1, except that $\mathcal{A}$ cannot ask the challenge ciphertext SCT^* that has been received.

Forgery: $\mathcal{A}$ generates a set of attributes $A_{\mathcal{A}}{}^*$ and a random partial ciphertext SCT^{P*}, which is not generated by the *PartialDecryption*. $\mathcal{C}$ executes the *TransformationKeyGen* to get $tk_{\mathcal{A}}$ and then recovers the message m. If $m \notin \{m^*, \perp\}$ and SCT^{P*} are verified to be valid, $\mathcal{A}$ wins the game.

Definition 4: If there is no polynomial adversary which can win the above game with a non-negligible probability, then the BVOABSC scheme meets the requirement of verifiability.

D. THE CONSTRUCTION OF BVOABSC

1) SYSTEM INITIALIZATION

In the initialization phase, the system generates public parameters. After the user registers the information in the system, the authority generates the public key and the master private key.

Phase 1: *GlobalSetup*

Define a bilinear group (G_1, G_2, G_T) of prime order p, an asymmetric bilinear map $\hat{e} : G_1 \times G_2 \to G_T$, two collusion-resistant hash functions $H : \{0,1\}^* \to Z_p$ and $H_1 : \{0,1\}^* \to G_1$, where g and g_1 are two generators of G_1 and G_2.Then, public the system parameters $GP = (\hat{e}, g, g_1, p, G_1, G_2, G_T, H)$.

Phase 2: *register*

First, different entities (such as patients, MDP, MDR, and auditor) register information to join the system. At the same time, each doctor creates an account in Ethereum and then publishes it to others. The cloud server also creates an externally owned account in Ethereum and sends it to doctors and auditors.

Phase 3: *AuthoritySetup*

After registration, each authority AA_j generates a public key and a master private key by performing the following process, where $j \in [1, N]$ and N is the number of authority.

1) AA_j defines a function $F : \tilde{U} \to (Z/pZ)^*$, where $\tilde{U} = \{a_{j,1}, \cdots, a_{j,k}\}$ is a set of attributes. For each attribute $a_{j,k} \in \tilde{U}$, the encoded attribute values $F(a_{j,k}) = x$ are pairwise different.
2) Randomly choose α_j, γ_j from $Z_p{}^*$ and calculate $\Delta_j = e(g^{\alpha_j}, g_1)$ and $\mu_j = g^{\alpha_j \gamma_j}$.
3) Let the public key $PK = \{\Delta_j, \{g_1{}^{\alpha_j \gamma_j{}^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k\}}, \mu_j, F\}$ be in public, and keep the master key $MSK = \{\alpha_j, \gamma_j\}$ secret.

Phase 4: *SecretGen*

The authority AA_j distributes the secret key for the users according to their attributes. The detailed description is as follows.

1) AA_j defines $\widetilde{A_w}$ as the attribute set of an entity, where $\widetilde{A_w} \in \tilde{U}$, ω represents the entity's type. A_O represents the data owner when $\omega = O$ and A_V means the data requester when $\omega = V$, and if $\omega = D$, A_D represents the doctor.
2) Randomly choose $r_{w,j}$ from $Z_p{}^*$, then $sk_{w,j} = (\{g^{\frac{r_{w,j}}{\gamma_j + F(a_{j,k})}}\}_{a_{j,k} \in \widetilde{AA_j} \cap \widetilde{A_w}}, \{g_1{}^{r_{w,j}\gamma_j{}^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k-2\}}, g_1{}^{\frac{r_{w,j}-1}{\gamma_j}}) = (sk_{w1,j}, sk_{w2,j}, sk_{w3,j})$ can be calculated.

2) APPOINTMENT

The following describes the process of interaction between patients, hospitals and doctors.

1) The patient randomly selects $\psi_j \in Z_p{}^*$ to compute $P_j = g^{\psi_j}$.
2) After receiving the patient's registration information, the hospital assigns a group of doctors $\{D_l\}_{(l \in I)}$, where I is the index of the designated doctors.
3) The patient entrusts D_l to generate EHRs by calculating a letter of authorization $(W_{j,\ell}{}', W_{j,\ell})$, such that $W_{j,\ell}{}' = tp_l || rea_l$ and $W_{j,\ell} = \psi_j \cdot H_1(W_{j,\ell}{}')$, tp_l is effective time, and rea_l represents medical-related information.

3) THE STORAGE OF EHRs

The generation of EHRs is divided into two situations, one is generated by a doctor, and the other is generated by multiple doctors in turn.

Case 1: The patient's EHRs are generated by only a doctor D_1.

Phase 1: The generation of the signcryption ciphertext SCT_1 and the transaction $Ts\,(D_1)$.

1) D_1 generates an EHR m_1 after the diagnosis and treatment.
2) The doctor signcrypts m_1 according to the encryption strategy $(t, R_{e,j})$ and signature strategy $(t, R_{s,j})$ formulated by the patient. $(t, R_{e,j})$ means the encryption strategy, where $R_{e,j} \subset \tilde{U}$, $s = |R_{s,j}|$ and $1 \leqslant t \leqslant |R_{e,j}|$. $(t, R_{s,j})$ represents the signature strategy, where $R_{s,j} \subset \tilde{U}$,$s = |R_{s,j}|$, and $1 \leqslant t \leqslant |R_{s,j}|$. So we can get $\delta_2 = g_1{}^{r_{D,j} P_{(\widetilde{A_D}, R_{s,j})}(\gamma_j)} / \delta_1$. The construction of signcryption ciphertext is executed by the doctor D_1, and the specific process is as follows.

TABLE 1. Notations.

Notation	Meaning
$\tilde{U}$	The attribute set
k	The size of attribute set $\tilde{U}$
$a_{j,k}$	kth attribute of the authority AA_j
$R_{e,j}$	An encryption strategy
$R_{s,j}$	An signature strategy
$\widetilde{A_O}$	An attribute universe of the data owner
$\widetilde{A_D}$	An attribute universe of the doctor
$\widetilde{A_V}$	An attribute universe of the user
E_1	The cost of an exponential operations in group G_1
E_2	The cost of an exponential operations in group G_2
E_T	The cost of an exponential operations in group G_T
E_Z	The exponential operations in ring E_P
P	The overhead of a pairing
S_s	The size of encryption policy $R_{e,j}$
S_e	The size of signature policy $R_{s,j}$
l_e	The number of encryption key attributes
l_s	The number of decryption key attributes
l_d	An pairing overhead
B_{G_1}	The length of an element of the group G_1 in bits
B_{G_2}	The length of an element of the group G_2 in bits
B_{Z_q}	The length of an element of the ring Z_q in bits

- D_1 uses aggregate functions $Aggreg$ [60] and the key $sk_{D,j}$ to calculate $X_1 = Aggreg(\{g^{\frac{r_{D,j}}{\gamma_j+F(a_{j,k})}}, F(a_{j,k})\}_{a_{j,k}\in\widetilde{AA_j}\cap\widetilde{A_D}}) = g^{\frac{r_{D,j}}{\prod_{a_{j,k}\in\widetilde{AA_j}\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))}}$.
- Compute the polynomial $P_{(\widetilde{A_D},R_{s,j})}(\gamma_j) = \frac{1}{\gamma_j}(\prod_{a_{j,k}\in R_s\cup B_{k+t-1-s}\setminus\widetilde{A_D}}(\gamma_j+F(a_{j,k})-\delta_1)$, where $\delta_1 = \prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D} F(a_{j,k})$ and $B_f = \{b_1,\cdots,b_f\}_{f\leqslant k-1}$. Then, D_1 can get $\delta_2 = g_1^{r_{D,j}P_{(\widetilde{A_D},R_{s,j})}(\gamma_j)}/\delta_1$ by using $sk_{D2,j}$.
- Calculate $sig_{1,1} = X_1 \cdot g^{\frac{H(m_1)}{\prod_{a_{j,k}\in\widetilde{AA_j}\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))}}$, $sig_{1,2} = sk_{D3,j}\cdot\delta_2\cdot g_1^{H(m_1)P_{(\widetilde{A_D},R_{s,j})}(\gamma_j)/\delta_1}$ and $sig_{1,3} = {g_1}^{\alpha_j\cdot H(m_1)}$, and generate a signature $sig_1 = (sig_{1,1}, sig_{1,2}, sig_{1,3})$.
- Choose η_1 from $Z_p{}^*$, then calculate $SCT_{1,1} = g^{-\eta_1\cdot\alpha_j\cdot\gamma_j}$, $SCT_{1,2} = g_1^{\eta_1\cdot\alpha_j\cdot\prod_{a_{j,k}\in R_{e,j}}(\gamma_j+F(a_{j,k}))}$ and $SCT_{1,3} = e(g,g_1)^{\alpha_j\cdot\eta_1}e(g,g_1)^{\alpha_j\cdot H(m_1)}\cdot m_1$. Consequently, the signcryption ciphertext $SCT_1 = (SCT_{1,1}, SCT_{2,1}, SCT_{3,1})$ is generated.

3) D_1 first generates an index $Inx(SCT_1)$ for the signcryption ciphertext SCT_1. Later, D_1 extracts the latest block hash value $Bhash_{T_1}$ based on the current time T_1. Finally, D_1 creates a transaction $Ts\,(D_1)$ and sends the service fee to CS's account and calculates the transaction data value, which is $Bhash_{T_1}||H_1(Inx(SCT_1)||W_{j,1}{}'||W_{j,1})$.
4) Compute the transaction data value $Bhash_{T_1}||H_1(Inx(SCT_1)||W_{j,1}{}'||W_{j,1})$ and send $(Bhash_{T_1}, SCT_1, W_{j,1}{}', W_{j,1})$ to the CS.

Phase 2: Verification and storage.

1) The CS verifies the service fee that has been received, checks the validity of tp_l and $Bhash_{T_{|I|}}$, and verifies whether the following two equations hold:

$$e(W_{j,1}{}', g) = e(H(W_{j,1}), P_j),$$

$\Delta_j = e(u_j{}^{-1}, sig_{1,2})\cdot e(g^{\alpha_j}, g_1)^{H(m_1)(1-Q_1-\frac{1}{Q_1})}\cdot e({sig_{1,1}}^{\delta}, {g_1}^{\alpha_j\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}(\gamma_j+F(a_{j,k}))})$, where $Q_1 = \prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}\frac{\gamma_j+F(a_{j,k})}{F(a_{j,k})}$. If the verification fails, CS refuses to store he signcryption ciphertext SCT_1; otherwise, it accepts it and returns the storage address CL_1 of SCT_1 to D_1.

2) After receiving CL_1 sent by D_1, the patient calculates $H_1(CL_1)$ and generates $Inx(H_1(CL))$. Finally, the patient writes $Inx(H_1(CL))$ into the smart contract.

Case 2: The patient's EHRs are sequentially generated by multiple doctors. Suppose D_1 is the first doctor who generates the EHRs, and $D_{|I|}$ is the last doctor who generates the EHRs.

Phase 1: The generation of the signcryption ciphertext $SCT_{|I|}$ and the transaction $Ts\,(D_{|I|})$.

1) First, D_1 generates SCT_1 of m_1 and the corresponding transaction (the process is the same as Case 1). Then, $(Bhash_{T_1}, SCT_1, W_{j,1}{}', W_{j,1})$ is sent to D_2.
2) After receiving $(Bhash_{T_{l-1}}, SCT_{l-1}, W_{j,\ell-1}{}', W_{j,\ell-1})$ from D_{l-1}, D_l verifies its validity through the equation $e(W_{j,\ell-1}{}', g) = e(H(W_{j,\ell-1}), P_j)$, where $l = 2,\cdots,|I-1|$. Then, $\{m_1,\cdots,m_{l-1}\}$ can be obtained by decrypting SCT_{l-1}.
3) D_1 generates the current EHR m_l, and then signcrypts it through the following process.

- Calculate $sig_{l,1} = X_1\cdot g^{\frac{H(m_1||\cdots||m_l)}{\prod_{a_{j,k}\in\widetilde{AA_j}\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))}}$, $sig_{l,2} = sk_{D3,j}\cdot\delta_2\cdot g_1^{H(m_1||\cdots||m_l)P_{(\widetilde{A_D},R_{s,j})}(\gamma_j)/\delta_1}$ and $sig_{l,3} = {g_1}^{\alpha_j\cdot H(m_1||\cdots||m_l)}$ to generate the signature $sig_l = (sig_{l,1}, sig_{l,2}, sig_{l,3})$.

– Choose $\eta_l \in Z_p{}^*$, and then calculate $SCT_{l,1} = g^{-\eta_l \cdot \alpha_j \cdot \gamma_j}$, $SCT_{l,2} = g_1^{\eta_l \cdot \alpha_j \cdot \prod_{a_{j,k} \in R_e} (\gamma_j + F(a_{j,k}))}$ and $SCT_{l,3} = e(g, g_1)^{\alpha_j \cdot \eta_l} e(g, g_1)^{\alpha_j \cdot H(m_1)} \cdot (m_1 || \cdots || m_l)$. Finally, the signature $SCT_l = (SCT_{l,1}, SCT_{l,2}, SCT_{l,3})$ is generated.

4) D_l generates index $Inx(SCT_l)$ of the SCT_l. At the same time, D_l extracts the hash value $Bhash_{T_l}$ of the block newly added to the blockchain based on the current time T_l. Then, D_l creates a transaction $Ts\,(D_l)$, and transfers 0 service fee to the next doctor's account. At the end, $(Bhash_{T_l}, SCT_l, W_{j,l}{}', W_{j,l})$ is sent to D_{l+1}.

5) For $D_{|I|}$, she/he first checks the validity of $(Bhash_{T_{|I|-1}}, SCT_{|I|-1}, W_{j,|I|-1}{}', W_{j,|I|-1})$ received by $D_{|I|-1}$, and it can be verified through the equation $e(W_{j,|I|-1}{}', g) = e(H(W_{j,|I|-1}), P_j)$. Then, $D_{|I|}$ decrypts $SCT_{|I|-1}$ and obtains $\{m_1, \cdots, m_{|I|-1}\}$.

6) $D_{|I|}$ signcrypts the EHR $m_{|I|}$ that is currently generated, the specific process is as follows.

– Calculate $sig_{|I|,1} = X_1 \cdot g^{\frac{H(m_1||\cdots||m_{|I|})}{\prod_{a_{j,k} \in \widetilde{AA_j} \cap \widetilde{A_D}} (\gamma_j + F(a_{j,k}))}}$, $sig_{|I|,2} = sk_{D3,j} \cdot \delta_2 \cdot g_1^{H(m_1||\cdots||m_{|I|}) P_{(\widetilde{A_D}, R_{s,j})}(\gamma_j)/\delta_1}$, and $sig_{|I|,3} = g_1^{\alpha_j \cdot H(m_1||\cdots||m_{|I|})}$, and generate the signature $sig_{|I|} = (sig_{|I|,1}, sig_{|I|,2}, sig_{|I|,3})$.

– $D_{|I|}$ selectes $\eta_{|I|}$ from $Z_p{}^*$, then calculates $SCT_{|I|,1} = g^{-\eta_{|I|} \cdot \alpha_j \cdot \gamma_j}$, $SCT_{|I|,2} = g_1^{\eta_{|I|} \cdot \alpha_j \cdot \prod_{a_{j,k} \in R_{e,j}} (\gamma_j + F(a_{j,k}))}$ and $SCT_{|I|,3} = e(g, g_1)^{\alpha_j \cdot \eta_{|I|}} e(g, g_1)^{\alpha_j \cdot H(m_{|I|})} \cdot (m_1 || \cdots || m_{|I|})$. Finally, $SCT_{|I|} = (SCT_{|I|,1}, SCT_{|I|,2}, SCT_{|I|,3})$ is generated.

7) $D_{|I|}$ computes the index $Inx(SCT_{|I|})$, then extracts the hash value $Bhash_{T_{|I|}}$ of the block newly added to the blockchain based on the current time $T_{|I|}$. Afterwards, $D_{|I|}$ creates a transaction $Ts\left(D_{|I|}\right)$ and transfers the service fee to the CS's account.

8) Finally, $D_{|I|}$ sends $(Bhash_{T_{|I|}}, SCT_{|I|}, W_{j,|I|}{}', W_{j,|I|})$ to the CS.

Phase 2: Verification and storage.

1) CS verifies the service fee, checks the validity of tp_l, $Bhash_{T_{|I|}}$ through the following equations.

$$e(W_{j,|I|}{}', g) = e(H(W_{j,|I|}), P_j),$$

$$\Delta_j = e(u_j{}^{-1}, sig_{1,2}) \cdot e(g^{\alpha_j}, g_1)^{H(m_1)(1-Q_1-\frac{1}{Q_1})} \cdot e(sig_{1,1}{}^{\delta}, g_1^{\alpha_j \prod_{a_{j,k} \in R_{s,j} \cup B_{k+t-1-s} \setminus A_D} (\gamma_j + F(a_{j,k}))}),$$

where $Q_1 = \prod_{a_{j,k} \in R_{s,j} \cup B_{k+t-1-s} \setminus A_D} \frac{\gamma_j + F(a_{j,k})}{F(a_{j,k})}$.

2) If the verification fails, CS refuses to store the $SCT_{|I|}$; otherwise, it accepts and returns its storage address CL to $D_{|I|}$.

3) After receiving CL sent by$D_{|I|}$, the patient calculates $H_1(CL)$ and generates $H_1(CL)$. Finally, the patient will write $Inx(H_1(CL))$ into smart contract.

4) EHRs ACCESS

Phase 1: *TransformationKeyGen* .

1) MDR pick a random number $z \in Z_p{}^*$ as a retrieval key $tsk = z$, and calculate $tpk = (sk_{w1,j}{}^{\frac{1}{z}}, sk_{w2,j}{}^{\frac{1}{z}}, sk_{w3,j}{}^{\frac{1}{z}})$, where $sk_{w1,j}{}^{\frac{1}{z}} = (\{g^{\frac{r_{w,j}}{z(\gamma_j + F(a_{j,k}))}}\}_{a_{j,k} \in \widetilde{AA_j} \cap \widetilde{A_w}}$, $sk_{w3,j}{}^{\frac{1}{z}} = g_1^{\frac{r_{w,j}-1}{z\gamma_j}}$, and $sk_{w2,j}{}^{\frac{1}{z}} = \{g_1^{\frac{r_{w,j}\gamma_j{}^{\varepsilon}}{z}}\}_{\{\varepsilon=0,\cdots,k-2\}}$.

2) Generate the transformed key $tk = (tpk, tsk)$.

Phase 2: *PartialDecryption*.

1) MDR browse the address index of the EHRs on the smart contract.

2) The MDR ask the CS to query the EHRs by sending the decryption attribute, address index and transformed key.

3) If the decryption attributes of the MDR meet the encryption strategy $(t, R_{e,j})$ formulated by the patient, the CS can use the transformed key tpk sent by MDR to partially decrypt the ciphertext of the EHRs. The process is as follows.

– For all $a_{j,k} \in \widetilde{AA_j} \cap \widetilde{A_V}$, the CS uses aggregate functions $Aggreg$ to aggregate the user's key $X_2 = Aggreg(\{g^{\frac{r_{V,j}}{z(\gamma_j + F(a_{j,k}))}}, F(a_{j,k})\}_{a_{j,k} \in \widetilde{AA_j} \cap \widetilde{A_V}}) = g^{\frac{r_{V,j}}{z \prod_{a_{j,k} \in \widetilde{AA_j} \cap \widetilde{A_V}} (\gamma_j + F(a_{j,k}))}}$.

– The CS calculates the partial signcryption ciphertext $SCT^P = e(SCT_1, sk_{V3,j}{}^{\frac{1}{z}}) \cdot e(g, sig_3{}^{\frac{1}{z}}) \cdot e(X_2, SCT_2)$.

4) The CS sends SCT^P to MDR.

Phase 3: *FullDecryption*.

1) After receiving SCT^P, MDR recover the EHR m by calculating $\frac{SCT_3}{(SCT^P)^z}$. Then, MDR calculate $\theta = g_1^{\alpha_j \cdot H(m)}$ to verify the correctness of m, which is the correctness of the partial ciphertext generated by CS.

2) If $\theta = sig_3$, it means that the partial ciphertext SCT^P calculated by CS is correct, otherwise, MDR discard it.

5) AUDIT

The auditor verifies the correctness and timeliness of the EHRs through the following steps.

- Extract the transaction from the blockchain and obtain the corresponding account information.
- Verify that the number of created transactions is consistent with the number of EHRs.
- Check the validity of the letter of authorization $W_{j,\ell}$.
- Verify the timeliness of medical data by checking transaction time.
- Calculate $(Bhash_T, SCT, W_j{}', W_j)$, and check whether it matches the transaction information.

E. CORRECTNESS

If the set $\widetilde{A_V}$ of attributes of the MDR meets the encryption policy $(t, R_{e,j})$ and signature policy $(t, R_{s,j})$ formulated by the patient, the MDR can use their attributes and the corresponding private key to retrieve EHRs.

First, the MDR compute the transformed key $tk = (tpk, tsk)$ and send tpk to the CS. Then CS verifies the

correctness of the signature of the doctor who uploads the EHRs, as shown below.

$$\begin{aligned}\Delta_j &= e(g^{-\alpha_j\cdot\gamma_j}, g_1^{\frac{r_D-1}{\gamma_j}} g_1^{\frac{(r_D+H(m))P_{(\widetilde{A_D},R_{s,j})}(\gamma_j)}{\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}(\gamma_j+F(a_{j,k}))}})\\ &\cdot e(g^{\alpha}, g_1)^{H(m)\cdot(1-Q_1-\frac{1}{Q_1})}\\ &\cdot e(g^{\frac{r_D+H(m)}{\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}F(a_{j,k})}},\\ &\quad g_1^{\alpha_j\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}}\gamma_j+F(a_{j,k})})\\ &= e(g^{-\alpha_j\cdot\gamma_j}, g_1^{H(m)\cdot\frac{P_{(\widetilde{A_D},S_s)}(\gamma_j)}{\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}F(a_{j,k})}})\\ &\cdot e(g^{-\alpha_j\cdot\gamma_j}, g_1^{\frac{r_D-1}{\gamma_j}} g_1^{r_D\cdot\frac{P_{(\widetilde{A_D},R_{s,j})}(\gamma_j)}{\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}(\gamma_j+F(a_{j,k}))}})\\ &\cdot e(g^{\alpha_j}, g_1)^{H(m)}\cdot e(g^{\alpha_j}, g_1)^{H(m)\cdot Q_1}\cdot e(g^{\alpha_j}, g_1)^{\frac{-H(m)}{Q_1}}\\ &\cdot e(g^{\frac{r_D}{\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}F(a_{j,k})}},\\ &\quad g_1^{\alpha_j\cdot\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}}(\gamma_j+F(a_{j,k}))})\\ &\cdot e(g^{\frac{H(m)}{\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}F(a_{j,k})}},\\ &\quad g_1^{\alpha_j\cdot\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}}(\gamma_j+F(a_{j,k}))})\\ &= e(g^{\alpha_j}, g_1)^{H(m)}\cdot e(g^{\alpha_j}, g_1)^{-H(m)\cdot Q_1}\\ &\cdot e(g^{\alpha_j}, g_1)^{\frac{-H(m)}{Q_1}}\cdot e(g^{\alpha_j}, g_1)\\ &\cdot e(g^{\alpha_j}, g_1)^{H(m)\cdot\prod_{\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}F(a_{j,k})}\frac{\gamma_j+F(a_{j,k})}{F(a_{j,k})}}\\ &\cdot e(g^{\alpha_j}, g_1)^{-H(m)\cdot\prod_{\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}F(a_{j,k})}\frac{F(a_{j,k})}{F(a_{j,k})}}\\ &\cdot e(g_1^{\frac{H(m)\cdot\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}}(\gamma_j+F(a_{j,k}))}{\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))\prod_{a_{j,k}\in R_{s,j}\cup B_{k+t-1-s}\setminus A_D}F(a_{j,k})}},\\ &\quad g^{\alpha_j}) = e(g^{\alpha_j}, g_1).\end{aligned}$$

Next, CS uses the aggregation key X_2 and the transformaton key tpk to calculate the partial ciphertext SCT^P. The computation process is as follows.

$$\begin{aligned}CT^P &= e(CT_1, sk_{V3,j}^{\frac{1}{z}})\cdot e(g, sig_3^{\frac{1}{z}})\cdot e(X_2, CT_2)\\ &= e(g^{-\eta\cdot\alpha_j\cdot\gamma_j}, g_1^{\frac{r_{V,j}-1}{z\cdot\gamma_j}})\cdot e(g, g_1^{\frac{\alpha_j\cdot H(m)}{z}})\\ &\cdot e(g^{\frac{r_{V,j}}{\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))}},\\ &\quad g_1^{\eta\cdot\alpha_j\cdot\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_D}}(\gamma_j+F(a_{j,k}))})\\ &= e(g, g_1)^{-\eta\cdot\alpha_j\cdot\frac{r_{V,j}-1}{z}}\cdot e(g, g_1)^{\frac{\alpha_j\cdot H(m)}{z}}\cdot e(g, g_1)^{\frac{r_{V,j}\cdot\eta\cdot\alpha_j}{z}}\\ &= e(g, g_1)^{\frac{\eta\cdot\alpha_j}{z}}\cdot e(g, g_1)^{\frac{\alpha_j\cdot H(m)}{z}}\end{aligned}$$

After receiving the SCT^P sent by CS, MDR use the retrieval key tsk to calculate $\frac{SCT_3}{(SCT^P)^z}$. Consequently, they can obtain the EHR m.

V. SECURITY ANALYSIS

In this part, we analyze the confidentiality, unforgeability, privacy, verifiability, immutability, and timeliness of the BVOABSC scheme.

A. CONFIDENTIALITY

Theorem 1: If the aMSE-CDH problem is true, BVOABSC based on IND-CCA2 is safe.

Proof: Based on the proof method in the literature [61], the following proves that the confidentiality of our scheme can be reduced to the difficulty of the aMSE-CDH problem under the selected ciphertext attack.

The following game is completed by adversary $\mathcal{A}$ and challenger $\mathcal{C}$ interactively.

Initialization: $\mathcal{A}$ selects a signcryption attribute set S^* and shares it with $\mathcal{C}$.

Setup: This process is the same as the definition of confidentiality in the security model.

AuthoritySetup: $\mathcal{C}$ defines $F(a_{j,\xi}) = x_\xi$ for $\xi = 1, \ldots, k$ as the encoding function of the attributes. Note that the encoding of the first k-s element is opposite to the root of $f(X)$, while the encoding of the middle attribute in S^* is opposite to some roots of $g(X)$. Then, $\mathcal{C}$ defines $B_f = \{b_1, \cdots, b_f\}$, where $f \leq k-1$. If $c = 1, \ldots, k+t-1-s$, $b_c = x_{k+c}$, the value of b_c is randomly selected from $(Z/pZ)^*$ for $c = k+t-1-s, \ldots, k-1$. Therefore, $\{x_1, \ldots, x_{2k+t-1-2}, b_{k+t-s}, \ldots, b_{c-1}\}$ is different between each other. $\mathcal{C}$ defines $g = g_0^{f(\gamma_j)}$ and $g_1 = g_1'$, then calculates $\mu_j = g^{\alpha_j\gamma_j} = g_0^{\alpha_j\gamma_j\cdot f(\gamma)}$, $\Delta_j = e(g^{\alpha_j}, g_1) = e(g_0^{\alpha_j\cdot f(\gamma)}, g_1')$ and $\{g_1^{\alpha_j\gamma_j^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k\}}$. Finally, $\mathcal{C}$ sends the generated public key $PK = \{\Delta_j, \{g_1^{\alpha_j\gamma_j^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k\}}, \mu_j, F\}$ to $\mathcal{A}$.

Query Phase 1: $\mathcal{C}$ creates an empty list Tab. $\mathcal{A}$ can apply for the following queries at several times.

- **Secret Key Query O_{sk}.** $\mathcal{A}$ asks $\mathcal{C}$ for the signcryption attribute set $A_{\mathcal{A}}{}^*$,where $|A_{\mathcal{A}}{}^* \cap R_{e,j}{}^*| < t$, $|A_{\mathcal{A}}{}^* \cap R_{s,j}{}^*| < t$. $\mathcal{C}$ randomly chooses γ from $(Z/pZ)^*$. If $|A_{\mathcal{A}}{}^*| = 0$ or $Q(X) = \lambda\cdot\prod_{a\in A_A{}^*}(X+F(a_{j,k}))$, $\mathcal{C}$ defines $Q(\gamma_j) = 1$, where $\lambda = (\prod_{a\in A_A{}^*}F(a_{j,k}))^{-1}$. Then, $\mathcal{C}$ obtains $r = (\omega y\gamma_j + 1)Q(\gamma_j)$ and calculates $sk_{\mathcal{A}} = (\{g^{\frac{r}{\gamma_j+F(a_{j,k})}}\}_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_A}}, \{g_1{}^{r\gamma_j{}^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k-2\}}, g_1{}^{\frac{r-1}{\gamma_j}})$. Finally, the generated $sk_{\mathcal{A}}$ will be sent to $\mathcal{A}$.
- **Transformation Key Query O_{tk}.** $\mathcal{A}$ requests the transformation key $tk_{\mathcal{A}}$ associated with the attribute $A_{\mathcal{A}}{}^*$. $\mathcal{C}$ first looks up $(A_{\mathcal{A}}{}^*, sk_{\mathcal{A}}, tk_{\mathcal{A}})$ in Tab. If it exists, $tk_{\mathcal{A}}$ is returned to $\mathcal{A}$; otherwise, *TransformationKeyGen* is performed to generate $tk_{\mathcal{A}} = (tpk_{\mathcal{A}}, tsk_{\mathcal{A}})$, where $tpk_A = (\{g^{\frac{r}{z(\gamma_j+F(a_{j,k}))}}\}_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_A}}, \{g_1{}^{\frac{r\gamma_j{}^{\varepsilon}}{z}}\}_{\{\varepsilon=0,\cdots,k-2\}}, g_1{}^{\frac{r-1}{z\gamma_j}})$, $tsk_A = z$. Finally, $tk_{\mathcal{A}}$ is sent to $\mathcal{A}$.
- **Signcryption Query O_{SC}.** $\mathcal{A}$ requests to signcrypt m by submitting the encryption strategy $(t, R_{e,j}{}^*)$ and signature strategy $(t, R_{s,j}{}^*)$. $\mathcal{C}$ first executes *SecretGen* to generate $sk_{\mathcal{C}}$. Then, $\mathcal{C}$ calculates $X_1 = g^{\frac{r}{\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_C}}(\gamma_j+F(a_{j,k}))}}$ and $P_{(\widetilde{A_C},R_{s,j})}(\gamma_j) = \frac{1}{\gamma_j}(\prod_{a_{j,k}\in R_s\cup B_{k+t-1-s}\setminus\widetilde{A_C}}(\gamma_j+F(a_{j,k}))-\delta_1)$ to generate the signature $sig = (sig_1, sig_2, sig_3)$, where $sig_1 = X_1\cdot g^{\frac{H(m)}{\prod_{a_{j,k}\in\widetilde{AA}_j\cap\widetilde{A_C}}(\gamma_j+F(a_{j,k}))}}$, $sig_2 = sk_C\cdot\delta_2\cdot$

$g_1^{H(m)P_{(\widetilde{A_C},R_{s,j})}(\gamma_j)/\delta_1}$, and $sig_3 = g_1^{\alpha_j \cdot H(m)}$. Finally, $\mathcal{C}$ selects η_1 from Z_p^*, and returns the generated SCT to $\mathcal{A}$.

- **Decsigncryption Query O_{DS}.** $\mathcal{A}$ sends a requirement for designcrypting SCT to $\mathcal{C}$. First, $\mathcal{C}$ verifies the attribute set $A_{\mathcal{A}}{}^*$ submitted by $\mathcal{A}$. If the verification fails, $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ executes *SecretGen* to generate $sk_{\mathcal{A}}$ and runs **Designcryption** algorithm to obtain the message m. Finally, $\mathcal{C}$ sends it to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two messages of equal length m_0, m_1 and the attribute set $A_{\mathcal{A}}{}^*$ to $\mathcal{C}$. Later, $\mathcal{C}$ selects a random bit $b \in \{0, 1\}^*$, and signcrypts m_b based on the attribute set $A_{\mathcal{A}}{}^*$. Finally, the generated challenge ciphertext SCT^* is returned to $\mathcal{A}$.

Query Phase 2: Repeat the **Phase 1**. Except that the ciphertext SCT^* and attribute set $A_{\mathcal{A}}{}^*$ have been challenged cannot be asked.

Guess: $\mathcal{A}$ outputs a guessed bit b'. If $b' = b$, then $\mathcal{C}$ answers the aMSE-CDH problem with the solution of the given instance, which means $Y = e(g_0, g_1{}')^{(\kappa + Hm_b))} \cdot f^{\gamma_j}$. Otherwise, $\mathcal{C}$ answers 0, which means Y is a random element.

Due to the difficulty of the aMSE-CDH problem, it is difficult for $\mathcal{A}$ to guess γ and r selected randomly during the key generation phase. Therefore, the adversary $\mathcal{A}$ cannot guess the message m correctly. We can get the advantage of $\mathcal{C}$ as $Adv_{\mathcal{C}}^{aMSE-CDH} \geq Adv_{\mathcal{A}}$, and the advantage of $\mathcal{A}$ can be ignored.

Ultimately, it shows that our BVOABSC scheme is safe based on the aMSE-CDH problem, under the selected ciphertext attack, and satisfies the characteristics of confidentiality.

B. UNFORGEABILITY

Theorem 2: If the aMSE-CDH problem holds, BVOABSC is unforgeable under the selected message attack.

Proof: $\mathcal{C}$ exploits the adversary $\mathcal{A}$ to solve the aMSE-CDH problem. $\mathcal{A}$ tries to calculate a signcryption ciphertext, and $\mathcal{C}$ can verify the correctness of it. Our scheme inherits the unforgeability of the scheme proposed in [61], which is described in detail as follows.

The **Initialization** and **Setup** phase are consistent with the definition of confidentiality in the security model. $\mathcal{A}$ selects a set of signcryption attributes $R_j{}^*$ and t^*. Then, $\mathcal{A}$ requests the secret key by changing the threshold t^*, and utilizes different signcryption attribute sets to ask the sign ciphertext of m. $\mathcal{A}$ tries to get the secret value by executing **Secret Key Query** and **Signcryption Query** multiple times.

- **Secret Key Query O_{sk}.** $\mathcal{A}$ asks $\mathcal{C}$ for the signcryption attribute set $A_{\mathcal{A}}{}^*$ and t^*, where $|A_{\mathcal{A}}{}^* \cap R_{e,j}{}^*| < t$, $|A_{\mathcal{A}}{}^* \cap R_{s,j}{}^*| < t$. $\mathcal{C}$ obtains $sk_{\mathcal{A}} = (\{g^{\frac{r}{\gamma_j + F(a_{j,k})}}\}_{a_{j,k} \in \widetilde{AA}_j \cap \widetilde{A_{\mathcal{A}}}}, \{g_1{}^{r\gamma_j{}^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k-2\}}, g_1{}^{\frac{r-1}{\gamma_j}})$ by running the *SecretGen* algorithm and sends the generated $sk_{\mathcal{A}}$ to $\mathcal{A}$.
- **Signcryption Query O_{SC}.** $\mathcal{A}$ submits a message m, $(t, R_{e,j}{}^*)$ and $(t, R_{s,j}{}^*)$. $\mathcal{C}$ executes *SecretGen* algorithm to generate $sk_{\mathcal{C}}$. $\mathcal{C}$ first calculates $Q(\gamma_j)/\lambda = \prod_{a \in A_{\mathcal{A}}{}^*}(\gamma_j + F(a_{j,k}))$, and then executes **Signcryption** algorithm to generate the signcrypyion ciphertext SCT, as follows:

$$\begin{cases} sig_1 = g_0{}^{(\omega y \gamma_j + 1) f(\gamma_j)\lambda} \cdot g_0{}^{f(\gamma_j)\frac{\lambda H(m)}{Q(\gamma)}} \\ sig_2 = g_1{}^{\frac{(\omega y \gamma_j + 1) Q(\gamma) \cdot P_{(\widetilde{A_B},R_{s,j})}(\gamma_j)}{\prod_{a_{j,k} \in R^* \cup B_{k+t-1-s} \setminus A_B} F(a_{j,k})}} \\ \quad \cdot g_1{}^{\frac{H(m) \cdot P_{(\widetilde{A_B},R_{s,j})}(\gamma_j)}{\prod_{a_{j,k} \in R^* \cup B_{k+t-1-s} \setminus A_B} F(a_{j,k})}} \\ \quad \cdot g_1{}^{\omega y Q(\gamma)} \cdot g_1{}^{\frac{Q(\gamma)-1}{\gamma_j}} \\ sig_3 = g_1{}^{\alpha_j \cdot H(m)} \end{cases}$$

Finally, $\mathcal{C}$ calculates $SCT_1 = g^{-\eta_1 \cdot \alpha_j \cdot \gamma_j}$, $SCT_2 = g_1{}^{\eta_1 \cdot \alpha_j \cdot \prod_{a_{j,k} \in R_{e,j}} (\gamma_j + F(a_{j,k}))}$ and $SCT_3 = e(g, g_1)^{\alpha_j \cdot \eta_1} e(g, g_1)^{\alpha_j \cdot H(m)} \cdot m$. Consequently, the signcrypyion ciphertext $SCT = (SCT_1, SCT_2, SCT_3)$ is returned to $\mathcal{C}$.

Forgery. $\mathcal{A}$ attempts to generate a valid signcryption ciphertext $SCT_1{}^*, SCT_2{}^*, SCT_3{}^*$ and $sig_3{}^*$ related to the encryption strategy $(t^*, R_{e,j}{}^*)$ and the signature strategy $(t^*, R_{s,j}{}^*)$. $\mathcal{A}$ must calculate $sig_1{}^*$ and $sig_2{}^*$ if he/she wants to win the game. Therefore, $\mathcal{A}$ must solve the aMSE-CDH problem to prove that the attributes he asks satisfy $(t^*, R_j{}^*)$. Similarly, $\mathcal{A}$ has to solve the CDH problem to guess the random value in the generated signature.

Therefore, based on the difficulty of the aMSE-CDH problem, our BVOABSC scheme is unforgeable under the selected message attack.

C. PRIVACY

Theorem 3: The following proof shows that the BVOABSC scheme has computational privacy.

Proof: The security game has been introduced in security model. The adversary $\mathcal{A}$ tries to distinguish the correct signatures generated by the two same attributes that have been obtained. That is, an adversary cannot obtain the identity of the corresponding signature entity through the signature message. If the probability of $\mathcal{A}$ winning the game is negligible, it indicates that the BVOABSC scheme is computationally private.

After receiving the public key generated by $\mathcal{C}$, $\mathcal{A}$ selects the attribute set $\widetilde{A_{\mathcal{A},1}}$ and $\widetilde{A_{\mathcal{A},2}}$ that satisfy the access control policy $(t^*, R_{e,j}{}^*)$ and $(t^*, R_{s,j}{}^*)$, and sends them to $\mathcal{C}$. Then, $\mathcal{C}$ generates the private key related to the attribute set $\widetilde{A_{\mathcal{A},1}}$ and $\widetilde{A_{\mathcal{A},2}}$ as follows:

$$sk_{C,1} = (\{g^{\frac{r}{\gamma_j + F(a_{j,k})}}\}_{a_{j,k} \in \widetilde{AA}_j \cap \widetilde{A_{C,2}}}, \{g_1{}^{r\gamma_j{}^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k-2\}}, g_1{}^{\frac{r-1}{\gamma_j}})$$

$$sk_{C,2} = (\{g^{\frac{r}{\gamma_j + F(a_{j,k})}}\}_{a_{j,k} \in \widetilde{AA}_j \cap \widetilde{A_{C,2}}}, \{g_1{}^{r\gamma_j{}^{\varepsilon}}\}_{\{\varepsilon=0,\cdots,k-2\}}, g_1{}^{\frac{r-1}{\gamma_j}}).$$

Challenge: First, $\mathcal{A}$ asks the challenger $\mathcal{C}$ to inquire about the signcryption ciphertext of the message m by using $sk_{C,1}$ or $sk_{C,2}$. Then, $\mathcal{C}$ selects a random bit $b \in \{0, 1\}^*$, and generates a signcryption ciphertext CT_b by executing the **Signcryption** algorithm. $\mathcal{C}$ can obtain a valid signcryption ciphertext about

m_b since $|\widetilde{A_{\mathcal{C},b}}^* \cap R_j^*| = t^*$. Therefore, as long as the signcryption ciphertexts generated by $sk_{\mathcal{C},1}$ or $sk_{\mathcal{C},2}$ are the same, the privacy of our scheme can be proved.

$\mathcal{C}$ uses $sk_{\mathcal{C},b}$ to generate signcryption ciphertext as follows:

$$SCT_b \begin{cases} sig_1 = X_1 \cdot g^{\frac{H(m)}{\prod_{a_{j,k} \in \widetilde{AA_j} \cap \widetilde{A_{C,b}}} (\gamma_j + F(a_{j,k}))}} \\ sig_2 = sk_{C,b} \cdot \delta_2 \cdot g_1^{H(m) P_{(\widetilde{A_{C,b}}, R_{s,j})}(\gamma_j)/\delta_1} \\ sig_3 = g_1^{\alpha_j \cdot H(m)} \\ SCT_1 = g^{-\eta_1 \cdot \alpha_j \cdot \gamma_j} \\ SCT_2 = g_1^{\eta_1 \cdot \alpha_j \cdot \prod_{a_{j,k} \in \widetilde{A_{C,b}}} (\gamma_j + F(a_{j,k}))} \\ SCT_3 = e(g, g_1)^{\alpha_j \cdot \eta_1} e(g, g_1)^{\alpha_j \cdot H(m)} \cdot m \end{cases}$$

After receiving SCT_b sent by $\mathcal{C}$, the adversary $\mathcal{A}$ verifies the validity of the signature by the following formula.

$$\Delta_j = e(u_j^{-1}, sig_{1,b}) \cdot e(g^{\alpha_j}, g_1)^{H(m_1||\cdots||m_{|I|})(1 - Q_1 - \frac{1}{Q_1})}$$

$$\cdot e({sig_{1,b}}^{\delta}, {g_1}^{\alpha_j \prod_{a_{j,k} \in R_{s,j} \cup B_{k+t-1-s}} (\gamma_j + F(a_{j,k}))}).$$

We can prove the generated signatures are the same by using the attribute sets $\widetilde{A_{\mathcal{A},1}}$ and $\widetilde{A_{\mathcal{A},2}}$ since $|\widetilde{A_{\mathcal{C},b}}^* \cap R_j^*| = |\widetilde{A_{\mathcal{C},1}}^* \cap R_j^*| = |\widetilde{A_{\mathcal{C},2}}^* \cap R_j^*| = t^*$. Thus, A does not know which attribute set to use to generate the signature due to the difficulty of the CDH problem.

Therefore, our BVOABSC scheme achieves computational privacy based on the CDH assumption.

D. VERIFIABILITY

Theorem 4: This part can verify the operation of the CS.

Proof: The purpose of the adversary $\mathcal{A}$ is to forge a partially decrypted ciphertext, which can successfully pass the veryfication of the challenger $\mathcal{C}$.

We define a safe game between $\mathcal{A}$ and $\mathcal{C}$. $\mathcal{A}$ initiates to a collusion attack of hash function H to $\mathcal{C}$, which purpose is to prove that the advantage of $\mathcal{A}$ in winning the game is less than $\mathcal{C}$. The following is a specific description of the game.

$\mathcal{A}$ attempts to obtain the private information of the unsigncryption process by separately requesting *SecretGen*, *PartialDecryption* and *FullDecryption*.

Query Phase 1: The challenger $\mathcal{C}$ first creates an empty list Tab. The adversary $\mathcal{A}$ can initiate **Secret Key Query**, **Transformation Key Query**, **Signcryption Query** and **Designcryption Query** multiple times, and these queries are consistent with the definition in security model.

Query Phase 2: $\mathcal{A}$ repeats **Phase 1** many times, except that it cannot inquire the decryption of the message m_b that has been challenged.

Forgery. $\mathcal{A}$ attempts to generate a set of signcryption attributes $\widetilde{A_{\mathcal{A}}}^*$ and a valid partially decrypted ciphertext without performing *PartialDecryption* algorithm. If $\mathcal{A}$ wants to win the game, he/she must break the collusion resistance of H. Due to the collusion-resistant nature of the hash function, $\mathcal{A}$ cannot generate $H(m_{part})$, so that $V = sig_3$ can be obtained without knowing m. So if $m_{part} \neq m'$, then $V \neq sig_3$.

Therefore, based on the collusion resistance of the hash function, the BVOABSC scheme is verifiable against the attacks of CS.

E. IMMUTABILITY

The generation of EHRs in our scheme is divided into two situations, one is a doctor, and the other is multiple doctors. In the case of a doctor, it can prove to be resistant to tampering attacks. On the one hand, the BVOABSC scheme uses a secure delegation mechanism. The authorization letter prepared by the patient is sent to the doctor for diagnosis and treatment. The establishment of the authorization is based on a secure signature algorithm, which satisfies unforgeability. On the other hand, the authorization letter and medical-related information are written into the transaction on the Ethereum by using the immutability of the blockchain. If $\mathcal{A}$ wants to successfully tamper the EHRs, the security of the Ethereum blockchain must be break. The situation of multiple doctors is an extension of a single doctor. On the basis of the above description, it is necessary to ensure the chronological order of the EHRs generated by multiple doctors.

F. TIMELINESS

Each EHR generated corresponds to a transaction on the blockchain. The timeliness of an EHR reflects the time of corresponding transactions on the blockchain. Therefore, anyone can extract the time when the EHRs are generated from the transaction in the Ethereum. In addition, the EHRs of each patient are typically generated by multiple doctors. The timeliness of the EHRs can reflect the order in which the EHRs are generated, and the identity of each doctor can be traced back in order. As the medical-related information stored on the blockchain is as difficult to fork, the adversary $\mathcal{A}$ cannot destroy the timeliness of the EHRs.

VI. PERFORMANCE EVALUATION

The purpose of this section is to compare the performance of our BVOABSC scheme with the existing relevant ABSC schemes [42], [47], [49], [50], [52], [54]. The complete EHRs owned by a patient are usually generated by multiple doctors, but most EHRs protection schemes only discuss the process of generating the EHR by one doctor. Therefore, our next discussion will only focus on the generation of an EHR, which is generated after a doctor performs diagnosis and treatment. Next, we mainly evaluate the performance from two aspects of communication overhead and computational overhead. The definition of notations are shown in table 1.

Table 2 summarizes the security and functional requirements related to signer privacy protection, outsourcing computing, multiple authorities, verifiability, and immutability. The results show that our scheme meets all the above requirements. Scheme [52] does not support the privacy protection of signcryptors, [42], [50], [52], [54] cannot realize the outsourcing operations, and [52], [54] is not verifiable. Only scheme [49] and our scheme have multiple authorities. Scheme [52], [54] and our scheme use blockchain technology to ensure that EHRs cannot be tampered with, which is a feature that other schemes do not have.

TABLE 2. The functional comparison of some similar ABSC schemes.

Schemes	Signcryptor Privacy	Computation Outsourcing	Multi-Authority	Verifiablity	Immutablity
[42]	✓	✗	✗	✓	✗
[47]	✓	✓	✗	✓	✗
[49]	✓	✓	✓	✓	✗
[50]	✓	✗	✗	✓	✗
[52]	✗	✗	✗	✗	✓
[54]	✓	✗	✗	✗	✓
Ours	✓	✓	✓	✓	✓

TABLE 3. Comparisons of computation cost.

Scheme	Signcryption	User Designcryption Cost	CS Designcryption cost
[42]	$(s_e + 2s_s + 3)E_1$	$(s_s + 5)P + (2s_e + s_s + 3)E_1$	/
[47]	$(10 + 2s_e + 6s_s)E_1 + E_T$	E_1	$(s_s + 2)E_1 + (2 + 2s_s)P$
[49]	$(kE_T) + (4k + 5ks_e)E_1$	$(3k + 3ks_e)E_1 + E_T + P$	$5E_1 + (10k + 3)E_2 + kE_T + 8kP$
[50]	$(4 + 3s_e + 3s_s)E_1$	$(5 + s_s)P + 2s_eP$	/
[52]	$(s_s + 3)E_1$	$3P$	/
[54]	$(2k + 2)E_1 + E_2$	P	/
Ours	$6E_1 + 2E_T$	E_T	$E_1 + 2E_2 + 3P$

TABLE 4. Comparison of communication cost.

Scheme	Key Size	Transformation Key Size	Ciphertext Size
[42]	$(l_s + 8B_{G_T} + 4)B_{G_1}$	/	$(l_s + l_e + 4)B_{G_1}$
[47]	$(l_s + l_d + 5)B_{G_1}$	$(l_d + 4)B_{G_1}$	$2(l_e + l_s + 2)B_{G_1}$
[49]	$(k + 9)B_{G_1}$	$(k + 13)B_{G_1}$	$(l_s + 3l_e + 4)B_{G_1} + B_{G_T}$
[50]	$(2l_s + 2l_d + 4)B_{G_1}$	/	$(3l_s + 3l_e + 5)B_{G_1}$
[52]	$(4 + l_d)B_{G_1}$	/	$(l_e + 3)B_{G_1}$
[54]	$2B_{G_1}$	/	$(l_e + 2)B_{G_T}$
Ours	kB_{G_1}	$(k + 1)B_{G_1}$	$8B_{G_T}$

In table 3, we compare these schemes in terms of computational overhead, mainly considering the cost of signcryption, user unsigncryption, and CS unsigncryption. For signcryption, our scheme has lower computational overhead than schemes [42], [47], [49], [50], [52], [54], and only needs to perform $6E_1 + 2E_T$ operations. Moreover, the BVOABSC scheme uses attribute-based outsourcing unsigncryption technology to allow CS to undertake greater computational pressure and let users perform only one E_T operation to complete decryption. Without this technology, the schemes [42], [50], [52], [54] make users face a huge computational burden. Although the schemes [47], [49] allow CS to perform the partial decryption operation, the calculation cost for users to complete the remaining part of the ciphertext is still higher than our scheme.

Table 4 describes the communication cost comparison. Since the CS performs a partial designcryption operation, it needs to generate the overhead of transformed key. Schemes [47], [49] and our scheme use outsourcing computing technology, but our scheme has a lower cost of generating signcryption ciphertext. The size of the ciphertext generated by the BVOABSC scheme is constant and will not change with the growth of the number of attributes. Therefore, our scheme has greater advantages in terms of communication overhead.

For the comparisons of computation effciency, a simulation experiment was implemented, which uses Stanford Pairing-Based Crypto (PBC) library [62] in VC++ 6.0. The computer provides 3GHz Intel Core i5-7400 CPU with 4GB RAM and 64-bit Windows 10 operating system for our implementation. Furthermore, type A bilinear [62] is used in our experiments, and the experimental results are shown in Fig.4. We can know that the time of P, E_1 and E_2 are 11.23ms, 5.62ms and 1.56ms, respectively, which was executed 10 trials.

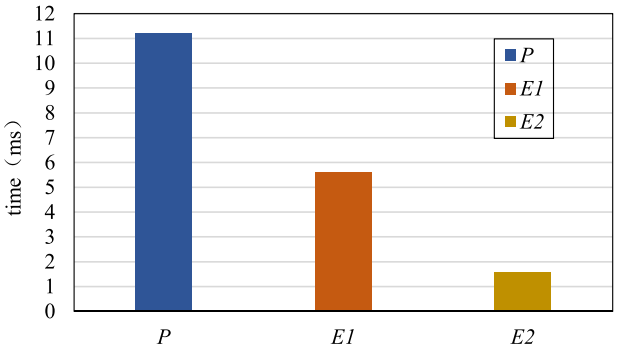

FIGURE 4. Performance evaluation benchmark.

EHRs usually contain many different types of data, that is, a large number of attributes. This will result in a lot of time on signcryption of these data. In response to this situation, our BVOABSC scheme provides a good solution. The

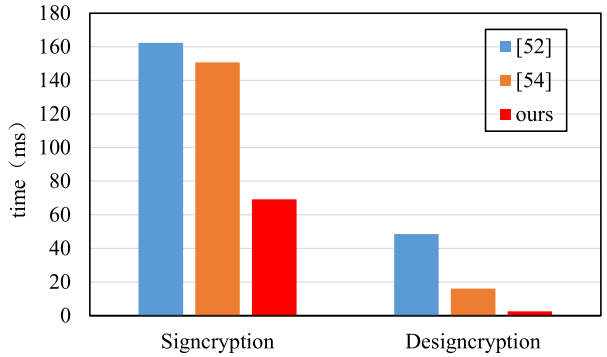

FIGURE 5. Time cost in signcryption and decryption.

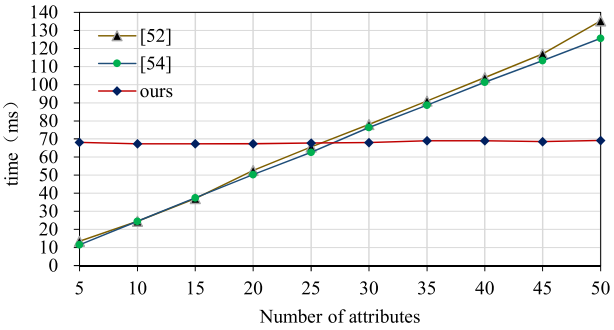

FIGURE 6. Time cost in signcryption.

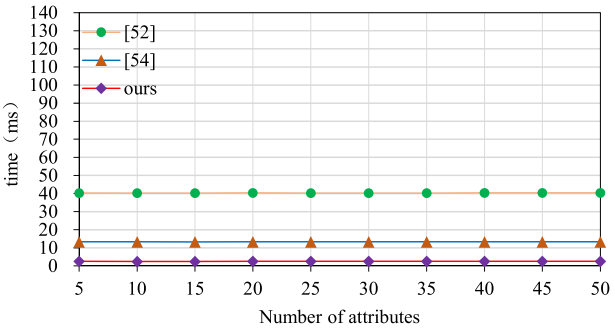

FIGURE 7. Time cost in decryption.

length of the signcryption ciphertext generated is constant, which has nothing to do with the number of attributes, so the signcryption time in our scheme is constant. In addition, users only need to perform an exponential calculation on G_T to obtain EHRs by using outsourcing computation. The BVOABSC scheme and schemes [52], [54] compare the time of signcryption and unsigncryption phase because they all use blockchain technology. It can be found from the Fig.5 that when the number of attributes is 60, our scheme requires 69.24ms and 2.53ms for signcryption and unsigncryption (user side), respectively, which is lower than the time spent on [52] and [54]. Therefore, when the number of attributes is large, our scheme has a considerable time advantage compared with similar schemes.

The Fig.6 shows the time taken to perform the signcryption operation. When the number of attributes is 25, our scheme takes more time than schemes [52] and [54]. Nevertheless, the length of the signcryption ciphertext of [52] and [54] is positively related to the number of attributes. Thus, as the number of attributes increases, they take longer. However, the signcryption time of our scheme remains constant. When the number of attributes is higher than 25, the advantages of our scheme gradually appear.

From the Fig.7, we can clearly see that the time taken for the three schemes to perform designcryption (user side) operations has nothing to do with the attributes, but the time spent advantage of our scheme has always been higher than schemes [52] and [54].

VII. CONCLUSION

We propose an attribute-based verifiable outsourcing signcryption scheme based on blockchain, which realizes the secure storage and sharing of EHRs in the cloud. The proposed scheme has the advantages of attribute-based signcryption, blockchain and cloud storage, which meets the characteristics of fine-grained access control, confidentiality, unforgeability, verifiability, privacy protection and non-tampering. Moreover, our scheme uses outsourcing computing technology to allow most of the calculations to be performed by CS, alleviating the user's computing burden. In addition, our scheme has been proven to be safe in the standard model. Consequently, performance analysis shows that our scheme has high efficiency and can be applied in practice.

REFERENCES

[1] D. Ivan, "Moving toward a blockchain-based method for the secure storage of patient records," in *Proc. ONC/NIST Blockchain Healthcare Res. Workshop*, 2016, pp. 1–11.

[2] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, p. 218, Oct. 2016.

[3] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Syst. J.*, vol. 11, no. 1, pp. 88–95, Mar. 2017.

[4] F. Xhafa, J. Feng, Y. Zhang, X. Chen, and J. Li, "Privacy-aware attribute-based PHR sharing with user accountability in cloud computing," *J. Supercomput.*, vol. 71, no. 5, pp. 1607–1619, May 2015.

[5] J. Hua, G. Shi, H. Zhu, F. Wang, X. Liu, and H. Li, "CAMPS: Efficient and privacy-preserving medical primary diagnosis over outsourced cloud," *Inf. Sci.*, vol. 527, pp. 560–575, Jul. 2020.

[6] S. Biswas, Anisuzzaman, T. Akhter, M. S. Kaiser, and S. A. Mamun, "Cloud based healthcare application architecture and electronic medical record mining: An integrated approach to improve healthcare system," in *Proc. 17th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dhaka, Bangladesh, Dec. 2014, pp. 286–291.

[7] C. He, X. Fan, and Y. Li, "Toward ubiquitous healthcare services with a novel efficient cloud platform," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 230–234, Jan. 2013.

[8] A. N. Khan, M. L. M. Kiah, M. Ali, S. A. Madani, A. U. R. Khan, and S. Shamshirband, "BSS: Block-based sharing scheme for secure data storage services in mobile cloud environment," *J. Supercomput.*, vol. 70, no. 2, pp. 946–976, Nov. 2014.

[9] S. Cao, G. Zhang, P. Liu, X. Zhang, and F. Neri, "Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain," *Inf. Sci.*, vol. 485, pp. 427–440, Jun. 2019.

[10] B. Bera, S. Saha, A. K. Das, N. Kumar, P. Lorenz, and M. Alazab, "Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled Internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9097–9111, Aug. 2020.
[11] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "BlocHIE: A BLOCkchain-based platform for healthcare information exchange," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2018, pp. 49–56.
[12] 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 (OBD)*, Aug. 2016, pp. 25–30.
[13] K. Peterson, R. Deeduvanu, P. Kanjamala, and K. Boles, "A blockchain-based approach to health information exchange networks," in *Proc. NIST Workshop Blockchain Healthcare*, vol. 1, 2016, pp. 1–10.
[14] 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, May 2018.
[15] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.
[16] A. Roehrs, C. A. da Costa, R. da Rosa Righi, V. F. da Silva, J. R. Goldim, and D. C. Schmidt, "Analyzing the performance of a blockchain-based personal health record implementation," *J. Biomed. Informat.*, vol. 92, Apr. 2019, Art. no. 103140.
[17] T. Aste, P. Tasca, and T. Di Matteo, "Blockchain technologies: The foreseeable impact on society and industry," *Computer*, vol. 50, no. 9, pp. 18–28, 2017.
[18] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 180–184.
[19] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.
[20] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K.-R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan. 2018.
[21] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.
[22] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016.
[23] M. A. Alsmirat, Y. Jararweh, I. Obaidat, and B. B. Gupta, "Internet of surveillance: A cloud supported large-scale wireless surveillance system," *J. Supercomput.*, vol. 73, no. 3, pp. 973–992, Mar. 2017.
[24] A. A. Boxwala, J. Kim, J. M. Grillo, and L. Ohno-Machado, "Using statistical and machine learning to help institutions detect suspicious access to electronic health records," *J. Amer. Med. Inform. Assoc.*, vol. 18, no. 4, pp. 498–505, 2011.
[25] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.
[26] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. EUROCRYPT*, Aarhus, Denmark, 2005, pp. 457–473.
[27] S. Chatterjee and A. K. Das, "An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks," *Secur. Commun. Netw.*, vol. 8, no. 9, pp. 1752–1771, Jun. 2015.
[28] X. Liang, M. Barua, R. Lu, X. Lin, and X. Shen, "HealthShare: Achieving secure and privacy-preserving health information sharing through health social networks," *Comput. Commun.*, vol. 35, no. 15, pp. 1910–1920, Sep. 2012.
[29] R. Lu, X. Lin, and X. Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 614–624, Mar. 2013.
[30] Y. Liu, Y. Zhang, J. Ling, and Z. Liu, "Secure and fine-grained access control on e-healthcare records in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 1020–1026, Jan. 2018.
[31] P. Yang, Z. Cao, and X. Dong, "Fuzzy Identity Based Signature," IACR Cryptol. EPrint Arch., Tech. Rep. 2008/002, 2008, p. 2. [Online]. Available: http://eprint.iacr.org/2008/002
[32] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Proc. Cryptogr. Track RSA Conf.*, 2011, pp. 376–392.
[33] S. F. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," in *Proc. Int. Conf. Cryptol.*, Africa, 2009, pp. 198–216.
[34] T. Okamoto and K. Takashima, "Decentralized attribute-based signatures," in *Public-Key Cryptography*. Berlin, Germany: Springer, 2013, pp. 125–142.
[35] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41678–41689, 2019.
[36] J. Liu, J. Ma, W. Wu, X. Chen, X. Huang, and L. Xu, "Protecting mobile health records in cloud computing: A secure, efficient, and anonymous design," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 2, pp. 1–20, Apr. 2017, Art. no. 57.
[37] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption)," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, 1997, pp. 165–179.
[38] M. Gagné, S. Narayan, and R. Safavi-Naini, "Threshold attribute-based signcryption," in *Security and Cryptography for Networks*, vol. 6280. Berlin, Germany: Springer, 2010, pp. 154–171.
[39] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K.-K.-R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3184–3197, Apr. 2020.
[40] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area networksecurity: A fuzzy attribute-based signcryption scheme," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 37–46, Sep. 2013.
[41] Y. S. Rao and R. Dutta, "Efficient attribute-based signature and signcryption realizing expressive access structures," *Int. J. Inf. Secur.*, vol. 15, no. 1, pp. 81–109, Feb. 2016.
[42] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: Ciphertext-policy attribute-based signcryption," *Future Gener. Comput. Syst.*, vol. 52, pp. 67–76, Nov. 2015.
[43] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Generat. Comput. Syst.*, vol. 67, pp. 133–151, Feb. 2017.
[44] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. USENIX Secur. Symp.*, no. 3, 2011, p. 34.
[45] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," *Inf. Sci.*, vol. 462, pp. 262–277, Sep. 2018.
[46] Y. Zhang, R. Deng, X. Liu, and D. Zheng, "Outsourcing service fair payment based on blockchain and its applications in cloud computing," *IEEE Trans. Services Comput.*, early access, Aug. 7, 2018, doi: 10.1109/tsc.2018.2864191.
[47] F. Deng, Y. Wang, L. Peng, H. Xiong, J. Geng, and Z. Qin, "Ciphertext-policy attribute-based signcryption with verifiable outsourced designcryption for sharing personal health records," *IEEE Access*, vol. 6, pp. 39473–39486, 2018.
[48] X. Liu, Y. Xia, Z. Sun, and X. Liu, "Provably secure attribute based signcryption with delegated computation and efficient key updating," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 5, pp. 2646–2659, 2017.
[49] Q. Xu, C. Tan, Z. Fan, W. Zhu, Y. Xiao, and F. Cheng, "Secure multi-authority data access control scheme in cloud storage system based on attribute-based signcryption," *IEEE Access*, vol. 6, pp. 34051–34074, 2018.
[50] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Gener. Comput. Syst.*, vol. 67, pp. 133–151, Feb. 2017.
[51] A. Ekblaw, A. Azaria, J. D. Halamka, and A. Lippman, "A case study for-blockchain in healthcare:medrec prototype for electronic health recordsand medical research data," in *Proc. IEEE Open Big DataConference*, 2016.
[52] H. Wang and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *J. Med. Syst.*, vol. 42, no. 8, pp. 152–161, Aug. 2018.
[53] X. Yang, T. Li, X. Pei, L. Wen, and C. Wang, "Medical data sharing scheme based on attribute cryptosystem and blockchain technology," *IEEE Access*, vol. 8, pp. 45468–45476, 2020.
[54] N. Eltayieb, R. Elhabob, A. Hassan, and F. Li, "A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud," *J. Syst. Archit.*, vol. 102, Jan. 2020, Art. no. 101653.
[55] D. Boneh, I. Mironov, and V. Shoup, "A secure signature scheme from bilinear maps," in *Proc. CT-RSA*, 2003, pp. 98–110.

[56] C. Hazay, A. López-Alt, H. Wee, and D. Wichs, "Leakage-resilient cryptography from minimal assumptions," *J. Cryptol.*, vol. 29, no. 3, pp. 514–551, Jul. 2016.
[57] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Public Key Cryptography-PKC*. Paris, France: Springer, 2010, pp. 19–34.
[58] C. Delerablée and D. Pointcheval, "Dynamic threshold public-key encryption," in *Proc. CRYPTO*, 2008, pp. 317–334.
[59] G. Wood, "Ethereum: A secure decentralised generalised transaction-ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.
[60] C. Delerablśe, P. Paillier, and D. Pointcheval, "Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys," in *Proc. 1st Int. Conf. Pairing-Based Cryptogr.*, 2007, pp. 39–59.
[61] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "Constant-size threshold attribute based signcryption for cloud applications," in *Proc. 14th Int. Joint Conf. e-Bus. Telecommun.*, Madrid, Spain, vol. 4, 2017, pp. 212–225.
[62] B. Lynn. *The Stanford Pairing Based Crypto Library*. Accessed: Nov. 2017. [Online]. Available: http://crypto.stanford.edu/pbc/

XIAODONG YANG (Member, IEEE) received the B.S. degree in mathematics from Northwest Normal University, China, in 2002, the M.S. degree in cryptography from Tongji University, China, in 2005, and the Ph.D. degree in cryptography from Northwest Normal University, in 2010. He is currently a Postdoctoral Fellow with the State Key Laboratory of Cryptology of China and a Professor in information and computer science with Northwest Normal University. His research interests include applied cryptography, network security, and cloud computing security. He is a member of the Chinese Cryptology and Information Security Association.

TING LI received the B.S. degree from Zhengzhou Normal University, Zhengzhou, China, in 2018. She is currently pursuing the master's degree in computer science with Northwest Normal University. Her current research interests include network security and blockchain technology and their applications.

WANTING XI received the B.S. degree from the Shaanxi University of Technology, Shaanxi, China, in 2019. She is currently pursuing the master's degree in computer science with Northwest Normal University. Her current research interest includes attribute-based encryption.

AIJIA CHEN was born in Lanzhou, China, in 1995. She is currently pursuing the master's degree with the College of Computer Science and Engineering, Northwest Normal University, Lanzhou. Her research interests include cryptology and information security.

CAIFEN WANG received the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2003. She is currently a Professor in computer science with Shenzhen Technology University. Her current research interests include network security, cryptographic protocols, and security engineering. She is a member of the Chinese Cryptology and Information Security Association.