Received 1 July 2023, accepted 13 July 2023, date of publication 19 July 2023, date of current version 10 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296781

RESEARCH ARTICLE

A Personal Privacy Data Protection Scheme for Encryption and Revocation of High-Dimensional Attribute Domains

CAIMEI WANG, JIANHAO LU, XINLU LI, PEI CAO, ZIJIAN ZHOU, AND QILUE WEN
School of Artificial Intelligence and Big Data, Hefei University, Hefei 230601, China

Corresponding author: Caimei Wang (wangcm@hfuu.edu.cn)

This work was supported in part by the Talent Research Fund Project of Hefei University under Grant 21-22RC19, and in part by the University Natural Sciences Research Project of Anhui Province under Project KJ2021ZD0118.

ABSTRACT With the frequent occurrence of private data breaches, it is now more necessary than ever to address how to protect private data. The combination of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and blockchain typically enables secure storage and sharing of data. However, in high-dimensional attribute domains, that is, the number of attributes is large, these schemes have issues such as low security of data protection, high computational overhead, and high cost of attribute revocation. This paper proposes a personal privacy data protection scheme for encryption and revocation of high-dimensional attribute domains to address these issues. The proposed scheme is made up of three components. Firstly, Fast High-dimensional Attribute Domain-based Message Encryption (HAD-FME) is proposed to improve data security and reduce computational cost. Secondly, an Attribute Revocation Mechanism Based on Sentry Mode (SM-ARM) is designed in combination with smart contracts. Lastly, a Blockchain-based Model for Personal Privacy Data Protection (BC-PPDP) is proposed by integrating HAD-FME with SM-ARM. The security analysis results show that HAD-FME proposed in this paper is secure under the DLIN assumption, and the attribute revocation satisfies both forward and backward security. Experiments show that HAD-FME has higher computational efficiency than existing schemes in the high-dimensional attribute domains, SM-ARM has lower revocation cost than existing attribute revocation mechanisms, and smart contracts and blockchain work well.

INDEX TERMS Privacy data, blockchain, attribute-based encryption, data storage and sharing, attribute revocation.

I. INTRODUCTION

The rapid development of technologies such as cloud computing and the Internet of Things (IoT) has led to the generation of a large amount of personal data worldwide. Enterprises continuously collect and analyze these personal data, providing them with professional services and generating significant economic benefits, enabling users and enterprises to gain huge profits from the information society. Unfortunately, in recent years, enterprises' lack of data protection measures, such as storing data in plaintext on their centralized servers, has led to an increasing number of personal data leakage incidents. Therefore, sharing and storing private data in a secure manner is critical. Blockchain, as a decentralised ledger database, due to its characteristics of decentralization and difficulty in tampering with data, can provide a trustworthy data storage and sharing environment. Currently, many researchers have used blockchain in various fields, including data storage, the Internet of Things, healthcare, transactions, and payments [1], [2], [3], [4], [5]. At the same time, many scholars have done a lot of research on tamper-resistance ledger databases [6], [7], [8].

However, if the data owner explicitly stores information related to private data on the blockchain, any user can access the data information. This may result in the data owner losing control over personal data. Ciphertext-Policy

The associate editor coordinating the review of this manuscript and approving it for publication was Rahim Rahmani.

Attribute-Based Encryption (CP-ABE) [9] was proposed as a solution by Bethencourt. In CP-ABE, the data owner can choose the ciphertext access method, in which the access policy is included in the ciphertext and the user attribute set is embedded in the key. The decryption process can only be completed when the user attribute set meets the access policy. In this paper, the set of attributes and the number of attributes is referred to as the attribute domain and attribute domain dimension, respectively. The Fast Attribute-based Message Encryption (FAME) [10] and other CP-ABE algorithms were subsequently proposed [11], [12]. Compared with the scheme in [9], FAME is constructed using asymmetric prime-order bilinear groups, which have stronger security requirements.

In order to achieve secure storage, distribution, and control of personal data, several data protection schemes have been proposed by combining blockchain and CP-ABE. In order to provide access control for IoT data, Zhang et al. [13] proposed a blockchain-based access control list and paired it with CP-ABE. In order to provide users with a safe storage environment, Sharma et al. [14] proposed a decentralized cloud storage architecture based on blockchain, which uses CP-ABE to encrypt data. Assuring data security by hybrid encryption of CP-ABE and AES, Lu and Fu [15] proposed a data access control mechanism based on attribute-based encryption and blockchain. The CP-ABE schemes used in [13], [14], and [15] are based on symmetric prime-order bilinear groups. If they are directly transformed into an even more secure scheme based on asymmetric prime-order bilinear groups, the computational cost will be significantly greater than that of the present schemes. Therefore, they are not suitable for data sharing in high-dimensional attribute domains.

The information in the attribute domain of the data owner usually changes dynamically, when the attributes change, the access rights for data users also fluctuate dynamically. Therefore, the issue of attribute revocation is another highly concerning research topic. Attribute revocation can be divided into three types based on the influence range of attribute revocation: system attribute revocation, user revocation, and user attribute revocation. Yang et al. [16] proposed a blockchain-based crowdsourced data storage and sharing scheme. In this scheme, the CP-ABE algorithm is improved to determine whether the user has access rights by using the matching mechanism of version key and ciphertext. Therefore, when the attribute is revoked, the key and ciphertext must be updated simultaneously, which also leads to a large revocation cost. Liu et al. [17] proposed a CP-ABE scheme that supports outsourcing decryption and attribute revocation. In this paper, when the attribute is revoked, the user's upgrade ciphertext and conversion key will be modified based on the attribute version number. Zheng et al. [18] proposed a cloud-assisted CP-ABE framework, where user revocation is accomplished by developing a time-based conversion key and embedding a revocation list in the ciphertext. Therefore, a low-cost and fine-grained user attribute revocation mechanism urgently needs to be designed.

In summary, to address these issues, such as low security of data protection, high computational overhead, and high cost of attribute revocation, we propose a personal privacy data protection scheme for encryption and revocation of high-dimensional attribute domains. The main contributions of the paper are as follows:

1) A Blockchain-based Model for Personal Privacy Data Protection (BC-PPDP) is proposed, which is combined with cryptographic algorithms. The model can ensure that data users retain control over information related to their private data in the blockchain, and use smart contracts to determine the access rights of data users.
2) Based on FAME and SM4 [19], we present the Fast High-dimensional Attribute Domain-based Message Encryption (HAD-FME) algorithm. This algorithm has stronger security and better data-sharing performance than existing algorithms in high-dimensional attribute domains.
3) We propose an Attribute Revocation Mechanism Based on Sentry Mode (SM-ARM) based on HAD-FME. When generating secret keys for data users, HAD-FME adds a fixed timestamp element and provides a passing attribute for the version key. When a data user accesses data, smart contracts can automatically determine the validity of the pass. This access method is referred to as sentinel mode in this paper. Since SM-ARM only needs to update the key when revoking a data user's attributes, the cost of attribute revocation is reduced.
4) To evaluate the effectiveness of our scheme, we conduct security analysis and extensive experiments. The scheme proposed in this paper is secure under the DLIN assumption and satisfies tamper resistance. Compared with existing algorithms, HAD-FME has faster processing performance, while SM-ARM provides forward and backward security and achieves low-cost attribute revocation.

II. RELATED WORK

This paper focuses on the current data protection schemes based on blockchain and CP-ABE. Data protection schemes, CP-ABE-based attribute revocation and verifiable ledger databases as related work of this paper will be introduced in the following.

A. DATA PROTECTION SCHEMES

Many privacy data security schemes have been promoted to more expansive fields like medical care and scientific research [20], [21], as user privacy data is gathered and used in an increasing number of companies. Chen et al. [22] presented an efficient CP-ABE scheme in cloud storage with shared decryption. Instead of simply one specified user, this scheme uses numerous alternate users to decrypt the ciphertext. By utilising an integrated access tree, this decryption approach improves the scheme's security while also reducing the computational cost. Technologies like CP-ABE and symmetrical encryption were used by Lee et al. [23] to protect the privacy and secrecy of blockchain. Wang et al. [24] proposed the RCP-ABE personal privacy data protection system, which substituted conventional third parties with smart contracts to accomplish access control of user data. Kang et al. [25] proposed a traceable and forward-secure attribute-based signature scheme with constant size, it solves the issues of abusing signature and key exposure in existing Attribute-Based Signature (ABS) schemes. Zhang et al. [26] proposed an agricultural products supply chain traceability system based on blockchain and CP-ABE. However, once the security of these schemes in the [23], [24], [26] is enhanced, they will suffer from high computing overhead during the encryption, decryption and key generation phase when used in high dimensional attribute domains.

B. CP-ABE-BASED ATTRIBUTE REVOCATION

One of the main research points of CP-ABE is attribute revocation, Qian et al. [27] proposed a privacy-preserving personal health record using multi-authority attribute-based encryption with revocation, which supports efficient revocation at both the user and attribute levels. An attribute-revocation-compliant cloud storage system was designed by Chen et al. [28], which refreshed the data user's right to access the private data only if their attribute was non-revoking. The ciphertext was updated by randomly creating a one-time re-encryption key that was connected with the data user's attributes. Lian et al. [29] proposed a CP-ABE scheme with user attribute revocation. They divided the master key into a delegation key and a secret key and updated the ciphertext and the delegation key by setting the data re-encryption algorithm. Li et al. [30] presented user collusion avoidance CP-ABE with efficient attribute revocation for cloud storage, which makes use of attribute groups and binds users' private keys with group keys. It solves the issue of a user's single attribute revocation affecting other users in the system who have the same attributes. A method for using CP-ABE in resource-constrained IoT devices was presented by Fischer et al. [31], which called for an attribute delegation centre to carry out a user key update algorithm and a proxy server to carry out a ciphertext update algorithm. In the attribute revocation schemes [28], [29], [31], the proxy server performs a second encryption on the relevant ciphertexts and updates the user keys, the computational cost of these schemes needs to be reduced.

C. VERIFIABLE LEDGER DATABASES

Regarding the ledger databases, Fekete and Kiss [32] point out they can be divided into two categories. The first is permission blockchain technology-based Decentralised Ledger Technology (DLT). Centralised Ledger Databases (CLD)-based Centralised Ledger Technology (CLT) is the second of them. Gorbunova et al. [33] stated that one of the vital DLT aspects is the capacity to offer an immutable and widely verifiable ledger for larger-scale and highly complex systems. However, DLT has low performance and transaction throughput. To address the problem of low throughput, high latency, and large storage overhead in systems, Yang et al. [7] proposed LedgerDB, a centralised ledger database with tamper-evidence and non-repudiation features similar to blockchain. Based on [7], Yang et al. [34] proposed ubiquitous verification in centralised ledger databases to address the shortcoming of high verification cost. In addition, researchers have begun to consider how to construct distributed ledger data with high performance and throughput. Three current types of verifiable ledger databases, such as blockchain, a certificate transparency log, and Amazon's Quantum Ledger Database (QLDB) [35], suffer from a lack of transaction support and inefficiency. To address these issues, Yue et al. [8] design a distributed database system GlassDB, an efficient verifiable ledger database system through transparency. However, how to construct a blockchain-based decentralised high-performance and throughput ledger database remains a challenge in current blockchain and CP-ABE-based decentralised privacy data protection systems.

III. PRELIMINARIES

A. BILINEAR MAPPING

Suppose $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are finite multiplicative cyclic groups of prime order p respectively. The bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is established then:

1) Bilinear: $\forall g_1 \in \mathbb{G}_1, \quad g_2 \in \mathbb{G}_2$ and $\forall a, b \in \mathbb{Z}_p$ satisfy $e\left(g_1^a, g_2^b\right) = e\left(g_1, g_2\right)^{ab}$.

2) Non-degeneration: $\exists g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ such that $e\left(g_1, g_2\right) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ represents the identity element in group $\mathbb{G}_T$.

3) Computability: There exists an algorithm $\forall g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$ that can get $e\left(g_1, g_2\right)$ through calculation.

If $\mathbb{G}_1 = \mathbb{G}_2$ the above bilinear mapping is symmetric, otherwise it is asymmetric.

B. LINEAR SECRET-SHARING SCHEMES (LSSS)

A secret-sharing scheme $\prod$ over a set of parties P is called liner (over $\mathbb{Z}_p$) if

1) The shares for each party from a vector over $\mathbb{Z}_p$.

2) There exists a matrix an M with l row and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, l$, the i-th row of M, we let the function ρ defined the party labelling row i as $\rho(i)$. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Mv is the vector of l shares of the secret s according to Π. The share $(Mv)_i$ belongs to party $\rho(i)$.

The [36] shows that LSSS maintains a linear reconstruction. If Π is an LSSS defined on the access structure $\mathbb{A}, I \subset \{1, 2 \ldots, l\}$ and $I = \{i : \rho(i) \in S\}$. Then there exists a set of constants $\left\{y_i \in \mathbb{Z}_p\right\}_{i \in I}$ such that Eq. 1.

$$\sum_{i \in I} y_i (M)_i = (1, 0, \ldots, 0). \tag{1}$$

where $(M)_i$ denotes the i-th row of the matrix M. Thus, for any valid sharing $\{\lambda_i \in Mv_i\}_{i \in I}$ of the secret s, there is $\sum_{i \in I} y_i \lambda_i = s$. In this paper, (M, ρ) is used to denote the access structure.

C. FAST ATTRIBUTE-BASED MESSAGE ENCRYPTION

The FAME [10] employs an asymmetric prime order bilinear group, which is more secure than the CP-ABE method of the [9]. The computational cost is decreased by only needing six pairing operations to finish the decryption. The following four methods provide the message space's *msg*.

- Setup $(1^\lambda) \rightarrow (pk, msk)$: Outputs a public key and a master key with the security field 1^λ as input.
- KeyGen $(msk, S) \rightarrow sk$: Input the master key *msk* and the set of attributes S, and output the secret key sk.
- Encrypt $(pk, \mathbb{A}, msg) \rightarrow CT_{FAME}$: Input pk, access structure $\mathbb{A}$ and plaintext *msg*, output ciphertext CT_{FAME}.
- Decrypt $(pk, CT_{FAME}, sk) \rightarrow msg$: Input pk, CT_{FAME}, and sk, output message *msg* for successful decryption, otherwise output special symbol $\perp$.

D. DECISIONAL LINEAR ASSUMPTION (DLIN)

An asymmetric pairing group generator GroupGen satisfies the decisional linear (DLIN) assumption if for all Probabilistic Polynomial-Time (PPT) adversaries,

$$\begin{aligned} Adv^{\mathcal{A}}_{DLIN}(\lambda) =| & Pr\left[\mathcal{A}\left(1^\lambda, par, D, T_0\right) = 1\right] \\ & - Pr\left[\mathcal{A}\left(1^\lambda, par, D, T_1\right) = 1\right] | \end{aligned} \quad (2)$$

is negligible in λ, where $par = (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h) \leftarrow \text{GroupGen}\left(1^\lambda\right), a_1, a_2 \in \mathbb{Z}_p^*, s_1, s_2, s \in \mathbb{Z}, D = (g^{a_1}, g^{a_2}, h^{a_1}, h^{a_2}, g^{a_1 s_1}, g^{a_2 s_2}, h^{a_1 s_1}, h^{a_2 s_2}), T_0 = \left(g^{s_1+s_2}, h^{s_1+s_2}\right)$, $T_1 = (g^s, h^s)$.

IV. SCHEME DEFINITION

A. MODEL DEFINITION

A personal privacy data protection scheme for encryption and revocation of high-dimensional attribute domains, including BC-PPDP, HAD-FME, and SM-ARM, is proposed in this paper. The BC-PPDP is shown in Figure 1. The proposed model consists of five entities, HAD-FME, Blockchain and SM-ARM as its core technologies. The model can be properly expressed as

$$\begin{aligned} \text{BC} - \text{PPDP} = \{ & \text{Entities (DO, DU, STSS, HF, IPFS)}, \\ & \text{CoreTechnologies (HAD} - \text{FME}, \\ & \text{SM} - \text{ARM, Blockchain)}\}. \end{aligned}$$

There are five entities mentioned, namely Data Owners (DO), Data Users (DU), Semi-trusted System Servers (STSS), Hyperleger Fabric (HF), and InterPlanetary File System (IPFS). The following are the roles played by each entity.

1) **DO.** DO is trustworthy and is mostly in charge of symmetric key encryption, private data uploading to IPFS, and data list uploading to HF.
2) **DU.** After registering, DU will acquire the version secret key linked to its attribute domain, calculate its hash value as a pass, and only the DU secret key will be decrypted to obtain the symmetric key once the AccessDecision contract determines that it is current and complies with the access policy, and then obtain the privacy data.
3) **STSS.** Semi-trusted STSS mainly performs two parts: the first is initialization and creation of DU's secret key, and the second is decryption to obtain a symmetric key. Semi-trustworthy is the assumption that the attacker will illegally tamper with the DU secret key data within the STSS. The attribute is revoked for the DU based on the access right decision made in this document using STSS as a sentinel in conjunction with the pass.
4) **HF.** The blockchain makes use of the Hyperleger Fabric. The main goal of HF is to enable smart contract-based storage, query, and updating of private data and symmetric key index data, as well as access decision-making based on version keys. The specific content of the smart contracts are shown in Figure 1.
5) **IPFS.** IPFS is mainly responsible for storing personal privacy data ciphertext.

B. ALGORITHM DEFINITION

This section proposes HAD-FME based on the FAME. Using SM4 to encrypt private data and FAME to encrypt the symmetric key, BC-PPDP uses HAD-FME as a key technology to ensure the secure storage and sharing of private data. In order to accomplish low-cost attribute revocation, HAD-FME further adds a fixed timestamp attribute to DU. Then, we use the SM3 algorithm [37] to obtain the DU key hash and use it as a pass to evaluate whether the DU has access rights. HAD-FME is mainly composed of the following eight algorithms.

- Setup $(1^\lambda) \rightarrow (pk, msk)$: Output a public key and a master key with the security field 1^λ as input.
- KeyGen$_{\text{FAME}}$ $(msk, S) \rightarrow sk$: Input the master key *msk* and the attribute domain S, and output the secret key sk. Where the attribute domain contains the fixed timestamp attribute field y_{time}.
- SM3 $(sk) \rightarrow Hash_{sk}$: Input the sk, output the hash value of the sk $Hash_{sk}$.
- KeyGen$_{\text{SM4}}$ $\rightarrow key$: Randomly generate symmetric key *key*.
- Enc$_{\text{SM4}}$ $\rightarrow$ *(Data, key) CT*: Input personal privacy data Data and symmetric key *key*, output ciphertext CT.
- Enc$_{\text{FAME}}$ $(pk, \mathbb{A}, key) \rightarrow Ckey$: Input pk, access structure $\mathbb{A}$ and *key*, output symmetric key cipher *Ckey*.
- Dec$_{\text{FAME}}$ $(pk, Ckey, sk) \rightarrow key$: Input the pk, the *Ckey* and the sk, and output the key.
- Dec$_{\text{SM}_4}$$(CT, key) \rightarrow Data$: Input CT and *key*, output *Data*.

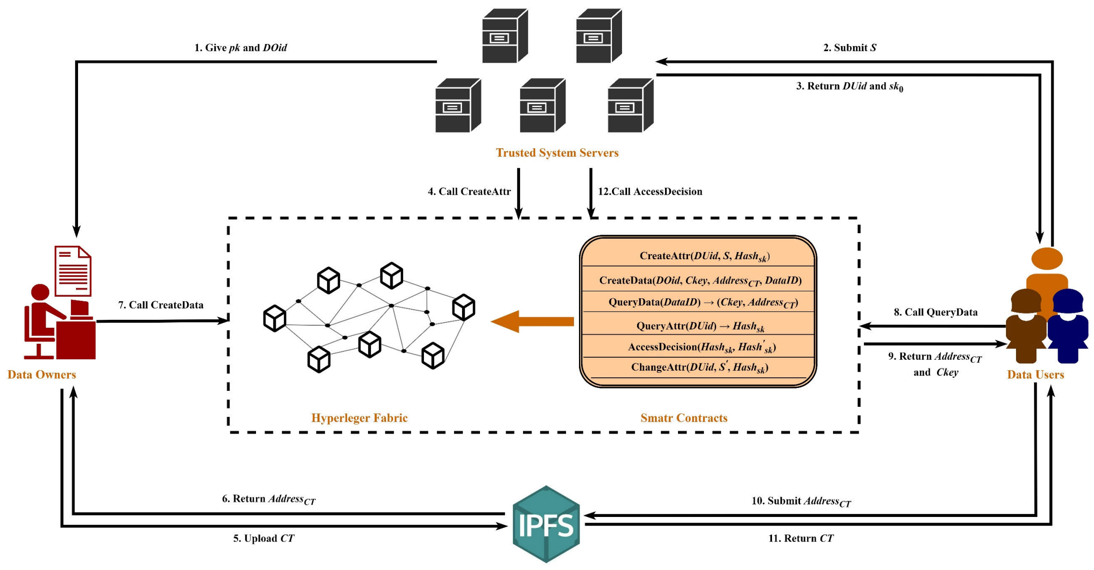

FIGURE 1. Blockchain-based model for personal privacy data protection.

C. SELECTIVE MODEL

In our system, DO is honest, and DU is honest but curious. STSS are semi-trusted system servers. We assume that STSS can securely generate a private key for DU and that attackers cannot obtain complete DU private key information. However, attackers may tamper with some private key information within STSS. All parameters in this paper are transmitted using TLS. The blockchain adopts Hyperledger Fabric. Essentially, all entities will perform according to the rules set by the scheme. We consider the chosen plaintext attack (CPA), which can be represented as a game between the adversaryand challenger.

Initialization Phase: Adversary $\mathcal{A}$ declares an accepted access policy $\mathbb{A}^*$ for the challenge and sends $\mathbb{A}^*$ to the challenger $\mathcal{C}$.

Setup Phase: Challenger $\mathcal{C}$ runs the initialization Setup $\left(1^\lambda\right)$ algorithm to obtain the public key pk and master key msk, and then sends the public key pk to adversary $\mathcal{A}$.

Query Phase 1: Adversary $\mathcal{A}$ sends attribute domain S, which includes a timestamp attribute and S does not satisfy the adversary's access policy $\mathbb{A}^*$. Then, challenger $\mathcal{C}$ runs the key generation algorithm $\text{KeyGen}_{\text{FAME}}$ $(msk, S\ )$ to obtain the private key sk and sends it to adversary $\mathcal{A}$. In addition, challenger $\mathcal{C}$ runs SM3 (sk) to obtain the hash value of sk $Hash_{sk}$ and sends it to adversary $\mathcal{A}$. This step will be repeated multiple times according to the needs of adversary $\mathcal{A}$.

Challenge Phase: Adversary $\mathcal{A}$ submits two equal-length messages M_0 and M_1, and then sends these two messages to challenger $\mathcal{C}$. Challenger $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$, runs the encryption algorithm Enc_{FAME} $(pk, \mathbb{A}^*, M_b) \rightarrow CT^*$. Finally, challenger $\mathcal{C}$ sends and CT^* to adversary $\mathcal{A}$.

Query Phase 2: Adversary $\mathcal{A}$ requests keys as in Phase 1.

Guessing Phase: Adversary $\mathcal{A}$ guesses b with b'. If $b' = b$, then adversary $\mathcal{A}$ wins the game.

TABLE 1. Definition of parameters.

Parameter	Meaning
P	large prime number
$\mathbb{G},\ \mathbb{H}, \mathbb{G}_T$	p-order cyclic groups
g	generating element of the group $\mathbb{G}$
h	generating element of the group $\mathbb{H}$
e	bilinear mapping
$\mathbb{Z}_p^*$	group of integers of order p without 0
$\mathbb{Z}_p$	p-order integer group

If adversary $\mathcal{A}$ can win the game with non-negligible advantage, we consider HAD-ABE is secure. The advantage is defined as:

$$Adv_{\mathcal{A}}^{CPA}(\lambda) = \left|Pr\left[b' = b\right] - 1/2\right| \tag{3}$$

V. SCHEME DESIGN

The definitions of some of the parameters involved in the scheme of this paper are shown in Table 1.

A. SCHEME PROCESS

The curriculum described here is divided into five main stages. They include initialization, DU identity registration, data encryption and upload, access decision and decryption, and attribute revocation based on sentinel mode.

1) INITIALIZATION

① Initialization of HAD-FME. STSS generates the master key msk and the public key pk during the Setup procedure.

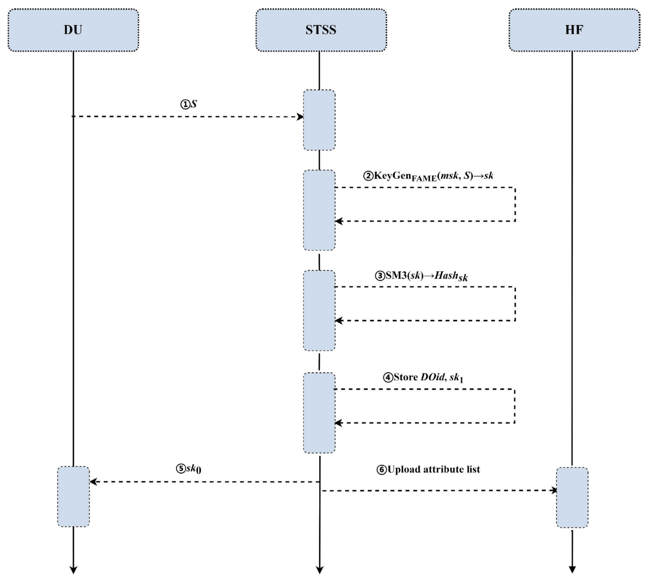

FIGURE 2. DU identity registration timeline.

Namely, Setup $(1^\lambda) \rightarrow (pk, msk)$. Take security field 1^λ as input and output $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h)$. Subsequently STSS picks $a_1, a_2 \in \mathbb{Z}_p^*, d_1, d_2, d_3 \in \mathbb{Z}_p$, and calculates pk.

$$pk = (h, H_1, H_2, T_1, T_2) \tag{4}$$

where $H_1 = h^{a_1}$, $H_2 = h^{a_2}$, $T_1 = e(g, h)^{d_1 a_1 + d_3}$, $T_2 = e(g, h)^{d_2 a_2 + d_3}$. Then STSS calculates msk.

$$msk = \left(g, h, a_1, a_2, b_1, b_2, g^{d_1}, g^{d_2}, g^{d_3}\right) \tag{5}$$

where $b_1, b_2 \in \mathbb{Z}_p.\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{G}$ is a hash function. Function mainly maps any string into members in group $\mathbb{G}$, which is used in the key generation and encryption stage.

② Initialization of HF. It is mainly used to generate smart contracts and then publish them on the HF network.

③ Initialization of DO. The STSS generates a special ID for the DO called *DOid*. STSS uses Transport Layer Security (TLS) to communicate the *DOid* and *pk* to the DO.

2) DU IDENTITY REGISTRATION

DU identity registration consists of six stages, the steps are shown in Figure 2.

① DU offers identifying details. DU gives STSS attribute domain $S(y_1, y_2, \cdots, y_{time})$ and STSS creates a special ID for the DU: *DUid*. The timestamp field y_{time}, a specific field in the collection of characteristics, is used to determine whether the secret key is the most recent DU key.

② The DU's secret key *sk* is generated via STSS. In accordance with the attribute domain *S*, STSS creates *sk* for DU and runs KeyGen$_{\text{FAME}}$ $(msk, S) \rightarrow sk$. To begin, KeyGen$_{\text{FAME}}$ randomly chooses $r_1, r_2 \in \mathbb{Z}_p, h, b_1, b_2 \in msk$, and then computes sk_0.

TABLE 2. Attribute list.

ID of the data user	Attribute Domain	Hash value of *sk*
DUid	*S*	$Hash_{sk}$

Where $sk_{0,1} = h^{b_1 r_1}$, $sk_{0,2} = h^{b_2 r_2}$, $sk_{0,3} = h^{r_1 + r_2}$. Next KeyGen$_{\text{FAME}}$ calculates $sk_{y,t}$.

$$sk_{y,t} = \mathcal{H}_1 \cdot \mathcal{H}_2 \cdot \mathcal{H}_3 \cdot g^{\frac{\sigma_y}{a_t}} \tag{6}$$

where $\mathcal{H}_1 = \mathcal{H}(y\|1\|t)^{\frac{b_1 r_1}{a_t}}$, $\mathcal{H}_2 = \mathcal{H}(y\|2\|t)^{\frac{b_2 r_2}{a_t}}$, $\mathcal{H}_3 = \mathcal{H}(y\|3\|t)^{\frac{r_1 + r_2}{a_t}}$.

In the above and following steps $y \in S, t = 1, 2$. Then, KeyGen$_{\text{FAME}}$ chooses $\sigma_y \in \mathbb{Z}_p$ at random, and sets up $sk_y = \left(sk_{y,1}, sk_{y,2}, sk_{y,3}\right)$. KeyGen$_{\text{FAME}}$ calculates sk'_t next.

$$sk'_t = g^{d_t} \cdot \mathcal{H}'_1 \cdot \mathcal{H}'_2 \cdot \mathcal{H}'_3 \cdot g^{\frac{\sigma'}{a_t}} \tag{7}$$

where $t = 1, 2$, $\sigma' \in \mathbb{Z}_p.\mathcal{H}'_1 = \mathcal{H}(0\|1\|1\|t)^{\frac{b_1 r_1}{a_t}}$, $\mathcal{H}'_2 = \mathcal{H}(0\|1\|2\|t)^{\frac{b_2 r_2}{a_t}}$, $\mathcal{H}'_3 = \mathcal{H}(0\|1\|3\|t)^{\frac{r_1 + r_2}{a_s}}$.

In addition, set up $sk'_3 = g^{d_3} \cdot g^{-\sigma'}$, $sk' = \left(sk'_1, sk'_2, sk'_3\right)$. Lastly, KeyGen$_{\text{FAME}}$ obtains sk_1.

$$sk_1 = \left(\left\{sk_y\right\}_{y \in S}, sk'\right) \tag{8}$$

STSS outputs (sk_0, sk_1) as DU's secret key *sk*.

③ STSS obtains the hash value of *sk*. The SM3 algorithm is used by STSS to obtain the hash value of *sk* $Hash_{sk}$. If *sk* has been tampered with or whether DU qualifies for decryption, will be decided later.

④ STSS managements *sk*. After storing the *DUid* and its associated attributes for the secret key sk_1, STSS transmits sk_0 to the DU through the TLS secure channel.

⑤ The attribute list is uploaded to HF by STSS. As seen in Table 2, when STSS wants to upload the list of the DU's attributes to HF for storage, it executes the CreateAttr contract.

3) DATA ENCRYPTION AND UPLOAD

Figure 3 illustrates the six stages involved in encrypting and uploading personal private data.

① The private data are encrypted using HAD-FME. DO runs KeyGen$_{\text{SM4}}$ $\rightarrow$ *key*, which chooses the symmetric key *key* at random. Afterwards, DO runs Enc$_{\text{SM4}}$ (*key*, *Data*) $\rightarrow$ *CT*, which employs the *key* to encrypt the user's private data in order to produce the ciphertext *CT*.

② DO uploads *CT* to IPFS. DO uploads the *CT* to IPFS for storage and returns the $Address_{CT}$.

③ The symmetric key is encrypted using HAD-FME. DO formulates an access policy $\mathbb{A}$, and executes Enc$_{\text{FAME}}$ $(pk, \mathbb{A}, key) \rightarrow Ckey$ to encrypt the *key*.

The LSSS matrix *M* is created during the encryption procedure from the Boolean formula used to describe the $\mathbb{A}$, n_1 rows and n_2 columns make up the matrix *M*, as stated

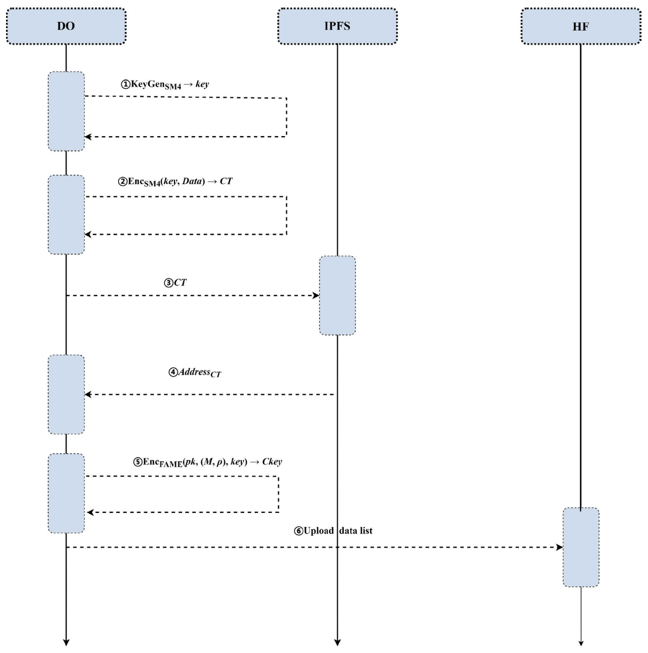
DO
IPFS
HF
①KeyGen$_{SM4}$ → key
②Enc$_{SM4}$(key, Data) → CT
③CT
④Address$_{CT}$
⑤Enc$_{FAME}$(pk, (M, ρ), key) → Ckey
⑥Upload data list

FIGURE 3. Data encryption and upload timing diagram.

in Section III. The LSSS imagines a Boolean formula as an access tree, where the leaf nodes represent characteristics and the inside nodes represent with or gates. Then, the LSSS establishes the global counter variable c to 1 and declares the access tree's root node as a vector (1). After that, it traverses the tree hierarchy breadth-first while setting the vector that identifies its parent node by that vector. The vector of marked leaf nodes then form M by row after marking the whole access number node.

Enc$_{\text{FAME}}$ firstly selects $s_1, s_2 \in \mathbb{Z}_p$ at random and calculates ct_0.

$$ct_0 = \left(H_1^{s_1}, H_2^{s_2}, h^{s_1+s_2}\right) \tag{9}$$

where $H_1, H_2 \in pk$, $h \in msk$. Secondly, for $i = 1, 2 \ldots n_1$, $j = 1, 2 \ldots n_2$, and $l = 1, 2, 3$, Enc$_{\text{FAME}}$ calculates $ct_{i,l}$.

$$ct_{i,l} = \mathcal{H}_4 \cdot \mathcal{H}_5 \cdot \prod_{j=1}^{n_2} (\mathcal{H}_6 \cdot \mathcal{H}_7) \tag{10}$$

where $\mathcal{H}_4 = \mathcal{H}(\rho(i)\|l\|1)^{s_1}$, $\mathcal{H}_5 = \mathcal{H}(\rho(i)\|l\|2)^{s_2}$, $\mathcal{H}_6 = \mathcal{H}(0\|j\|l\|1)^{s_1}$, $\mathcal{H}_7 = \mathcal{H}(0\|j\|l\|2)^{s_2}$. Then Enc$_{\text{FAME}}$ calculates ct'.

$$ct' = T_1^{s_1} \cdot T_2^{s_2} \cdot key \tag{11}$$

where $T_1, T_2 \in pk$. Finally, a symmetric key cipher $Ckey$ is composed of ct_0, ct_i and ct'.

$$Ckey = \left(ct_0, ct_1, ct_2 \ldots, ct_{n_1}, ct'\right) \tag{12}$$

④ DO provides the data list to HF. As seen in Table 3, DO creates the data list and sends it via the CreateData contract to HF for storage.

TABLE 3. Data list.

Data owner's ID	Ciphertext of key	Address of IPFS	ID of data
$DOid$	$Ckey$	$Address_{CT}$	$DataID$

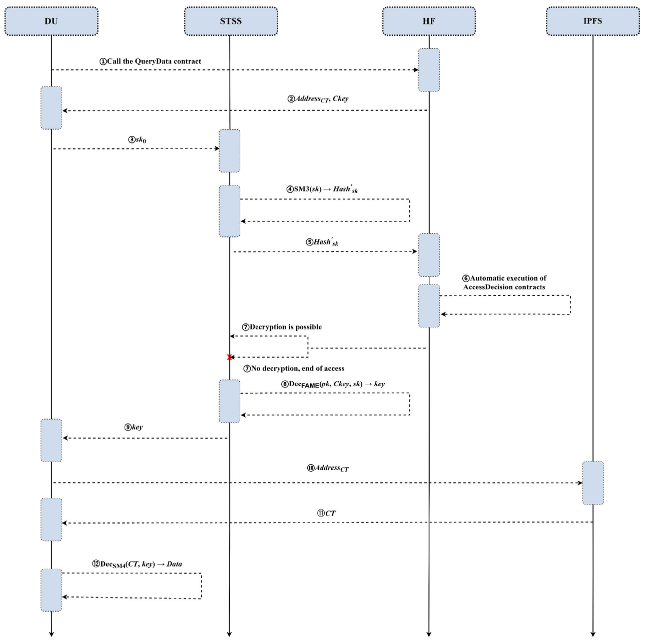
DU
STSS
HF
IPFS
①Call the QueryData contract
②Address$_{CT}$, Ckey
③sk_0
④SM3(sk) → Hash'$_{sk}$
⑤Hash'$_{sk}$
⑥Automatic execution of AccessDecision contracts
⑦Decryption is possible
⑦No decryption, end of access
⑧Dec$_{FAME}$(pk, Ckey, sk) → key
⑨key
⑩Address$_{CT}$
⑪CT
⑫Dec$_{SM4}$(CT, key) → Data

FIGURE 4. Access decision and decryption timing diagram.

4) ACCESS DECISION AND DECRYPTION

The access decision and decryption process consists of twelve phases, and Figure 4 depicts their precise order.

① Smart Contract Queries. To retrieve the IPFS address $Address_{CT}$ of the CT and the encrypted symmetric key information $Ckey$, DU invokes the QueryData contract.

② DU asks for admission. The DU transmits to STSS a decryption request that includes its $DUid$ and sk_0.

③ Access choice with STSS first decryption. After computing the hash value of sk twice, STSS runs SM3(sk) $\rightarrow$ $Hash'_{sk}$ and transmits $Hash'_{sk}$ to HF, which activates the QueryAttr contract and gets the $Hash_{sk}$ from the attribute list. In the following step, the AccessDecision contract will automatically compare if the two values are the same. If they match, STSS performs the first decryption operation, Dec$_{\text{FAME}}$(pk, $Ckey$, sk) $\rightarrow$ key to obtain the key.

If the attributes of the DU fulfil the access policy, as determined by LSSS in Section III then the DU has the value indicated by the attribute. The matrix M s rows that fulfil the set of qualities S are referred to as the set I. According to Eq. 1, there is $\{y_i \in \mathbb{Z}_p\}_{i \in I}$. The detailed calculation process is as follows.

$$num = ct' \cdot e_1 \cdot e_2 \cdot e_3 \tag{13}$$

$$dec = e_1' \cdot e_2' \cdot e_3' \tag{14}$$

where $e_1 = e\left(\prod_{i\in I} ct_{i,1}^{y_i}, sk_{0,1}\right)$, $e_2 = e\left(\prod_{i\in I} ct_{i,2}^{y_i}, sk_{0,2}\right)$, $e_3 = e\left(\prod_{i\in I} ct_{i,3}^{y_i}, sk_{0,3}\right) \cdot e'_1 = e\left(sk'_1 \prod_{i\in I} sk_{p(i),1}^{y_i}, ct_{0,1}\right)$, $e'_2 = e\left(sk'_2 \prod_{i\in I} sk_{p(i),2}^{y_i}, ct_{0,2}\right)$, $e'_3 = e\left(sk'_3 \prod_{i\in I} sk_{p(i)}^{y_i}, ct_{0,3}\right)$.

Finally, $\mathrm{Dec}_{\mathrm{FAME}}$ gets the *key*.

$$key = \frac{ct' \cdot e_1 \cdot e_2 \cdot e_3}{e'_1 \cdot e'_2 \cdot e'_3} \tag{15}$$

$ct_0, ct_i, ct' \in Ckey$, $sk_0, sk_{\rho(i)}, sk' \in sk$. The first three elements are represented as $sk_{0,1}, sk_{0,2}, sk_{0,3}$ respectively, and the same for ct_0.

④ STSS sends the *key* securely. Over the secure channel TLS, STSS transmits the *key* to the DU.

⑤ DU receives the *CT*. Using $Address_{CT}$ from IPFS, DU retrieves *CT*.

⑥ DU second decryption. To access the user's personal information, DU uses $\mathrm{Dec}_{\mathrm{SM4}}(CT, key)$ and decrypts using *key*.

5) ATTRIBUTE REVOCATION BASED ON SENTINEL MODE

In this part, the idea of a sentinel and pass is introduced. STSS, which is primarily in charge of outsourcing decryption, serves as the sentinel, while the most recent iteration of DU's $Hash_{sk}$ serves as the pass. The automated DU identification verification is completed through the AccessDecision contract.

SM-ARM only updates *sk* to generate attribute revocation effect, which reduces the cost of revocation. The precise flow of this SM-ARM is given in Figure 5 when the attribute of DU is altered. When the attributes of DU are revoked or the private key information of DU is tampered with, the system performs the following steps.

① STSS generates the latest *sk* for DU. If the DU attributes change, the timestamp field is modified to y'_{time}. Then, STSS runs $\mathrm{KeyGen}_{\mathrm{FAME}}(msk, S') \rightarrow Newsk$, and the attribute domain of DU is $S'(y'_1, y'_2, \cdots, y'_{time})$. Firstly, $\mathrm{KeyGen}_{\mathrm{FAME}}$ chooses $r'_1, r'_2 \in \mathbb{Z}_p$, $h, b_1, b_2 \in msk$, and calculates sk'_0.

$$sk'_0 = \left(h^{b_1 r'_1}, h^{b_2 r'_2}, h^{r'_1 + r'_2}\right) \tag{16}$$

Secondly, $\mathrm{KeyGen}_{\mathrm{FAME}}$ calculates $sk'_{y,t}$.

$$\begin{aligned} sk'_{y,t} = \mathcal{H}\left(y' \| 1 \| t\right)^{\frac{b_1 r'_1}{a_t}} \cdot \mathcal{H}\left(y' \| 2 \| t\right)^{\frac{b_2 r'_2}{a_t}} \\ \cdot \mathcal{H}\left(y' \| 3 \| t\right)^{\frac{r'_1 + r'_2}{a_t}} \cdot g^{\frac{\sigma_y}{a_t}} \end{aligned} \tag{17}$$

where $y \in S'$, $t = 1, 2$, $\sigma_y \in \mathbb{Z}_p$. So $sk'_y = \left(sk'_{y,1}, sk'_{y,2}, g^{-\sigma_y}\right)$. Then $\mathrm{KeyGen}_{\mathrm{FAME}}$ calculates sk'_t.

$$\begin{aligned} sk'_t = g^{d_t} \cdot \mathcal{H}(0\|1\|1\|t)^{\frac{b_1 r'_1}{a_t}} \\ \cdot \mathcal{H}(0\|1\|2\|t)^{\frac{b_2 r'_2}{a_t}} \\ \cdot \mathcal{H}(0\|1\|3\|t)^{\frac{r'_1 + r'_2}{a_t}} \cdot g^{\frac{\sigma'}{a_t}} \end{aligned} \tag{18}$$

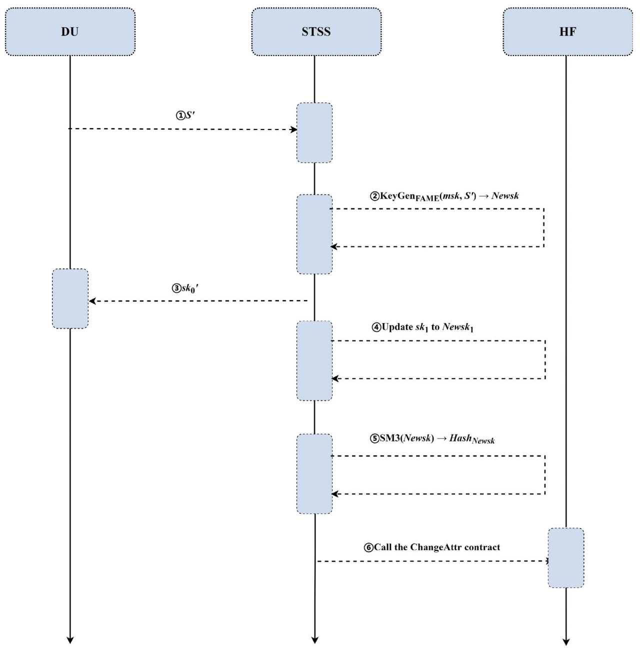
DU
STSS
HF
①S'
②$\mathrm{KeyGen}_{\mathrm{FAME}}(msk, S') \rightarrow Newsk$
③sk_0'
④Update sk_1 to $Newsk_1$
⑤SM3($Newsk$) $\rightarrow Hash_{Newsk}$
⑥Call the ChangeAttr contract

FIGURE 5. Timing diagram for attribute revocation based on sentinel mode.

where $\sigma' \in \mathbb{Z}_p$. In addition to setting $sk' = (sk'_1, sk'_2, g^{d_3} \cdot g^{-\sigma'})$. So $Newsk_1 = (\left\{sk'_y\right\}_{y\in S'}, sk')$. Finally, STSS outputs $Newsk = (sk'_0, Newsk_1)$ as the latest version *sk* for DU.

② STSS modifies the DU's *sk*. STSS updates the $Newsk_1$ in the server depending on the *DOid* after sending sk'_0 to the DU.

③ The hash of the *Newsk* is determined by STSS. The hash for *Newsk* is calculated by STSS using the SM3 algorithm by running SM3 $(Newsk) \rightarrow Hash_{Newsk}$.

④ The attribute list is updated by HF. The attribute domain in the DU's attribute list is changed by STSS to S', and the hash value is changed to $Hash_{Newsk}$.

B. SECURITY ANALYSIS

1) SECURITY OF HAD-FME

Theorem: If the DLIN assumption holds, then the adversary will break the proposed scheme with a negligible advantage $Adv_{\mathcal{A}}$.

Proof: At first, the challenger $\mathcal{C}$ generates three groups $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$. At the same time, a bilinear map $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$ is also created. Then $\mathcal{C}$ selects a generate g for $\mathbb{G}$ and a generate h for $\mathbb{H}$. $\mathcal{C}$ randomly chooses $a_1, a_2 \in \mathbb{Z}_p^*$, $s_1, s_2, s \in \mathbb{Z}_p$ and randomly chooses $U \in \mathbb{G}$, $V \in \mathbb{H}$. Then $\mathcal{C}$ gets group elements $U = g^s$, $V = h^s$, $S_1 = g^{s_1}$, $S_2 = g^{s_2}$, $S'_1 = h^{s_1}$, $S'_2 = h^{s_2}$.

$\mathcal{C}$ gets $par = (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e, g, h)$, $D = (g^{a_1}, g^{a_2}, h^{a_1}, h^{a_2}, g^{a_1 s_1}, g^{a_2 s_2}, h^{a_1 s_1}, h^{a_2 s_2})$. A polynomial-time algorithm $\mathcal{B}$ can be constructed with advantage $Adv_{\mathcal{B}}$ to break the DLIN assumption. $\mathcal{C}$ flips a fair coin $b \in \{0, 1\}$. If, $\mathcal{C}$ sets $T = (U, V)$, else it sets $T = \left(S_1 S_2, S'_1 S'_2\right)$. $\mathcal{C}$ sends to $\left(1^\lambda, par, D, T\right)$ to $\mathcal{B}$. $\mathcal{B}$ outputs his/her guess b' on b.

Initialization Phase: Adversary $\mathcal{A}$ declares an accepted access matrix $(M, \rho)^*$ for the challenge and sends $(M, \rho)^*$ to $\mathcal{B}$.

Setup: $\mathcal{B}$ randomly selects $a_1, a_2 \in \mathbb{Z}_p^*$, $d_1', d_2', d_3' \in \mathbb{Z}_p$, and sets $d_t = d_t'(s_1 + s_2)$, where $t = 1, 2, 3$. $\mathcal{B}$ computes:

$$\begin{aligned} e(g, h)^{d_1 a_1 + a_3} &= e(g, h)^{d_1 a_1} e(g, h)^{d_3} \\ &= e(g, h)^{(s_1+s_2) d_1' a_1} e(g, h)^{d_3'(s_1+s_2)} \\ &= e\left(S_1 S_2, h\right)^{d_1' a_1} e\left(S_1 S_2, h\right)^{d_3'} \end{aligned} \tag{19}$$

Then $\mathcal{B}$ also computes $e(g, h)^{d_2 a_2 + d_3} = e(S_1 S_2, h)^{d_2' a_2}$ $e(S_1 S_2, h)^{d_3'}$. $\mathcal{B}$ responses $pk = (h, h^{a_1}, h^{a_2}, e(S_1 S_2, h)^{d_1' a_1}$ $e(S_1 S_2, h)^{d_3'}, e(S_1 S_2, h)^{d_2' a_2} e(S_1 S_2, h)^{d_3'})$ as the public key of $\mathcal{A}$.

Query Phase 1: $\mathcal{A}$ Chooses the attribute domain S, including a timestamp attribute that does not satisfy the access matrix $(M, \rho)^*$. $\mathcal{B}$ randomly selects $b_1, b_2 \in \mathbb{Z}_p^*$ and generates the secret key for each S as follows. $\mathcal{B}$ sets $h^{r_1} = h^{s_1}$, $h^{r_2} = h^{s_2}$ and computes $sk_0 = (h^{b_1 r_1}, h^{b_2 r_2}, h^{r_1 + r_2}) = (h^{b_1 s_1}, h^{b_2 s_2}, h^{s_1 + s_2})$. Then $\mathcal{B}$ randomly selects $\sigma_y' \in \mathbb{Z}_p$ and sets $\sigma_y = \sigma_y'(s_1 + s_2)$, the secret key parameter $sk_{y,t}$ is computed by $\mathcal{B}$ as follows:

$$\begin{aligned} sk_{y,t} &= \mathcal{H}(y\|1\|t)^{\frac{b_1 s_1}{a_t}} \cdot \mathcal{H}(y\|2\|t)^{\frac{b_2 s_2}{a_t}} \\ &\quad \cdot \mathcal{H}(y\|3\|t)^{\frac{s_1+s_2}{a_t}} \cdot g^{\frac{\sigma_y'(s_1+s_2)}{a_t}} \\ &= \mathcal{H}(y\|1\|t)^{\frac{b_1 s_1}{a_t}} \cdot \mathcal{H}(y\|2\|t)^{\frac{b_2 s_2}{a_t}} \\ &\quad \cdot \mathcal{H}(y\|3\|t)^{\frac{s_1+s_2}{a_t}} \cdot (S_1 S_2)^{\frac{\sigma_y'}{a_t}} \end{aligned} \tag{20}$$

where $y \in S$, $t = 1, 2$. $\mathcal{B}$ sets $sk_y = (sk_{y,1}, sk_{y,2}, (S_1 S_2)^{-\sigma_y'})$. Then $\mathcal{B}$ randomly selects $\sigma' \in \mathbb{Z}_p$ and sets $\sigma' = s_1 + s_2$. $\mathcal{B}$ computes:

$$\begin{aligned} sk_t' &= g^{d_t'(s_1+s_2)} \cdot \mathcal{H}(0\|1\|1\|t)^{\frac{b_1 s_1}{a_t}} \cdot \mathcal{H}(0\|1\|2\|t)^{\frac{b_2 s_2}{a_t}} \\ &\quad \cdot \mathcal{H}(0\|1\|3\|t)^{\frac{s_1+s_2}{a_t}} \cdot g^{\frac{s_1+s_2}{a_t}} \\ &= (S_1 S_2)^{d_t + \frac{1}{a_t}} \cdot \mathcal{H}(0\|1\|1\|t)^{\frac{b_1 s_1}{a_t}} \\ &\quad \cdot \mathcal{H}(0\|1\|2\|t)^{\frac{b_2 s_2}{a_t}} \cdot \mathcal{H}(0\|1\|3\|t)^{\frac{s_1+s_2}{a_t}} \end{aligned} \tag{21}$$

$\mathcal{B}$ sets $sk' = \left(sk_1', sk_2', (S_1 S_2)^{d_3' - 1}\right)$. Finally, $\mathcal{B}$ delivers the secret key $sk = ((S_1')^{b_1}, (S_2')^{b_2}, S_1' S_2', \{sk_{y,1}, sk_{y,2}, (S_1 S_2)^{-\sigma_y'}\}_{y \in S}, sk_1', sk_2', (S_1 S_2)^{d_3' - 1})$ to $\mathcal{A}$.

Challenge Phase: Adversary $\mathcal{A}$ sends two messages M_0 and M_1 with equal length to $\mathcal{B}$. $\mathcal{B}$ flips a fair coin $b \in \{0, 1\}$. Next, $\mathcal{A}$ selects the message M_b to encrypt under $(M, \rho)^*$. $\mathcal{B}$ randomly chooses $s_1, s_2 \in \mathbb{Z}_p$, then $\mathcal{B}$ using pk computes:

$$\begin{aligned} ct_0 &= \left(h^{a_1 s_1}, h^{a_2 s_2}, h^{s_1 + s_2}\right) \\ &= \left((S_1')^{a_1}, (S_2')^{a_2}, S_1' S_2'\right) \end{aligned} \tag{22}$$

Then $\mathcal{B}$ supposes M has n_1 rows and n_2 columns. For $i = 1, \ldots, n_1$ and, $\mathcal{B}$ computes:

$$\begin{aligned} ct_{i,l} &= \mathcal{H}(\rho(i)\|l\| \mid 1)^{s_1} \cdot \mathcal{H}(\rho(i)\|l\|2)^{s_2} \\ &\quad \cdot \prod_{j=1}^{n_2} \left[\mathcal{H}(0||j||l||1)^{s_1} \cdot \mathcal{H}(0||j||l||2)^{s_2}\right]^{(M)_{i,j}} \end{aligned} \tag{23}$$

Finally, $\mathcal{B}$ computes:

$$\begin{aligned} ct' &= \left(e(g, h)^{d_1 a_1 + d_3}\right)^{s_1} \cdot \left(e(g, h)^{d_2 a_2 + d_3}\right)^{s_2} \cdot M_b \\ &= \left(e\left(S_1 S_2, S_1'\right)^{d_1' a_1} e\left(S_1 S_2, S_1'\right)^{d_3'}\right) \\ &\quad \cdot \left(e\left(S_1 S_2, S_2'\right)^{d_2' a_2} e\left(S_1 S_2, S_2'\right)^{d_3'}\right) \cdot M_b \end{aligned} \tag{24}$$

$\mathcal{B}$ delivers $CT^* = \left(ct_0, ct_1, \ldots, ct_{n_1}, ct'\right)$ to $\mathcal{A}$.

Query Phase 2: This phase is similar to phase 1.

Guess: Adversary $\mathcal{A}$ gives its guess b' about b. If $b' = b$, $\mathcal{B}$ returns 0, which means that $T = \left(S_1 S_2, S_1' S_2'\right) = \left(g^{s_1 + s_2}, h^{s_1 + s_2}\right)$. Otherwise, it returns 1, which means that $T = (U, V) = (g^s, h^s)$.

The advantage of adversary $\mathcal{A}$ break scheme is $Adv_{\mathcal{A}}^{CPA} = Pr\left[b' = b\right] - \frac{1}{2}$. Then the probability of $\mathcal{B}$ breaking the DLIN assumption is:

$$\begin{aligned} Adv_B &= Pr\left[\mathcal{B}\left(1^\lambda, par, D, T = \left(g^{s_1+s_2}, h^{s_1+s_2}\right)\right)\right] \\ &= Pr\left[b' = b\right] = \frac{1}{2} + Adv_{\mathcal{A}} \end{aligned} \tag{25}$$

Since the DLIN assumption is a hard problem, the advantage $Adv_{\mathcal{B}}$ of $\mathcal{B}$ to break it is negligible. In the proposed scheme, SM3 is used to obtain the hash value of the secret key, which is used as a passport for DU to access private data. The adversary $\mathcal{A}$ can make queries to the $\mathcal{B}$ to obtain the hash value $Hash_x$. According to [38], SM3 has strong collision resistance, the adversary has a negligible advantage in distinguishing the outputs $Hash_x$ and $Hash_{x'}$. So, the advantage $Adv_{\mathcal{A}}$ of $\mathcal{A}$ to break the proposed scheme is also negligible.

2) USER ATTRIBUTE REVOCATION

The attribute revocation mechanism proposed in this paper satisfies both forward security and backward security. When the attribute domain of DU does not comply with the access policy, the previous *sk* cannot be used to decrypt the later ciphertext, which is called forward security. The scheme ensures forward security even when DU or part of the attributes are revoked. When the attributes of an object do not match the access policy, SM-ARM will generate a new version of *sk* for DU. If DU attempts to decode subsequent ciphertext using outdated versions of *sk*, or fails to decrypt or retrieve ciphertext, the AccessDecision contract will not be fulfilled. As a result, the SM-ARM guarantees the forward security of personal data.

When the newly registered DU attribute domain satisfies the access policy, they can use their own new key to decrypt the previously encrypted ciphertext, which is called backward security. The most recent version of *sk* is generated for a new DU when they register in the scheme or when more

TABLE 4. Comparison of scheme features.

Schemes	Bilinear group	Access structure	Blockchain	Standard assumption	Constant decryption	Attribute revocation
[13]	Symmetric prime group	LSSS	Ethereum	×	×	×
[15]	Symmetric prime group	Tree	Fabric	×	×	×
[16]	Symmetric prime group	Tree	Fabric	×	×	√
[18]	Symmetric prime group	LSSS	×	×	×	×
[24]	Symmetric prime group	Tree	Fabric	×	×	√
[29]	Symmetric prime group	LSSS	×	√	×	√
Ours	Asymmetric prime group	LSSS	Fabric	√	√	√

characteristics are added to an existing DU. If the attribute domain of DU satisfies the access policy, and the Access Decision contract judges that the user's private key *sk* is the latest version, then DU can access the private data. As a result, the scheme proposed in this paper guarantees the backward security of personal data.

3) ANTI-TAMPERING

The tamper-proof solution mainly consists of two elements:

① The anti-tampering property of blockchain and smart contracts. The data list and attribute list are stored on the blockchain. The blockchain's data can only be altered when 51% or more of the nodes are compromised, which is extremely unlikely.

② The anti-tampering property of DU secret key *sk*. SM3 calculates the hash value of the DU *sk*, and $Hash_{sk}$ is stored in HF. In order to establish if *sk* is the most recent state and whether it has been altered, STSS will recalculate the hash value of *sk* whenever DU accesses confidential data. The decryption procedure is impossible to complete if some keys stored in STSS have been altered.

VI. SCHEME ANALYSIS

This section mainly demonstrates the scheme of the paper from three aspects: function comparison, theoretical analysis of scheme efficiency, and various performance tests compared with other schemes.

A. FUNCTION COMPARISON

We compare the features of HAD-FME with the schemes in [13], [15], [16], [18], [24], and [29]. As shown in Table 4, the schemes in [13], [15], [16], [18], [24], and [29] are all constructed under the symmetric prime group, but the HAD-FME is constructed under the asymmetric prime group, which is more secure. The schemes of [15], [16], and [24] are based on the Access Tree structure, compared with the LSSS, which is not flexible. Most of the solutions in Table 4 utilize blockchain technology to improve the reliability of access control management. In addition, only [29] follows the standard assumption, as well as HAD-FME. The most regrettable thing is that [13], [15], [18] did not consider user attribute revocation. However, the cost of attribute revocation schemes for [16], [24], and [29] needs to be reduced.

B. THEORETICAL ANALYSIS

In this section, two schemes are selected for comparison. Scheme 1 is the data protection scheme of [15] and [16], and scheme 2 is the mixed encryption scheme of CP-ABE [12] and AES. To highlight the proposed scheme, the schemes are all based on the LSSS structure.

We use t_m, t_e, t_h, t_p to represent the computational cost for multiplication, exponentiation, hashing and bilinear pairing operations respectively, n_1 and n_2 are the dimensions of the MSP. The subscripts $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ respectively indicate that these operations are performed in the groups $\mathbb{G}$, $\mathbb{H}$, and $\mathbb{G}_T$. Table 5 and Table 6 list the number of various group operations involved in implementations of these schemes.

As shown in Table 5, the number of elements in $\mathbb{G}$ and $\mathbb{H}$ list the sizes of the ciphertext and the key, with one element in $\mathbb{H}$ being three times of $\mathbb{G}$. Therefore, the key size of HAD-FME is smaller than both Scheme 1 and Scheme 2, the ciphertext size is comparable to Scheme 2, but smaller than Scheme 1. Regarding key generation, T denotes the attribute domain dimensions input to KeyGen, we can see that HAD-FME consumes more multiplications, exponentiations, and hash operations, compared to Scheme 1 and Scheme 2. However, most of the calculations are performed in group $\mathbb{G}$. Operations in group $\mathbb{G}$ are faster than those in group $\mathbb{H}$.

As shown in Table 6, during the encryption phase, HAD-FME needs to perform 3 exponentiations in group $\mathbb{H}$. However, the computational cost of Scheme 1 will increase with the increase of MSP dimensions. At the same time, in group $\mathbb{G}$, Scheme 2 requires $6n_1 + 9n_2$ exponentiations in each encryption process, resulting in higher computational cost. In the decryption phase, I is the number of attributes used in $\mathrm{Dec}_{\mathrm{FAME}}$, HAD-FME uses a constant 6 bilinear operations, while the number of bilinear operations in Scheme 1 increases with the increase of I. In addition, compared to Scheme 2, most of the calculations in HAD-FME are performed in the faster group $\mathbb{G}$. Overall, considering the key and ciphertext sizes, key generation, encryption, and decryption, HAD-FME outperforms Scheme 1 and Scheme 2.

TABLE 5. Comparison of key size, ciphertext size and key generation.

Schemes	Key size	Ciphertext size	Key generation
Scheme 1	$(T+1)_{\mathbb{G}}+T_{\mathbb{H}}$	$(n_1)_{\mathbb{G}}+(n_1+1)_{\mathbb{H}}$	$[(T+1)_{t_m}+(T+2)_{t_e}+T_{t_h}]_{\mathbb{G}}+\ (T_{t_e})_{\mathbb{H}}$
Scheme 2	$3(T+2)_{\mathbb{H}}$	$3(n_1+1)_{\mathbb{G}}$	$[3(T+2)_{t_e}]_{\mathbb{H}}$
HAD-FME	$3(T+1)_{\mathbb{G}}+3_{\mathbb{H}}$	$(3n_1)_{\mathbb{G}}+3_{\mathbb{H}}$	$[(8T+9)_{t_m}+\big(9T+9)_{t_e}+6(T+1)_{t_h}\big]_{\mathbb{G}}\ +\ (3_{t_e})_{\mathbb{H}}$

TABLE 6. Comparison of encryption and decryption.

Schemes	Encryption	Decryption
Scheme 1	$[(n_1)_{t_e}+\ (n_1)_{t_h}]_{\mathbb{G}}+[(n_1+1)_{t_e}]_{\mathbb{H}}$	$[(2I+1)_{\mathbb{G}_T}]_{t_m}+(2I+1)_{t_p}$
Scheme 2	$[(6n_1n_2)_{t_m}+(6n_1+9n_2)_{t_e}]_{\mathbb{G}}$	$[(3I)_{\mathbb{G}}+(3I)_{\mathbb{H}}+(6)_{\mathbb{G}_T}]_{t_m}+(6)_{t_p}$
HAD-FME	$[(12n_1n_2+6n_1)_{t_m}+(6n_1)_{t_e}+6(n_1+n_2)_{t_h}]_{\mathbb{G}}+[(3)_{t_e}]_{\mathbb{H}}$	$[(6I+3)_{\mathbb{G}}+(6)_{\mathbb{G}_T}]_{t_m}+(6)_{t_p}$

TABLE 7. Phases and overall time overhead.

Schemes	Encrypt/ms	KeyGen$_{\text{CP-ABE}}$/ms	Decrypt/ms	Overall overhead/ms
Scheme 1	420.99	210.00	296.36	927.35
Scheme 2	741.10	550.28	22.30	1313.68
HAD-FME	360.96	240.00	24.43	**625.39**

C. PERFORMANCE TEST

The experiments in this section cover two main aspects: cryptographic algorithm performance and blockchain performance test. Performance test schemes are three schemes in theoretical analysis.

The maximum attribute domain dimension selected for the experimental part of the literature [15], [16] is 20. In order to highlight the performance advantages of our scheme in high-dimensional attribute domains, the maximum attribute domain dimension selected for the experiment is 100.

1) EXPERIMENTAL ENVIRONMENT

The environment for running the experiment is Intel Core i7-11700 processor, 16GB RAM, and 64-bit version of Ubuntu 20.04. In our development, the encryption library used charm-crypto0.50, the elliptic curve type used MNT224, the attribute-based encryption algorithm implemented in Python, and the symmetric encryption algorithm implemented in GO. Blockchain uses Hyperledger Fabric 2.3. The blockchain network consists of two organizations, each containing two peer nodes and a sorting node Order. For this scheme, a federated chain is created, and interaction with IPFS and Hyperledger Fabric is developed using tools like go-ipfs, Fabric's Fabric SDK, and Docker.

2) ENCRYPTION ALGORITHM PERFORMANCE TEST

To test the effect of attribute domain dimension on HAD-FME, attribute domains with 4 to 100 attributes each are developed in this experiment. Figure 6 shows the relationship between the dimension of the attribute domain and the time required for various encryption algorithms when the data size is 500 KB. According to the experimental results, all schemes' encryption times grow linearly as the attribute domain dimension rises. In the experiment, the encryption time of HAD-FME is less than that of other schemes when the attribute domain dimension is greater than 20, and the higher the attribute domain dimension, the bigger the difference between the encryption times of these two schemes.

To evaluate the impact of data size on data encryption, we constructed 20 personal privacy data samples with data sizes ranging from 500 KB to 1000 KB and attribute domain dimensions ranging from 50 to 100. Figure 7 shows that HAD-FME performs better than others when the attribute domain dimension is 50 and the data volume is less than 4000 KB. The encryption time of the HAD-FME is minimum. when the attribute domain volume is less than 8500 KB. Therefore, our scheme is more suitable for small-scale data encryption in high-dimensional attribute domains.

3) DECRYPTION ALGORITHM PERFORMANCE TEST

To test the impact of attribute domain dimension on data decryption, we created user attributes with attribute domain dimensions ranging from 4 to 100. When the data size is 500 KB, Figure 8 shows the relationship between the decryption time and the attribute domain dimension for different schemes. The experimental results show that as the

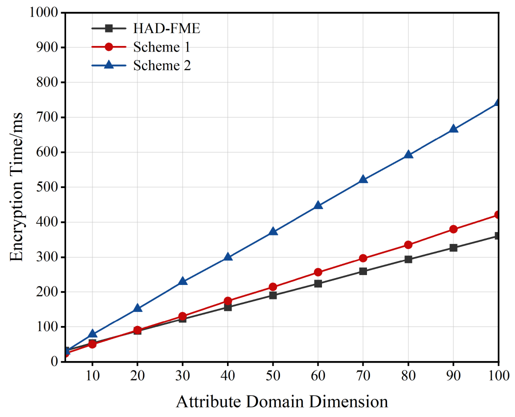

FIGURE 6. The relationship between encryption time and attribute domain dimension of different schemes.

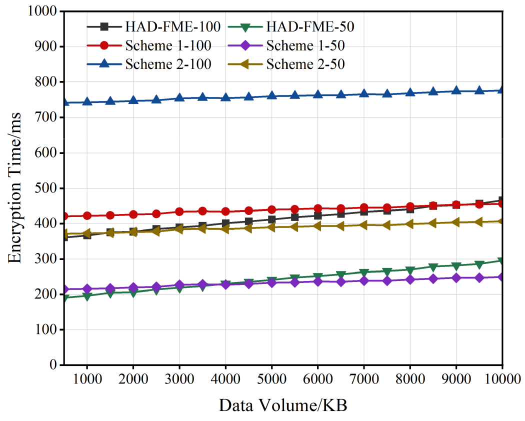

FIGURE 7. The relationship between encryption time and data volume of different schemes.

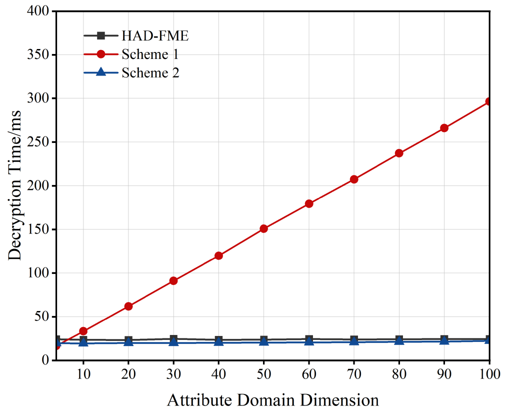

FIGURE 8. The relationship between decryption time and attribute domain dimension of different schemes.

dimension of the attribute domain increases, the decryption time of HAD-FME is significantly faster than scheme 1 and only slightly slower than scheme 2.

In order to investigate the impact of data volume on data decryption, we created 20 pieces of personal privacy data during the experiment, with data sizes ranging from 500 KB to 10000 KB and attribute domain dimensions of 50 and 100, respectively. The experimental results are displayed in Figure 9. Scheme 1 performs worse than other schemes and takes much longer to decrypt data as the size of the data increases. When the attribute domain dimension is between 50 and 100, the difference in decryption times between the schemes presented in this study and Scheme 2 widens.

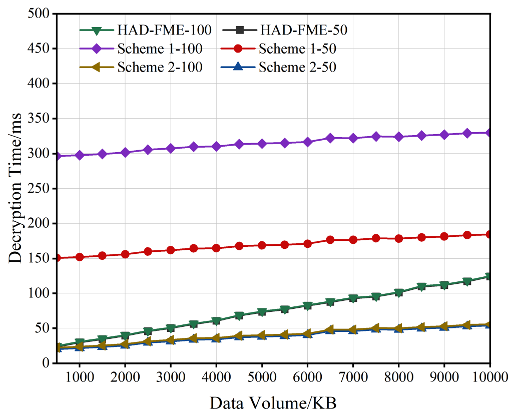

FIGURE 9. The relationship between decryption time and data volume of different schemes.

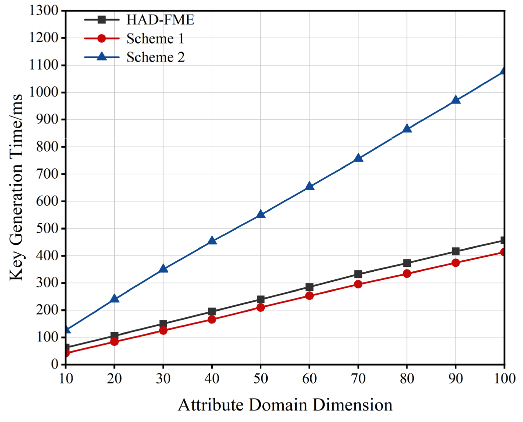

FIGURE 10. The relationship between key generation time and attribute domain dimension of different schemes.

4) KEY GENERATION ALGORITHM PERFORMANCE TEST

In this section, the attribute domain dimension is increased from 10 to 100, and the average value is used as the result of the experiment. Figure 10 shows that Scheme 2 performs poorly in terms of key generation and its key generation time rises the quickest when the attribute domain dimension is raised. Even with an attribute field size of 100, the key generation time of HAD-FME is just significantly slower than that of Scheme 1 (457 ms) and is somewhat longer than that of Scheme 2.

5) PHASES AND OVERALL TIME OVERHEAD

In this section, we summarize the performance test time of the previous three sections. Table 7 shows the various stages

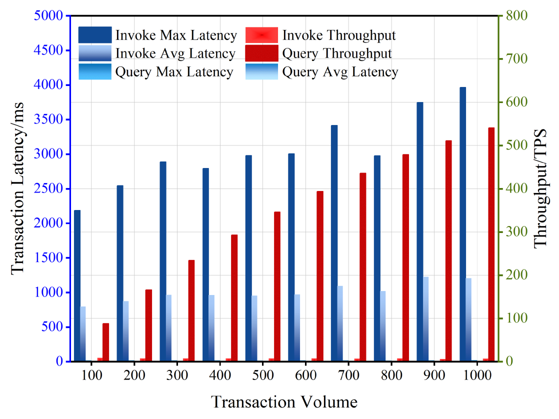

FIGURE 11. **Transaction delay and throughput diagram under 100-1000 transaction volume.**

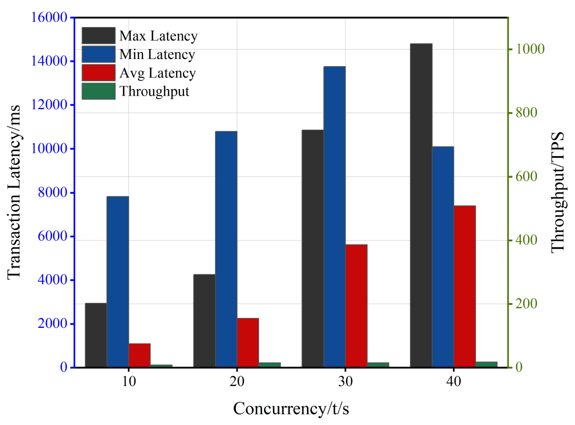

FIGURE 12. **Transaction delay and throughput diagram under 10t/s to 40t/s concurrency.**

and overall time cost of different schemes with an attribute domain dimension of 50 and a data volume of 500 KB. The encryption time of Scheme 1 is slightly higher than that of HAD-FME, while the encryption time overhead of Scheme 2 is twice that of HAD-FME. In addition, the key generation time of the HAD-FME is slightly higher than that of Scheme 1, and the key generation overhead of Scheme 2 is much higher than the HAD-FME. Finally, the decryption overheads of the HAD-FME and Scheme 2 are only slightly different, but the decryption overhead of Scheme 1 is 10 times higher than the HAD-FME. In terms of overall overhead, the performance of the HAD-FME is better than other schemes, and it is better in the scenario of the high-dimensional attribute domain and small data volume.

6) BLOCKCHAIN PERFORMANCE TEST

Transaction latency and transaction throughput are key metrics for blockchain system performance evaluation, where transaction latency is the time from issuance to final confirmation of a transaction on the chain, and transaction throughput is the number of transactions per second.

In this experiment, we have used the Hyperledger Caliper testing tool to test the invoke and query transactions of smart contracts. Among them, the query transactions include QueryAttr contract, QueryData contract, and AccessDecision contract. The invoke transactions include the CreateAttr contract, ChangeAttr and CreateData contract.

Figure 11 displays the latency and throughput of query and invoke transactions for transaction volumes ranging from 100 to 1000. According to the experimental results, the throughput is around 8 TPS, and the average latency of invoke transactions grow from 797 ms to 1208 ms with an increase in transaction volume. The throughput improves from 88 TPS to 541 TPS, with query transactions having a maximum latency of roughly 18 ms and an average delay of about 1 ms. Because query transactions only involve one Peer node while invoke transactions perform data uplink operations involving multiple Peer nodes and require Order nodes to participate in transaction sorting and packaging, the invoke transaction latency is higher and throughput is lower under different transaction volumes.

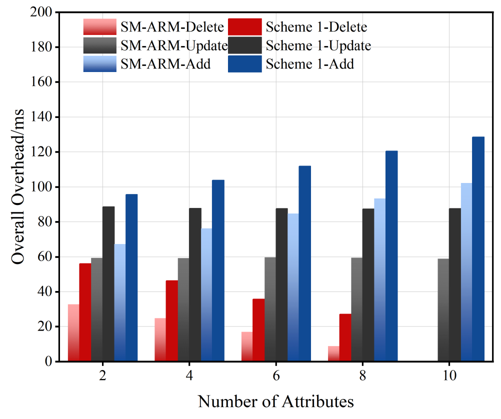

FIGURE 13. **Relationship between attribute revocation type and overall overhead.**

This experiment continues to examine the transaction latency and throughput of invoke transactions with a concurrency of 10 TPS to 40 TPS to research the throughput of invoke transactions with tolerable transaction latency. Figure 12 illustrates that the highest average transaction latency and throughput for invoke transactions, both within acceptable bounds, are 7386 ms and 17 transactions per second, respectively.

7) PROPERTY REVOCATION OVERHEAD TEST

When compared to key update time, the SM-ARM requires the SM3 algorithm to compute and get the key hash value as a single pass, which is insignificant.

Contrary to the attribute revocation mechanism designed in Scheme 1, SM-ARM determines the user's access right by checking the key version through a smart contract, rather than confirming the key and ciphertext version. SM-ARM achieves attribute revocation by simply updating the key, without updating the ciphertext. The AccessDecision contract time is negligible, according to the results of the blockchain performance test. Therefore, the SM-ARM is superior to Scheme 1 in terms of ciphertext updates. Our scheme has a slightly higher key update overhead than Scheme 1. As shown in Figure 13, the overall cost of SM-ARM is lower than that of Scheme 1, which includes the computational cost when DU performs attribute deletion, update, and addition. In addition, our proposed attribute revocation mechanism based on sentinel mode is also superior to Scheme 1.

VII. CONCLUSION

We have proposed a personal privacy data protection scheme for encryption and revocation of high-dimensional attribute domains, which addresses the issues of low security, significant computational overhead, and high attribute revocation cost of current schemes in high-dimensional attribute domains. Compared with existing data protection schemes, HAD-FME is based on FAME and SM4 with high security, which can reduce the computing overhead and meet the requirements of secure storage and sharing of data in high-dimensional attribute domains. We also designed an Attribute Revocation Mechanism Based on Sentry Mode (SM-ARM) to reduce the cost of attribute revocation by updating only the user version key. We have assumed in this paper that STSS is unable to obtain a complete DU private key, and the blockchain system exhibits performance limitations. In the future, we will plan to research multi-authority-based key generation schemes that ensure DU security, while exploring privacy data protection schemes based on high-performance tamper-proof systems to improve the throughput and performance of the schemes.

REFERENCES

[1] P. Patil and M. Sangeetha, "Blockchain-based decentralized KYC verification framework for banks," *Proc. Comput. Sci.*, vol. 215, pp. 529–536, Jan. 2022, doi: 10.1016/j.procs.2022.12.055.

[2] V. Mani, M. M. Ghonge, N. K. Chaitanya, O. Pal, M. Sharma, S. Mohan, and A. Ahmadian, "A new blockchain and fog computing model for blood pressure medical sensor data storage," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108202, doi: 10.1016/j.compeleceng.2022.108202.

[3] X. Qin, Y. Huang, Z. Yang, and X. Li, "A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing," *J. Syst. Archit.*, vol. 112, Jan. 2021, Art. no. 101854, doi: 10.1016/j.sysarc.2020.101854.

[4] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, G. Srivastava, and H. Karimipour, "Secure intelligent fuzzy blockchain framework: Effective threat detection in IoT networks," *Comput. Ind.*, vol. 144, Jan. 2023, Art. no. 103801, doi: 10.1016/j.compind.2022.103801.

[5] L. Zhou, K. Qin, C. F. Torres, D. V. Le, and A. Gervais, "High-frequency trading on decentralized on-chain exchanges," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 428–445, doi: 10.1109/sp40001.2021.00027.

[6] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Apr. 2018, pp. 1–15, doi: 10.1145/3190508.3190538.

[7] X. Yang, Y. Zhang, S. Wang, B. Yu, F. Li, Y. Li, and W. Yan, "LedgerDB: A centralized ledger database for universal audit and verification," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 3138–3151, Aug. 2020, doi: 10.14778/3415478.3415540.

[8] C. Yue, T. T. A. Dinh, Z. Xie, M. Zhang, G. Chen, B. C. Ooi, and X. Xiao, "GlassDB: An efficient verifiable ledger database system through transparency," *Proc. VLDB Endowment*, vol. 16, no. 6, pp. 1359–1371, Feb. 2023, doi: 10.14778/3583140.3583152.

[9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334, doi: 10.1109/SP.2007.11.

[10] S. Agrawal and M. Chase, "FAME: Fast attribute-based message encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 665–682, doi: 10.1145/3133956.3134014.

[11] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptogr.*, 2011, pp. 53–70, doi: 10.1007/978-3-642-19379-8_4.

[12] J. Chen, R. Gay, and H. Wee, "Improved dual system ABE in prime-order groups via predicate encodings," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2015, pp. 595–624, doi: 10.1007/978-3-662-46803-6_20.

[13] Y. Zhang, D. He, and K.-K.-R. Choo, "BaDS: Blockchain-based architecture for data sharing with ABS and CP-ABE in IoT," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, Nov. 2018, doi: 10.1155/2018/2783658.

[14] P. Sharma, R. Jindal, and M. D. Borah, "Blockchain-based decentralized architecture for cloud storage system," *J. Inf. Secur. Appl.*, vol. 62, Nov. 2021, Art. no. 102970, doi: 10.1016/j.jisa.2021.102970.

[15] X. Lu and S. Fu, "A trusted data access control scheme combining attribute based encryption and blockchain," *Netinfo Secur.*, vol. 21, no. 3, pp. 7–8, 2021, doi: 10.3969/j.issn.1671-1122.2021.03.002.

[16] Z. Yang, S. Huang, and C. Y. Zheng, "Study on crowdsourced testing intellectual property protection technology based on blockchain and improved CP-ABE," *Comput. Sci.*, vol. 49, no. 5, pp. 325–332, 2022, doi: 10.11896/jsjkx.210900075.

[17] P. Liu, Q. He, and W. Y. Liu, "CP-ABE scheme supporting attribute revocation and outsourcing decryption," *Netinfo Secur.*, vol. 20, no. 3, pp. 90–97, 2020, doi: 10.3969/j.issn.1671-1122.2020.03.012.

[18] D. Zheng, B. Qin, Y. Li, and A. Tian, "Cloud-assisted attribute-based data sharing with efficient user revocation in the Internet of Things," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 18–23, Jun. 2020, doi: 10.1109/MWC.001.1900433.

[19] F. Liu, W. Ji, L. Hu, J. Ding, S. Lv, A. Pyshkin, and R.-P. Weinmann, "Analysis of the SMS4 block cipher," in *Proc. 12th Australas. Conf. Inf. Secur. Privacy*, 2007, pp. 158–170, doi: 10.1007/978-3-540-73458-1_13.

[20] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *J. Med. Syst.*, vol. 43, no. 1, Jan. 2019, Art. no. 5, doi: 10.1007/s10916-018-1121-4.

[21] Z. Ullah, B. Raza, H. Shah, S. Khan, and A. Waheed, "Towards blockchain-based secure storage and trusted data sharing scheme for IoT environment," *IEEE Access*, vol. 10, pp. 36978–36994, 2022, doi: 10.1109/ACCESS.2022.3164081.

[22] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Trans. Comput.*, vol. 71, no. 1, pp. 175–184, Jan. 2022, doi: 10.1109/TC.2020.3043950.

[23] T. Lee, H.-S. Moon, and J. Jang, "Data encryption method using CP-ABE with symmetric key algorithm in blockchain network," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 1371–1373, doi: 10.1109/ICTC52510.2021.9620889.

[24] Y. J. Wang, C. T. Cao, and L. You, "A novel personal privacy data protection scheme based on blockchain and attribute-based encryption," *J. Cryptol. Res.*, vol. 8, no. 1, pp. 14–27, 2021, doi: 10.13868/j.cnki.jcr.000416.

[25] Z. Kang, J. Li, J. Shen, J. Han, Y. Zuo, and Y. Zhang, "TFS-ABS: Traceable and forward-secure attribute-based signature scheme with constant-size," *IEEE Trans. Knowl. Data Eng.*, early access, Feb. 3, 2023, doi: 10.1109/TKDE.2023.3241198.

[26] G. Zhang, X. Chen, B. Feng, X. Guo, X. Hao, H. Ren, C. Dong, and Y. Zhang, "BCST-APTS: Blockchain and CP-ABE empowered data supervision, sharing, and privacy protection scheme for secure and trusted agricultural product traceability system," *Secur. Commun. Netw.*, vol. 2022, pp. 1–11, Jan. 2022, doi: 10.1155/2022/2958963.

[27] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *Int. J. Inf. Secur.*, vol. 14, no. 6, pp. 487–497, Nov. 2015, doi: 10.1007/s10207-014-0270-9.

[28] G. Chen, Z. Xu, J.-J. Zhang, G.-J. Wang, H. Jiang, and M.-Q. Huang, "Generic attribute revocation systems for attribute-based encryption in cloud storage," *Frontiers Inf. Technol. Electron. Eng.*, vol. 20, no. 6, pp. 773–786, Jun. 2019, doi: 10.1631/fitee.1800512.

[29] H. Lian, Q. Wang, and G. Wang, ''Large universe ciphertext-policy attribute-based encryption with attribute level user revocation in cloud storage,'' *Int. Arab J. Inf. Technol.*, vol. 17, no. 1, pp. 107–117, Jan. 2020, doi: 10.34028/iajit/17/1/13.
[30] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, ''User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage,'' *IEEE Syst. J.*, vol. 12, no. 2, pp. 1767–1777, Jun. 2018, doi: 10.1109/jsyst.2017.2667679.
[31] M. Fischer, A. Scheerhorn, and R. Tönjes, ''Using attribute-based encryption on IoT devices with instant key revocation,'' in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2019, pp. 126–131, doi: 10.1109/PERCOMW.2019.8730784.
[32] D. L. Fekete and A. Kiss, ''A survey of ledger technology-based databases,'' *Future Internet*, vol. 13, no. 8, p. 197, Jul. 2021, doi: 10.3390/fi13080197.
[33] M. Gorbunova, P. Masek, M. Komarov, and A. Ometov, ''Distributed ledger technology: State-of-the-art and current challenges,'' *Comput. Sci. Inf. Syst.*, vol. 19, no. 1, pp. 65–85, 2022, doi: 10.2298/csis210215037g.
[34] X. Yang, S. Wang, F. Li, Y. Zhang, W. Yan, F. Gai, B. Yu, L. Feng, Q. Gao, and Y. Li, ''Ubiquitous verification in centralized ledger database,'' in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 1808–1821, doi: 10.1109/ICDE53745.2022.00181.
[35] *Amazon Quantum Ledger Database*. Accessed: Jun. 29, 2023. [Online]. Available: https://aws.amazon.com/qldb/
[36] A. Lewko and B. Waters, ''Decentralizing attribute-based encryption,'' in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2011, pp. 568–588, doi: 10.1007/978-3-642-20465-4_31.
[37] X. Wang and H. Yu, ''SM3 cryptographic hash algorithm,'' *J. Inf. Secur. Res.*, vol. 2, no. 11, pp. 983–994, 2016.
[38] G. Wang and Y. Shen, ''Preimage and pseudo-collision attacks on step-reduced SM3 hash function,'' *Inf. Process. Lett.*, vol. 113, no. 8, pp. 301–306, Apr. 2013, doi: 10.1016/j.ipl.2013.02.006.

CAIMEI WANG received the Ph.D. degree in computer science from the University of Science and Technology of China, in 2018. She is currently an Associate Professor with the School of Artificial Intelligence and Big Data, Hefei University. Her research interests include computer networks, trusted computing, and information security.

JIANHAO LU was born in 1998. He is currently pursuing the master's degree with the School of Artificial Intelligence and Big Data, Hefei University, China. His research interests include blockchain and attribute-based encryption.

XINLU LI received the M.S. degree in computer science from Anhui University, Anhui, China, in 2009, and the Ph.D. degree in computer science from Technological University (TU Dublin), Dublin, Ireland, in 2019. From 2014 to 2018, he was a Teaching Assistant with the School of Computing, TU Dublin. Since 2019, he has been an Associate Professor with the School of Artificial Intelligence and Big Data, Hefei University. His research interests include artificial intelligence and natural language processing.

PEI CAO received the B.Eng. degree in software engineering from Shandong University, in 2016, and the Ph.D. degree from the University of Science and Technology of China, in 2022. She is currently a Lecturer with Hefei University. Her research interests include the cross-application of AI in engineering and multimodal application technology.

ZIJIAN ZHOU received the B.S. degree from Hefei University, in 2023. His research interests include information security, decentralizing systems, and cloud computing.

QILUE WEN was born in Anhui, China. She is currently pursuing the master's degree with the School of Artificial Intelligence and Big Data, Hefei University, China. Her research interest includes cloud storage security.

• • •