Efficient and Flexible Multi-Authority Attribute-Based Authentication for IoT Devices

Ye Su, Xi Zhang, Jing Qin and Jixin Ma.

***Abstract*—The correctness and reliability of data sources are the keys to the practicality of data collected by IoT devices. Attribute-based signature(ABS) is a cryptographic primitive for users to sign with their own attributes, which can be applied to the authentication process in IoT scenarios. The attribute authority is responsible for issuing the attribute key to the user in ABS. Multiple authorities can complete attribute management tasks to avoid the threat of a single authority. However, attribute authorities need to execute multiple interactions to collaborate to generate attribute keys for users, which brings a large transmission burden. In addition, a lot of resource-constrained terminals in the IoT mostly play the role of signer or verifier in authentication protocols. The signature generation and verification algorithms often have heavy pairing and exponentiation operations. Currently, no ABS scheme takes into account the efficiency of all participating entities simultaneously. In this paper, we present an aggregated anonymous key issue (AAKI) protocol to reduce the transmission burden between multiple authorities. Meanwhile, the non-interactive zero-knowledge proof aggregate exponentiation (NI-ZKPoKAE) protocol is designed to aggregate the transmitted secret values in AAKI. To reduce the computational burden of signers and verifiers, Blakley secret sharing, where the Hadamard matrix is used more efficiently to handle the (n, n)-threshold, is used to construct an efficient and fine-grained multi-authority ABS (EFMA-ABS) scheme. This brings high efficiency to all three types of parties involved in IoT authentication. Our above-mentioned protocols have been proven to be feasible and effective.**

***Index Terms*—attribute-based signature, multi-authority, IoT authentication, proof of knowledge, aggregated anonymous key issue, Blakley secret sharing.**

I. Introduction

THE Internet of Things (IoT) is a growing global Internet object network. Objects in the network can collect data, transmit data, process data and communicate with each other. It allows people to reach out to anywhere with sensors connected to the network. Anywhere. Thanks to the current 5G network with lower latency and large bandwidth, industries related to the Internet of things, such as automatic driving and smart wear have achieved rapid development. The number of devices and data in the Internet of Things will also increase exponentially. However, the control and management of each device still need to be achieved through the transmission and connection of information. The data generated by these smart devices releases numerous business opportunities. Considering the complex demands and high fragmentation of the Internet of Things, in order to provide stable and high-quality network connection services for IoT terminal customers, many service vendors, such as Alibaba Cloud [1], have launched the Internet of Things wireless connection service. Such services are characterized by wide variety, intelligence and ecological connectivity. With the help of cloud platform technology, it provides flexibility, safety and stability for many scenarios such as the Internet of Vehicles, smart wearable devices, mobile payments, digital media economic solutions, environmental monitoring, and smart agriculture.

However, currently hindering people's widespread acceptance of Internet of Things services are concerns about the related security issues that follow. Smart devices that have traditionally been isolated or disconnected are now fully exposed to the external environment and the Internet, which makes them vulnerable to attacks from the Internet. In 2021, Zscaler ThreatLabz [2], as a threat research team, gave an annual report on IoT security. It pointed out that during the Covid-19 epidemic, as more people work from home, they needed to connect with corporate networks through IoT devices. Resulting in a 700% increase in malicious attacks on IoT devices in the enterprise in 2020 compared to the previous year. The range of IoT devices connected to corporate networks continues to expand, including from smartwatches and IP cameras to cars and music devices. The communication of these IoT devices can be easily accessed by attackers. Worse, the communication is easily intercepted and modified, allowing attackers to exploit IoT devices for malicious attacks. Therefore, people call for stronger security in the IoT network, which includes the security of the protocols, communication transmission process, and the privacy of data and identity.

Attribute-based signature [3] is proposed as a cryptographic primitive to protect the identity privacy of the signer. The signer uses a set of attributes he owns to sign the message. As a signature protocol, it can not only provide identity authentication but also protect the identity of the signer. In the process of signature verification, the verifier can only get the attributes that the signer satisfies, but cannot locate the specific identity. Previous attribute signature protocols [4]–[6] designed for (t, n) threshold attributes, and later protocols [7]–[10] designed for fine-grained AND gate, OR gate and tree

Ye Su is with School of Information Science and Engineering, Shandong Normal University, Jinan, 250358, China, and School of Mathematics, Shandong University, Jinan, 250100, China. E-mail: (sy0422@163.com)

Xi Zhang is with School of Mathematics, Shandong University, Jinan, 250100, China. E-mail: (mathxizhang@mail.sdu.edu.cn)

Jing Qin (corresponding author) is with School of Mathematics, Shandong University, Jinan, 250100, China, and State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China). E-mail:(qinjing@sdu.edu.cn)

Jixin Ma is with Centre for Computer and Computational Science at School of Computing and Mathematical Sciences, University of Greenwich, London, UK. E-mail:(j.ma@greenwich.ac.uk)

access structure. Attribute signature protocols can be used to solve the problem of fine-grained identity authentication that protects user identity privacy in IoT.

The two main participants to the authentication algorithm in IoT systems, the signer and the verifier, are usually low-performing entities that are sensitive to the amount of computation and transmission. However, the current attribute signature protocol requires a lot of computationally heavy bilinear pairing and modular exponential operations, which appear in signature algorithm and verify algorithm in which the signer and verifier are mainly involved. They are not suitable for current application scenarios based on 5G networks, such as vehicle-mounted Internet scenarios [11]. In the vehicle-mounted Internet scenario, multiple message transmission and authentication tasks need to be completed quickly and accurately in a short period of time, which brings great challenges to sensor devices with low performance in intelligent vehicles and edge terminals in base stations. Therefore, designing an attribute signature protocol with both efficient signature and verification algorithms is the key to the practical application of IoT system.

A large number of devices will be connected to an IoT network system, and there will be many subdivided attributes. If only one attribute authority is used to manage the distribution of attribute keys, it will impose a huge computational burden on this single authority. At the same time, if the attribute authority is corrupted by an adversary, the security of the entire system will no longer exist. Based on this consideration, the attribute cryptosystem of multi-attribute authorities is proposed [12]–[16]. In these efforts, they decentralized a single attribute management center and used multiple attribute authorities to collaborate to complete the key distribution. Although the multi-attribute authority improves the robustness of the attribute cryptosystem, more computation and communication overhead will be incurred. On the one hand, the increased overhead comes from the interaction between the user and multiple attribute authorities to generate a set of attribute keys. On the other hand, it comes from the interaction between multiple attribute authorities when generating attribute keys for the same set of attributes. Current multi-attribute authority cryptographic schemes use an anonymous key distribution protocol constructed with the help of zero-knowledge proof technology proposed by Chow et al. [17] in the process of generating attribute keys by multiple attribute authorities interactively. The number of interactions and throughput of anonymous key distribution will increase obviously, when the number of attribute authorities participating in protocol execution increases. How to reduce the communication burden of the attribute authorities in this process has also become an urgent need for the smooth implementation of the multi-attribute authority cryptosystem.

In the existing multi-attribute authority cryptosystem, multiple attribute authorities are in equal status, and they are all at the top of the defined access structure. In other words, the multiple attribute authorities form the first level child nodes of the root node of the tree access structure. However, in real scenes, there are more possibilities for the relationship between attribute authorities. One situation is that in the access structure, there may still be a threshold gate between the attribute authorities up to the root. At this point, the attribute authorities is no longer at the first level child of the root node of the tree access structure. For example, there are three attribute authorities A, B and C in an IoT system. When the user needs to apply for the attribute key from these three attribute authorities, the defined access structure meets the requirements from the attribute authorities to the root: authority A and authority B need to meet the OR gate, and their result needs to meet the AND gate with authority C. This scenario, which requires richer attribute access structures to express, was not considered in previous work. Designing a more flexible multi-attribute authority cryptographic scheme will make the attribute cryptosystem more suitable for the real scenario.

A. Our Contribution

In view of the above discussion, we design an efficient, flexible and fine-grained multi-attribute authority signature scheme for the authentication process in IoT. Our proposal takes into account the efficiency of all parties involved in IoT authentication, including signers, verifiers, and multi-attribute authorities. The main contributions are as follows:

- Firstly, to reduce the transmission burden between multiple authorities, we present a non-interactive zero-knowledge proof aggregate exponentiation (NI-ZKPoKAE) protocol, which can help aggregate multiple secret values into one value in a secret transfer. Then, we use it to construct an aggregated anonymous key issue (AAKI) protocol to complete the interaction between multiple attribute authorities to generate a set of attribute keys for users.
- Secondly, to reduce the computational burden of signers and verifiers, we construct an efficient and fine-grained multi-authority attribute-based signature (EFMA-ABS) scheme based on the co-CDH assumption. The Blakely's secret sharing is used to construct the signature scheme, and the Hadamard matrix is used to characterize (n, n) gate to reduce computation cost in both signature and verification phase.
- Thirdly, we analyze the computing and storage efficiency of the scheme in theory and experimental simulation. An IoT authentication system based on our proposed multi-authority attribute-based signature scheme is presented.

B. Related Works

The Internet of Things (IoT) is a communication paradigm hosting a growing number of devices that can sense, collect, connect, and exchange data. The IoT model has been increasingly used worldwide to enhance the quality of daily life. A relatively new case is that during the COVID-19 pandemic, medical IoT improved the service level in the healthcare field by simplifying accessibility and increasing efficiency, enabling doctors to connect on-demand with patients in hospitals and isolated at home. These situations are forcing scientists and researchers to increase the use of IoT systems. However, the widespread deployment of IoT

systems has brought new security challenges, one of which is how IoT devices achieve authentication. In recent years, the methods used by researchers to construct IoT authentication mechanisms mainly include blockchain-based methods, multi-factor authentication, physically unclonable functions (PUFs) and public key infrastructure (PKI) [18].

Using the public key infrastructure, Almalki et al. [19] proposed an efficient and privacy-preserving authenticated data aggregation scheme based on additive homomorphic encryption in the healthcare IoT system. In order to design and build an access control and access detection model in a distributed IoT environment, Zhang et al. [20] proposed an IND-CCA secure multi-authority ciphertext-policy ABE scheme with outsourced decryption and progressive mode attribute-based authentication. Sun et al. [16] designed an outsourcing decentralized multi-authority Attribute-based signature scheme (ODMA-ABS) and used it to construct the IoT authentication protocol. Our proposal continues the research of the work [16] and focuses on developing the application of PKI in IoT authentication.

In order to maintain the privacy of the signer's identity, the attribute signature scheme (ABS) was first proposed by Maji et al. [3] in 2008. In the attribute signature scheme, the user can use a set of attributes to replace his own identity to generate a signature. During signature verification, the verifier can only know the attributes that the signer satisfies, but not the identity of the signer. In [3], they proposed a scheme to support the predicate described by the monotone span program. Later, Shahandashti et al. [4] and Li et al. [5] respectively proposed attribute signature schemes that support (k, n) threshold predicates. The schemes [7]–[9] were designed for more fine-grained attribute signature protocols for AND gates, OR gates and tree access structures. However, the above schemes are all constructed for the case of a single-authority setting. On the one hand, the single-authority setting will cause the attribute authority to face greater computational pressure; on the other hand, if the authority is corrupted or damaged, the whole system will be paralyzed.

Therefore, some early works [12], [13], [15] proposed attribute signature schemes for multi-authority. In order to prevent multiple users from combining their attributes to forge a set of attributes to sign, some attribute signature schemes [16], [21] adopt the method of Chow et al. [17], which increases the interaction among multiple attribute authorities to negotiate the secret value, and hides the secret value in the attribute key of the user. The above schemes are based on CDH assumption. However, these schemes only take into account the efficiency of one of the signature algorithm and the verify algorithm at most, and do not reduce the computation for both low-performance signers and verifiers in the IoT authentication process.

In order to relieve the computational pressure of low-performance signers, Sun et al. [16] outsourced a large amount of the signer's calculations to cloud servers, and constructed an outsourced multi-authority attribute signature scheme. And their scheme is based on the co-CDH assumption, which makes the scheme obtain higher security. Li et al. [22] outsourced a large amount of computation in the signature and verification phase to the server and constructed a server-assisted multi-authority ABS scheme. However, the method of outsourcing computing requires IoT terminals to interact with cloud server, which is a burden for IoT devices with limited transmission capacity. It can be seen that attribute signature schemes not only needs to take security and robustness into account, but also needs to pursue higher computing and storage efficiency in both signature and verification phase, and needs to have richer access structure expression.

Chase and Chow [23] constructed a generic anonymous key issuing (AKI) protocol, then the AKI protocol is widely used in the construction of multi-authority attribute cryptosystems [16], [17], [24], [25]. This protocol provides a way for users and attribute authorities to interact with each other for trusted secret computing. The AKI protocol is one of the important components of the multi-authority attribute signature scheme. In this protocol, proof of knowledge (PoK) is utilized in the interactive transmission of secret values. In the existing AKI protocol, each transmission of a secret value has to execute a PoK protocol, which makes the amount of transmission and the number of interactions proportional to the total number of attribute authorities. When the number of attribute authorities is large, the transmission amount between attribute authorities will bring a great burden to each authority during executing the AKI protocol. Boneh et al. [26] discussed the PoK protocol that supports aggregated transport when improving transmission efficiency in the blockchain. In order to relieve the transmission pressure of attribute authorities, how to design an aggregate PoK protocol suitable for the AKI protocol, so as to obtain a more efficient AKI protocol under the multi-authority setting has become an urgent problem to be solved.

From the above description, the IoT authentication solutions constructed based on the existing ABS schemes have some degree of deficiencies in bandwidth performance, security assurance and practical deployment possibilities, they cannot be deployed effectively yet. IoT systems call for an authentication scheme that can take into account security, robustness, fine-grained access structure, computational efficiency and rapid response in both signing and verification phases, and efficient transmission by attribute authorities. In particular, schemes [3]–[9] only consider the single-attribute authority scenario. Although schemes [12], [13], [15], [21] use multi-attribute authorities to provide robustness for authentication protocols, none of them consider reducing the computation cost of both the signer and verifier simultaneously. Schemes [16], [22] consider improving the efficiency of both signature and verification phases, but introduce a third party to share the computational pressure. This increases the transmission burden on IoT low-performance terminals, which is unsuitable for IoT scenarios requiring a fast response. In addition, the above multi-authority attribute-based signature schemes all use the traditional AKI protocol in [17] to generate the signature key interactively, so the transmission burden is heavy. To the best of our knowledge, there is not yet a complete scheme that can satisfy all the above requirements. In this paper, we aim to propose a new scheme to solve the above issues thoroughly.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263497

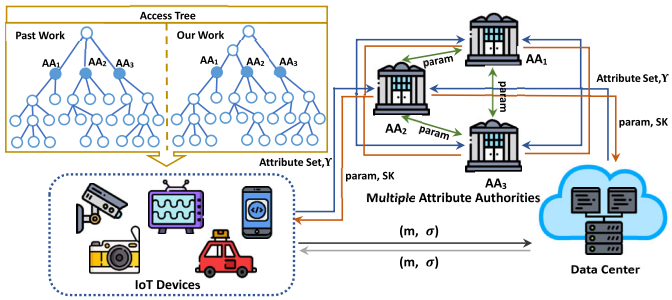


Fig. 1. System Architecture: "AA_1, AA_2, AA_3" refer to different attribute authorities, "IoT Devices" refer to the entities that collect and transfer data and sign the data with attribute keys, "Data Center" refers to a high-performance entity that collects, analyzes and sends authenticated commands.

C. IoT Authentication System Model

In the Internet of Things environment, the authenticity of the collected data and the reliability of the commands issued by the data analysis are crucial to the robust operation of the IoT system. Therefore, our IoT identity authentication system model focuses on the authenticity of the data and the reliability of the data source, as well as improving the flexibility of the access structure, and reducing the computation and communication costs. Figure 1 shows our IoT identity authentication system model, which contains three types of entities, namely multiple attribute authorities, data centers and IoT devices.

- **Multiple attribute authorities**: Multiple attribute authorities jointly generate system parameters for data centers and IoT devices. Each attribute authority is responsible for maintaining a subset of attributes. When an entity in the system wants to generate a set of attribute keys for its own attribute set, multiple attribute authorities will generate them together through interaction.
- **Data center**: This entity generally has powerful computing, storage, and transmission capabilities. It is mainly responsible for collecting and storing data from IoT devices, deriving new commands by analyzing these data, and returning the new commands to the IoT devices. In this process, the data center first uses its own attribute set to apply for system parameters and attribute keys from multiple attribute authorities, then stores them locally. Second, the data center verifies the data and its signature sent by the IoT device to ensure the authenticity of the data. Third, the data center uses its own attribute key to sign the commands generated after analyzing the data and sends them to the IoT devices.
- **IoT devices**: Sensor-equipped entities have limited computing and storage capabilities. IoT devices use their own attribute set to apply for system parameters and attribute private keys from multiple attribute authorities, and store them locally. When the device collects data, it signs the data with its own attribute key and sends them to the data center. Whenever the data center sends a new command to the IoT device, the IoT device verifies that the signature of the command is correct before executing it.

D. Design Idea

In the existing multi-authority attribute cryptosystem, whether it is a threshold attribute structure or a more fine-grained access structure, multiple attribute authorities are in equal status, and they are all at the top of the defined access structure. This means that multiple attribute authorities can only be located at the first level child of the root node of the access structure. However, in the real world, there are more possibilities for the relationship between attribute authorities. Our multi-attribute authority scheme considers a tree-based access structure in which each node can be an AND gate, an OR gate, or a (t, n) gate. Compared with the previous multi-attribute authority signature schemes, our scheme also considers that there can be some gates between the attribute authority and the root of the access structure, which makes the authentication method more flexible. We illustrate it in figure 1. "AA_1, AA_2, AA_3" respectively represent three attribute authorities. In the previous works on the tree access structure of multi-authority attribute cryptosystem, the three attribute authorities can only be at the top of the tree, and they constitute an AND gate together. Our proposal considers that there can be some AND gates, OR gates or (t, n) gates between each attribute authority and the root node of the tree. For example, an OR gate can be run between attribute authority AA_1 and AA_2, and their result will be an AND gate run with attribute authority AA_3.

Although schemes such as ePass [8] satisfy the attribute privacy security model defined by Li et al. [5]. However, in the signature verification phase, just a part of the verification is performed using the public key, and the attribute value that is actually used to recover the secret will be found. Compared with the attribute privacy achieved by the threshold attribute cryptography schemes such as Li's [5] or Sun's [16], the attribute privacy satisfied by these more fine-grained attribute signature schemes becomes less meaningful. The main reason for this problem is the more fine-grained access structure that the scheme is intended to implement. The verifier must know which real attributes the signer used to sign, and then know the real path to recover the secret in the tree structure. It seems inevitable. Therefore, our scheme abandoned this weak attribute privacy in exchange for a more efficient signature verification algorithm.

Blakley's secret sharing [27] was proposed at the same time as Shamir's secret sharing [28], which is a more general case of Shamir's protocol. Just as the Shamir secret sharing is initialized by a polynomial, Blakley's protocol is initialized by a matrix. Xia et al. [29] pointed out that when Blakley's secret sharing is initialized by a special matrix, the Hadamard matrix, it makes the secret recovery process much easier. Define matrix multiplication for matrices $A = (a_{i,j})_{m\times p}$ and $B = (b_{i,j})_{p\times n}$ as $A \times B = \sum_{k=1}^{p} a_{ik}b_{kj}$. Let the Hadamard matrix of order q be $H = (b_{i,j})_{q\times q}$. Any two rows in H are perpendicular, and H is made up of the elements 1 and -1. The Hadamard matrix satisfies $H \times H^T = n \cdot I_n$, where H^T represents H's transpose and I_n represents the $n \times n$ identity matrix. Since the largest amount of computation in the secret recovery phase of Blakley's secret sharing comes from

inverting the matrix, the use of the Hadamard matrix makes this process require only a small amount of computation. Because we can get the inverse matrix by dividing Hadamard's transpose H^T by n. This greatly improves the efficiency of secret sharing protocols. The only problem, however, is that the Hadamard matrix is a square matrix and therefore only applies to the (n, n) threshold case. To take full advantage of this good property, we set all nodes in the access tree in our scheme as (n, n) gates to obtain more efficient signature verification phase.

E. Organization

The remainder of the paper is organized as follows. In Section II, we introduce the preparatory knowledge needed for subsequent schemes. We construct a new aggregated anonymous key issue protocol in Section III. In this section, we also present a new non-interactive zero-knowledge proof aggregate exponentiation protocol as one component of the new AAKI protocol, and give the security proof of these protocols. In Section IV, we present our efficient and flexible multi-authority attribute signature (EFMA-ABS) scheme, as well as security proof, theoretical analysis and experimental performance. We use our EFMA-ABS scheme to construct an authentication protocol for IoT system in Section V. We sum up our paper in Section VI.

II. Preliminaries

We present the definitions of bilinear map and access structure. After that, we introduce the co-CDH assumption and adaptive root assumption that our schemes are based on. Finally, the definitions of Blakely secret sharing and Hadamard matrix are given.

A. Bilinear Pairing

Bilinear map, as the main tool of our protocol, are an important tool in many cryptographic-based non-interactive authentication protocols.

Definition 1 (Bilinear pairing)**.** *Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three multiplicative cyclic groups with the prime order p. Let g_1 be a generator of group $\mathbb{G}_1$ and g_2 be a generator of group $\mathbb{G}_2$. A map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear pairing if the following conditions hold:*

- *Bilinearity. $e(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $a, b \in \mathbb{Z}_p$.*
- *Non-degeneracy. For any generators $G_1 \in \mathbb{G}_1, G_2 \in \mathbb{G}_2$, $e(G_1, G_2) \neq 1_{G_T}$.*
- *Computability. It is efficient to compute $e(G_1, G_2)$ for any $G_1 \in \mathbb{G}_1, G_2 \in \mathbb{G}_2$.*

Our scheme is constructed based on asymmetric group, that is, there is either no computable isomorphism ϕ from $\mathbb{G}_1$ to $\mathbb{G}_2$ or there is no computable isomorphism ϕ' from $\mathbb{G}_2$ to $\mathbb{G}_1$.

B. Access Structure

Our construct is suitable for the access tree scenario. Let $\mathcal{U}$ be the universe attribute set. $\mathcal{AA}$ denotes the set of all attribute authorities. Let $\mathcal{U}_{k\in\{1,\cdots,N\}} \subset \mathcal{U}$ be the attribute universe of each attribute authority $A_{k\in\{1,\cdots,N\}} \in \mathcal{AA}$.

Definition 2 (Access Structure)**.** *Let $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ be a set of parties. A collection $\Upsilon \subseteq 2^{\mathcal{P}} \backslash \{\emptyset\}$ satisfies monotone property, if for any $X, Y \subseteq \mathcal{P}$ such that $X \in \Upsilon$ and $X \subseteq Y$, then $Y \in \Upsilon$ holds. A collection Υ is called the monotone access structure, where the sets in Υ are the authorized sets and the sets not in Υ are the unauthorized sets.*

Definition 3 (Access Tree)**.** *An access tree Υ is an access structure represented by a tree structure. The leaves are associated with attributes and each non-leaf node of the tree defines a threshold gate, which is represented by its child nodes and threshold value. Let num_x be the number of children of node x and $k_x \in [0, num_x]$ is its threshold value. The threshold gate is an AND gate when $k_x = num_x$, and OR gate when $k_x = 1$. Let R be the root of the access tree Υ. Υ_x is the subtree of Υ rooted at the node x. If the set of attributes γ satisfies Υ_x, denote it as $\Upsilon_x(\gamma) = 1$.*

In our multi-attribute authority signature scheme, define the access tree as $\Upsilon = \Upsilon_{ex} \bigcup_{k\in\{1,\cdots,N\}} \Upsilon_k$. Υ_{ex} is the access tree between the attribute authorities, which takes the attribute authorities as the leaves. Υ_k is the internal access tree of attribute authority k, where authority k as the root node and the attributes are associated with the leaves. To facilitate our description of the access tree, we define the following two notations. Let the function $p(\cdot)$ be a mapping from the child node x to its parent node in Υ. The function $l(\cdot)$ maps the child node to its location in its parent node.

C. Computational Assumption

Our construction is based on the co-CDH assumption and adaptive root assumption. The co-CDH assumption was first proposed by boneh et al. [30] to deal with the CDH assumption when $\mathbb{G}_1 \neq \mathbb{G}_2$ in the bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. The adaptive root assumption was first proposed by Wesolowski [31] from the root finding problem. The adaptive root assumption implies that the adversary can't compute the order of any non-trivial element.

Definition 4 (co-CDH assumption)**.** *Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three multiplicative cyclic groups with the prime order p. Let g_1 be a generator of group $\mathbb{G}_1$ and g_2 be a generator of group $\mathbb{G}_2$. A map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map. We say that the co-CDH assumption is (t, ϵ)-hold if for any probabilistic polynomial t-time algorithm $\mathcal{A}$ there are*

$$|Pr[a, b \leftarrow_R \mathbb{Z}_p : \mathcal{A}(p, g_1, g_2, g_1^a, g_2^b) = g_2^{ab}]| \leq \epsilon.$$

Definition 5 (Adaptive Root Assumption)**.** *Let $\lambda \in \mathbb{N}$ be the security parameter and $GGen(\cdot)$ be a randomized group generation algorithm. Let the group $\mathbb{G} \xleftarrow{\$} GGen(\lambda)$ and a random prime number $l \xleftarrow{\$} H_{Prime}(\lambda)$. There are efficient polynomial time adversaries $(\mathcal{A}_0, \mathcal{A}_1)$, where $x \xleftarrow{\$} \mathcal{A}_{i\in\{0,1\}}(\cdot)$ denotes the random variable that is the output of a randomized algorithm $\mathcal{A}_{i\in\{0,1\}}$. They run $(\omega, \text{state}) \xleftarrow{\$} \mathcal{A}_0$, where ω is an*

element of $\mathbb{G}$. *Run* $u \xleftarrow{\$} \mathcal{A}_1(l, \text{state})$, *where* $u \in \mathbb{G}$ *satisfies* $u^l = \omega \neq 1$. *We say that the adaptive root assumption holds for* $GGen(\cdot)$ *if for all pairs of adversaries* $(\mathcal{A}_0, \mathcal{A}_1)$, *the probability* $Pr[(\omega, \text{state}) \xleftarrow{\$} \mathcal{A}_0;\ u \leftarrow \mathcal{A}_1(l, \text{state}) : u = \omega^{1/l} \neq 1] \leq \text{negl}(\lambda)$.

D. Blakely's Secret Sharing and Hadamard Matrix

The basic idea of Blakely et al.'s work [27] is that, in a (t, n) secret sharing protocol, each of the n parties owns a t-dimensional hyperplane different from others'. The secret that the parties need to recover is a point in the t-dimensional space. When t or more parties come together to recover the secret, they can get the secret by solving a system of equations. When there are fewer than t parties, they cannot get any information about the secret.

Definition 6 ((t, n) Blakely's Secret Sharing)**.** *A* (t, n) *Blakely's secret sharing has two phases which is sharing phase and reconstruction phase, defined as follows:*

- ***Share****: In order to share a secret* a_0, *a dealer* $\mathcal{D}$ *selects* $t-1$ *random values* a_i, *where* $i \in \{1, \cdots, t-1\}$. *Next,* $\mathcal{D}$ *generates a* $n \times t$ *matrix* M *whose any two rows are independent and* (i, j)*-th element is defined as* b_{ij}. *Then, the dealer broadcasts* M. $\mathcal{D}$ *generates the shares* $s_i = a_0 b_{i,1} + a_1 b_{i,2} + ... + a_{t-1} b_{i,t}$, *where* $i \in \{1, \cdots, n\}$.
- ***Reconstruct****: The secret can be recovered if any* t *parties holding the secret share work together. Suppose the secret share vector is* $\overline{s} = [s_1, s_2, \cdots, s_t]$, *and the corresponding rows in* M *forms a* $t \times t$ *matrix* M_S. *Then, the vector* $\overline{a} = [a_0, a_1, \cdots, a_{t-1}]$ *can be reconstructed as* $\overline{a}^T = M_S^{-1} \cdot \overline{s}^T$. *Note that only the first element* a_0 *which is the secret need to be recovered.*

When the Blakely's secret sharing is initialized by a special matrix, i.e., the Hadamard matrix, it makes the secret recovery process requires very little computation. Define matrix multiplication for matrices $A = (a_{i,j})_{m \times p}$ and $B = (b_{i,j})_{p \times n}$ as $A \times B = \sum_{k=1}^{p} a_{ik} b_{kj}$. Let the Hadamard matrix of order q be $H = (b_{i,j})_{q \times q}$. Any two rows in H are perpendicular, and H is made up of the elements 1 and -1. The Hadamard matrix satisfies $H \times H^T = n \cdot I_n$, where $H^T = (b_{j,i})_{q \times q}$ represents the matrix H's transpose and I_n represents the $n \times n$ identity matrix. The inverse matrix can be obtained by dividing the Hadamard's transpose H^T by n.

III. Aggregated Anonymous Key Issue Protocol

The anonymous key issue (AKI) protocol was proposed in [17], which provides a way for users and attribute authorities to perform trusted secret computation through interaction. The AKI protocol is one of the important building block for multi-attribute authority signature protocol.

In the attribute key generation algorithm of the previous scheme, the user needs to execute $N-1$ AKI protocol with N attribute authorities respectively, which leads to too much interaction. In order to reduce the interaction, we propose an aggregated anonymous key issue (AAKI) protocol. It reduces the number of AKI executions between the user and each attribute authority from $N-1$ to 1.

In previous AKI protocols, each time the user and server transmitted a secret parameter, the proof of knowledge (PoK) protocol need to be executed once. It means that the original PoK protocol cannot be applied directly to our new AAKI protocol when parameters are transferred as the form of aggregation. Therefore, in this section, we first construct a non-interactive zero-knowledge proof aggregate exponentiation (NI-ZKPoKAE) protocol. Compared with previous PoK protocols, our protocol has the advantages of zero knowledge, non-interaction and fewer parameters. Then, we present the construction of our AAKI protocol with NI-ZKPoKAE protocol as a component.

A. NI-ZKPoKAE Protocol

Algorithm 1 $NI-ZKPoKAE$

$\underline{Params}$:
$\mathbb{G} \xleftarrow{\$} GGen(\lambda), (g_0, g_1, \ldots, g_n) \in \mathbb{G}^{n+1}, B > 2^{2\lambda}|\mathbb{G}|$.
$\underline{Prove(w = \prod_{i=1}^{n} u_i^{x_i}, \boldsymbol{u} \in \mathbb{G}^n, \boldsymbol{x} \in \mathbb{Z}^n)}$:
$k_1, \ldots, k_n, \theta_x, \theta_k \xleftarrow{\$} [-B, B]$,
$z = g_0^{\theta_x} \prod_{i=1}^{n} g_i^{x_i}, A_g = g_0^{\theta_k} \prod_{i=1}^{n} g_i^{k_i}, A_u = \prod_{i=1}^{n} u_i^{k_i}$,
$l \leftarrow H_{\text{Prime}}(\boldsymbol{u}, w, z, A_g, A_u), c \leftarrow H(l)$.
For $i = 1, \ldots, n$, let
$q_i \leftarrow \lfloor (k_i + c \cdot x_i)/l \rfloor, r_i \leftarrow (k_i + c \cdot x_i) \mod l$.
Let $q_0 \leftarrow \lfloor (\theta_k + c \cdot \theta_x)/l \rfloor, r_0 \leftarrow (\theta_k + c \cdot \theta_x) \mod l$.
Let $Q_1 = \prod_{i=0}^{n} g_i^{q_i}, Q_2 = \prod_{i=1}^{n} u_i^{q_i}$.
Set $\boldsymbol{r} \leftarrow (r_0, r_1, \ldots, r_n) \in [l]^{n+1}$.
Let $\pi = (l, z, Q_1, Q_2, \boldsymbol{r})$.
$\underline{Verify(w \in \mathbb{G}, \boldsymbol{u} \in \mathbb{G}^n, \pi)}$:
Parse $(l, z, Q_1, Q_2, \boldsymbol{r}) \leftarrow \pi, c \leftarrow H(l)$,
$A_g \leftarrow Q_1^l \cdot \prod_{i=0}^{n} g_i^{r_i} \cdot z^{-c}, A_u \leftarrow Q_2^l \cdot \prod_{i=1}^{n} u_i^{r_i} \cdot w^{-c}$.
Check: $\boldsymbol{r} \in [l]^{n+1}, l = H_{\text{Prime}}(\boldsymbol{u}, w, z, A_g, A_u)$.

Algorithm 1 shows our NI-ZKPoKAE protocol, which satisfies zero knowledge and non-interaction.

Theorem 1. *Protocol* $NI-ZKPoKAE$ *is a statistical zero-knowledge argument of knowledge for the relation* $\{(\boldsymbol{u} \in \mathbb{G}^n, \boldsymbol{x} \in \mathbb{Z}^n) : w = \prod_{i=1}^{n} u_i^{x_i} \in \mathbb{G}\}$ *in random oracle model.*

The proof of Theorem 1 is in Appendix A available in the online supplemental material.

B. AAKI Protocol

In order for user u to get the attribute key, we construct an aggregated anonymous key issue protocol. User u needs to execute the AAKI protocol with each attribute authority k in N attribute authorities. $(\mathbb{G}, g, q)$ is a multiplicative group and $\{g_i\}_{i \in I_k}, h$ are generated by the group generator g, where set $I_k = \{1, \cdots, N\} \backslash \{k\}$. The user takes $u \in \mathbb{Z}_p$ as private input and the attribute authority k takes $\{\alpha_i, \beta_i, \gamma_i\}_{i \in I_k}$ as private input to jointly compute $h^{\sum_{i \in I_k} \alpha_i} \prod_{i \in I_k} g_i^{\gamma_i/(\beta_i + u)}$. This protocol ensures that only the user knows the secret calculated value, while all other information remains hidden.

User u		Attribute Authority k
$\{\rho_i \in_R \mathbb{Z}_p\}_{i\in I_k}$	$\overset{2PC}{\leftrightarrow}$	$\forall i \in \{1,\cdots,N\}\backslash\{k\}, x_i := (\beta_i + u)\rho_i, \tau \in_R \mathbb{Z}_p$
$\rho' \in_R \mathbb{Z}_p$	$\xleftarrow[X_1,X_2,PoK(\{x_i\}_{i\in I_k},\{\gamma_i\}_{i\in I_k},\tau)]{\{X_{1,i}\}_{i\in I_k},PoK(\{\alpha_i\}_{i\in I_k},\tau)}$	$\{X_{1,i} := g_i^{\gamma_i\cdot\tau/x_i}\}_{i\in I_k}$, $X_1 := \prod_{i\in I_k} X_{1,i}$, $X_2 = h^{\sum_{i\in I_k}\alpha_i\cdot\tau}$
$Y := (\prod_{i\in I_k} X_{1,i}^{\rho_i} \cdot X_2)^{\rho'}$	$\xrightarrow{Y,PoK(\rho')}$	
$D := Z^{1/\rho'}$	$\xleftarrow{Z,PoK(\tau)}$	$Z := Y^{1/\tau}$

Fig. 2. Aggregated Anonymous Key Issue Protocol

Figure 2 shows our AAKI protocol. The 2PC protocol represents a secure computing protocol in which a user inputs $u, \{\rho_i\}_{i\in I_k}$ and an attribute authority inputs $\{\beta_i\}_{i\in I_k}$. The two parties secretly compute $\{x_i := (\beta_i + u)\rho_i\}_{i\in I_k}$, which are obtained by the attribute authority. The PoK protocol in each interaction represents the proof of knowledge about the secret values in the computation. Call our NI-ZKPoKAE protocol for these PoK protocols to ensure that the aggregation exponentiation is correct. $PoK(\{\alpha_i\}_{i\in I_k}, \tau)$ executes the protocol twice as $NI-ZKPoKAE(X_2, h^{\tau}, \{\alpha_i\}_{i\in I_k})$ and $NI-ZKPoKAE(X_2, h^{\sum_{i\in I_k}\alpha_i}, \tau)$. Although in the AAKI protocol, $\{X_{1,i}\}_{i\in I_k}$ are not sent to the user aggregatively, their proof of knowledge $PoK(\{x_i\}_{i\in I_k}, \{\gamma_i\}_{i\in I_k}, \tau)$ can still be completed by running the NI-ZKPoKAE protocol three times in the same way. Compared with performing a knowledge proof once for each $X_{1,i}$, it reduces the number of executing the NI-ZKPoKAE protocol and the number of data transfers required to implement $PoK(\{x_i\}_{i\in I_k}, \{\gamma_i\}_{i\in I_k}, \tau)$ from $O(n)$ to $O(1)$. Respectively, $PoK(\rho')$ and $PoK(\tau)$ run the NI-ZKPoKAE protocol once at $n = 1$.

Theorem 2. *Based on Diffie-Hellman assumption and the assumption that the general 2-party computation protocol and knowledge proofs are secure, the proposed AAKI protocol for computing* $h^{\sum_{i\in I_k}\alpha_i}\prod_{i\in I_k} g_i^{\gamma_i/(\beta_i+u)}$ *is a secure 2PC protocol.*

The proof of Theorem 2 is in Appendix B available in the online supplemental material.

C. Performance

To evaluate the performance of our proposed NI-ZKPoKAE protocol and AAKI protocol, we compare our protocols with Boneh's protocol [26] and Chow's protocol [17]. Through theoretical analysis and experimental simulation, the storage, communication, and computation complexity of these protocols are presented.

1) Theoretical Analysis: We contrast the NI-ZKPoKAE protocol with the non-interactive zero-knowledge exponentiation proof protocol proposed by Boneh et al. [26] in 2019. Boneh's protocol [26] and our NI-ZKPoKAE protocol both are non-interactive and zero-knowledge. The difference is that Boneh's protocol [26] is designed for the exponentiation of a single element, while ours is designed for the aggregating exponentiation of multiple elements. Boneh's protocol [26] is a special case of our protocol when $n = 1$.

In a non-interactive knowledge proof protocol, the computation overhead of the prover and the verifier comes from the prove algorithm and the verify algorithm, respectively, and only the prover generates the communication overhead. We show these comparisons between our NI-ZKPoKAE protocol and Boneh's protocol [26] in Table I. $|G|$ represents the size of the group element in the group $\mathbb{G}$ and Z represents the element size in the field $\mathbb{Z}$, n represents the total number of exponentiations the protocol needs to prove, E stands for exponential operation in the group $\mathbb{G}$, H stands for hash operation in the group $\mathbb{G}$, and M stands for multiplication operation in the group $\mathbb{G}$.

TABLE I
COMPARISON WITH RESPECT TO COMPUTATION AND COMMUNICATION OF NON-INTERACTIVE ZERO-KNOWLEDGE PROOF PROTOCOL

	Protocol	Boneh's protocol [26]	Our NI-ZKPoKAE
Comp.	*Prove*	$8nE + 3nM + 2nH$	$(5n-2)M + 2H + (5n+3)E$
	Verify	$7nE + 5nM + 2nH$	$(2n+3)M + 2H + (2n+5)E$
Comm.	Prover	$3n\|G\| + 3n\|Z\|$	$3\|G\| + (n+1)\|Z\|$

As shown in Table I, in terms of computation overhead, our aggregated protocol requires a constant number of hash operations per transfer. In both prove algorithm and verify algorithm, our NI-ZKPoKAE protocol needs to perform much less than the protocol [26] for the computationally expensive exponential operations in the group. In terms of communication overhead, the protocol [26] needs to transfer the number of group $\mathbb{G}$ elements related to n. Our protocol is better than protocol [26] to aggregate the elements of the group $\mathbb{G}$ into constant elements for communication. Meanwhile, the number of group $\mathbb{Z}$ element transfers in our NI-ZKPoKAE protocol is $n+1$, which is less than the $3n$ in protocol [26]. Overall, our NI-ZKPoKAE protocol outperforms the protocol [26] in both computation and communication.

We compare our AAKI protocol with Chow et al. 's [17] anonymous key issuing protocol, which is used by almost all existing multi-authority attribute-based cryptosystems. To facilitate the presentation of the efficiency of the protocols, we show in Table II the amount of computation and communication in addition to the 2PC protocol in the first step of both protocols and the PoK protocol used in the interaction. This is due to the following considerations: the two anonymous key issuing protocols can execute the same 2PC protocol to complete the first step, and the same PoK protocol can be implemented in the interaction, which makes the two protocols perform the same in the two processes respectively.

According to Table II, in terms of computation, after ex-

TABLE II
COMPARISON WITH RESPECT TO COMPUTATION AND COMMUNICATION OF ANONYMOUS KEY ISSUING PROTOCOL

	Protocol	Chow's Protocol [17]	Our AAKI Protocol
comp.	User	$3(N-1)E+NM$	$(N+1)E+NM$
	AA	$3(N-1)E$	$NE+(N-2)M$
comm.	User	$(N-1)\|G\|$	$\|G\|$
	AA	$3(N-1)\|G\|$	$(N+2)\|G\|$

cluding PoK protocol and 2PC protocol, no matter the user or authorities, the exponential operation of protocol [17] is nearly two times more than our AAKI protocol. In terms of communication, in our AAKI protocol, the user transmission amount is only a single group $\mathbb{G}$element, and the authority transmission amount is $N+2$ group $\mathbb{G}$ elements. This is due to the fact that our AAKI protocol aggregates all elements in the group $\mathbb{G}$ into constant elements for transmission. However, the communication overhead in protocol [17] is all related to N, and the transmission amount of authority is about 3 times of our AAKI protocol. In general, our AAKI protocol is superior to protocol [17] in terms of both user and authority computation and communication.

2) Experimental Simulation: The experimental simulations of NI-ZKPoKAE protocol and AAKI protocol are in Appendix C available in the online supplemental material.

IV. EFMA-ABS SCHEME

In this section, we present the system model, algorithm construction, security proof and performance analysis of our efficient and flexible multi-authority attribute signature (EFMA-ABS) scheme.

A. EFMA-ABS System Model

The EFMA-ABS scheme contains two types of entities: some attribute authorities $\{A_k\}_{k\in\{1,2,\cdots,N\}}$ and user u. $\mathcal{AA}$ denotes the set of all attribute authorities. Let $\mathcal{U}$ be the universe attribute set and each attribute authority $A_{k\in\{1,\cdots,N\}} \in \mathcal{AA}$ owns the attributes in universe $\mathcal{U}_k$. The attributes in different attribute authorities are mutually disjoint. $\mathcal{GID}$ stands for the set of all users' global identities. The user u owns attributes from these attribute authorities and can obtain the corresponding attribute key from them. $\mathcal{NID}$ is denoted as the node identities of access tree.

The EFMA-ABS scheme consists of four probabilistic polynomial time algorithms (**Setup**, **AkeyGen**, **Sign**, **Verify**).

- $(PP, MSK) \leftarrow$ **Setup**(λ, N): Given the system security parameter λ, the number of attribute authorities N, the setup algorithm outputs system public parameter PP and all attribute authority master secret key MSK.
- $SK \leftarrow$ **AKeyGen**$(PP, MSK, \omega_u, \Upsilon)$: Given the system public parameter PP, the attribute authority master secret key MSK, the attribute set ω_u owned by user u and access tree structure Υ, the probabilistic attribute key generation algorithm outputs signature secret key SK. Here we have $\omega_u = \bigcup_{k\in\{1,\cdots,N\}} \omega_{k,u}$, where $\omega_{k,u}$ is the attribute set owned by user u and distributed by the attribute authority A_k, and any two sets do not intersect each other.
- $\sigma \leftarrow$ **Sign**$(SK, \omega_u, m, \overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}})$: A signer with the signature secret key SK owns a attribute set ω_u to generate a signature on message m with a specific access tree predicate $\overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}}$, where C_x is the candidate attribute set and d_x is the threshold value for node x in the predicate. The signature algorithm outputs the signature σ of message m.
- $1/0 \leftarrow$ **Verify**$(PP, \sigma, m, \overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}})$: Using the signature σ, the message m, the access structure $\overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}}$ and the system parameter PP as input, the verify algorithm outputs the signature verification result $1/0$. If the signature is valid, the algorithm outputs 1; otherwise, it outputs 0.

B. EFMA-ABS Construction

The pseudorandom function (PRF) is applied in our protocol construction. A pseudorandom function family is a collection of functions which are computational indistinguishable from the really random functions. Our AAKI protocol can be used to implement the AKI protocol in the attribute signature scheme we will construct below to reduce the transmission amount. We use Blakely secret sharing to construct our scheme, where the Hadamard matrix is used to reduce the computing cost of our EFMA-ABS scheme. We give our new multi-authority attribute-based signature scheme as follow.

- **Setup**(λ, N):
  1) The algorithm input security parameter λ and a public random string $\zeta \in^{poly(\lambda)}$, the N attribute authorities generate the same admissible asymmetric bilinear group parameters respectively, which is denoted by $e = (p, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}(\cdot,\cdot), \phi(\cdot))$. Among them, the bilinear map is $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are three multiplicative cyclic groups of prime order p. g_1 and g_2 are the generators of groups $\mathbb{G}_1$ and $\mathbb{G}_2$.
  2) The N attribute authorities generate from the random string ζ. Define some collision resistant hash function as follow. $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$ maps user's global identity $gid \in \mathcal{GID}$ to an element in $\mathbb{Z}_p$, which is denoted by $u = H(gid)$. $H' : \{0,1\}^* \rightarrow \mathbb{G}_2$ maps the message to be signed to an element in $\mathbb{G}_2$. $H_k : \{0,1\}^* \rightarrow \mathbb{G}_2$ maps each attribute to a unique element in $\mathbb{G}_2$, where $k \in \{1,\cdots,N\}$. Deifine $\psi(\cdot)$ is a mapping that maps the node uniformly to an unique attribute-based on its position in the access tree. The image space of $\psi(\cdot)$ is disjoint with the attribute space $\mathcal{U}$ that the user can own.
  3) Select admissible $h \in_R \mathbb{G}_1$, $G_2 \in_R \mathbb{G}_2$ and a value n among the attribute authorities. Then, for each attribute authority A_k, define a value n_k. Select $v_k \in_R \mathbb{Z}_p$ as A_k's master key msk_k, compute $Y_k = g_1^{v_k}$, and obtain its public key $Z_k = \hat{e}(Y_k, G_2)$.
  4) A two-party key exchange protocol is carried out between every two attribute authorities. The attribute authority A_k secretly shares a pseudo-random seed

$s_{kj} \in \mathbb{Z}_p$ with another attribute authority A_j. Specially, define $s_{kj} = s_{jk}$. A_k shares a pseudo-random seed $x_k \in_R \mathbb{Z}_p$ and sends $y_k = G_2^{x_k}$ to all the other authorities. For user u, the pseudo-random function between authority k and j is defined as

$$\text{PRF}_{kj}(u) = G_2^{x_k x_j/(s_{kj}+u)}, u \in \mathbb{Z}_p$$

5) The system public parameter is

$$PP = (\{y_k, Y_k, Z_k, H_k, n_k\}_{k\in\{1,\cdots,N\}}, H, H', \psi, e, h, G_2, n),$$

and the master secret key is

$$MSK = (\{x_k, v_k, \{s_{kj}\}_{j\in\{1,\cdots,N\}\setminus\{k\}}\}_{k\in\{1,\cdots,N\}}).$$

- **AKeyGen**$(PP, MSK, \omega_u, \Upsilon)$: To obtain the attribute private key of attribute set $\omega_u = \bigcup_{k\in\{1,\cdots,N\}} \omega_{k,u}$ on access structure Υ, where $\Upsilon = \Upsilon_{ex} \bigcup_{k\in\{1,\cdots,N\}} \Upsilon_k$, the user u sets a injection between his own attribute set and leaf node set of Υ. The set of all nodes in Υ is defined as $NID \subseteq \mathcal{NID}$. When the node belongs to the internal access tree Υ_k, the number of children of node x is $n_x = n_k$. When it belongs to the external access tree Υ_{ex}, $n_x = n$. Then, the user u sends the key generation request consisting of the access structure $\Upsilon_{ex} \bigcup \Upsilon_k$ to each attribute authority A_k. The interaction between the attribute authorities and the user is as follows.
  1) For each $A_{j\in\{1,\cdots,N\}\setminus\{k\}}$, the user u independently runs the anonymous key issuing protocol with each authority by using the tuple $(g_j, h, \alpha_j, \beta_j, \gamma_j) = (y_j^{x_k}, G_2, \delta_{kj}R_{kj}, s_{kj}, \delta_{kj})$, where R_{kj} is selected randomly from $\mathbb{Z}_p$ by A_k, $\delta_{kj} = 1$ if $k > j$ and $\delta_{kj} = -1$ otherwise. Finally, the user gets $D_{kj} = G_2^{R_{kj}}\text{PRF}_{kj}(u)$, if $k > j$. Otherwise, the user obtains $D_{kj} = G_2^{R_{kj}}/\text{PRF}_{kj}(u)$.
  2) Build the external access tree Υ_{ex} between multiple attribute authorities A_k, where $k \in \{1, \cdots, N\}$. N attribute authorities as the leaves of access tree Υ_{ex} randomly select and share all Hadamard matrix $M_x = (b_{i,j})_{n\times n}$ for all non-leaf nodes x of Υ_{ex}. Each $r_{k,x}$ randomly selected from $\mathbb{Z}_p$, authority A_k computes $V_k = v_k - \sum_{j\in\{1,\cdots,N\}\setminus\{k\}} R_{kj}$. For each node x of Υ_{ex} except the root node R and the leaf node A_k, A_k sets each external node's secret value is $V_k/\prod_{j\in\Omega_x} M_{p(j)}^{-1}(1, l(j))$, where Ω_x is a collection that contains all nodes on the path from the node x to the root node. Hence, A_k has the secret share

$$X_k = V_k/\prod_{x\in\Omega_k} M_{p(x)}^{-1}(1, l(x)).$$

  3) Attribute authority A_k build the internal access tree Υ_k in a top-down manner. First, A_k randomly selects $n_k - 1$ values $(a_{x,1}, a_{x,2}, \cdots, a_{x,n_k-1})$ and a Hadamard matrix $M_x = (b_{l,m})_{n_k\times n_k}$ for each non-leaf node x of Υ_k. Second, A_k sets the root node R_k's secret value $a_{R_k,0} = X_k$. For the other nodes x of Υ_k except the root node R_k, use the information of their parent nodes to define their secret values $a_{x,0} = a_{p(x),0}\cdot b_{l(x),1}^{p(x)} + a_{p(x),1}\cdot b_{l(x),2}^{p(x)} + \cdots + a_{p(x),n_k-1}\cdot b_{l(x),n_k}^{p(x)}$. Third, A_k generates the leaf node key for all leaf nodes x of Υ. Each $r_{k,x}$ randomly selected from $\mathbb{Z}_p$, A_k computes the default attribute keys of all leaf nodes in Υ as

$$d_{k,x0} = G_2^{\sum_{s=0}^{n_k-1} a_{p(x),s}^k \cdot b_{l(x),s+1}^{p(x)}} H_k(\psi(x))^{r_{k,x}},$$
$$d_{k,x1} = g_1^{r_{k,x}}.$$

where $\psi(\cdot)$ maps the node (its position in the tree) to an attribute value. We define this default attribute set generated by authority A_k consisting of all the leaf node keys as F_k.

  4) For all the attribute $i \in \omega_{k,u}$, A_k sets $a'_{x,0} = 0$ and randomly selects $n_k - 1$ values $(a'_{x,1}, a'_{x,2}, \cdots, a'_{x,n_k-1})$ for each first level parent node. Then, A_k randomly selects $r_{k,i}$ from $\mathbb{Z}_p$, and computes

$$d_{k,i0} = G_2^{\sum_{s=0}^{n_k-1} (a_{p(i),s}^k + a'_{x,s})\cdot b_{l(i),s+1}^{p(i)}} H_k(i)^{r_{k,i}},$$
$$d_{k,i1} = g_1^{r_{k,i}}.$$

Authority A_k outputs the user's secret key

$$SK_k = (\{D_{kl}\}_{l\in\{1,\cdots,N\}\setminus\{k\}},$$
$$\{d_{k,i0}, d_{k,i1}\}_{i\in\omega_{k,u}},$$
$$\{d_{k,x0}, d_{k,x1}\}_{x\in F_k}).$$

The user reconstructs the leaf node keys of Υ_{ex} as follows:

$$d_{x0} = \prod_{k\in\omega_x} d_{k,x0}, d_{x1} = \prod_{k\in\omega_x} d_{k,x1},$$

where ω_x is the collection of attribute authorities contained in a subtree Υ_x.
Finally, the user's signature key is

$$SK = \{SK_k\}_{k\in\{1,\cdots,N\}}.$$

- **Sign**$(SK, \omega_u, m, \overline{\Upsilon}_{\{(d_x, C_x)\}_{x\in NID}})$: A signer owns a attribute set ω_u to generate a signature on message m with a specific access tree predicate $\overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}}$, where C_x is the candidate attribute set and d_x is the threshold value for node x in the predicate.
  1) Select the attribute set. The signer selects a subset $\omega'_u \subseteq \omega_u$ that satisfies $\overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}}(\omega'_u) = 1$.
  2) Select the complement set. For leaf nodes in the access structure that do not have corresponding attributes in set ω'_u, select attributes from the default attribute set $\cup_{k\in\{1,\cdots,N\}} F_k$ to form the complement set S_u, so that each leaf node in the access structure has one corresponding attribute. It makes each node x in $\overline{\Upsilon}$ satisfy $|S_{u,x} \cup \omega'_{u,x}| = d_x$.
  3) Sign the message with the access tree. The signer chooses a random value $s_{k,i}$ for each $i \in w'_u$ and a random value $s_{k,x}$ for each $x \in S_u$. Let Ω_x be the set of all nodes on the path from node x to the root node. First compute $D_u = \prod_{(k,j)\in\{1,\cdots,N\}\times\{1,\cdots,N\}\setminus\{k\}} D_{k,j}$ and $\Phi = H'(m||\overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}})$. Then, randomly choose $s \leftarrow_R \mathbb{Z}_p$ and computes

$$\sigma_0 = D_u \prod_{1\le k\le N} [\prod_{i\in\omega'_u} (d_{k,i0}^{\prod_{x\in\Omega_i} M_{k,x}^{-1}(1,l(x))} \cdot H_k(i)^{s_{k,i}}) \cdot \prod_{x\in S_u} (d_{k,x0}^{\prod_{j\in\Omega_x} M_{k,j}^{-1}(1,l(j))} \cdot H_k(\psi(x))^{s_{k,x}})] \cdot \Phi^s$$

$$\sigma_{k,i} = d_{k,i1}^{\prod_{x\in\Omega_i} M_{k,i}^{-1}(1,l(x))} g_1^{s_{k,i}}, i \in \omega'_u$$

$$\sigma_{k,x} = d_{k,x1}^{\prod_{j\in\Omega_x} M_{k,x}^{-1}(1,l(j))} g_1^{s_{k,x}}, x \in S_u$$

$$\sigma_s = g_1^s.$$

Finally, the signer outputs the signature

$$\sigma = (\sigma_0, \sigma_s, \{\{\sigma_{k,i}\}_{i\in\omega'_u}, \{\sigma_{k,x}\}_{x\in S_u}\}_{k\in\{1,\cdots,N\}}).$$

- **Verify**$(PP, \sigma, m, \overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}})$: Using the signature σ, the message m and the access structure $\overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}}$ as input. Compute $\Phi = H'(m\|\overline{\Upsilon}_{\{(d_x,C_x)\}_{x\in NID}})$. Verify the following equation

$$\hat{e}(g_1,\sigma_0) \stackrel{?}{=} \hat{e}(\sigma_s,\Phi) \cdot \prod_{1\le k\le N} [Z_k \prod_{i\in\omega'_u} \hat{e}(\sigma_{k,i}, H_k(i)) \cdot \prod_{x\in S_u} \hat{e}(\sigma_{k,x}, H_k(\psi(x)))]$$

If the equation outputs 1, the signature is accepted; otherwise 0 and the signature is rejected.

Remark 1. *Note that the orders of the Hadamard matrices are all multiples of 2 or 4. When the number of a node's child nodes in the access tree is less than the pre-defined order of the Hadamard matrix, that is n or n_k, our scheme is equivalent to adding dummy nodes to the node by the attribute authority. When the dummy node is an internal node of A_k, it is generated by the attribute authority A_k. When the dummy node is an external node, it can be generated by randomly appointing an attribute authority through negotiation of multiple attribute authorities. The key generation method of the newly added dummy node is the same as that of the original node, which is defined in the* ***AKeyGen*** *algorithm.*

Remark 2. *The inverse matrix in our construction uses the simple computing method of Hadamard matrix inversion $H^{-1} = H^T/n$, which greatly reduces the computing cost.*

Theorem 3. *The proposed EFMA-ABS scheme is correct.*

Theorem 4. *The proposed EFMA-ABS scheme is existentially unforgeable under co-CDH assumption.*

The relevant security definitions and proofs in Appendix D available in the online supplemental material.

C. Performance

To illustrate the performance of our fine-grained multi-authority attribute signature scheme, we compare our scheme with ePass scheme [8], Li's ABS scheme [21], ODMA-ABS [16] and DABSAS scheme [22]. The ePass scheme [8] is a representative scheme of fine-grained attribute signature schemes. Li's ABS scheme [21] is a representative scheme of multi-authority attribute signature schemes. ODMA-ABS scheme [16] is an outsourced multi-authority attribute signature. DABSAS scheme [22] is a server-assisted distributed attribute signature. The storage and computational complexity comparisons of these five schemes are given through theoretical analysis and experimental simulation respectively.

1) Theoretical Analysis: We show the storage complexity and computational complexity comparison of the five schemes in Table III. $|G_1|$, $|G_2|$, $|G_T|$ represent the size of group elements in groups G_1, G_2, and G_T respectively, N represents the number of attribute authorities, w represents the number of user's attributes, l represents the number of attributes (or leaf nodes) contained in the access structure of all attribute authorities, τ represents the number of minimum internal nodes that all attribute authorities should satisfy the attribute tree, and d represents the sum of the threshold values of all attribute authorities in the threshold attribute structure. e stands for bilinear pairing operation, E stands for exponential operation in the group, H stands for hash operation in the group, and M stands for multiplication operation in the group.

We first focus on the theoretical analysis of storage cost. Since the ePass scheme is an attribute signature scheme for a single attribute authority, it has a constant number of system public parameters. Due to schemes [21], [16], [22] and our scheme are multi-attribute authorities signature schemes, the system parameters are related to the number of attribute authorities N. Since schemes [16], [22] and our scheme are constructed based on the co-CDH assumption and achieves higher security, the number of public parameters related to N is generated in the three groups. This makes schemes [16], [22] and our scheme more system parameters than the other two schemes based on CDH assumption. The master secret key is generated by the attribute authority and kept by it, or deleted directly after the attribute secret key is generated. The number of our master keys is the same as schemes [21], [16], [22] which are also multi-authority scenarios. In the ePass scheme and DABASAS scheme, the attribute secret keys used to generate the signature and the verification keys used to verify the signature are essentially the user's keys. The size of the user key generated by our scheme is the same as that of the ePass scheme, which is also a fine-grained attribute signature scheme. Schemes [16], [22] also have outsourcing keys that need to be given to the server, the size of which is related to w. The length of the signature in our scheme is the same as that in schemes [21], [16], [22], while the signature length of the ePass scheme is about twice that of our scheme. Schemes [16], [22] also have outsourcing signatures associated with l.

Then, we focus on the theoretical analysis of the computation cost. The above five attribute signature schemes are all based on amortized model, that is, after performing system initialization once to generate system parameters, the scheme can be run multiple times. In addition, user can apply to the attribute authority to generate an attribute key once, and the key can be used to generate signatures for different messages multiple times. Therefore, in the practical application of the attribute signature scheme, the computational efficiency of

TABLE III
COMPARISON WITH RESPECT TO KEY SIZE AND SIGNATURE SIZE AND THE COMPUTATION OF THE ALGORITHM

Scheme	ePass Scheme [8]	Li's ABS [21]	Our Scheme	DABSAS [16]	ODMA-ABS [22]
PP size	$3\|G_1\|+\|G_2\|$	$(N+1)\|G_1\|+N\|G_2\|$	$(N+2)\|G_1\|+N\|G_T\|$ $+(N+2)\|G_2\|$	$N\|G_1\|+N\|G_T\|$ $+(N+1)\|G_2\|$	$N\|G_1\|+(N+1)\|G_2\|$ $+N\|G_T\|$
MK size	$\|Z_p\|$	$(N^2+N)\|Z_p\|$	$(N^2+N)\|Z_p\|$	$(N^2+N)\|Z_p\|$	$(N^2+N)\|Z_p\|$
OK size	–	–	–	$w\|G_1\|+w\|G_2\|$	$w\|G_1\|+w\|G_2\|$
SK size	$(2w+1)\|G_1\|$	$(2w+1)\|G_1\|$	$(l+w+1)\|G_1\|$ $+(l+w)\|G_2\|$	$(N-1)\|G_2\|$	$(N-1)\|G_2\|$
VK size	$2l\|G_1\|$	–	–	$(l+1)\|G_1\|+\|G_2\|$ $+\|G_T\|$	–
σ_{out} size	–	–	–	$(l+1)\|G_1\|+\|G_2\|$	$l\|G_1\|+\|G_2\|+2\|G_T\|$
σ size	$2(l+1)\|G_1\|$	$(l+2)\|G_1\|$	$(l+1)\|G_1\|+\|G_2\|$	$(l+1)\|G_1\|+\|G_2\|$	$(l+1)\|G_1\|+\|G_2\|$
$Sign_{out}$ Gen	–	–	–	$2(l+d)E+(l+1)H$ $+(l+2d-2)M$	$(2l+N+2d-4)M$ $+2(l+d)E+2lH$ $+(l+1)e$
$Sign$ Gen	$2(l+1)E+(l+1)H$ $+(3l+1)M$	$2(l+d+1)E+(l+1)H$ $+(l+2d-1)M$	$2(l+1)E+(l+1)H$ $+(3l+1)M$	$(N^2-N+2)M$ $+2E+H$	$(N^2-N+2)M$ $+e+2E+H$
$Verify_{out}$	–	–	–	$(l+1)e+lH$ $+(l+N)M$	–
$Verify$	$2(l+1)e+\tau E$ $+H+(l+2)M$	$(l+2)e+(l+1)H$ $+(l+1)M$	$(l+2)e+(l+1)H$ $+(l+N)M$	$e+(l+2)E$ $+(l+1)H$	$(l+2)e+lH$ $+(l+N)M$

the signature generation and verification phase is worth more attention. The main computation of the user comes from the signature generation phase, while the computation of the verifier comes from the verification signature phase. The computation cost in the signature phase mainly comes from the exponential operation in the group. The calculation amount of our scheme and ePass scheme in the signature phase is the same, which is only related to the number l of attributes contained in the access structure, and lower than that of Li's ABS scheme. The signature phase of Li's scheme is also affected by threshold value d. Schemes [16] and [22] outsource most of the computation of signature generation to the server, so the user's computation amount in the signature phase is small. The verification cost mainly comes from the bilinear pairing operation. The calculation amount of our scheme is almost the same as that of schemes [21], [16], and is about half of the ePass scheme. The ePass scheme also performs group exponential calculation in the verification phase, which is not in schemes [21], [16] and our scheme. Schemes [22] outsources part of the verification calculation to the server, and the user performs a small number of calculations in the final verification phase. To this end, user needs to pre-process the original data before outsourcing, which brings new computation overload.

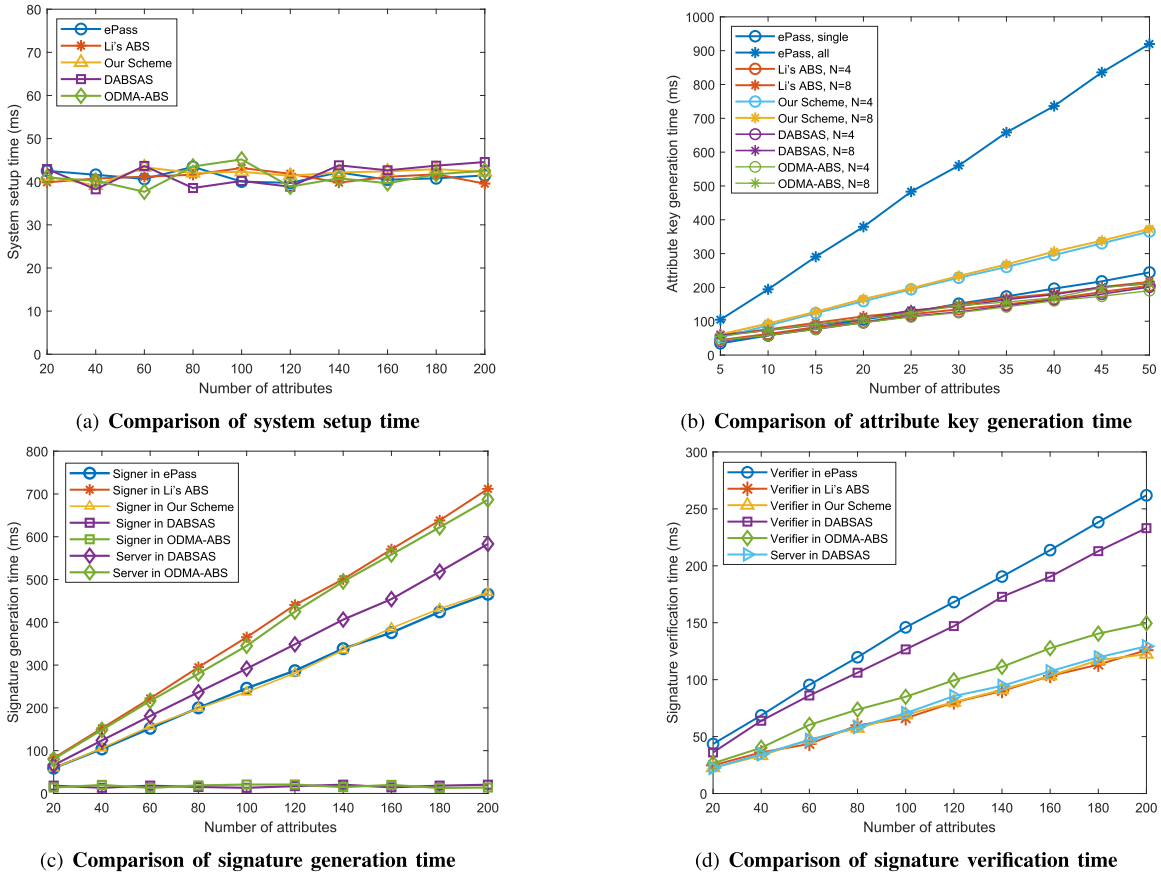
2) Experimental Simulation: We simulate the above five attribute signature schemes to compare their performance. Notice that there are four entities involved in the experiment, the attribute authorities, the signer, the verifier and the server. The attribute authorities are simulated on a Linux machine equipped with an Intel Core i78550U CPU 2.00 GHz processor and 16GB RAM. The signer and verifier are simulated on a virtual machine with Intel (R) Core (TM) i7- 8550U CPU processor running at 1.80 GHz and 512 MB RAM. The server is simulated on the HPC Cloud Platform of Shandong University with Intel(R) Xeon(R) Gold 6132 CPU at 2.60 GHz and 20.0G RAM. When implementing each algorithm in above five schemes, C programming language with the Pairing Based Cryptography Library (PBC) and GNU Multiple Precision Arithmetic Library (GMP) are used. We set the basic field size as 512 bits, and the element size in Z_p is 160 bits. Type-A pairing constructed on the elliptic curve $y^2 = x^3 + x$ over the field F_q, where q is a large prime. The hash functions in signature are instantiated with the full-domain hash proposed by Bellare et al [32]. In the experiment, we assume that the number of attributes, access structure and threshold value owned by each attribute authority are the same.

In Figure 3 (a), we simulated the computational overhead of the setup phase, taking the number of attributes contained in the access structure as the independent variable. It can be seen that the computational overhead in the setup phase of the five schemes is constant, and the amount of calculation is very small and almost the same, about 41ms.

Figure 3 (b) shows the efficiency of attribute key generation phase, taking the number w_k of attributes that the user applies to a single attribute authority as the independent variable, and w_k varies from 5 to 50. Since the key generation phase involves interactions between attribute authorities, the computation overhead in this phase of the multi-authority attribute signature scheme is related to the number N of attribute authorities. We simulated the computation time required for a single attribute authority to generate an attribute key when the number of attribute authorities is $N = 4$ and $N = 8$ for schemes [21], [16], [22] and our scheme respectively. Since the ePass scheme is a single-authority attribute signature scheme, we have performed the following two simulations on it. One is to simulate and generate the same number of attribute keys as an attribute authority in the multi-authority attribute signature scheme. The second is the total number of attribute keys that a single attribute authority should generate when running a real scheme. In the second case, we assume $N = 4$. That is, in this scenario, the number of attributes that the ePass scheme needs to generate varies from 20 to 200.

The attribute key generation time of the five schemes all have a linear growth relationship with w_k. When $N = 4$

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3263497



(a) **Comparison of system setup time**

(b) **Comparison of attribute key generation time**

(c) **Comparison of signature generation time**

(d) **Comparison of signature verification time**

Fig. 3. Efficiency Comparison for Setup, KeyGen, Signing and Verify among ePass scheme, Li's scheme and our scheme

and $w_k = 50$, the ePass, Li's ABS, our scheme, DABSAS and ODMA-ABS scheme require 244.393ms, 205.066ms, 365.204ms, 201.198ms and 190.524ms respectively in this phase. The ePass scheme and our scheme are less efficient than schemes [21], [16], [22], because ours need to perform calculations related to l. When generating the key for each attribute, our scheme performs one more exponential operation in the group than ePass scheme, which results in slightly lower efficiency of our scheme than the ePass scheme. And because of our multi-authority setting, the key generation process also increases the time required to implement the AAKI protocol. But our key generation time is still within an acceptable range. When $N = 8$ and $w_k = 50$, schemes [21], [22], [16] and our scheme require 216.598ms, 373.001ms, 211.021ms and 212.519ms respectively. It can be seen that adding attribute authorities has little effect on running the AAKI protocol, and adding four attribute authorities only needs to increase the running time of about 10ms. The ePass scheme is a single authority attribute signature scheme, which requires 919.713ms to generate all 200 attribute keys. At the same time, each attribute authority in the multi-authority scheme only needs to generate 50 attribute keys, that is, our scheme needs 365.204ms and other three schemes take less time. Therefore, it can be seen that apportioning the attribute key generation process to multiple attribute authorities is a good choice to relieve the calculation pressure of attribute authority and improve the robustness of the system.

The user signature generation time is shown in Figure 3 (c). Without loss of generality, we assume that $N = 4$, the threshold d in the threshold structure is half of the number l of attributes contained in the access structure, and let l vary from 20 to 200. We can see that the signature generation time of the ePass, Li's ABS, our scheme and the outsourcing signature calculation of the DABSAS and ODMA-ABS scheme increase with the increase of l. When $l = 200$, the ePass scheme and our scheme take almost the same time, which are 465.571ms and 469.778ms, respectively. Because of the additional calculations associated with the threshold value d, Li's ABS scheme and outsourcing signature calculation of schemes [22], [16] respectively take 711.96 ms, 583.103ms and 686.645ms. In outsourcing schemes [16], [22], the user signature requires only a small amount of computation, about 20ms. In the signing phase, our scheme is about 34% faster than Li's ABS scheme.

Figure 3 (d) shows the time required by the verifier during the verification phase. The verification process of the five schemes all increase with the increase of l. The calculation amount of the ePass scheme is also affected by τ. Let l in our experiment vary from 20 to 200. When $l = 20$, $\tau = 8$, and for every 20 increase in l, τ increases by 2. From this we can see that when $l = 200$, the verification time of Li's ABS, our scheme, ODMA-ABS scheme and server in DABSAS scheme are almost the same, taking 124.882ms, 124.351ms, 129.295ms and 149.516ms respectively, while the

ePass scheme takes 261.903ms. The verification time of our scheme is about 52.5% faster than the ePass scheme. The computation of the verifier in DABSAS scheme comes from two phases, one is the generation of the outsourced verify key, the other is the signature verification phase. Therefore, the verifier calculation of DABSAS scheme is not small, which is 233.07ms.

In summary, it can be seen from the above experimental simulation that our scheme has the optimal efficiency in the setup phase, user's signature phase and verifier's verification phase. Although the attribute key generation phase run by the attribute authority is slightly slower than the other two schemes, the efficiency is still acceptable.

V. Attribute-based IoT Authentication System

There are five types of entities in attribute-based IoT authentication systems, namely, user who initiate authentication requests, the authentication verifier, and the attribute authority that issue attribute permissions to users. As shown in Figure 1, both the IoT devices and the data center need to act as both the authentication request initiator and the authentication verifier. Therefore, in our IoT system model, we need an authentication protocol that makes both of them computationally efficient.

There are five algorithms in the IoT authentication system, namely, **System Setup** algorithm, **User Grant** algorithm and **Authentication** algorithm. In the **System Setup** algorithm, system public parameters and master secret keys will be generated. When there are multiple attribute authorities, the **System Setup** algorithm may require interaction between them. In the **User Grant** algorithm, the user and the attribute authority obtain the attribute key through interaction. When a user needs to send data to other entities, the **Authentication** algorithm is used to send proof that the data satisfies a specific access tree to the receiver for verification to prove the reliability of the transmitted data, which is guaranteed by our signature technique. We design the following attribute-based authentication protocol in a multi-authority setting.

- **System Setup**. This setup algorithm is same as the **Setup** algorithm for our ABS scheme in section IV-B. The system public key contains $e = (p, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}(\cdot,\cdot), \phi(\cdot))$, and several hash functions H, H', $\{H_k\}_{k\in[1,N]}$ and $\psi(\cdot)$. Select $h \in_R \mathbb{G}_1$, $G_2 \in_R \mathbb{G}_2$ and the value n for multiple attribute authorities. Each attribute authority A_k defines the value n_k and randomly selects v_k to calculate $Y_k = g_1^{v_k}$. Obtain the public key $Z_k = \hat{e}(Y_k, G_2)$. The attribute authority A_k shares the random seed s_{kj} with each attribute authority A_j. A_k shares the random seed x_k and sends $y_k = G_2^{x_k}$ to other attribute authorities. For user u, define the pseudo-random function $\mathrm{PRF}_{kj}(u) = G_2^{x_k x_j/(s_{kj}+u)}$.
- **User Grant**. Suppose the user u has the property set ω_u. Run **AKeyGen** algorithm in our ABS scheme in section IV-B, and each attribute authority A_k generates user attribute private key $SK_k = (\{D_{kl}\}_{l\in[1,N]\setminus\{k\}}, \{d_{k,i0}, d_{k,i1}\}_{i\in\omega_{k,u}}, \{d_{k,x0}, d_{k,x1}\}_{x\in F_k})$.
- **Authentication**. Suppose one have a attribute private key for the property set ω_u. To prove that his identity satisfies the access tree $\overline{\Upsilon}$, the user performs the **Sign** algorithm in section IV-B. The user outputs the signature $\sigma = (\sigma_0, \sigma_s, \{\{\sigma_{k,i}\}_{i\in\omega'_u}, \{\sigma_{k,x}\}_{x\in S_u}\}_{k\in[1,N]})$, and sends the message and signature to the message receiver. The receiver decides whether to accept the data by performing the **Verify** algorithm in section IV-B.

The security of the attribute-based authentication protocol is guaranteed by the security of our multi-authority attribute signature scheme.

VI. Conclusion

In this work, we have attempted to construct a fine-grained multi-attribute authority IoT authentication system that is efficient for all the participants in this system, including multiple attribute authorities, signers, and verifiers. To reduce the transmission burden between multiple authorities, we have presented an aggregated anonymous key issue protocol. In order to aggregate transfer secret values in AAKI protocol, we have designed a non-interactive zero-knowledge proof aggregate exponentiation. To reduce the computational burden of signers and verifiers, we have used Blakley secret sharing to construct a fine-grained multi-authority attribute-based signature scheme. We have carried out theoretical analysis and comparative experiments to demonstrate the security and efficiency of our protocol. Based on our proposed multi-authority attribute signature scheme, we have presented an IoT authentication system.

Acknowledgment

This work is supported by the National Nature Science Foundation of China under Grant No.: 62072276 and 61772311. The scientific calculations in this paper have been done on the HPC Cloud Platform of Shandong University.

References

[1] "Alibaba cloud solutions for internet of things." https://www.alibabacloud.com/fr/solutions/IoT.

[2] Z. ThreatLabz, "Iot in the enterprise: Empty office edition." https://info.zscaler.com/resources-reports-threatlabz-iot-2021, 2021.

[3] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: Achieving attribute-privacy and collusion-resistance," *Iacr Cryptology Eprint Archive*, 2008.

[4] S. F. Shahandashti and R. S. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," *Springer, Berlin, Heidelberg*, 2009.

[5] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Acm Symposium on Information*, 2010.

[6] L. Jin and K. Kim, "Hidden attribute-based signatures without anonymity revocation," *Information Sciences*, vol. 180, no. 9, pp. 1681–1689, 2010.

[7] T. Okamoto and K. Takashima, "Efficient attribute-based signatures for non-monotone predicates in the standard model," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 409–421, 2014.

[8] J. S. A, D. C. A, B. Z. A, X. W. A, and I. Y. B, "epass: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the internet of things - sciencedirect," *Future Generation Computer Systems*, vol. 33, no. 2, pp. 11–18, 2014.

[9] K. Gu, W. Jia, G. Wang, and S. Wen, "Efficient and secure attribute-based signature for monotone predicates," *Acta Informatica*, vol. 54, no. 5, pp. 521–541, 2017.

[10] Y. Li, X. Chen, Y. Yin, J. Wan, and Z. Dong, "Sdabs: A flexible and efficient multi-authority hybrid attribute-based signature scheme in edge environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–15, 2020.

[11] R. Lu, L. Zhang, J. Ni, and Y. Fang, "5g vehicle-to-everything services: Gearing up for security and privacy," *Proceedings of the IEEE*, vol. PP, no. 99, pp. 1–17, 2019.

[12] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Springer Berlin Heidelberg*, pp. 376–392, 2010.

[13] D. Cao, B. Zhao, X. Wang, J. Su, and G. Ji, "Multi-authority attribute-based signature," in *Third International Conference on Intelligent Networking & Collaborative Systems*, 2012.

[14] Y. Chen, J. Chen, and G. Yang, "Provable secure multi-authority attribute based signatures," *Journal of Convergence Information Technology*, vol. 8, no. 2, pp. 545–553, 2013.

[15] T. Okamoto and K. Takashima, *Decentralized Attribute-Based Signatures*. Springer Berlin Heidelberg, 2013.

[16] J. Sun, Y. Su, J. Qin, J. Hu, and J. Ma, "Outsourced decentralized multi-authority attribute based signature and its application in iot," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 1195–1209, 2021.

[17] S. Chow, "New privacy-preserving architectures for identity-/attribute-based encryption," in *New York University*, 2010.

[18] M. Mamdouh, A. I. Awad, A. A. Khalaf, and H. F. Hamed, "Authentication and identity management of ioht devices: Achievements, challenges, and future directions," *Computers & Security*, vol. 111, p. 102491, 2021.

[19] F. A. Almalki and S. B. Othman, "Eppda: An efficient and privacy-preserving data aggregation scheme with authentication and authorization for iot-based healthcare applications," *Wirel. Commun. Mob. Comput.*, vol. 2021, pp. 5594159:1–5594159:18, 2021.

[20] Z. Zhang and S. Zhou, "A decentralized strongly secure attribute-based encryption and authentication scheme for distributed internet of mobile things," *Computer Networks*, vol. 201, p. 108553, 2021.

[21] J. Li, X. Chen, and X. Huang, "New attributecbased authentication and its application in anonymous cloud access service," *International Journal of Web & Grid Services*, vol. 11, no. 1, p. 125, 2015.

[22] J. Li, Y. Chen, J. Han, C. Liu, Y. Zhang, and H. Wang, "Decentralized attribute-based server-aid signature in the internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4573–4583, 2022.

[23] S. Chow, "Removing escrow from identity-based encryption: new security notions and key management techniques," *Public Key Cryptographypkc*, pp. 256–276, 2009.

[24] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009*, 2009.

[25] M. Hua, E. Dong, Z. Liu, and L. Zhang, "Privacy-preserving multi-authority ciphertext-policy attribute-based encryption with revocation," in *International Conference on Broadband & Wireless Computing*, 2017.

[26] D. Boneh, B. Bünz, and B. Fisch, *Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains*. Springer International Publishing, 2019.

[27] G. R. Blakley, "Safeguarding cryptographic keys," in *Afips*, 1979.

[28] A. Shamir, "How to share a secret," *Communications of the ACM*, 1979.

[29] Z. Xia, B. Yang, Y. Zhou, M. Zhang, and Y. Mu, *Improvement of Attribute-Based Encryption Using Blakley Secret Sharing*. Springer, Cham, 2020.

[30] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in Cryptology — EUROCRYPT 2003*, (Berlin, Heidelberg), pp. 416–432, Springer Berlin Heidelberg, 2003.

[31] B. Wesolowski, "Efficient verifiable delay functions." Cryptology ePrint Archive, Paper 2018/623, 2018. https://eprint.iacr.org/2018/623.

[32] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *CCS '93*, 1993.

Ye Su received her bachelor degree in School of Mathematical Sciences from the University of Jinan, China, in 2014. She received her Ph.D. degree in school of mathematics in Shandong University, Jinan, China, in 2022. She is currently a lecturer with School of Information Science and Engineering, Shandong Normal University. Her research interests include cryptography and cloud security.

Xi Zhang received the BS degree from the School of Mathematical Sciences, Hebei Normal University, P.R.China, in 2018. She is a visiting student advised by Professor Sherman Shen, university of waterloo in 2022. She is currently working toward the PhD degree in the School of Mathematics, Shandong University. Her research interests include searchable encryption and cloud computing.

Jing Qin is a professor in School of Mathematics, Shandong University, China. Her research interests include computational number theory, information security, design and analysis of security about cryptologic protocols. She received her B.S. degree from Information Engineering University, Zhenzhou, China in 1982 and Ph.D. degree from School of Mathematics in Shandong University, China in 2004. She has coauthored 2 books and has published about 30 professional research papers. Prof. Qin is a senior member of the Chinese Association for Cryptologic Research (CACR) as well as the China Computer Federation (CCF).

JIXINMA (Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics from Zhengzhou University, Zhengzhou, China, in 1982 and 1988, respectively, and the Ph.D. degree in computer sciences from the University of Greenwich, London, U.K., in 1994. He is currently a Reader in computer science with the School of Computing and Mathematical Sciences, University of Greenwich, U.K., and a Visiting Professor with Beijing Normal University, Auhui University, and Zhengzhou Light Industrial University, China. He has published more than 100 research articles in international journals and conferences. His main research areas include artificial intelligence and information systems, with special interests in temporal logic and information security.