Hindawi
Security and Communication Networks
Volume 2022, Article ID 9287026, 19 pages
https://doi.org/10.1155/2022/9287026

Research Article

SPCABS: Signature-Policy Comparable Attribute-Based Signatures

Hongying Chen,[1] **Zhenjie Huang**,[2,3] **Hui Huang**,[4] **and Yafeng Guo**[1]

[1] *School of Electronic and Information Engineering, Zhangzhou City University, Zhangzhou 363000, China*
[2] *Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, Zhangzhou 363000, Fujian, China*
[3] *School of Mathematics and Statistics, Minnan Normal University, Zhangzhou 363000, Fujian, China*
[4] *School of Computer Science, Minnan Normal University, Zhangzhou 363000, Fujian, China*

Correspondence should be addressed to Zhenjie Huang; zjhuang@mnnu.edu.cn

Received 20 November 2021; Accepted 23 March 2022; Published 18 June 2022

Academic Editor: Youwen Zhu

Attribute-based signature is an attractive cryptographic primitive and finds broad applications in many fields. Existing attribute-based signature schemes deal with attributes in the way of "with" or "without," and there is no attribute-based signature scheme that supports comparing attributes. Using the 0-encoding and 1-encoding, we propose an access structure algorithm and an attribute expansion algorithm, enabling the attribute-based signature scheme to effectively deal with the comparative attributes. Then, we propose a signature-policy comparable attribute-based signature scheme using the proposed expansion algorithms. The proposed scheme is existentially unforgeable under the computational Diffie–Hellman exponent (CDHE) assumption and achieves privacy in the sense of information theory. Theoretical analysis and simulation experiments show that our method is practical and has significant advantages in storage and computation overhead compared with the trivial way. Comparable attribute-based signature dramatically expands the application scenarios of attribute-based signature.

1. Introduction

Attribute-based signature is a very attractive cryptographic primitive [1]. ABS is divided into key-policy ABS (KP-ABS) and signature-policy ABS (SP-ABS). The former KP-ABS uses the access policy (structure) to generate the signing key, and the message can be signed only when its attribute set satisfies the access policy. The latter SP-ABS is the opposite. The signer possesses the signing key corresponding to his attributes and the message with an access policy. In the signature generation stage, a valid signature can be generated if and only if the attributes of the signer satisfy the access policy. In the signature verification phase, the verifier can only ensure that the signer's attributes satisfy the message's access policy but cannot distinguish the signer's identity. These are the unforgeability and privacy (anonymity) of ABS. Since ABS has fine-grained access control, anonymous authentication, privacy protection, and other good properties, it finds broad applications in many fields, such as private access control and anonymous credential.

The concept of ABS was introduced by Maji et al. [1]. They presented the definition and security model and proposed concrete schemes with security proof under the general group model. Later, Li et al. [2, 3], Shahandashti et al. [4], and Gagné et al. [5] constructed the ABS schemes under the selection model. These schemes only support threshold predicates. Maji et al. [6] and Gu et al. [7] proposed ABS schemes for monotone predicates. In 2011, Okamoto et al. [8, 9] proposed ABS schemes supporting nonmonotone predicates, improving access control flexibility, and satisfying adaptive security. In 2012, Herranz et al. [10] constructed a threshold ABS scheme with constant signature length, and its security is improved from the original selective unforgeability to adaptive unforgeability. In the same year, Chen et al. [11] combined ABS with attribute-based encryption (ABE) to construct a hybrid

ABS/ABE scheme. The advantage is that ABS and ABE share the same user private key, which reduces the cost of key generation. Su et al. [12] proposed an attribute signature scheme that supports the threshold tree access structure. While the expression and security of ABS continue to improve, its functions are also constantly evolving. Wang and Chen [13] constructed a lattice-based ABS scheme to resist quantum computing attacks. Escala et al. [14] introduced the concept of traceability, allowing a trusted authority to control the signer's identity and hold the signer accountable when the signer breaks the law. Tang et al. [15] proposed an ABS scheme for circuits from multilinear, and Sakai et al. [16] proposed an ABS scheme for circuits from bilinear maps. Based on lattices, Kaafarani et al. [17] proposed an ABS scheme for unbounded circuits. Datta et al. [18] proposed an ABS scheme for unbounded arithmetic branching programs.

All ABS schemes mentioned above have a single attribute authority. This attribute authority knows the signing keys of all users, so it must be trustworthy. Moreover, this attribute authority may become the bottleneck of the system. To overcome this shortcoming, the concepts of multiauthority attribute-based signature [19–21] and decentralized attribute-based signature [22–24] were introduced.

The existing works are summarized in Table 1.

1.1. Related Works. Generally, the computational overhead of ABS is too large, making it unsuitable for resource-constrained equipment. To this end, using cloud computing outsourcing technology, Chen et al. [25] introduced the concept of outsourced attribute-based signature. After that, several outsourced attribute-based signature schemes were proposed [26, 27]. In addition, several ABS schemes with various additional properties have also been proposed, such as group signature [28], signcryption [29], proxy signature [30], traceability [23], revocation [14], hierarchical [31], linkability [21], message recovery [32], and self-revealability [33].

So far, attribute-based signatures are still receiving widespread attention. In 2021, Perera et al. [34] constructed an attribute-based group signature (ABGS) scheme with verifier-local revocation (VLR). In the same year, Chen et al. [35] presented a novel ABS scheme using the attribute tree as an access policy that expresses flexible access control. They utilized the server-aid technique to verify signatures and reduce the computation burden. Luo et al. [36] introduced attribute-based proxy resignatures (ABPRS), which allows a semitrusted proxy to transform a signature of one entity into a signature of another, without revealing any signing key and information about the signer. Zhao et al. [37] constructed a novel attribute-based signcryption (ABSC) scheme realizing multiauthority access control and constant-size ciphertext that does not depend on the number of attributes or authorities.

In recent years, attribute-based signature has found new applications in many fields. Yang et al. [38] and Guo et al. [39] construct medical record management systems based on attribute-based signature and blockchain, respectively. Liu et al. [40] proposed a secure vehicular crowdsensing scheme based on multiauthority attribute-based signature (TRAMS), which allows the publisher to flexibly customize a fine-grained policy that the potential participants must satisfy and uses the attribute-based signature to authenticate sensed messages while protecting the privacy of the sensing vehicle. Also, they proposed a multiauthority key management scheme, which can improve vehicle-based sensing efficiency on the Internet of vehicles.

1.2. Motivation and Contributions. So far, the existing attribute-based signature schemes have dealt with attributes in a way that is "with" or "without". No attribute-based signature scheme supports comparative attributes or more complex relationship attributes. Consider such a simple illustrative example. On the forum of a game community, it is required that only players who meet the following conditions can postexperience and guide novices:

Member of the community AND

((Register before 2018 AND More than 10 million game points) OR Top 32 in first-class competitions).

Suppose member Alice was registered in 2017 and has 20 million game points. We can easily determine that her attributes meet the above access structure, but none of the existing attribute-based signature schemes can handle it directly. The previous processing method is to expand the access structure or attributes. For example, expand "Register before 2018" and "More than 10 million game points" to

"Registered in 2000 OR Registered in 2002 OR $\cdots$ OR Registered in 2017" and

"11 million game points OR 12 million game points OR $\cdots$ OR 127 million game points".

Although this trivial method can solve the problem, it brings $O(N)$ level attribute amount expansion, where N is the size of the value space of the attribute. This will make the storage overhead and computation overhead increase linearly with N. The trivial method is not practical, so it is urgent to propose a practical comparative attribute management method.

The main contributions of this paper are as follows:

(i) Using the 0-encoding and 1-encoding of Lin and Tzeng [41], we propose an access structure algorithm and an attribute expansion algorithm, which reduce the data expansion from $O(N)$ to $O(\log_2 N)$. These two algorithms enable the attribute-based signature scheme to deal with the comparative attributes effectively.

(ii) Using the proposed expansion algorithms, we propose an efficient attribute-based signature scheme that supports comparative attributes. Assuming that the computational Diffie–Hellman exponent (CDHE) problem is hard, the proposed scheme is existentially unforgeable under adaptive chosen message but selective access structure attack. The proposed scheme achieves privacy in the sense of information theory, and the adversary cannot break the privacy even if he has infinite capabilities.

Table 1: Summary of existing works.

Scheme	Access structure	Type	Assumption	Model
Maji et al. [1]	LSSS	KP	CDH	GGM
Jin and Kim [2]	Threshold	KP	CDH	ROM
Jin et al. [3]	Threshold	SP	CDH	ROM&SM
Shahandashti and Safavi-Naini [4]	Threshold	KP	CDH	SM
Gagné et al. [5]	Threshold	KP	CDH	ROM
Maji et al. [6]	Threshold	SP	CDH&DLIN	GGM
Gu et al. [7]	LSSS	SP	CDH	SM
Okamoto and Takashima [8]	Threshold	KP	DLIN	GGM
Okamoto and Takashima [9]	LSSS	SP	DLIN	GGM
Herranz et al. [10]	Threshold	SP	DLIN	GGM
Chen et al. [11]	LSSS	SP	CDHE	ROM
Su et al. [12]	Threshold	SP	CDH	ROM
Wang and Chen [13]	Threshold	SP	SIS	ROM
Escala et al. [14]	LSSS	SP	SIS	SM
Tang et al. [15]	Circuit	KP	MCDH	ROM
Sakai et al. [16]	Circuit	KP	CDH	ROM
Kaafarani and Ghadafi [17]	Circuit	SP	DDH	GGM
Datta et al. [18]	LSSS	SP	DLIN	SM
Cao et al. [19]	Threshold	SP	CDH	SM
Cao et al. [20]	Threshold	KP	CDH	ROM
El Kaafarani et al. [21]	LSSS	SP	DDH	ROM
Okamoto and Takashima [22]	LSSS	SP	DLIN	ROM
El Kaafarani et al. [23]	LSSS	SP	DLIN&CDH	ROM&SM
Ghadafi [24]	LSSS	SP	DDH&DLIN	SM

LSSS = linear secret sharing scheme, SP=signature-policy, KP=key-policy, DLIN = decisional linear assumption, SIS=short integer solution, CDH=computational Diffie–Hellman assumption, MCDH = multilinear CDH, DDH = decisional DH, CDHE=computational Diffie–Hellman exponent assumption, GGM = generic group model, ROM = random oracle model, and SM=standard model.

(iii) Theoretical analysis and simulation experiments show that our method is practical and has significant advantages in storage and computation overhead compared with the trivial way.

(iv) Comparable attribute-based signature dramatically expands the application scenarios of attribute-based signature.

1.3. Organization. The rest of the paper is organized as follows. The necessary background and notations are presented and reviewed in Section 2. Section 3 reviewed the attribute-based signature with its security model. Section 4 describes comparative attribute management. Our CABS constructions are proposed in Section 5. The security proof and performance analysis of the proposed scheme are given in Sections 6 and 7, respectively. Finally, the paper is concluded in Section 8.

2. Preliminaries

2.1. Notations. The notations are summarized in Table 2.

2.2. Bilinear Mapping and the Complexity Assumptions. In this section, we introduce the notions of bilinear maps, complexity assumption, access structure, and linear secret sharing scheme.

Definition 1 (bilinear maps). *Let p be a prime number. Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of order p. A map $e\colon \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ is called a bilinear map or (bilinear) pairing if the following hold:*

(i) *Bilinearity.* $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}, \forall g_1, g_2 \in \mathbb{G}, a, b \in \mathbb{Z}_p$.

(ii) *Nondegeneracy.* $e(g_1, g_2) \neq 1_T$, *whenever* $g_1, g_2 \neq (1, 1)$, *where 1 (or 1_T) is the identity element in $\mathbb{G}$ or $(\mathbb{G}_T)$.*

(iii) *Computability.* $e(g_1, g_2)$ *is efficiently computable,* $\forall g_1, g_2 \in \mathbb{G}$.

Definition 2 (computational Diffie–Hellman exponent (CDHE) assumption). *The challenger chooses $g \in \mathbb{G}$, $\alpha \in \mathbb{Z}_p$ at random and outputs $(g, g^{\alpha}, g^{\alpha^2}, \dots, g^{\alpha^n}, g^{\alpha^{n+2}}, \dots, g^{\alpha^{2n}})$. The CDHE problem is to compute $g^{\alpha^{n+1}}$ according to $(g, g^{\alpha}, g^{\alpha^2}, \dots, g^{\alpha^n}, g^{\alpha^{n+2}}, \dots, g^{\alpha^{2n}})$. The (t, ε)-CDHE assumption holds if no t-time adversary has at least ε advantage to solve the CDHE problem.*

2.3. 0-Encoding and 1-Encoding. The 0-encoding and the 1-encoding are used by Lin et al. to solve the millionaire problem [41]. Let $s = s_n s_{n-1} \dots s_1 \in \{0, 1\}^n$ be an n-length binary string of a value:

(i) The *0-encoding* of s is defined as a set

$$S_s^0 = \{s_n s_{n-1} \cdots s_{i+1} 1 | s_i = 0, 1 \leq i \leq n\}. \tag{1}$$

(ii) The *1-encoding* of s is the set

Table 2: Notations.

Notation	Description
$x\in_R X$	Picking an element x uniformly at random from the set X
$[n]$	Positive integer set $\{1, 2, \ldots, n\}$ for any positive integer n
$\mathbf{a}$	A vector $\mathbf{a} = (a_1, \ldots, a_j) \in \mathbb{Z}_p^j$
$\mathbf{ab}$	Inner product of vectors $\mathbf{a}$ and $\mathbf{b}$, $\mathbf{ab} = a_1b_1 + \cdots + a_jb_j$
$\lvert A\rvert$	The number of elements in the set A
u_i	Usual attribute
$\widetilde{u}_i$	Comparative attribute
$\ddot{u}_i$	Comparative attribute name
$\ddot{a}_i$	Comparative attribute threshold
$\underset{\cdot\cdot}{b}_i$	User attribute value

$$S_s^1 = \{s_n s_{n-1} \cdots s_i | s_i = 1, 1 \le i \le n\}. \tag{2}$$

Intuitively, the 1-encoding of s is the set of all its odd prefix substrings, and the 0-encoding is the set of all of its modified even prefix substrings, where the least significant bit is flipped from "0" to "1". For example, $20 = (10100)_2$, its 0-encoding $S_{20}^0 = \{11, 1011, 10101\}$, and 1-encoding $S_{20}^1 = \{1, 101\}$.

Lemma 1 (see [41]). *$x > y$ if and only if $S_x^1 \cap S_y^0 \neq \varnothing$.*

2.4. Access Structure and Linear Secret Sharing Scheme. Let $\mathscr{U} = \{u_1, u_2, \ldots, u_n\}$ be an attribute universe; an access structure is a Boolean function f over $\mathscr{U}$. An attribute set $A \subseteq \mathscr{U}$ is an authorized set, if $f(A) = 1$. An access structure is monotone if $f(A) = 1$ and $A \subseteq B$ implies $f(B) = 1$ for all $A, B \subseteq \mathscr{U}$.

A Linear Secret Sharing Scheme (LSSS) for monotone access structure f over $\mathbb{Z}_p$ is a matrix $\mathbf{M}_{l\times k}$ along with a function $\rho(i)$ to indicate the i-th row of $\mathbf{M}$ as an attribute in f, which consists of the following polynomial time operations:

(i) *Distribution of Shares* . The distribution of a secret $a \in \mathbb{Z}_p$ is performed by the dealer. The dealer first samples $a_2, a_3, \ldots, a_k \in \mathbb{Z}_p$ and sets $\mathbf{v} = (a, a_2, a_3, \ldots, a_k) \in \mathbb{Z}_p^k$. Then, the dealer outputs a set $\{\lambda_i\colon \lambda_i = \mathbf{M}_i\mathbf{v}\}_{i\in[l]}$, where $\mathbf{M}_i$ is the i^{th} row of the matrix $\mathbf{M}$.

(ii) *Reconstruction of the Secret* . Suppose that A is an authorized set. The secret reconstruction constants $\{w_i\}_{i\in I} \subset \mathbb{Z}_p$, where $I = \{i\colon \rho(i) \in A\}$, satisfying $\sum_{i\in I} w_i \mathbf{M}_i = \mathbf{1}$. Hence, $\sum_{i\in I} w_i \lambda_i = a$.

3. Attribute-Based Signature

3.1. Algorithms. An attribute-based signature (ABS) scheme consists of the following algorithms:

(i) **Setup**: it takes as input the security parameter λ and returns the system public parameters *PP* and master secret key *msk*.

(ii) **KeyGen**: it takes the master secret key msk and an attribute set A as inputs and returns the signing key *skA*.

(iii) **Sign**: it takes a signing key sk_A, a message M, and an access structure f as inputs and returns a signature σ if $f(A) = 1$.

(iv) **Verify**: it takes a signature σ, a message M, and an access structure f as inputs and returns 1 or 0.

3.2. Security. A secure ABS scheme should have the properties of correctness, unforgeability, and privacy. We present formal definitions of them in the following.

Definition 3 (correctness). *An ABS scheme is correct, if*

$$\Pr\left[\mathbf{Verify}(\sigma, M, f) \to 1 \;\middle|\; \begin{array}{l} (\mathrm{PP}, \mathrm{msk}) \leftarrow \mathbf{Setup}(1^\lambda) \\ sk_A \leftarrow \mathbf{KeyGen}(msk, A) \\ \sigma \leftarrow \mathbf{Sign}(sk_A, M, f) \end{array}\right] = 1, \tag{3}$$

for any M, f, and A such that $f(A) = 1$.

The popular notion of unforgeability for ABS is unforgeable under adaptive chosen message and selective access structure (EUF-sA-CMA). We describe the EUF-sA-CMA attack by the following game between challenger $\mathscr{C}$ and adversary $\mathscr{A}$.

GAME 1.(EUF-sA-CMA):

(i) **Init**. $\mathscr{A}$ sends a challenge access structure f^* to $\mathscr{C}$.

(ii) **Setup**. $\mathscr{C}$ generates and sends the system public parameters to $\mathscr{A}$.

(iii) **Queries Phase**. $\mathscr{A}$ can access the following oracles.

  (a) **KeyGen-Oracle**. A sends an attribute set A to C, C returns a signing key skA.

  (b) **Sign-Oracle**. A sends a message M and an access structure A to C, C returns a signature.

(iv) **Forgery**. $\mathscr{A}$ outputs a triple (σ^*, M^*, f^*)

$\mathscr{A}$ wins the GAME 1, if

(i) $\mathbf{Verify}(\sigma^*, M^*, f^*) \longrightarrow 1$.

(ii) (M^*, f^*) has never been queried to **Sign-Oracle.**

(iii) Any attribute set A queried to **KeyGen-Oracle** does not satisfy the challenge access structure f^*.

The advantage $Adv_{\mathscr{A}}^{\mathrm{EUF}}(1^\lambda)$ is defined as the probability of $\mathscr{A}$ winning the game above.

Definition 4. (unforgeability). *An ABS scheme is existentially unforgeable under adaptive chosen message but selective attribute attack if the advantage $Adv_{\mathscr{A}}^{EUF}(1^\lambda)$ is negligible for any PPT adversary $\mathscr{A}$.*

For a secure ABS scheme, an adversary cannot find the attribute set used to generate the signature. We describe privacy by the following game between challenger $\mathscr{C}$ and adversary $\mathscr{A}$.

GAME 2.(privacy):

(i) **Setup** and **Queries Phase 1** are the same as **Setup** and **Queries Phase** in GAME 1, respectively

(ii) **Challenge**. $\mathscr{A}$ chooses and sends (M, f, A_0, A_1) such that $f(A_0) = f(A_1) = 1$ to $\mathscr{C}$. $\mathscr{C}$ chooses $b \in_R \{0,1\}$, runs **Sign** with inputs $(\mathrm{SK}_{A_b}, M, f)$ to generate σ_b, and returns σ_b to $\mathscr{A}$.

(iii) **Queries Phase 2**. The same as the **Queries Phase** 1 above.

(iv) **Guess**. $\mathscr{A}$ outputs his guess $b' \in \{0,1\}$.

The advantage of $\mathscr{A}$ is $Adv_{\mathscr{A}}^{Pri}(1^\lambda) = |\Pr[b' = b] - 1/2|$.

Definition 5 (privacy). *An ABS scheme achieves privacy if the advantage $Adv_{\mathscr{A}}^{Pri}(1^\lambda)$ is negligible for any adversary $\mathscr{A}$.*

4. Comparative Attribute Management

4.1. Expansion Algorithms. Denote the usual attribute as u_i and the comparative attribute as $\tilde{u}_i$. The comparative attribute is further expressed as $\ddot{u}_i > \ddot{a}_i$ or $\ddot{u}_i < \ddot{a}_i$, where $\ddot{u}_i$ is its attribute name and $\ddot{a}_i$ is its threshold. Denote the user attribute corresponding to the comparative attribute $\tilde{u}_i$ as $\ddot{u}_i \| \ddot{b}_i$.

In the above example, "Member of the community" is a usual attribute, "Register before 2018," "More than 10 million game points", and "Top 32 in the first-class competition" are comparative attributes; they can be denoted as u_1, $\tilde{u}_2 = \ddot{u}_2 < 18$, $\tilde{u}_3 = \ddot{u}_3 > 10$, and $\tilde{u}_4 = \ddot{u}_4 \le 32 = \ddot{u}_4 < 33$. Then, the access structure is $f = u_1 \wedge ((\tilde{u}_2 \wedge \tilde{u}_3) \vee \tilde{u}_4)$. Alice's attributes can be denoted as $A = (u_1, \ddot{u}_2 \| 17, \ddot{u}_3 \| 20)$.

We can easily see that Alice's attributes A satisfy the access structure f, but the algorithm cannot. It is necessary to extend the comparative attributes and access structure so that the algorithm can use Lemma 1 to determine whether it is satisfied. We propose the following algorithms to extend the access structure and user attributes.

4.1.1. AccStruExpan. The access structure expansion algorithm inputs an access structure f with its matrix $\mathbf{M}$ and outputs a new access structure f' with its matrix $\mathbf{M}'$.

For all comparative attributes $\tilde{u}_i$,

If $\tilde{u}_i = \ddot{u}_i > \ddot{a}_i$, encode it to 0-coding $\mathscr{U}_{\tilde{u}_i} = U_{\ddot{a}_i}^0 = \left\{\tilde{u}_{i,1}, \tilde{u}_{i,2}, \ldots, \tilde{u}_{i,n_{\tilde{u}_i}}\right\}$.

If $\tilde{u}_i = \ddot{u}_i < \ddot{a}_i$, encode it to 1-coding $\mathscr{U}_{\tilde{u}_i} = U_{\ddot{a}_i}^1 = \left\{\tilde{u}_{i,1}, \tilde{u}_{i,2}, \ldots, \tilde{u}_{i,n_{\tilde{u}_i}}\right\}$.

Sets

$$\rho'(j) = \begin{cases} \rho(j), & j < k, \\ \tilde{u}_{i,l}, & j = k + l - 1, l \in \left\{1, 2, \ldots, n_{\tilde{u}_i}\right\}, \\ \rho\left(j - n_{\tilde{u}_i} + 1\right), & j \ge k + n_{\tilde{u}_i}, \end{cases} \tag{4}$$

where k: $\rho(k) = \tilde{u}_i$.

Repeat the k-th row of the matrix $n_{\tilde{u}_i}$ times as rows $k, k+1, \ldots, k + n_{\tilde{u}_i} - 1$ to obtain a new matrix.

Replace each comparative attribute $\tilde{u}_i$ in f with $\vee_{j=1}^{n_{\tilde{u}_i}} \tilde{u}_{i,j}$ to get the new access structure f'.

4.1.2. UserAttExpan. The user attribute expansion algorithm inputs a user attribute set A and outputs a new attribute set A'.

(i) Encode each attribute $\ddot{u}_i \| \ddot{b}_i$ to 0-coding $A_{\ddot{u}_i} \| \ddot{b}_i^{0}$ and 1-coding $A_{\ddot{u}_i} \| \ddot{b}_i^{1}$

(ii) Replace each attribute $\ddot{u}_i \| \ddot{b}_i$ in A with $A_{\ddot{u}_i} \| \ddot{b}_i^{0} \cup A_{\ddot{u}_i} \| \ddot{b}_i^{1}$ to get the new attribute set A'

4.2. Example. To facilitate understanding the above algorithms, we use the above example to execute the algorithms as follows.

AccStruExpan: Take as input $(f, \mathbf{M}, \rho)$, where $f = u_1 \wedge ((\tilde{u}_2 \wedge \tilde{u}_3) \vee \tilde{u}_4)$,

$$\mathbf{M} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \begin{matrix} \rho(1) = u_1, \\ \rho(2) = \tilde{u}_2, \\ \rho(3) = \tilde{u}_3, \\ \rho(4) = \tilde{u}_4, \end{matrix} \tag{5}$$

(i) For $\tilde{u}_2$,Encode $\tilde{u}_2 = \ddot{u}_2 < 18$ to 1-coding:

$$\mathscr{U}_{\tilde{u}_2} = \mathscr{U}_{\ddot{u}_2 < 18} = U_{18}^1 = \{(1, \ddot{u}_2 <), (1001, \ddot{u}_2 <)\} = \left\{\tilde{u}_{2,1}, \tilde{u}_{2,2}\right\}. \tag{6}$$

Set $\rho'(1) = \rho(1) = u_1$, $\rho'(2) = \tilde{u}_{2,1}, \rho'(3) = \tilde{u}_{2,2}$, $\rho'(4) = \rho(3) = \tilde{u}_3, \rho'(5) = \rho(4) = \tilde{u}_4$. Repeat the 2-th row 2 times as rows 2, 3 to obtain a new matrix

$$\mathbf{M}' = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \begin{matrix} \rho'(1) = u_1, \\ \rho'(2) = \tilde{u}_{2,1}, \\ \rho'(3) = \tilde{u}_{2,2}, \\ \rho'(4) = \tilde{u}_3, \\ \rho'(5) = \tilde{u}_4, \end{matrix} \tag{7}$$

(ii) For $\tilde{u}_3$, Encode $\tilde{u}_3 = \ddot{u}_3 > 10$ to 0-coding:

$$\begin{aligned}\mathcal{U}_{\tilde{u}_3} &= \mathcal{U}_{\ddot{u}_3>10} = U_{10}^{0} = \{(1,\ddot{u}_3>),\ (01,\ddot{u}_3>),\ (001,\ddot{u}_3>),\ (00011,\ddot{u}_3>),\ (0001011,\ddot{u}_3>)\}\\ &= \left\{\tilde{u}_{3,1},\tilde{u}_{3,2},\tilde{u}_{3,3},\tilde{u}_{3,4},\tilde{u}_{3,5}\right\}.\end{aligned} \tag{8}$$

Set $\rho''(1)=\rho'(1)=u_1$, $\rho''(2)=\rho'(2)=\tilde{u}_{2,1}, \rho''(3)=\rho'(3)=\tilde{u}_{2,2}$, $\rho''(4)=\tilde{u}_{3,1}, \rho''(5)=\tilde{u}_{3,2}, \rho''(6)=\tilde{u}_{3,3}, \rho''(7)=\tilde{u}_{3,4}, \rho''(8)=\tilde{u}_{3,5}, \rho''(9)=\rho'(9)=\tilde{u}_4$. Repeat the 4-th row 5 times as rows $4,5,6,7,8$ to obtain a new matrix:

$$\mathbf{M}'' = \begin{pmatrix} 1&1&0\\ 0&1&1\\ 0&1&1\\ 0&0&1\\ 0&0&1\\ 0&0&1\\ 0&0&1\\ 0&0&1\\ 0&1&0 \end{pmatrix}, \begin{array}{l} \rho''(1)=u_1,\\ \rho''(2)=\tilde{u}_{2,1},\\ \rho''(3)=\tilde{u}_{2,2},\\ \rho''(4)=\tilde{u}_{3,1},\\ \rho''(5)=\tilde{u}_{3,2},\\ \rho''(6)=\tilde{u}_{3,3},\\ \rho''(7)=\tilde{u}_{3,4},\\ \rho''(8)=\tilde{u}_{3,5},\\ \rho''(9)=\tilde{u}_4. \end{array} \tag{9}$$

(iii) For $\tilde{u}_4$,Encode $\tilde{u}_4=\ddot{u}_4<33$ to 1-coding:

$$\begin{aligned}\mathcal{U}_{\tilde{u}_4} &= \mathcal{U}_{\ddot{u}_4<33} = U_{33}^{1} = \{(1,\ddot{u}_4<),\ (100001,\ddot{u}_4<)\}\\ &= \left\{\tilde{u}_{4,1},\tilde{u}_{4,2}\right\}.\end{aligned} \tag{10}$$

Set $\rho'''(1)=\rho''(1)=u_1$, $\rho'''(2)=\rho''(2)=\tilde{u}_{2,1}, \rho'''(3)=\rho''(3)=\tilde{u}_{2,2}$, $\rho'''(4)=\rho''(4)=\tilde{u}_{3,1}, \rho'''(5)=\rho''(5)=\tilde{u}_{3,2}, \rho'''(6)=\rho''(6)=\tilde{u}_{3,3}, \rho'''(7)=\rho''(7)=\tilde{u}_{3,4}, \rho'''(8)=\rho''(8)=\tilde{u}_{3,5}, \rho'''(9)=\tilde{u}_{4,1}, \rho'''(10)=\tilde{u}_{4,2}$. Repeat the 9-th row 2 times as rows $9,10$ to obtain the final matrix which is

$$\mathbf{M}''' = \begin{pmatrix} 1&1&0\\ 0&1&1\\ 0&1&1\\ 0&0&1\\ 0&0&1\\ 0&0&1\\ 0&0&1\\ 0&0&1\\ 0&1&0\\ 0&1&0 \end{pmatrix}, \begin{array}{l} \rho'''(1)=u_1,\\ \rho'''(2)=\tilde{u}_{2,1},\\ \rho'''(3)=\tilde{u}_{2,2},\\ \rho'''(4)=\tilde{u}_{3,1},\\ \rho'''(5)=\tilde{u}_{3,2},\\ \rho'''(6)=\tilde{u}_{3,3},\\ \rho'''(7)=\tilde{u}_{3,4},\\ \rho'''(8)=\tilde{u}_{3,5},\\ \rho'''(9)=\tilde{u}_{4,1},\\ \rho'''(10)=\tilde{u}_{4,2}. \end{array} \tag{11}$$

Replace $\tilde{u}_2,\tilde{u}_3$, and $\tilde{u}_4$ with $\tilde{u}_{2,1}\vee\tilde{u}_{2,2},\tilde{u}_{3,1}\vee\tilde{u}_{3,2}\vee\tilde{u}_{3,3}\vee\tilde{u}_{3,4}\vee\tilde{u}_{3,5}$, and $\tilde{u}_{4,1}\vee\tilde{u}_{4,2}$ respectively; the final access structure is

$$\begin{aligned} f' = u_1\wedge\Big(\big(\big((\tilde{u}_{2,1}\vee\tilde{u}_{2,2})\wedge(\tilde{u}_{3,1}\vee\tilde{u}_{3,2}\vee\tilde{u}_{3,3}\vee\tilde{u}_{3,4}\vee\tilde{u}_{3,5}\big)\big)\\ \vee(\tilde{u}_{4,1}\vee\tilde{u}_{4,2})\Big).\end{aligned} \tag{12}$$

UserAttExpan: take as input $A=\{u_1,\ddot{u}_2\|17,\ddot{u}_3\|20\}$.

(i) Encode attribute $\ddot{u}_2\|17$ to 0-coding:

$$\begin{aligned} A_{\ddot{u}_2}\|17^{0} &= \{(11,\ddot{u}_2<),\ (101,\ddot{u}_2<),\ (1001,\ddot{u}_2<)\}\\ &= \left\{u_{2,1},u_{2,2},u_{2,3}\right\}.\end{aligned} \tag{13}$$

and 1-coding

$$A_{\ddot{u}_2}\|17^{1} = \{(1,\ddot{u}_2>),\ (10001,\ddot{u}_2>)\} = \left\{u_{2,4},u_{2,5}\right\} \tag{14}$$

(ii) Encode attribute $\ddot{u}_3\|20$ to 0-coding:

$$\begin{aligned} A_{\ddot{u}_3}\|20^{0} &= \{(1,\ddot{u}_3<),\ (01,\ddot{u}_3<),\ (0011,\ddot{u}_3<),\\ &\quad\cdot(001011,\ddot{u}_3<),\ (0010101,\ddot{u}_3<)\}\\ &= \left\{u_{3,1},u_{3,2},u_{3,3},u_{3,4},u_{3,5}\right\},\end{aligned} \tag{15}$$

and 1-coding

$$A_{\ddot{u}_3}\|20^{1} = \{(001,\ddot{u}_3>),\ (00101,\ddot{u}_3>)\} = \left\{u_{3,6},u_{3,7}\right\}. \tag{16}$$

(iii) The new user attribute set $A'=\{u_1,u_{2,1},u_{2,2},u_{2,3}, u_{2,4},u_{2,5},u_{3,1},u_{3,2},u_{3,3},u_{3,4},u_{3,5},u_{3,6},u_{3,7}\}$.

When running the algorithms above, we assume that the value spaces of the comparative attributes $\tilde{u}_2,\tilde{u}_3$, and $\tilde{u}_4$ are $[0,31]$, $[0,127]$, and $[0,63]$, respectively. If the trivial expansion method is used, the matrix of the access structure will increase from 4 rows to 168 rows, while using our expansion method, the matrix only grows to 10 rows. There is a 15.8 times gap between the two, which shows that our expansion method is effective.

5. Comparable Attribute-Based Signature Scheme

Based on the above comparative attribute management method and Chen et al.'s attribute-based signature scheme [11], we propose a practical attribute-based signature scheme that supports comparative attributes.

5.1. The Overall Framework.

5.1. The Overall Framework. The overall framework of our scheme is shown in Figure 1. The **Setup** algorithm generates the public parameters and master private key for the system. The **KeyGen** algorithm calls the **UserAttExpan** algorithm to generate the private key. The **Sign** algorithm and the **Verify** algorithm call the **AccStruExpan** algorithm to generate a signature and **verify** the signature, respectively.

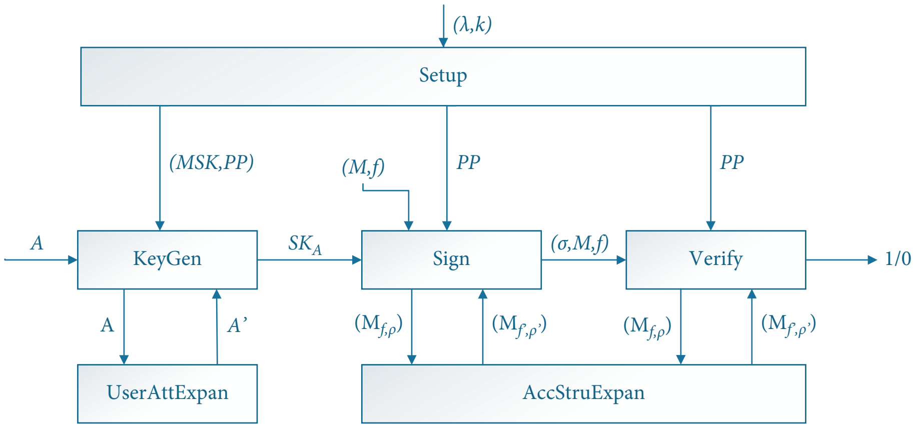

Figure 1: The framework of SPCABS.

5.2. The Proposed Scheme. Chen et al.'s scheme inputs the attribute universe in the **Setup** phase and its public key is related to the attribute universe, so it is challenging to support attribute expansion. We use the hash value $H_1(u)$ of attribute u instead of the public key h_u in Chen et al.'s scheme. So the scheme can support the dynamic attribute universe and solve the problem of not supporting attribute expansion. Another advantage is that the size of the public key is significantly reduced.

(i) **Setup** $(1^\lambda, 1^k)$: Choose two prime order $p > 2^\lambda$ multiplicative cyclic groups $\mathbb{G}$, $\mathbb{G}_T$ with a generator g and a bilinear map $e\colon \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$. Choose two collision-resistant hash functions $H_1\colon \{0,1\}^* \longrightarrow \mathbb{G}$, $H_2\colon \{0,1\}^* \longrightarrow \{0,1\}^k$Choose $a, \alpha \in_R \mathbb{Z}_p$, then compute $y = g^a, Z = e(g,g)^\alpha$. Choose $u_0, u_i \in_R \mathbb{G}, i = 1, 2, \ldots, k$. The public parameters $PP = (p, g, \mathbb{G}, \mathbb{G}_T, e, y, Z, H_1, H_2, u_0, \{u_i\}_{i=1}^k)$ and the master secret key $MSK = \alpha$.

(ii) **KeyGen** (PP, MSK, A): Run the user attribute expansion algorithm **UserAttExpan**, and extend A to A'. Pick a random value $t \in_R \mathbb{Z}_p$, and compute

$$D = g^\alpha \cdot y^t, L = g^t, \left\{D_u = H_1^t(u)\right\}_{u \in A'}. \tag{17}$$

The private key $SK_A = (D, L, \{D_u\}_{u \in A'})$.

(iii) **Sign** (PP, f, M, SK_A): Run the access structure expansion algorithm **AccStruExpan**, extend $(\mathbf{M}_f, \rho)$ to $(\mathbf{M}_{f'}, \rho')$.Let $I = \left\{i \in [l_{f'}]\colon \rho'(i) \in A'\right\}$, and find $\{w_i\colon i \in I\}$ such that $\sum_{i \in I} w_i \mathbf{M}_{f'i} = \mathbf{1}$, where $\mathbf{1} = (1, 0, \ldots, 0)$. Set $w_i = 0$ for $i \in [l_{f'}] \backslash I$. Pick random $\{v_i\}_{i=1}^{l_{f'}}$ satisfying $\sum_{i=1}^{l_{f'}} v_i \mathbf{M}_{f'i} = \mathbf{0}$, where $\mathbf{0} = (0, 0, \ldots, 0)$. Choose $t' \in_R \mathbb{Z}_p$, randomize part of the private key

$$D' = D \cdot y^{t'}, L' = L \cdot g^{t'}, \left\{D'_u = D_u \cdot H_1^{t'}(u)\right\}_{u \in \left\{\rho'(i)\colon i \in I\right\}}. \tag{18}$$

Compute $h = H_2(f \| M) = h_1 h_2 \ldots h_k$, and $H_M = u_0 \cdot \sum_{i=1}^k u_i^{h_i}$.Choose $r, s \in_R \mathbb{Z}_p$, and compute

$$\begin{aligned} \sigma_0 &= D' \cdot H_M^r, \\ \sigma'_0 &= g^r, \\ \sigma_{0,i} &= L'^{w_i} \cdot g^{sv_i}, i \in \left[l_{f'}\right], \\ \sigma_{1,i} &= D'^{w_i}_{\rho'(i)} \cdot H_1^{sv_i}(\rho'(i)), i \in \left[l_{f'}\right]. \end{aligned} \tag{19}$$

The final signature $\sigma = (\sigma_0, \sigma'_0, \left\{\sigma_{0,i}, \sigma_{1,i}\right\}_{i=1}^{l_{f'}})$.

(iv) **Verify** (PP, σ, M, f): Run the access structure expansion algorithm **AccStruExpan**, and extend $(\mathbf{M}_f, \rho)$ to $(\mathbf{M}_{f'}, \rho')$. Pick a random vector $\mathbf{x} = (x, x_2, \ldots, x_n) \in \mathbb{Z}_p^n$, and compute the shares $\lambda_i = \langle \mathbf{x}, \mathbf{M}_{f'i} \rangle$ for $i = 1, \ldots, l_{f'}$. Compute $h = H_2(f \| M) = h_1 h_2 \ldots h_k$, and $H_M = u_0 \cdot \sum_{i=1}^k u_i^{h_i}$. The verifier checks the equation

$$\frac{e(g^x, \sigma_0)}{e(H_M^x, \sigma'_0) \cdot \prod_{i=1}^{l_{f'}} \left(e\left(y^{\lambda_i} H_1^{-x}(\rho'(i)), \sigma_{0,i}\right) \cdot e\left(g^x, \sigma_{1,i}\right)\right)} \overset{?}{=} Z^x. \tag{20}$$

6. Proofs of Security

Theorem 1 (correctness). *The proposed scheme is correct.*

Proof. Because

$$\begin{aligned} \sigma_0 &= D' \cdot H_M^r, \\ \sigma'_0 &= g^r, \\ \sigma_{0,i} &= L'^{w_i} \cdot g^{sv_i}, i \in \left[l_{f'}\right], \\ \sigma_{1,i} &= D'^{w_i}_{\rho'(i)} \cdot H_1^{sv_i}(\rho'(i)), i \in \left[l_{f'}\right], \end{aligned} \tag{21}$$

we have

$$\begin{aligned} e\left(g^{x},\sigma_{0}\right) &= e\left(g^{x},D'\cdot\left(H_{M}\right)^{r}\right)\\ &= e\left(g^{x},g^{\alpha}\cdot y^{t}\cdot y^{t'}\cdot H_{M}^{r}\right)\\ &= e\left(g^{x},g^{\alpha}\cdot g^{a\left(t+t'\right)}\cdot H_{M}^{r}\right)\\ &= e\left(g,g\right)^{x\alpha}\cdot e\left(g,g\right)^{xa\left(t+t'\right)}\cdot e\left(g,H_{M}\right)^{xr},\\ e\left(H_{M}^{x},\sigma_{0}'\right) &= e\left(H_{M}^{x},g^{r}\right) = e\left(g,H_{M}\right)^{xr},\\ e\left(y^{\lambda_{i}}H_{1}^{-x}\left(\rho'\left(i\right)\right),\sigma_{0,i}\right)\cdot e\left(g^{x},\sigma_{1,i}\right) &= e\left(g^{a\lambda_{i}}H_{1}^{-x}\left(\rho'\left(i\right)\right),L'^{w_{i}}\left(g^{s}\right)^{v_{i}}\right)\cdot e\left(g^{x},\left(D_{\rho'\left(i\right)}'\right)^{w_{i}}H_{1}^{sv_{i}}\left(\rho'\left(i\right)\right)\right)\\ &= e\left(g^{a\lambda_{i}}H_{1}^{-x}\left(\rho'\left(i\right)\right),\left(L\cdot g^{t'}\right)^{w_{i}}\left(g^{s}\right)^{v_{i}}\right)\cdot e\left(g^{x},D_{\rho'\left(i\right)}^{w_{i}}\cdot H_{1}^{t'w_{i}}\left(\rho'\left(i\right)\right)H_{1}^{sv_{i}}\left(\rho'\left(i\right)\right)\right)\\ &= e\left(g^{a\lambda_{i}}H_{1}^{-x}\left(\rho'\left(i\right)\right),g^{\left(t+t'\right)w_{i}}\cdot g^{sv_{i}}\right)\cdot e\left(g^{x},H_{1}^{\left(t+t'\right)w_{i}}\left(\rho'\left(i\right)\right)H_{1}^{sv_{i}}\left(\rho'\left(i\right)\right)\right)\\ &= e\left(g,g\right)^{a\lambda_{i}\left(\left(t+t'\right)w_{i}+sv_{i}\right)}. \end{aligned} \tag{22}$$

Then,

$$\begin{aligned} \prod_{i=1}^{l_{f'}}\left(e\left(y^{\lambda_{i}}H_{1}^{-x}\left(\rho'\left(i\right)\right),\sigma_{0,i}\right)\cdot e\left(g^{x},\sigma_{1,i}\right)\right) &= \prod_{i=1}^{l_{f'}}e\left(g,g\right)^{a\lambda_{i}\left(\left(t+t'\right)w_{i}+sv_{i}\right)}\\ &= e\left(g,g\right)^{a\left(t+t'\right)\sum_{i\in I}\lambda_{i}w_{i}}\cdot e\left(g,g\right)^{as\sum_{i=1}^{l_{f'}}\lambda_{i}v_{i}}\\ &= e\left(g,g\right)^{xa\left(\left(t+t'\right)}. \end{aligned} \tag{23}$$

Therefore,

$$\frac{e\left(g^{x},\sigma_{0}\right)}{e\left(H_{M}^{x},\sigma_{0}'\right)\cdot\prod_{i=1}^{l_{f'}}\left(e\left(y^{\lambda_{i}}H_{1}^{-x}\left(\rho'\left(i\right)\right),\sigma_{0,i}\right)\cdot e\left(g^{x},\sigma_{1,i}\right)\right)} = \frac{e\left(g,g\right)^{x\alpha}\cdot e\left(g,g\right)^{xa\left(t+t'\right)}\cdot e\left(g,H_{M}\right)^{xr}}{e\left(g,H_{M}\right)^{xr}\cdot e\left(g,g\right)^{xa\left(t+t'\right)}} = Z^{x}. \tag{24}$$

The verification equation is established, so the proposed scheme is correct. □ □

Theorem 2 (unforgeability). *If the CDHE problem is difficult, then the proposed scheme above is existentially unforgeable under adaptive chosen message but selective access structure attack.*

Proof. Let $\mathscr{A}$ be an adversary against our scheme, and let $\mathscr{C}$ be a challenger who generates a random instance of the CDHE problem. We construct an adversary $\mathscr{B}$, which uses $\mathscr{A}$ as a subroutine, to solve the CDHE problem as follows. Here, $\mathscr{B}$ acts as a challenger of our scheme for $\mathscr{A}$ as well. Let $\widehat{k}$ be the maximum number of columns of the access structure matrixes and let q_s be the maximum number of **Sign-Oracle** queries.

(i) **CDHE Problem Gen**. $\mathscr{C}$ generates a random instance of the CDHE problem $(p, \mathbb{G}, \mathbb{G}_T, e, g, \{g_i = g^{a^i}\}_{i=1,i\neq\widehat{k}+1}^{\widehat{2k}})$ and sends it to $\mathscr{B}$.

(ii) **Init**. $\mathscr{A}$ sends a challenge access structure $f^* = (\mathbf{M}_{l^*\times k^*}^*, \rho^*)$ to $\mathscr{B}$. $\mathscr{B}$ extends f^* to $f^{*'} = (\mathbf{M}_{l_{\prime}^*\times k_{\prime}^*}^*, \rho\prime^*)$ by running **AccStruExpan**.

(iii) **Setup**. Choose $\alpha' \in_R \mathbb{Z}_p$ and set $y = g^a, Z = e(g^a, g^{a^{\widehat{k}}})e(g,g)^{\alpha'} = e(g,g)^{a^{\widehat{k}+1}+\alpha'} = e(g,g)^{\alpha}$, where $\alpha = a^{\widehat{k}+1} + \alpha'$. Let $l_M = 4q_s, k_M \in_R [k]$, choose $\phi_0, \phi_1, \ldots, \phi_k \in \mathbb{Z}_{l_M}$ and $\pi_0, \pi_1, \ldots, \pi_k \in \mathbb{Z}_p$, and set $u_0 = (g^a)^{p-l_M k_M+\pi_0} \cdot g^{\phi_0}$ and $u_i = (g^a)^{\pi_i} \cdot g^{\phi_i}$, $i = 1, \ldots, k$. Forward $\mathrm{PP} = (p, g, \mathbb{G}, \mathbb{G}_T, e, y, Z, u_0, \{u_i\}_{i=1}^{k})$ to $\mathscr{A}$.

(iv) **$\mathbf{H}_1$-Oracle**. $\mathscr{A}$ sends an attribute u to $\mathscr{B}$.$\mathscr{B}$ chooses $z_u \in_R \mathbb{Z}_p$. If there exists i such that $\rho\prime^* (i) = u$, then let

$$h_u = g^{z_u} g^{aM^*_{i,1}} g^{a^2 M^*_{i,2}} \dots g^{a^{k'^*} M^*_{i,k'^*}}. \tag{25}$$

Otherwise, let $h_u = g^{z_u}$.Return h_u to $\mathscr{A}$.

(v) **$\mathbf{H}_2$-Oracle**. $\mathscr{A}$ sends access structure f and a message M to $\mathscr{B}$. $\mathscr{B}$ chooses $h \in_R \{0,1\}^k$ and returns it to $\mathscr{A}$.

(vi) **KeyGen-Oracle**. $\mathscr{A}$ sends an attribute set A to $\mathscr{B}$. $\mathscr{B}$ extends A to A' by running **UserAttExpan**. Finds $\mathbf{v} = (v_1 = -1, v_2, \dots, v_{k\prime^*})$ such that $\mathbf{v}\mathbf{M}_i^* = 0$ for all $i \in [1, l'^*]$: $\rho'^* (i) \in A'$.Choose $r \in \mathbb{Z}_p$ and set

$$\begin{aligned} L &= g^r \prod_{i=1}^{k'^*} \left(g^{\widehat{a^{k-i+1}}} \right)^{v_i} = g^t, \\ D &= g^{\alpha'} g^{ar} \cdot \prod_{i=2}^{k'^*} \left(g^{\widehat{a^{k-i+2}}} \right)^{v_i} = g^{\alpha} \cdot y^t, \end{aligned} \tag{26}$$

where $t = r + \sum_{i=1}^{k'} v_i \widehat{a^{k-i+1}}$. For all $u \in A'$, request **$\mathbf{H}_1$-Oracle** on u. If there exists i such that $\rho\prime^* (i) = u$, then let

$$D_u = L^{z_u} \cdot \prod_{j=1}^{k'^*} \left(g^{a^j \cdot r} \prod_{d \neq j} \left(g^{\widehat{a^{k+1+j-d}}} \right)^{v_d} \right)^{\mathbf{M}^*_{i,j}} = H_1 (u)^t. \tag{27}$$

Otherwise, let

$$D_u = L^{z_u} = \left(g^t \right)^{z_u} = H_1 (u)^t. \tag{28}$$

Return $\mathrm{sk}_{A'} = (D, L, \{D_u\}_{u \in A'})$.

(v) **Sign-Oracle**. $\mathscr{A}$ sends a message M and an access structure f to $\mathscr{B}$.$\mathscr{B}$ extends f to f' by running **AccStruExpan**. Finds $\{w_i \colon i \in I\}$ such that $\sum_{i \in I} w_i \mathbf{M}_{f'i} = 1$, where $I = \left\{ i \in [l_{f'}] \colon \rho' (i) \in A' \right\}$, $\mathbf{1} = (1, 0, \dots, 0)$. Set $w_i = 0$ for $i \in [l_{f'}] \backslash I$.Request **$\mathbf{H}_2$ Oracle** on $f \| M$ and get $h = H_2 (f \| M) = h_1 h_2 \dots h_k$. Compute

$$\begin{aligned} F (M) &= p - l_M k_M + \pi_0 + \sum_{j=1}^{k} \pi_j \cdot h_j, \\ J (M) &= \phi_0 + \sum_{j=1}^{k} \phi_j \cdot h_j, \\ K (M) &= \begin{cases} 0, & \text{if } \pi_0 + \sum_{j=1}^{k} \pi_j \cdot h_j \equiv 0 \pmod{l_M} \\ 1, & \text{otherwise} \end{cases} \end{aligned} \tag{29}$$

If $K (M) = 0$, the simulation stops. If $K (M) \neq 0$, then $F (M) \neq 0 \pmod{p}$, because we can assume $l_M (k + 1) < p$ for any reasonable values of p, k and l_M. Choose $t, s \in_R \mathbb{Z}_p$, and $\{v_i\}_{i=1,\dots,l'_f}$ such that $\sum_{i=1}^{l'_f} v_i \cdot \mathbf{M}_i = 0$, and then compute

$$\begin{aligned} \sigma_0 &= (g^a)^{t+\alpha'} \cdot g^{\alpha'} \cdot \left(g^{\widehat{a^k}} \right)^{-J(M)/F(M)} = g^{\alpha} \cdot y^t \cdot \left(u_0 \prod_{i=1}^{k} (u_i)^{h_i} \right)^{-\widehat{a^k}/F(M)} \\ \sigma_0' &= \left(g^{\widehat{a^k}} \right)^{-1/F(M)} = g^{-\widehat{a^k}/F(M)} \\ \sigma_{0,i} &= g^{w_i t + v_i s}, i \in \left[l_{f'} \right], \\ \sigma_{1,i} &= H_1 (\rho (i))^{w_i t + v_i s}, i \in \left[l_{f'} \right], \end{aligned} \tag{30}$$

where $r' = -\widehat{a^k}/F(M)$.Return the signature $\sigma = (\sigma_0, \sigma_0', \{\sigma_{0,i}, \sigma_{1,i}\}_{i=1,\dots,l'_f})$.

(vi) **Output**: $\mathscr{A}$ outputs a triple (σ^*, M^*, f^*). If $F (M^*) \neq 0$, then the simulation stops. Otherwise, $\mathscr{B}$ computes and outputs

$$\sigma_0^* \cdot \frac{\prod_{i=1}^{l'^*} \left(\sigma_{0,i}^* \right)^{z_{\rho'^*(i)}}}{\left(\prod_{i=1}^{l'^*} \sigma_{1,i}^* \right) \cdot \left(\sigma_0'^* \right)^{J(M^*)} \cdot g^{\alpha\prime}} = g^{\widehat{a^{k+1}}}. \tag{31}$$

The calculation of the above equation is as follows:

$$\begin{aligned}\frac{\sigma_0^* \cdot \prod_{i=1}^{l_{\prime}^*}\left(\sigma_{0,i}^*\right)^{z_{\rho_{\prime}^*(i)}}}{\left(\prod_{i=1}^{l_{\prime}^*}\sigma_{1,i}^*\right)\cdot\left(\sigma_0^{'*}\right)^{J(M^*)}\cdot g^{\alpha'}} &= \frac{g^{\alpha}\cdot y^{t^*}\cdot H_M^{r^*}\cdot\left(\prod_{i=1}^{l_{\prime}^*}g^{w_i^*t^*+v_i^*s^*}\right)^{z_{\rho_{\prime}^*(i)}}}{\left(\prod_{i=1}^{l_{\prime}^*}H_1\left(\rho\prime^*(i)\right)^{w_i^*t^*+v_i^*s^*}\right)\cdot\left(g^{r^*}\right)^{J(M^*)}\cdot g^{\alpha_{\prime}}},\\ &= \frac{g^{\widehat{a^{k+1}}+\alpha_{\prime}}\cdot g^{at^*}\cdot\left(g^{r^*}\right)^{aF(M^*)+J(M^*)}\cdot\left(\prod_{i=1}^{l_{\prime}^*}g^{w_i^*t^*+v_i^*s^*}\right)^{z_{\rho_{\prime}^*(i)}}}{g^{at^*}\cdot\left(\prod_{i=1}^{l_{\prime}^*}g^{w_i^*t^*+v_i^*s^*}\right)^{z_{\rho_{\prime}^*(i)}}\cdot\left(g^{r^*}\right)^{J(M^*)}\cdot g^{\alpha_{\prime}}},\\ &= g^{\widehat{a^{k+1}}}.\ (\text{Note that } F(M^*)=0.).\end{aligned} \tag{32}$$

According to Claim 2 of Waters [42], the probability that the simulation is not aborted is $1/8q_s(k+1)$. Therefore, if $\mathscr{A}$ can successfully forge a valid signature with probability ε, then $\mathscr{B}$ can solve the CDHE problem with probability $\varepsilon' \geq \varepsilon/8q_s(k+1)$. □

Theorem 3 (privacy). *The proposed scheme achieves privacy.*

Proof. The adversary $\mathscr{A}$ and the challenger $\mathscr{C}$ perform the following interactive game:

(i) $\mathscr{C}$ executes the **Setup** algorithm to set up the system and responds to the oracle requests by running the corresponding algorithm.

(ii) $\mathscr{A}$ chooses and sends (M, f, A_0, A_1) such that $f(A_0) = f(A_1) = 1$ to $\mathscr{C}$.

(iii) $\mathscr{C}$ chooses $b \in_R \{0,1\}$, runs **Sign** with inputs $(\mathrm{SK}_{A_b}, M, f)$ to generate σ_b, and returns it to $\mathscr{A}$.

(iv) $\mathscr{C}$ continues to respond to the oracle requests by running the corresponding algorithm.

Since $\sigma_b = (\sigma_{0b}, \sigma_{0b}', \{\sigma_{0b,i}, \sigma_{1b,i}\}_{i=1}^{l_f'})$ is a signature on (M, f) using SK_{A_b}, we have

$$\begin{aligned}\sigma_{0b} &= D_b' \cdot H_M^{r_b}\\ \sigma_{0b}' &= g^{r_b},\\ \sigma_{0b,i} &= L_b^{'w_{bi}}\cdot g^{s_b v_{bi}} = g^{(t_b+t_b')w_{bi}+s_b v_{bi}}, i \in \left[l_{f'}\right],\\ \sigma_{1b,i} &= D_{b\rho'(i)}^{'w_{bi}}\cdot H_1(\rho'(i))^{s_b v_{bi}} = H_1(\rho'(i))^{(t_b+t_b')w_{bi}+s_b v_{bi}}, i \in \left[l_{f'}\right],\end{aligned} \tag{33}$$

where $\mathbf{w}_b = (w_{b1}, w_{b2}, \ldots, w_{bl_{f'}})$ and $\mathbf{v}_b = (v_{b1}, v_{b2}, \ldots, v_{bl_{f'}})$ such that

$$\begin{aligned}\mathbf{w}_b\mathbf{M}_{f'} &= \sum_{i=1}^{l_{f'}} w_{bi}\mathbf{M}_{f'i} = \mathbf{1},\\ \mathbf{v}_b\mathbf{M}_{f'} &= \sum_{i=1}^{l_{f'}} v_{bi}\mathbf{M}_{f'i} = \mathbf{0}.\end{aligned} \tag{34}$$

Let $I_{\bar{b}} = \{i \in [l_{f'}]: \rho'(i) \in A_{\bar{b}}'\}$, and find $\{w_{\bar{b}i}: i \in I_{\bar{b}}\}$ such that $\sum_{i\in I_{\bar{b}}} w_{\bar{b}i}\mathbf{M}_{f'i} = \mathbf{1}$, where $\mathbf{1} = (1, 0, \ldots, 0)$. Set $w_{\bar{b}i} = 0$ for $i \in [l_{f'}] \setminus I_{\bar{b}}$.

Let $\mathbf{w}_{\bar{b}} = (w_{\bar{b}1}, w_{\bar{b}2}, \ldots, w_{\bar{b}l_{f'}})$ and $\mathbf{v}_{\bar{b}} = s_{\bar{b}}^{-1}(t_b + t_b')(\mathbf{w}_b - \mathbf{w}_{\bar{b}}) + \mathbf{v}_b$; then,

$$\mathbf{v}_{\bar{b}}\mathbf{M}_{f'} = s_{\bar{b}}^{-1}(t_b + t_b')\left(\mathbf{w}_b\mathbf{M}_{f'} - \mathbf{w}_{\bar{b}}\mathbf{M}_{f'}\right) + \mathbf{v}_b\mathbf{M}_{f'} = \mathbf{0}. \tag{35}$$

Now, we rewrite $\sigma_b = (\sigma_{0b}, \sigma_{0b}', \{\sigma_{0b,i}, \sigma_{1b,i}\}_{i=1}^{l_f'})$ as

$$\begin{aligned}\sigma_{0b} &= g^{\alpha}\cdot g^{a(t_b+t_b')}\cdot H_M^{r_b}\\ &= g^{\alpha}\cdot g^{a\left(t_{\bar{b}}+\left(t_b'-t_{\bar{b}}\right)\right)}\cdot H_M^{r_b}\\ &= g^{\alpha}\cdot g^{a\left(t_{\bar{b}}+t_{\bar{b}}'\right)}\cdot H_M^{r_b}\\ &= D_{\bar{b}}'\cdot H_M^{r_b},\\ \sigma_{0b}' &= g^{r_b},\\ \sigma_{0b,i} &= g^{(t_b+t_b')w_{bi}+s_b v_{bi}}\\ &= g^{\left(t_{\bar{b}}+t_{\bar{b}}'\right)w_{\bar{b}i}+s_b v_{\bar{b}i}}\\ &= L_{\bar{b}}^{'w_{\bar{b}i}}\cdot g^{s_b v_{\bar{b}i}}, i \in \left[l_{f'}\right]\\ \sigma_{1b,i} &= H_1(\rho'(i))^{(t_b+t_b')w_{bi}+s_b v_{bi}}\\ &= H_1(\rho'(i))^{\left(t_{\bar{b}}+t_{\bar{b}}'\right)w_{\bar{b}i}+s_b v_{\bar{b}i}}\\ &= D_{\bar{b}\rho'(i)}^{'w_{\bar{b}i}}\cdot H_1(\rho'(i))^{s_b v_{\bar{b}i}}, i \in \left[l_{f'}\right],\end{aligned} \tag{36}$$

where $t'_{\overline{b}} = t'_b - t_{\overline{b}}$.

This concludes that σ_b is also a signature using $\mathrm{SK}_{A_{\overline{b}}}$. Therefore, even if the adversary has an unlimited capability, it is impossible to distinguish which attribute set was used to generate the signature, and the advantage of $\mathscr{A}$ of winning GAME 2 is 0. The proposed scheme achieves privacy. □

7. Performance Analysis and Experimental Simulation

7.1. Performance Analysis. We compare our method with the trivial method in terms of data size and computational overhead. Without loss of generality, we assume that the attribute universe $\mathscr{U} = \{\widetilde{u}_1, \ldots, \widetilde{u}_{n_1}, u_{n_1+1}, \ldots, u_{n_1+n_2}\}$, where $\widetilde{u}_i$ is a comparative attribute and its value space is $[N_i]$; $\ddot{a}_i$ takes the number near the median of the value space; the access structure f contains attributes $\{\widetilde{u}_1, \ldots, \widetilde{u}_{l_1}, u_{n_1+1}, \ldots, u_{n_1+l_2}\}$ and each attribute only appears once; the user's attribute set $A = \{\ddot{u}_1 \| \ddot{b}_1, \ldots, \ddot{u}_{t_1} \| \ddot{b}_{t_1}, u_{n_1+1}, \ldots, u_{n_1+t_2}\}$. Let k be the length of the output of the Hash function, and let $\mathbb{G}$ and $\mathbb{Z}_p$ be an element of $\mathbb{G}$ and $\mathbb{Z}_p$, respectively. Denote by P a pairing operation and by E an exponentiation operation, respectively.

The comparison is carried out on five schemes, and the results are shown in Tables 3–10. In these tables, the left column shows the theoretical calculation results, and the right column shows the results in the case of the above example.

- $n'_1 = \sum_{i=1}^{n_1} \lceil N_i/2 \rceil, n''_1 = \sum_{i=1}^{n_1} \log_2 \lceil N_i/2 \rceil$.
- $n'_1 = \sum_{i=1}^{n_1} \lceil N_i/2 \rceil, n''_1 = \sum_{i=1}^{n_1} \log_2 \lceil N_i/2 \rceil$.
- $t''_1 = \sum_{i=1}^{n_1} \log_2 \lceil N_i/2 \rceil$.
- $l'_1 = \sum_{i=1}^{l_1} \lceil N_i/2 \rceil, l''_1 = \sum_{i=1}^{l_1} \lceil \log_2 (N_i/2) \rceil$.
- $n' = \sum_{i=1}^{n_1} \lceil N_i/2 \rceil + n_2, n'' = \sum_{i=1}^{n_1} \log_2 \lceil N_i/2 \rceil + n_2$.
- $t' = \sum_{i=1}^{t_1} \lceil N_i/2 \rceil + t_2, t'' = \sum_{i=1}^{t_1} \log_2 \lceil N_i/2 \rceil + t_2$.

The analysis shows the following:

(1) Our method has significant advantages in signature size, signature generation overhead, and signature verification overhead, which is less than 10% of the trivial method.

(2) There is also a significant advantage in master public key size. Except for [11] which is about half, the others are less than 10%.

(3) There are also advantages in the overhead of master public key generation, especially when applied to the schemes of [6, 9] and [22].

(4) The sizes of the master secret keys are the same except for [22]. For [22], our method can reduce the size of the master private key to 3.47%.

(5) In terms of signing key size and signing key generation overhead, our method is not as good as the trivial method, which is about three times theirs.

Generally, the system only needs to generate a signing key once for each user, and the user only needs one signing key. However, one user may sign multiple messages to generate multiple signatures, and a signature may be verified multiple times by different users. Our method has advantages in the case of "multiple," while the disadvantages are "one-time." Therefore, our method has significant advantages.

7.2. Experimental Simulation. Two main factors are affecting the system data size and computing overhead. One is the number of attributes, especially comparative attributes. The second is the size of the value space of the comparative attribute. We simulate and analyze these two cases, respectively. The simulation experiment is carried out on the windows 10 subsystem Debian-10.9, 11th Gen Intel(R) Core(TM) i7-11370H @3.30 GHz, CPU × 4, RAM 16 GB, and the PBC-0.5.14 library. We use "a.param" as the incoming parameter file of the PBC library.

The first scenario we simulated assumes that the access structure has l_1 comparative attributes and l_2 usual attributes, and the user has t_1 comparative attributes and t_2 usual attributes. (l_1, l_2, t_1, t_2) take five groups of data, which are $(3, 1, 2, 1)$, $(6, 2, 4, 2)$, $(9, 3, 6, 3)$, $(12, 4, 8, 4)$, and $(16, 6, 10, 6)$. The results are shown in Figures 2–5.

The simulation results show that our scheme has significant advantages in data size and computational overhead. The details are as follows:

(1) In terms of master public key size, using our method, the average sizes of the five schemes are 91, 51, 163, 794, and 1755 Kb, respectively. Using the trivial method, the average size of the five schemes increases to 2260, 1135, 1292, 19232, and 44054 Kb. In other words, our method reduces the master public key sizes to 4.03%, 4.49%, 12.61%, 4.13%, and 3.98%. See Figure 2.

(2) In terms of master secret key size, using our method, the average sizes of the five schemes are 72.4, 71.4, 70.4, 36.2, and 34.2 Kb, respectively. Using the trivial method, the average size of the five schemes increases to 1130, 1129 1128, 565, and 563 Kb. That is, our method reduces the master public key sizes to 6.41%, 6.32%, 6.24%, 6.41%, and 6.07%. See Figure 3. Because the data of the [6, 7] and [11] schemes are very close, and the data of the [9, 22] schemes are also very close, it seems that there are only 2 schemes and 4 polylines in Figure 3. In fact, there are 5 schemes and 10 polylines.

(3) The cost of signature generation for the 5 schemes grows linearly with the number of attributes, whether using our method or the trivial method. But the increase is much faster with the trivial method, and the 5 dashed lines representing the trivial method are all above the 5 solid lines representing our method. The signing overhead using our method is 0.10, 0.05, 0.08, 0.07, and 0.22 seconds on average, which are about 6% of the trivial method. See Figure 4.

(4) Similarly, the overhead of signature verification can be reduced to about 6% of the trivial method with

Table 3: Comparison of master public key sizes.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$(2n_1' + 2n_2 + 1)\lvert\mathbb{G}\rvert$	$289\lvert\mathbb{G}\rvert$	$(2n_1'' + 2n_2 + 1)\lvert\mathbb{G}\rvert$	$11\lvert\mathbb{G}\rvert$	3.81%
Gu et al. [7]	$(n_1' + n_2 + 6)\lvert\mathbb{G}\rvert$	$150\lvert\mathbb{G}\rvert$	$(n_1'' + n_2 + 6)\lvert\mathbb{G}\rvert$	$11\lvert\mathbb{G}\rvert$	7.33%
Chen et al. [11]	$(n_1' + n_2 + k + 3)\lvert\mathbb{G}\rvert$	$307\lvert\mathbb{G}\rvert$	$(k + 3)\lvert\mathbb{G}\rvert$	$163\lvert\mathbb{G}\rvert$	53.09%
Okamoto and Takashima [9]	$(17n_1' + 17n_2 + 29)\lvert\mathbb{G}\rvert$	$2477\lvert\mathbb{G}\rvert$	$(17n_1'' + 17n_2 + 29)\lvert\mathbb{G}\rvert$	$114\lvert\mathbb{G}\rvert$	4.60%
Okamoto and Takashima [22]	$(39n_1' + 39n_2)\lvert\mathbb{G}\rvert$	$5616\lvert\mathbb{G}\rvert$	$(39n_1'' + 39n_2)\lvert\mathbb{G}\rvert$	$195\lvert\mathbb{G}\rvert$	3.47%

Table 4: Comparison of master secret key sizes.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$3\lvert\mathbb{Z}_p\rvert$	$3\lvert\mathbb{Z}_p\rvert$	$3\lvert\mathbb{Z}_p\rvert$	$3\lvert\mathbb{Z}_p\rvert$	100%
Gu et al. [7]	$1\lvert\mathbb{Z}_p\rvert$	$1\lvert\mathbb{Z}_p\rvert$	$1\lvert\mathbb{Z}_p\rvert$	$1\lvert\mathbb{Z}_p\rvert$	100%
Chen et al. [11]	$1\lvert\mathbb{Z}_p\rvert$	$1\lvert\mathbb{Z}_p\rvert$	$1\lvert\mathbb{Z}_p\rvert$	$1\lvert\mathbb{Z}_p\rvert$	100%
Okamoto and Takashima [9]	$1\lvert\mathbb{G}\rvert$	$1\lvert\mathbb{G}\rvert$	$1\lvert\mathbb{G}\rvert$	$1\lvert\mathbb{G}\rvert$	100%
Okamoto and Takashima [22]	$(13n_1' + 13n_2)\lvert\mathbb{Z}_p\rvert$	$1872\lvert\mathbb{Z}_p\rvert$	$(13n_1'' + 13n_2)\lvert\mathbb{Z}_p\rvert$	$65\lvert\mathbb{Z}_p\rvert$	3.47%

Table 5: Comparison of signing key sizes.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$(t_1 + t_2 + 1)\lvert\mathbb{G}\rvert$	$4\lvert\mathbb{G}\rvert$	$(t_1'' + t_2 + 1)\lvert\mathbb{G}\rvert$	$14\lvert\mathbb{G}\rvert$	350%
Gu et al. [7]	$(t_1 + t_2 + 2)\lvert\mathbb{G}\rvert$	$5\lvert\mathbb{G}\rvert$	$(t_1'' + t_2 + 2)\lvert\mathbb{G}\rvert$	$15\lvert\mathbb{G}\rvert$	300%
Chen et al. [11]	$(t_1 + t_2 + 2)\lvert\mathbb{G}\rvert$	$5\lvert\mathbb{G}\rvert$	$(t_1'' + t_2 + 2)\lvert\mathbb{G}\rvert$	$15\lvert\mathbb{G}\rvert$	300%
Okamoto and Takashima [9]	$(t_1 + t_2 + 3)\lvert\mathbb{G}\rvert$	$6\lvert\mathbb{G}\rvert$	$(t_1'' + t_2 + 3)\lvert\mathbb{G}\rvert$	$16\lvert\mathbb{G}\rvert$	267%
Okamoto and Takashima [22]	$(13t_1 + 13t_2)\lvert\mathbb{G}\rvert$	$39\lvert\mathbb{G}\rvert$	$(13t_1'' + 13t_2)\lvert\mathbb{G}\rvert$	$169\lvert\mathbb{G}\rvert$	433%

Table 6: Comparison of signature sizes.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$2\,(l_1' + l_2 + 2)\lvert\mathbb{G}\rvert$	$230\lvert\mathbb{G}\rvert$	$2\,(l_1'' + l_2 + 2)\lvert\mathbb{G}\rvert$	$24\lvert\mathbb{G}\rvert$	10.43%
Gu et al. [7]	$(2l_1' + l_2 + 2)\lvert\mathbb{G}\rvert$	$227\lvert\mathbb{G}\rvert$	$(2l_1'' + l_2 + 2)\lvert\mathbb{G}\rvert$	$21\lvert\mathbb{G}\rvert$	9.25%
Chen et al. [11]	$2\,(l_1' + l_2 + 1)\lvert\mathbb{G}\rvert$	$228\lvert\mathbb{G}\rvert$	$2\,(l_1'' + l_2 + 1)\lvert\mathbb{G}\rvert$	$22\lvert\mathbb{G}\rvert$	9.65%
Okamoto and Takashima [9]	$(l_1' + l_2 + 2)\lvert\mathbb{G}\rvert$	$115\lvert\mathbb{G}\rvert$	$(l_1'' + l_2 + 2)\lvert\mathbb{G}\rvert$	$12\lvert\mathbb{G}\rvert$	10.43%
Okamoto and Takashima [22]	$(l_1' + l_2)\lvert\mathbb{G}\rvert$	$113\lvert\mathbb{G}\rvert$	$(l_1'' + l_2)\lvert\mathbb{G}\rvert$	$10\lvert\mathbb{G}\rvert$	8.85%

Table 7: Comparison of master public key generation overhead.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$(2n' + 1)E$	$73E$	$(2n'' + 1)E$	$11E$	15.07%
Gu et al. [7]	$1E$	$1E$	$1E$	$1E$	100%
Chen et al. [11]	$2E + 1P$	$2E + 1P$	$2E + 1P$	$2E + 1P$	100%
Okamoto and Takashima [9]	$(144n' + 162)E$	$144E$	$(144n'' + 162)E$	$5E$	3.47%
Okamoto and Takashima [22]	$8n'E$	$1152E$	$8n''E$	$40E$	3.47%

our method. Here, the advantage of the scheme [11] is obvious. The minimum overhead is 0.03 seconds, and the maximum overhead is only 0.09 seconds, which is only 1.28% of the trivial method. See Figure 5.

The second scenario we simulated uses the example above. The value space of comparative attributes $\widetilde{u}_2, \widetilde{u}_3$, and $\widetilde{u}_4$ takes five groups of data such as $(2^5, 2^7, 2^6)$, $(2^6, 2^8, 2^7)$, $(2^7, 2^9, 2^8)$, $(2^8, 2^{10}, 2^9)$, and $(2^9, 2^{11}, 2^{10})$. The results are shown in Figures 6–9. They show that the advantages of our method are more significant in the second case.

(1) When using our method, the signature size, signing overhead, and verification overhead all remain largely

Table 8: Comparison of signing key generation overhead.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$(t' + 1)E$	$4E$	$(t'' + 1)E$	$14E$	350.00%
Gu et al. [7]	$(t' + 2)E$	$5E$	$(t'' + 2)E$	$15E$	300.00%
Chen et al. [11]	$(t' + 3)E$	$6E$	$(t'' + 3)E$	$16E$	266.67%
Okamoto and Takashima [9]	$(4t' + 8)E$	$20E$	$(4t'' + 8)E$	$60E$	300.00%
Okamoto and Takashima [22]	$8t'E$	$24E$	$8t''E$	$192E$	800.00%

Table 9: Comparison of signature generation overhead.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$(4l_f + 2)E$	$678E$	$(4l_{f'} + 2)E$	$42E$	6.19%
Gu et al. [7]	$(2l_f + 5)E$	$343E$	$(2l_{f'} + 5)E$	$25E$	7.29%
Chen et al. [11]	$(2l_f + 2\|I\| + 3)E$	$347E$	$(2l_{f'} + 2\|I\| + 3)E$	$29E$	8.36%
Okamoto and Takashima [9]	$(3l_f + 3)E$	$510E$	$(3l_{f'} + 3)$	$33E$	6.47%
Okamoto and Takashima [22]	$9l_fE$	$1521E$	$9l_{f'}E$	$90E$	5.92%

Table 10: Comparison of signature verification overhead.

Schemes	Trivial method		Our method		Percentage
	Theoretical	Example	Theoretical	Example	
Maji et al. [6]	$3l_fE + (2l_f + 2)P$	$507E + 340P$	$3l_{f'}E + (2l_{f'} + 2)P$	$30E + 22P$	6.44%
Gu et al. [7]	$(l_f + 2)E + (2l_f + 4)P$	$171E + 342P$	$(l_{f'} + 2)E + (2l_{f'} + 4)P$	$12E + 24P$	7.01%
Chen et al. [11]	$(2l_f + 3)E + (2l_f + 2)P$	$341E + 340P$	$(2l_{f'} + 3)E + (2l_{f'} + 2)P$	$23E + 22P$	6.69%
Okamoto and Takashima [9]	$(4l_f + 6)E + (l_f + 3)P$	$682E + 172P$	$(4l_{f'} + 6)E + (l_{f'} + 3)P$	$46E + 13P$	7.45%
Okamoto and Takashima [22]	$(10l_f + 1)E + l_fP$	$1691E + 169P$	$(10l_{f'} + 1)E + l_{f'}P$	$101E + 10P$	5.93%

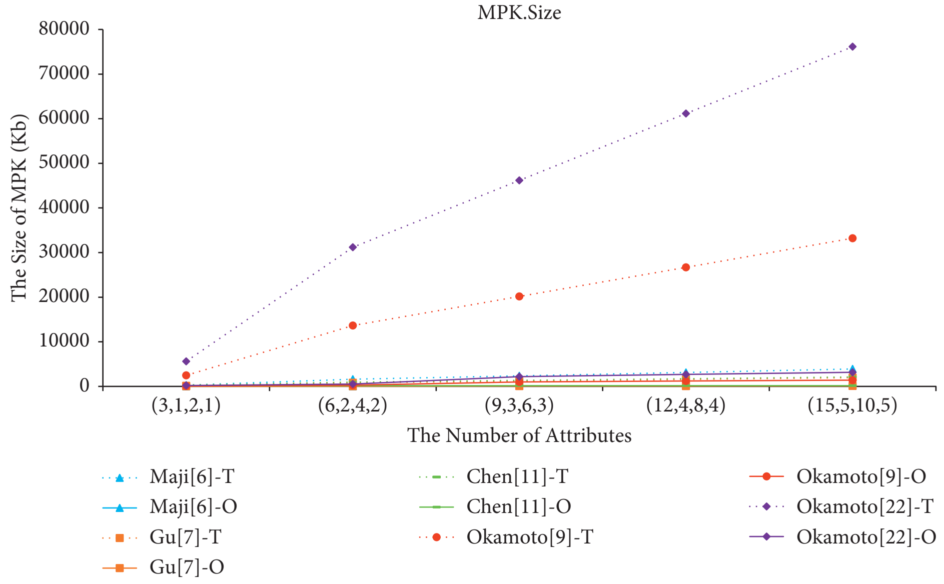

Figure 2: Comparison of master public key sizes on the attribute number.

unchanged. Instead, they both grow exponentially faster using the trivial method. In all three aspects, our method reduces them to within 2%.

(2) Our method also grows slower than the trivial method in terms of master public key size. It can be reduced to about 15% in all five schemes.

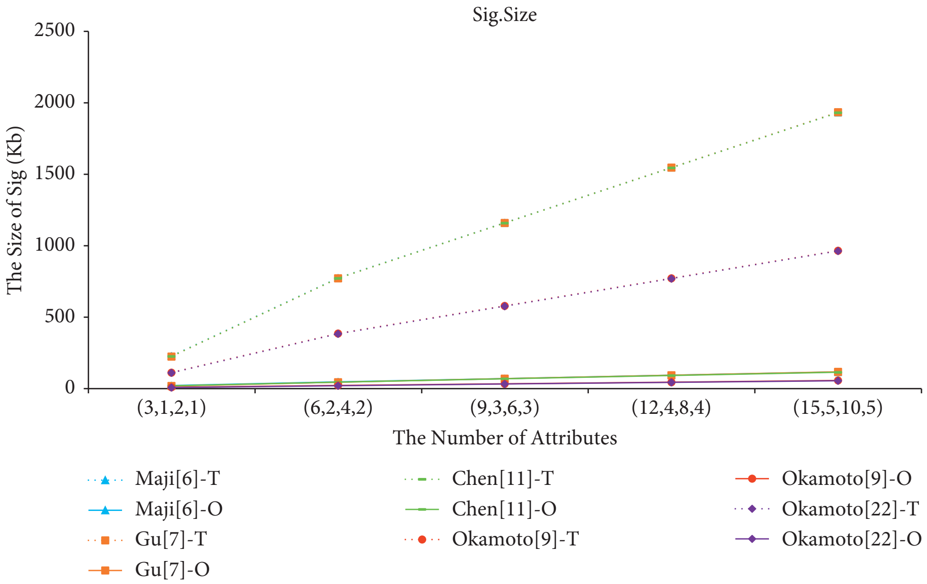

Figure 3: Comparison of signature sizes on the attribute number.

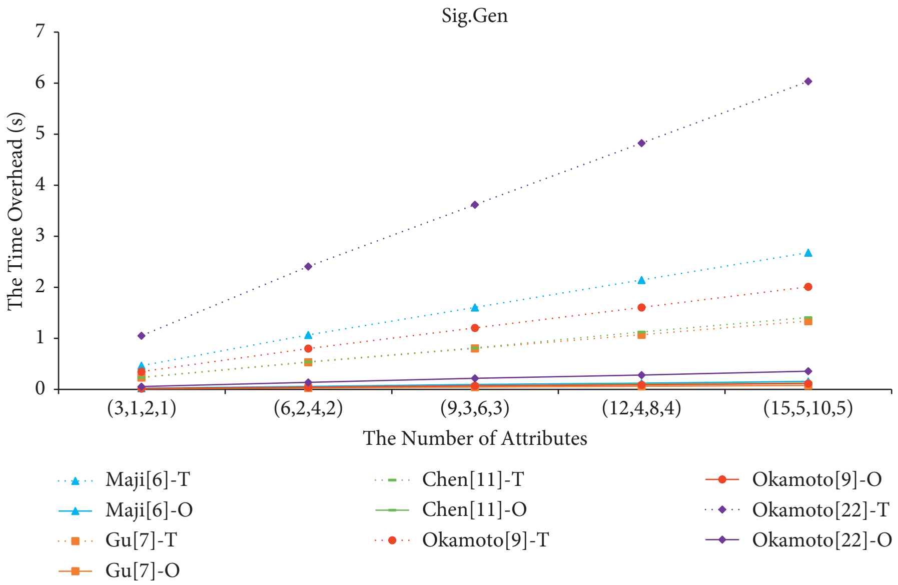

Figure 4: Comparison of signing overhead on the attribute number.

(3) Using our method, the average size of the master public key for the 5 schemes is 241, 126, 163, 2072, and 4687 Kb; the average size of the signature for the 5 schemes is 30, 27, 28, 15, and 13 Kb, respectively.

(4) Using our method, the average signature generation overhead for the 5 schemes is 0.04, 0.02, 0.03, 0.03, and 0.08 seconds; the average verification overhead for the 5 schemes is 0.56, 0.58, 0.04, 0.34, and 0.34 seconds, respectively.

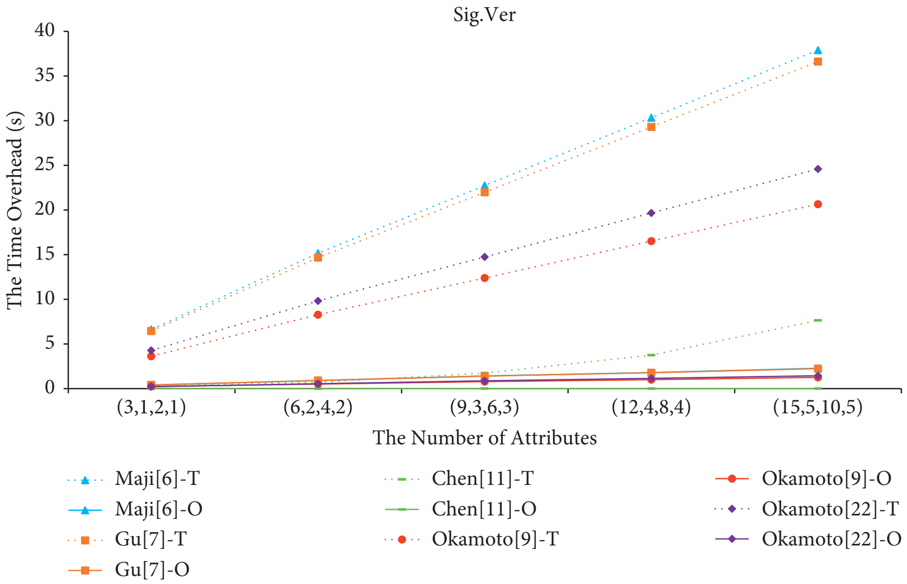

FIGURE 5: Comparison of verification overhead on the attribute number.

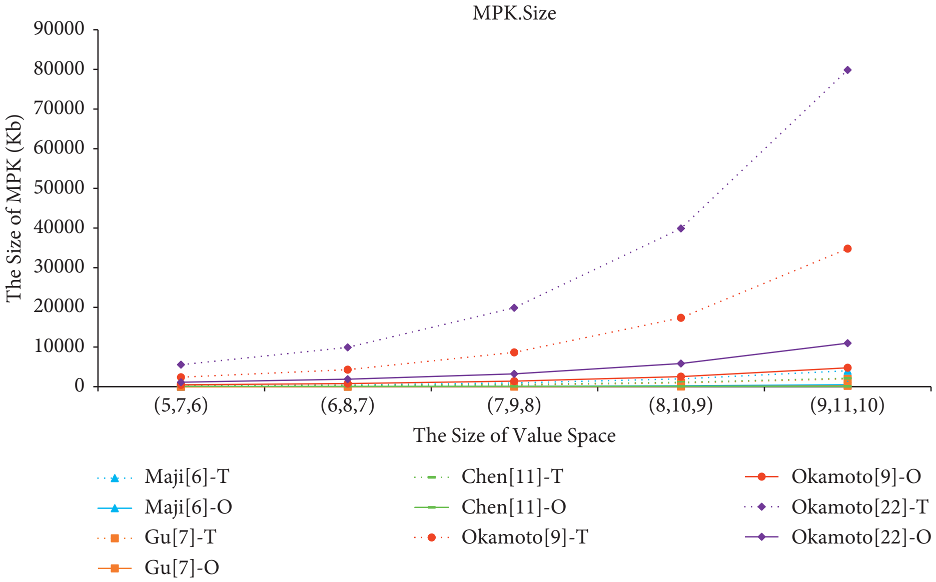

FIGURE 6: Comparison of master public key sizes on the value space.

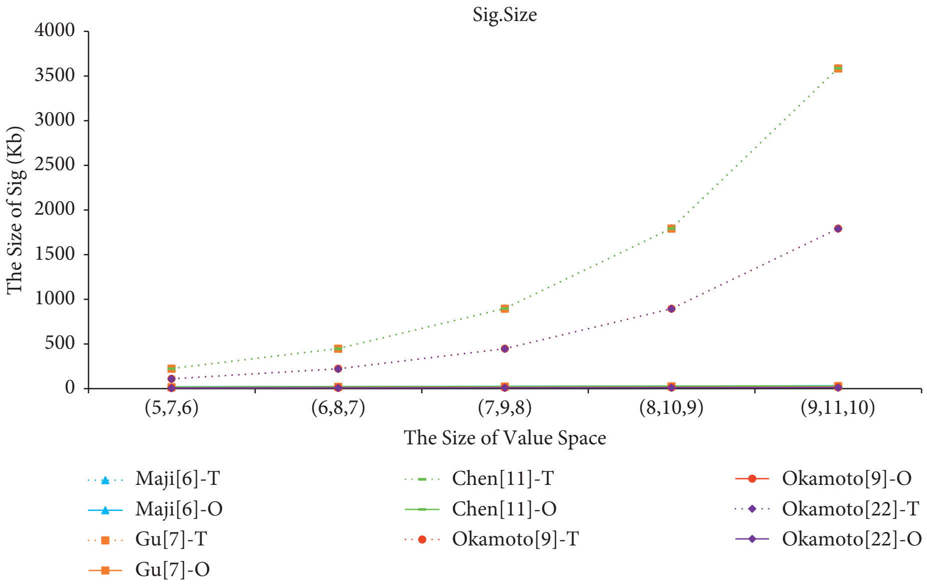

Figure 7: Comparison of signature sizes on the value space.

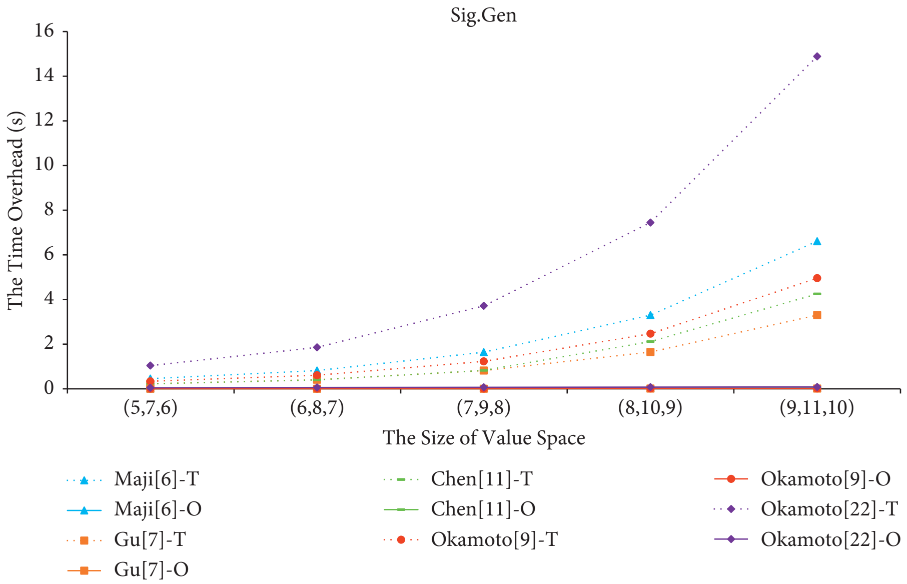

Figure 8: Comparison of signing overhead on the value space.

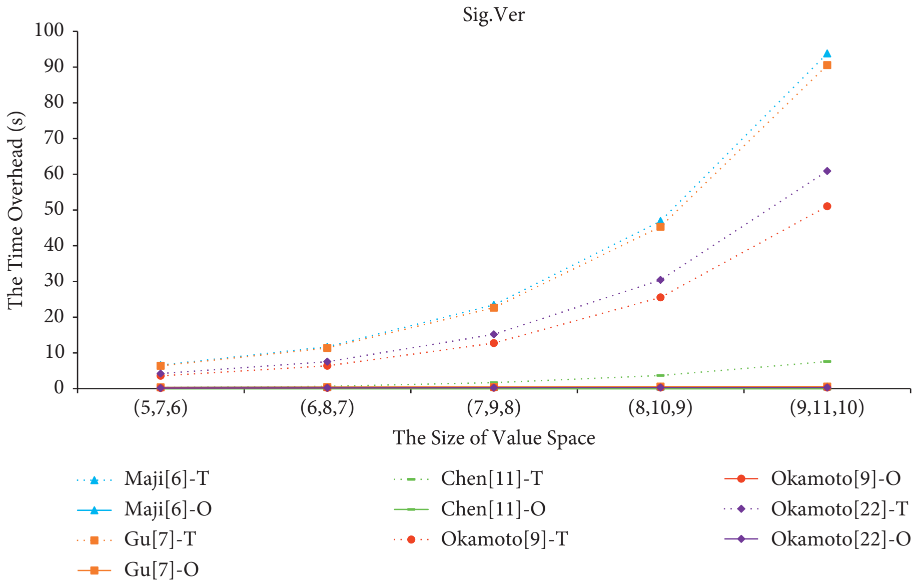

Figure 9: Comparison of verification overhead on the value space.

Overall, the simulation results show that our method has significant advantages over the trivial method and is then practicable and meaningful.

8. Conclusion

Attribute-based signature is a widely used cryptographic primitive and has been studied by many scholars. However, none of the existing attribute-based signature schemes supports comparing attributes. Using the 0-encoding and 1-encoding, we propose a signature-policy comparable attribute-based signature scheme, which effectively deals with the comparative attributes. Theoretical analysis and simulation experiments show that our method is practical and has significant advantages in storage and computation overhead compared with the trivial method. CABS may be combined with other technologies for wider application in the future [43–46].

The method proposed in this paper does not seem to be suitable for the case of key-policy, and it is one of the future research directions to propose a key-policy comparable attribute-based signature scheme.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province of China (Nos. 2019J01750, 2019J01752, and 2020J01814), Zhangzhou Natural Science Foundation of China (Nos. ZZ2019127 and ZZ2021J48), and Fujian Provincial Young and Middle-Aged Teacher Education Research Project (Science and Technology) (Nos. JAT201384 and JAT201387). The first author and the fourth author thank the Fujian Key Laboratory of Granular Computing and Application (Minnan Normal University) for its support of this work.

References

[1] H. Maji, M. Prabhakaran, and M. Rosulek, “Attribute-based signatures: achieving attribute-privacy and collusion-resistance,” *Iacr Cryptology Eprint Archive*, vol. 23, 2008.

[2] Li Jin and K. Kim, “Attribute-based ring signatures,” *IACR Cryptology ePrint Archive*, vol. 394, p. 01, 2008.

[3] Li Jin, M. Ho Au, W. Susilo, D. Xie, and K. Ren, “Attribute-based signature and its applications,” in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*Association for Computing Machinery. [Online]. Available:, New York, NY, USA, April 2010.

[4] S. F. Shahandashti and R. Safavi-Naini, “Threshold attribute-based signatures and their application to anonymous credential systems,” in *Progress in Cryptology - AFRICACRYPT 2009*, B. Preneel, Ed., vol. volume 5580, pp. 198–216, Springer Berlin Heidelberg, Berlin,Heidelberg, 2009.

[5] M. Gagné, S. Narayan, and R. Safavi-Naini, “Short pairing-efficient threshold-attribute-based signature,” in *Pairing-Based Cryptography - Pairing 2012 Pairing-Based Cryptography – Pairing 2012*, M. Abdalla and T. Lange, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 295–313, 2013.

[6] H. K. Maji, M. Prabhakaran, and M. Rosulek, “Attribute-based signatures,” in *Topics in Cryptology - CT-RSA 2011 Topics in Cryptology – CT-RSA 2011*, A. Kiayias, Ed., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 376–392, 2011.

[7] 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.

[8] T. Okamoto and K. Takashima, “Efficient attribute-based signatures for non-monotone predicates in the standard

model," in *Dario Catalano, Nelly Fazio, Rosario Gennaro and Antonio Nicolosi Public Key Cryptography – PKC 2011*Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[9] 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, Oct 2014.

[10] J. Herranz, F. Laguillaumie, B. Libert, and C. Ràfols, "Short attribute-based signatures for threshold predicates," in *Topics in Cryptology – CT-RSA 2012*, D. Orr, Ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[11] C. Chen, J. Chen, H. W. Lim, Z. Zhang, and D. Feng, "Combined public-key schemes: the case of ABE and ABS," in *Provable Security*, T. Takagi, G. Wang, Z. Qin, S. Jiang, and Y. Yu, Eds., vol. volume 7496, pp. 53–69, Springer Berlin Heidelberg, Berlin,Heidelberg, 2012.

[12] J. Su, D. Cao, B. Zhao, and X. Wang, "You: ePASS: an expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of Things," *Future Generation Computer Systems*, vol. 33, pp. 11–18, 2014.

[13] Q. Wang and S. Chen, "Attribute-based signature for threshold predicates from lattices," *Security and Communication Networks*, vol. 8, no. 5, pp. 811–821, 2015.

[14] A. Escala, J. Herranz, and P. Morillo, "Revocable attribute-based signatures with adaptive security in the standard model," in *Lecture Notes in Computer Science Progress in Cryptology – AFRICACRYPT 2011*, A. Nitaj and D. Pointcheval, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 224–241, 2011.

[15] F. Tang, H. Li, and B. Liang, "Attribute-based signatures for circuits from multilinear maps," in *Lecture Notes in Computer Science Information Security*, S. S. M. Chow, C. Jan, L. C. K. Hui, and M. Y. Siu, Eds., Springer International Publishing, Cham, pp. 54–71, 2014.

[16] Y. Sakai, N. Attrapadung, and G. Hanaoka, Chen-Mou Cheng, "Attribute-based signatures for circuits from bilinear map," in *Public-Key Cryptography - PKC 2016 Public-Key Cryptography – PKC 2016*, K.-M. Chung, G. Persiano, and Bo-Y. Yang, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 283–300, 2016.

[17] A. El Kaafarani and E. Ghadafi, "Attribute-based signatures with user-controlled linkability without random oracles," in *Cryptography and Coding Cryptography and Coding*, M. . O'Neill, Ed., Springer International Publishing, Cham, pp. 161–184, 2017.

[18] P. Datta, T. Okamoto, and K. Takashima, "Efficient attribute-based signatures for unbounded arithmetic branching programs," *IEICE - Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E104, 2021.

[19] D. Cao, B. Zhao, X. Wang, J. Su, and G. Ji, "Multi-authority attribute-based signature," in *Proceedings of the 2011 Third International Conference on Intelligent Networking and Collaborative Systems*, Fukuoka, Japan, November 2011.

[20] D. Cao, B. Zhao, X. Wang, and J. Su, "Flexible multi-authority attribute-based signature schemes for expressive policy," *Mobile Information Systems*, vol. 8, no. 3, pp. 255–274, 2012.

[21] A. El Kaafarani, L. Chen, E. Ghadafi, and J. Davenport, "Attribute-based signatures with user-controlled linkability," in *Cryptology and Network Security Dimitris Gritzalis, Aggelos Kiayias and Ioannis Askoxylakis Cryptology and Network Security*, pp. 256–269, Springer International Publishing, Cham, 2014.

[22] T. Okamoto and K. Takashima, "Decentralized attribute-based signatures," in *Kaoru Kurosawa and Goichiro Hanaoka Public-Key Cryptography – PKC 2013*vol. 125–142, Berlin, Heidelberg, Springer Berlin Heidelberg, 2013.

[23] A. El Kaafarani, E. Ghadafi, and D. Khader, "Decentralized traceable attribute-based signatures," in *Topics in Cryptology - CT-RSA 2014 Topics in Cryptology – CT-RSA 2014*, J. Benaloh, Ed., Springer International Publishing, Cham, pp. 327–348, 2014.

[24] E. Ghadafi, "Stronger security notions for decentralized traceable attribute-based signatures and more efficient constructions," in *Lecture Notes in Computer Science Topics in Cryptology — CT-RSA 2015*, K. Nyberg, Ed., Springer International Publishing, Cham, pp. 391–409, 2015.

[25] F. Chen, Y. Han, D. Jiang, X. Li, and X. Yang, "Outsourcing the unsigncryption of compact attribute-based signcryption for general circuits," in *Communications in Computer and Information Science Social Computing*, W. CheQ. Han et al., Eds., Springer Singapore, Singapore, pp. 533–545, 2016.

[26] Z. Huang, Z. Lin, Q. Chen, Y. Zhou, and H. Huang, "Outsourced attribute-based signatures with perfect privacy for circuits in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 10, Article ID e6173, 2021.

[27] Z. Huang and Z. Lin, "Secure server-aided attribute-based signature with perfect anonymity for cloud-assisted systems," *Journal of Information Security and Applications*, vol. 65, Article ID 103066, 2022.

[28] V. Kuchta, G. Sharma, R. A. Sahu, and O. Markowitch, "Generic framework for attribute-based group signature," in *Information Security Practice and Experience Information Security Practice and Experience*, J. K. Liu and P. Samarati, Eds., Springer International Publishing, Cham, pp. 814–834, 2017.

[29] Y. Song, Z. Li, Y. Li, and J. Li, "Attribute-based signcryption scheme based on linear codes," *Information Sciences*, vol. 417, pp. 301–309, 2017.

[30] C. Sun, Y. Guo, and Y. Li, "One secure attribute-based proxy signature," *Wireless Personal Communications*, vol. 103, no. 2, pp. 1273–1283, Nov 2018.

[31] C.-C. . Drǎgan, D. Gardham, and M. Manulis, "Hierarchical attribute-based signatures," in *Jan Camenisch and Panos Papadimitratos Cryptology and Network Security*, pp. 213–234, Springer International Publishing, Cham, 2018.

[32] Y. S. Rao and R. Dutta, "Bandwidth-efficient attribute-based key-insulated signatures with message recovery," *Information Sciences*, vol. 369, pp. 648–673, 2016.

[33] H. Cui, G. Wang, R. H. Deng, and B. Qin, "Escrow free attribute-based signature with self-revealability," *Escrow free attribute-based signature with self-revealability" Information Sciences*, Information Sciences, vol. 367-368, , pp. 660–672, 2016.

[34] Maharage Nisansala Sevwandi Perera, T. Nakamura, M. Hashimoto, H. Yokoyama, and K. Sakurai, "Almost fully anonymous attribute-based group signatures with verifier-local revocation and member registration from lattice assumptions," *Theoretical Computer Science*, vol. 891, pp. 131–148, 2021.

[35] Yu Chen, J. Li, C. Liu, J Han, Y Zhang, and P Yi, "Efficient attribute based server-aided verification signature," *IEEE Transactions on Services Computing*, vol. 1, 2021.

[36] F. Luo, S. Al-Kuwari, W. Susilo, and D. H. Duong, "Attribute-based proxy re-signature from standard lattices and its applications," *Computer Standards & Interfaces*, vol. 75, Article ID 103499, 2021.

[37] Y. Zhao, A. Ruan, G. Dan, J. Huang, and Yi Ding, "Efficient multi-authority attribute-based signcryption with constant-

size ciphertext," in *Proceedings of the 2021 IEEE Conference on Dependable and Secure Computing (DSC)*, Fukushima, Japan, January 2021.

[38] X. Yang, T. Li, X. Pei, L. Wen, and C. Wang, "Medical data sharing scheme based on attribute cryptosystem and blockchain technology," *IEEE Access*, vol. 8, pp. 45468–45476, 2020.

[39] H. Guo, W. Li, E. Meamari, C.-C. Shen, and M. Nejad, "Attribute-based multi-signature and encryption for ehr management: a blockchain-based solution," in *Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Toronto, ON, Canada, May 2020.

[40] X. Liu, W. Chen, Y. Xia, and R. Shen, "Trams: a secure vehicular crowdsensing scheme based on multi-authority attribute-based signature," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1–11, 2021.

[41] H.-Y. Lin and W.-G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," in *Applied Cryptography and Network Security John Ioannidis, Angelos Keromytis and Moti Yung Applied Cryptography and Network Security*, pp. 456–466, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[42] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in Cryptology – EUROCRYPT 2005*, R. Cramer, Ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[43] Li Qi, B. Xia, H. Huang, Y. Zhang, and T. Zhang, "Trac: traceable and revocable access control scheme for mhealth in 5g-enabled iiot," *IEEE Transactions on Industrial Informatics*, vol. 1, 2021.

[44] Y. Tian, Ta Li, and J. Xiong, "A blockchain-based machine learning framework for edge services in iiot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, March 2022.

[45] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in dwsns," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, Sep. 2020.

[46] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, June 2020.