Hindawi
Security and Communication Networks
Volume 2021, Article ID 7884007, 14 pages
https://doi.org/10.1155/2021/7884007

Research Article

Secure Outsourced Attribute-Based Signatures with Perfect Anonymity in the Standard Model

Zhenjie Huang,[1,2] **Runlong Duan**,[1,3] **Qunshan Chen**,[3] **Hui Huang**,[3] and **Yuping Zhou**[3]

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

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

Received 16 May 2021; Revised 7 September 2021; Accepted 11 September 2021; Published 16 October 2021

Academic Editor: Helena Rifà-Pous

Outsourced attribute-based signatures (OABS) enable users to sign messages without revealing specific identity information and are suitable for scenarios with limited computing power. Recently, Mo et al. proposed an expressive outsourced attribute-based signature scheme (Peer-to-Peer Networking and Applications, 11, 2017). In this paper, we show that Mo et al.'s scheme does not achieve any of the three security properties. Their scheme is incorrect. The adversary can collude with the malicious signing-cloud service provider (S-CSP) to forge valid signatures on any message and any attribute set. And the S-CSP could trace the access structures used to generate the signatures. Then, we treat the S-CSP as an adversary and present more accurate unforgeability and anonymity models for OABS to remedy the drawbacks of the previous ones. Finally, we propose a simple but significant improvement to fix our attacks. The improved scheme achieves correctness, unforgeability, and perfect anonymity while keeping the efficiency almost unchanged. We also prove the security of the improved scheme under the standard model.

1. Introduction

Attribute-based cryptography is a powerful cryptographic primitive, enabling us to design various cryptosystems with fine-grained access control in a multiuser environment [1, 2]. Attribute-based signature (ABS) is one of the leading research contents of attribute-based cryptography. ABS can provide fine-grained privacy protection for signers and finds applications in many fields, such as private access control, trust negotiations, and anonymous credentials [2, 3]. ABS may also be applied to mobile authentication and two-factor/multifactor authentication in the future [4–6]. Since it was introduced, numerous ABS schemes for different access structures have been proposed one after another [7–15].

However, with the continuous enhancement of the expressiveness of the access structure, the computational overhead of ABS is increasing, which makes it challenging to execute in devices with limited computing power. Using outsourcing technology of cloud computing, Chen et al. [16] introduced outsourced attribute-based signatures (OABS) to overcome this problem. In OABS, the signer can delegate most of his/her signing workload to a signing-cloud service provider (S-CSP). After receiving the semisignature from the S-CSP, the signer can generate the final signature by little computations. In this way, ABS can be used in resource-constrained devices.

1.1. Related Works. While introducing OABS, Chen et al. [16] proposed two concrete OABS schemes. Their schemes are signature-policy OABS schemes with threshold access structures. After, Mo et al. [17] proposed an OABS scheme and applied it to the medical cloud. Mo et al.'s scheme is a key-policy OABS scheme that supports a more expressive monotonic access structure. Sun et al. [18, 19] introduced decentralization into OABS and proposed an outsourcing decentralized multiauthority attribute-based signature scheme. Their scheme is a signature-policy scheme for threshold access structure. In 2021, Huang et al. [20] proposed a new key-policy OABS scheme for circuits. Their scheme is a short signature scheme, and its final signature has only one element of the group.

Chen et al.'s OABS model assumes that the S-CSP is honest-but-curious, i.e., the S-CSP always runs the algorithm honestly and outputs the semisignatures correctly, but the S-CSP may forge signatures. As a remedy for the overly strong assumption of S-CSP's honesty, Chen et al. [16] discussed the accountability of OABS, which provides an audit function for S-CSP's honesty. Liu et al. [21] studied OABS under the concept of server-assisted anonymous attribute authentication, added the correctness verification of the semisignature to OABS, and defined the outsourcing verifiability. After that, Ren and Jiang [22] formally introduced the concept of Verifiable Outsourced Attribute-Based Signatures (VOABS) with a concrete scheme supporting threshold access structure. Unfortunately, Uzunkol [23] presented two attacks on the verifiability of Ren et al.'s scheme. Moreover, one of the attacks enables the untrusted S-CSP to forge signatures.

In 2018, Cui et al. [24] introduced a new notion of Server-Aided Attribute-Based Signature (SA-ABS). SA-ABS outsources both signing tasks and verification tasks to cloud service providers, while OABS only outsources signing tasks. This is the main difference between the two. Cui et al. also proposed a signature-policy SA-ABS scheme for threshold access structure. But Hu et al. [25] pointed out that Cui et al.'s scheme [24] was forgeable and then proposed a new SA-ABS scheme for monotonic access structure.

Wang et al. studied the other side of ABS outsourcing and introduced Attribute-Based Server-Aided Verification Signature (ABSAVS) [26]. In ABSAVS, the signer outsources the verification workload to the server but does not outsource the signing workload. Wang et al. also proposed a ABSAVS scheme for threshold access structure. Recently, Chen et al. proposed a new ABSAVS scheme for tree access structure [27].

Previous schemes are summarized and compared in Table 1.

1.2. Contributions. The main contributions of this paper are as follows:

(i) We analyze the security of Mo et al.'s EOABS scheme [17] and show that it does not achieve any of the three security properties. The scheme is incorrect. The adversary can collude with the malicious S-CSP to forge valid signatures on any message and any attribute set. The S-CSP could trace the access structures used to generate the signatures.

(ii) We present more accurate security models for OABS. The main drawback of the previous security models is that the S-CSP's attacks are not considered, and our security models make up for it.

(iii) We propose a simple but significant improvement to fix our attacks. The improved scheme achieves correctness, unforgeability, and perfect anonymity while keeping the efficiency almost unchanged. We also prove its security under the standard model.

1.3. Organization. The rest of this paper is organized as follows. Section 2 presents preliminaries. Section 3 reviews Mo et al.'s EOABS scheme and analyzes its security. Section 4 presents a new definition and new security models for OABS. Section 5 proposes an improvement to fix our attacks with security proofs and performance analysis. Section 6 concludes this paper.

2. Preliminaries

Let $a \in_R A$ denote sampling a randomly from A. Let $[n] = \{1, 2, \ldots, n\}$ for $n \in \mathbb{N}$. For any vectors $\mathbf{v} = (v_1, v_2, \ldots, v_k) \in \mathbb{Z}_p^k$ and $\mathbf{w} = (w_1, w_2, \ldots, w_k) \in \mathbb{Z}_p^k$, their inner product $\mathbf{vw} = \sum_{i=1}^k v_i w_i$.

2.1. Bilinear Map. Let $\mathbb{G}$ and $\mathbb{G}_T$ be prime order p multiplication cyclic groups. Let $e\colon \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ be a map satisfying the following properties:

(i) For all $a, b \in_R \mathbb{Z}_p$ and $g_1, g_2 \in_R \mathbb{G}$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.

(ii) There exist $g_1, g_2 \in \mathbb{G}$ such that $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$.

(iii) For all $g_1, g_2 \in \mathbb{G}$, $e(g_1, g_2)$ can be computed efficiently.

2.1.1. Computational Diffie-Hellman Exponent (CDHE) Problem. Given $(g, g^a, g^{a^2}, \ldots, g^{a^n}, g^{a^{n+2}}, \ldots, g^{a^{2n}})$ to compute $g^{a^{n+1}}$, where $g \in \mathbb{G}$, $a \in_R \mathbb{Z}_p$ [28].

2.2. Linear Secret Sharing Scheme. Let $P = \{p_1, p_2, \ldots, p_n\}$ be a party set; a collection $\mathbb{A}$ of nonempty subsets of P is defined as an access structure. A set in $\mathbb{A}$ is an authorized set, and a set not in $\mathbb{A}$ is an unauthorized set. An access structure $\mathbb{A} \subseteq 2^P$ is monotone, if $B \in \mathbb{A}$ and $B \subseteq C$ implies $C \in \mathbb{A}$ for all B, C.

A linear secret sharing scheme (LSSS) for a monotone access structure $\mathbb{A}$ over $\mathbb{Z}_p$ is a matrix $\mathbf{M}_{l\times k}$ with a function $\pi(i)$ indicating the i th row of $\mathbf{M}$ as an attribute, and it satisfies the following properties:

(i) For any authorized set $A \in \mathbb{A}$, there are constants $\{w_i \in \mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I} w_i \mathbf{M}_i = (1, 0, \ldots, 0)$, where $I = \{i\colon \pi(i) \in A\}$, and $\mathbf{M}_i$ is the i th row of the matrix $\mathbf{M}$.

(ii) For any unauthorized set $B \notin \mathbb{A}$, there are no constants $\{w_i \in \mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I} w_i \mathbf{M}_i = (1, 0, \ldots, 0)$, where $I = \{i : \pi(i) \in B\}$.

The distribution and reconstruction algorithms of an LSSS are as follows:

(i) Distribution: it takes as inputs a matrix $\mathbf{M}_{l\times k}$ with a function $\pi(.)$ and a secret $s \in \mathbb{Z}_p$ to be shared. It chooses $r_2, r_3, \ldots, r_k \in_R \mathbb{Z}_p$, sets $\mathbf{v} = (s, r_2, r_3,$

Table 1: Previous schemes.

Scheme	Chen et al. [16]	Mo et al. [17]	Sun et al. [19]	Huang et al. [20]	Liu et al. [21]	Ren and Jiang [22]	Cui et al. [24]	Hu et al. [25]
Access structure	Threshold	LSSS	Threshold	Circuit	Threshold	Threshold	Threshold	LSSS
Type	SP	KP	SP	KP	SP	SP	SP	KP

*LSSS = Linear Secret Sharing Scheme, SP = signature-policy, KP = key-policy.

$\ldots, r_k) \in \mathbb{Z}_p^k$, and computes share set $\{\lambda_i : \lambda_i = \mathbf{M}_i\mathbf{v}_i\}$.

(ii) Reconstruction: it takes as inputs a matrix $\mathbf{M}_{l\times k}$ with a function $\pi(.)$ and an authorized set $A \in \mathbb{A}$ with its share set $\{\lambda_i\}_{i\in I}$. It finds constants $\{w_i \in \mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I} w_i\mathbf{M}_i = (1, 0, \ldots, 0)$ and then reconstructs the secret $s = \sum_{i\in I} w_i\lambda_i$.

Lemma 1 (see [29]). *Suppose that $\mathbb{A}$ is a monotone access structure with matrix $\mathbf{M}_{l\times k}$. For any unauthorized set $B \notin \mathbb{A}$, there is a vector $\mathbf{w} = (-1, w_2, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $\mathbf{M}_i\mathbf{w} = 0$ for all i: $\pi(i) \in B$.*

3. Cryptanalysis of Mo et al.'s EOABS Scheme

3.1. Review of Mo et al.'s EOABS Scheme. In this section, we review the EOABS scheme proposed by Mo et al. [17]. It comprises five algorithms and involves four entities: attribute authority (AA), S-CSP, signer, and verifier.

(i) Setup: Suppose U is the attribute universe, δ is the default attribute, and m is the maximal length of the message.

  (i) The AA chooses two prime order p cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ with a bilinear map e: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$.
  (ii) It selects a generator g of $\mathbb{G}$.
  (iii) It selects $a, b \in_R \mathbb{Z}_p$ and computes $Y = e(g, g)^{a+b}$.
  (iv) It samples $u_0, u_1, \cdots, u_m \in_R \mathbb{G}$.
  (v) It chooses $T_0 \in_R \mathbb{G}$ and $T_u \in_R \mathbb{G}$, $u \in U \bigcup \{\delta\}$.

The system public parameters:

$$PP = \left(g, p, \mathbb{G}, \mathbb{G}_T, Y, T_0, \{T_u\}_{u\in U\bigcup\{\delta\}}, u_0, u_1, \cdots, u_m\right), \tag{1}$$

the master secret key:

$$\mathrm{MSK} = (a, b). \tag{2}$$

(ii) KeyGen: it takes as inputs the master secret key MSK and an access structure $\mathbb{A}$ with its matrix $\mathbf{M}_{l\times k}$.

  (i) It chooses $v_2, \cdots, v_k \in_R \mathbb{Z}_p$, sets $\mathbf{v} = (a, v_2, \cdots, v_k)$, and computes $\lambda_i = \mathbf{M}_i\mathbf{v}, i \in [l]$.
  (ii) For each $i \in [l]$, it chooses $r_i \in_R \mathbb{Z}_p$ and computes

$$\begin{aligned} d_i &= g^{\lambda_i}\left(T_0 T_{\pi(i)}\right)^{r_i},\\ d_i' &= g^{r_i},\\ d_{iu}'' &= T_u^r,\\ & u \in U\backslash\pi(i). \end{aligned} \tag{3}$$

  (iii) It chooses $r_\delta \in_R \mathbb{Z}_p$ and then computes

$$\begin{aligned} d_\delta &= g^b\left(T_0 T_\delta\right)^{r_\delta},\\ d_\delta' &= g^{r_\delta}. \end{aligned} \tag{4}$$

The outsourced key:

$$\mathrm{OSK}_{\mathbb{A}} = \left(d_i, d_i', d_{iu}''\right). \tag{5}$$

The signer's signing key:

$$\mathrm{PSK}_{\mathbb{A}} = \left(d_\delta, d_\delta'\right) \tag{6}$$

(iii) **OutSign**: it takes as inputs an attribute set A and an outsourced key $\mathrm{OSK}_{\mathbb{A}}$.

  (i) If $A \in \mathbb{A}$, the S-CSP finds $\{w_i$: $i \in I = \{i \in [l]: \pi(i) \in A\}\}$ such that $\sum_{i\in I} w_i\mathbf{M}_i = (1, 0, \ldots, 0)$.
  (ii) It chooses r, $s \in_R \mathbb{Z}_p$, and computes

$$\begin{aligned} \sigma_1' &= \left(T_0 \prod_{u\in A} T_u\right)^r \left(\prod_{i\in I}\left(d_i \prod_{u\in A, u\neq\pi(i)} d_{iu}''\right)^{w_i}\right),\\ \sigma_2' &= g^r \prod_{i\in I} d_i'^{w_i},\\ \sigma_3' &= g^s. \end{aligned} \tag{7}$$

The outsourced signature $\sigma_{\mathrm{out}} = (\sigma_1, \sigma_2, \sigma_3)$.

(iv) Sign: it takes as inputs $\mathrm{PSK}_{\mathbb{A}}$, σ_{out}, and $\mathbf{M} = m_1 m_2 \cdots m_m \in \{0, 1\}^m$, and the signer selects $s_\delta \in_R \mathbb{Z}_p$ and computers

$$\sigma_1 = d_\delta \left(T_0 T_\delta\right)^{s_\delta} \sigma_1' \left(u_0 \prod_{i=1}^m u_i^{m_i}\right)^s, \sigma_2 = \sigma_2' d_\delta' g^{s_\delta}, \sigma_3 = \sigma_3'. \tag{8}$$

The final signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$.

(v) Verify: it takes as inputs $(PP, \sigma, \mathbf{M})$, and the verifier checks whether

$$e(g,\sigma_1) = Ye\left(\sigma_2, T_0 \prod_{u \varepsilon A \bigcup \delta} T_u\right) e\left(\sigma_3, u_0 \prod_{i=1}^{m} u_i^{m_i}\right). \tag{9}$$

outputs 1 if it holds; otherwise it outputs 0.

3.2. Attacks on Mo et al.'s EOABS Scheme. Mo et al.'s EOABS scheme [17] does not achieve any of the three security properties, although it was proven to be secure under their security models.

3.2.1. On Correctness. Mo et al.'s EOABS scheme is incorrect.

In Mo et al.'s scheme

$$\begin{aligned}
\sigma_1' &= \left(T_0 \prod_{u\in A} T_u\right)^r \left(\prod_{i\in I}\left(d_i \prod_{u\in A, u\neq \pi(i)} d_{iu}''\right)^{w_i}\right),\\
&= \left(T_0 \prod_{u\in A} T_u\right)^r \left(\prod_{i\in I}\left(g^{\lambda_i}\left(T_0 T_{\pi(i)}\right)^{r_i} \prod_{u\in A, u\neq \pi(i)} T_u^{r_i}\right)^{w_i}\right),\\
&= g^{\sum_{i\in I}\lambda_i w_i}\left(T_0 \prod_{u\in A} T_u\right)^{r+\sum_{i\in I} r_i w_i},\\
&= g^{a}\left(T_0 \prod_{u\in A} T_u\right)^{r+\sum_{i\in I} r_i w_i},\\
\sigma_2' &= g^r \prod_{i\in I} d_i'^{w_i},\\
&= g^r \prod_{i\in I} \left(g^{r_i}\right)^{w_i},\\
&= g^{r+\sum_{i\in I} r_i w_i},\\
\sigma_1 &= d_\delta \left(T_0 T_\delta\right)^{s_\delta} \sigma_1' \left(u_0 \prod_{i=1}^{m} u_i^{m_i}\right)^s,\\
&= g^b \left(T_0 T_\delta\right)^{r_\delta} \left(T_0 T_\delta\right)^{s_\delta} g^a \left(T_0 \prod_{u\in A} T_u\right)^{r+\sum_{i\in I} r_i w_i} \left(u_0 \prod_{i=1}^{m} u_i^{m_i}\right)^s,\\
&= g^{a+b} \left(T_0 T_\delta\right)^{r_\delta+s_\delta} \left(T_0 \prod_{u\in A} T_u\right)^{r+\sum_{i\in I} r_i w_i} \left(u_0 \prod_{i=1}^{m} u_i^{m_i}\right)^s,\\
\sigma_2 &= \sigma_2' d_\delta' g^{s_\delta},\\
&= g^{r+\sum_{i\in I} r_i w_i} g^{r_\delta} g^{s_\delta},\\
&= g^{r_\delta+s_\delta} g^{r+\sum_{i\in I} r_i w_i},\\
\sigma_3 &= g^s.
\end{aligned} \tag{10}$$

So we have

$$\begin{aligned}
&\frac{e(g,\sigma_1)}{Ye(\sigma_2, T_0\prod_{u\in A\cup\delta}T_u)e(\sigma_3, u_0\prod_{i=1}^m u_i^{m_i})},\\
&=\frac{e\left(g,g^{a+b}\right)(T_0T_\delta)^{r_\delta+s_\delta})(T_0\prod_{u\in A}T_u)^{r+\sum_{i\in I}r_iw_i}(u_0\prod_{i=1}^m u_i^{m_i})^s}{Ye\left(g^{r_\delta+s_\delta}g^{r+\sum_{i\in I}r_iw_i}, T_0\prod_{u\in A\cup\delta}T_u\right)e(g^s, u_0\prod_{i=1}^m u_i^{m_i})},\\
&=\frac{e\left(g,g^{a+b}\right)e(g,(T_0T_\delta)^{r_\delta+s_\delta})e(g, T_0\prod_{u\in A}T_u)^{r+\sum_{i\in I}r_iw_i}e\left(g,(u_0\prod_{i=1}^m u_i^{m_i})^s\right)}{Ye(g^{r_\delta+s_\delta}, T_0\prod_{u\in A\cup\delta}T_u)e\left(g^{r+\sum_{i\in I}r_iw_i}, T_0\prod_{u\in A\cup\delta}T_u\right)e(g^s, u_0\prod_{i=1}^m u_i^{m_i})},\\
&=\frac{1}{e(g^{r_\delta+s_\delta}, T_0\prod_{u\in A\cup\delta}T_u)e\left(g^{r+\sum_{i\in I}r_iw_i}, T_0, T_\delta\right)}\neq 1_{\mathbb{G}_T}.
\end{aligned} \tag{11}$$

Thus, the verification equation does not hold.

3.2.2. Forgery Attack. Mo et al.'s EOABS scheme is forgeable. Adversaries can collude with the malicious S-CSP to forge signatures.

Suppose that A is an attribute set, $\mathbb{A}$ is adversary $\mathscr{B}$'s access structure, and $A \notin \mathbb{A}$. Adversary $\mathscr{B}$ can collude with the malicious S-CSP to forge signatures for $(\mathbf{M}, A)$ as follows:

(i) The malicious S-CSP finds an outsourced key $\mathrm{OSK}_{\mathbb{A}'}$ with access structure $\mathbb{A}'$ satisfied by A. It runs the OutSign algorithm with $\mathrm{OSK}_{\mathbb{A}'}$ and generates and sends the outsourced signature $\sigma_{\mathrm{out}} = (\sigma_1', \sigma_2', \sigma_3')$ for A to adversary $\mathscr{B}$.

(ii) With the outsourced signature σ_{out} and message $\mathbf{M}$, adversary $\mathscr{B}$ runs the Sign algorithm with his signing key $\mathrm{PSK}_{\mathbb{A}}$ and then outputs a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ on $\mathbf{M}$ and A.

The attack above is executable for the following reasons:

(i) The signing key $\mathrm{PSK}_{\mathbb{A}}$ is only related to the master secret key b and the default attribute δ, but not to the access structure $\mathbb{A}$.

(ii) $A \in \mathbb{A}'$, so the outsourced signature σ_{out} for A can be generated correctly using $\mathrm{OSK}_{\mathbb{A}'}$.

Obviously, the output of adversary $\mathscr{B}$ above is a valid signature on the message $\mathbf{M}$ and the attribute set A. But the attribute set A does not satisfy $\mathscr{B}$'s access structure $\mathbb{A}$.

3.2.3. On Anonymity. Mo et al.'s EOABS scheme does not achieve anonymity. The S-CSP can identify the corresponding access structures of the signatures as follows:

(i) The S-CSP stores all outsourced signature σ_{out} with its corresponding access structure $\mathbb{A}$ into a list L in the form of $(\sigma_1', \sigma_2', \sigma_3', \mathbb{A})$.

(ii) Receiving a final signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, the S-CSP outputs the corresponding access structure $\mathbb{A}$ if there is $\sigma_3' = \sigma_3$ in L.

The attack above is practicable for the following reasons:

(i) The S-CSP needs to know the access structure $\mathbb{A}$ when using $\mathrm{OSK}_{\mathbb{A}}$ to generate outsourced signatures. So it can maintain the list L correctly.

(ii) Since $\sigma_3 = \sigma_3'$, the S-CSP can correctly establish the link between the final signature σ and the outsourced signature σ_{out}.

4. Outsourced Attribute-Based Signature

The attacks above suggest that the security models in [17] are not conforming to the actual. Their models are similar to the nonoutsourced models [2, 30]. We present more accurate security models in this section.

4.1. Definition. An outsourced attribute-based signature (OABS) scheme is composed of the following algorithms.

(i) $\mathrm{Setup}(1^\lambda) \longrightarrow (\mathrm{pp}, \mathrm{msk})$. It takes the security parameter λ as input and returns the public parameters pp and master key msk.

(ii) $\mathrm{KeyGen}(\mathrm{pp}, \mathrm{msk}, \mathbb{A}, f_u) \longrightarrow (\mathrm{OSK}_{\mathbb{A}f_u}, \mathrm{PSK}_{\mathbb{A}f_u})$. It takes the public parameters pp, master key msk, and an access structure $\mathbb{A}$ with a flag f_u as inputs and returns the outsourced key $\mathrm{OSK}_{\mathbb{A}f_u}$ and private signing key $\mathrm{PSK}_{\mathbb{A}f_u}$.

(iii) $\mathrm{Sign}_{\mathrm{out}}(\mathrm{pp}, \mathrm{OSK}_{\mathbb{A}f_u}, A) \longrightarrow \sigma_{\mathrm{out}}$. The outsourced signing algorithm takes the public parameters pp, an outsourced key $\mathrm{OSK}_{\mathbb{A}f_u}$, and an attribute set $A \in \mathbb{A}$ as inputs and returns an outsourced signature σ_{out}.

(iv) $\mathrm{Sign}(\mathrm{pp}, \mathrm{PSK}_{\mathbb{A}f_u}, \mathbf{M}, A, \sigma_{\mathrm{out}}) \longrightarrow \sigma$. The signing algorithm takes the public parameters pp, a private signing key $\mathrm{PSK}_{\mathbb{A}f_u}$, a message $\mathbf{M}$, and an outsourced signature σ_{out} as inputs and returns a signature σ for $(\mathbf{M}, A)$.

(v) $\mathrm{Verify}(\mathrm{pp}, \sigma, \mathbf{M}, A) \longrightarrow 1/0$. It takes the public parameters pp, a signature σ, a message $\mathbf{M}$, and an attribute set A as inputs. If σ is valid, it returns 1; otherwise, it returns 0.

Note: The flag f_u we introduced above is just an identifier used to match the outsourced key and private signing key correctly. It does not take part in any operation and does not affect efficiency and security.

4.2. Security. In this subsection, we present enhanced formal security models for OABS.

Definition 1 (correctness). *An OABS scheme is correct, if*

$$\Pr\left(\text{Verify}(\text{pp},\sigma,\mathbf{M},A)=1\;\middle|\;\begin{array}{l}(\text{pp},\text{msk})\leftarrow\text{Setup}\left(1^{\lambda}\right),\\ \left(\text{OSK}_{\mathbb{A}f_u},\text{PSK}_{\mathbb{A}f_u}\right)\leftarrow\text{KeyGen}\left(\text{pp},\text{msk},\mathbb{A},f_u\right),\\ \sigma_{\text{out}}\leftarrow\text{Sign}_{\text{out}}\left(\text{pp},\text{OSK}_{\mathbb{A}f_u},A\right),A\in\mathbb{A},\\ \sigma\leftarrow\text{Sign}\left(\text{pp},\text{PSK}_{\mathbb{A}f_u},\mathbf{M},A,\sigma_{\text{out}}\right)\end{array}\right)=1, \quad (12)$$

for any message $\mathbf{M}$*, any access structure* $\mathbb{A}$*, and any attribute set* A *such that* $A\in\mathbb{A}$*.*

4.2.1. Unforgeability. A trivial requirement for the unforgeability is that the adversary cannot possess the key required for signing because anyone who has the signing key can run the signing algorithm to generate a valid signature. In the scenario of outsourced signatures, all outsourced keys are sent to the S-CSP, and the S-CSP is not necessarily trusted. Therefore, it should be assumed that the adversary may have all the outsourced keys and only restrict him from possessing the required private signing key. To this end, we need to provide different oracles for the outsourced key and the private signing key. In addition, since the adversary is permitted to obtain all outsourced keys and can generate outsourced signatures by himself, he need not make any outsourced signing oracle query.

The unforgeability model of Mo et al. [17] does not reflect the above requirements and is therefore inaccurate. We present a more accurate unforgeability model in the following. There are two main differences between our model and Mo et al.'s model: First, our model provides the adversary with two oracles, OSK-Oracle and SK-Oracle, while their model only provides one oracle, KeyGen-Oracle. Second, our model restricts the adversary from possessing any private signing key of the access structure satisfied by the challenge attribute. In contrast, their model does not prohibit the adversary from obtaining the private signing key. These two improvements reflect the ideas mentioned above.

(1) GAME 1 (EUF-sA-CMA).

(i) *Initialization*. Adversary $\mathscr{A}$ selects and sends a challenge attribute set A^* to challenger $\mathscr{C}$.

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

(iii) *OSK-Oracle*. $\mathscr{A}$ chooses and sends an access structure $\mathbb{A}$ with a flag f_u to $\mathscr{C}$. $\mathscr{C}$ returns an outsourced key $\text{OSK}_{\mathbb{A}f_u}$ to $\mathscr{A}$.

(iv) *SK-Oracle*. $\mathscr{A}$ chooses and sends an access structure $\mathbb{A}$ with a flag f_u to $\mathscr{C}$. $\mathscr{C}$ returns a private signing key $\text{PSK}_{\mathbb{A}f_u}$ to $\mathscr{A}$.

(v) Sign-Oracle: $\mathscr{A}$ chooses and sends a message M, an attribute set A, and an outsourced signature σ_{out} with a flag f_u to $\mathscr{C}$. $\mathscr{C}$ returns a signature σ to $\mathscr{A}$.

(vi) Forgery. Adversary $\mathscr{A}$ outputs a triple $(\sigma^*,\mathbf{M}^*,A^*)$.

Adversary $\mathscr{A}$ wins the game, if

(i) $(\mathbf{M}^*,A^*)$ was not queried to Sign-Oracle;

(ii) any access structure $\mathbb{A}$ queried to SK-Oracle is not satisfied by A^*;

(iii) $\text{Verify}(\text{pp},\sigma^*,\mathbf{M}^*,A^*)=1$.

Adversary $\mathscr{A}$'s advantage is defined as its probability of winning the above game, denoted as $\text{Adv}_{\text{OABS},\mathscr{A}}^{\text{EUF}-sA-\text{CMA}}(1^{\lambda})$.

Definition 2 (unforgeability). *An OABS scheme is existentially unforgeable under selective attribute set but adaptive chosen message attack, if* $\text{Adv}_{\text{OABS},\mathscr{A}}^{\text{EUF}-sA-\text{CMA}}(1^{\lambda})$ *is negligible in the security parameter* λ *for any PPT adversary* $\mathscr{A}$*.*

4.2.2. Perfect Anonymity. In the outsourced attribute-based signature, the untrusted S-CSP generates the outsourced signature, and then the signer generates the final signature. This is the essential difference from the general attribute-based signature, which must be reflected in the security model. In the model of Mo et al., the outsourced signature is generated by the challenger, and the adversary has no way of knowing it. This makes it impossible for the adversary to determine the access structure through the outsourced signature. But in the outsourced attribute-based signature scheme, the outsourced signatures are calculated by the S-CSP, so that the S-CSP may track the access structures corresponding to the signatures through the outsourced signatures. This is why Mo et al.'s scheme is anonymous under their model, but the above attack exists. In our model, the outsourced signatures are generated by the adversary instead and then sent to the challenger. Under such a model, Mo et al.'s scheme does not achieve anonymity. Our model reflects the difference between outsourced attribute-based signatures and general attribute-based signatures.

We formalize our definition by a game between challenger $\mathscr{C}$ and adversary $\mathscr{A}$ as follows.

(1) GAME 2 (Perfect Anonymity).

(i) *Setup*. It is the same as that of GAME 1.

(ii) *Phase 1*. The adversary is allowed to request OSK-Oracle, SK-Oracle, and Sign-Oracle for any access structure or message he/she chooses. OSK-Oracle, SK-Oracle, and Sign-Oracle are the same as those of GAME 1.

(iii) *Challenge*.

  (i) Adversary $\mathscr{A}$ chooses a message $\mathbf{M}$, an attribute set A, and two challenge access structures $\mathbb{A}_0$ and $\mathbb{A}_1$ such that $A \in \mathbb{A}_0$ and $A \in \mathbb{A}_1$ and generates two outsourced signatures σ_{out}^0 and σ_{out}^1 using outsourced keys $\text{OSK}_{\mathbb{A}_0 f_{u_0}}$ and $\text{OSK}_{\mathbb{A}_1 f_{u_1}}$, respectively. Then he sends $(\mathbf{M}, A, \mathbb{A}_0, \mathbb{A}_1, \sigma_{\text{out}}^0, \sigma_{\text{out}}^1, f_{u_0}, f_{u_1})$ to challenger $\mathscr{C}$.

  (ii) $\mathscr{C}$ flips a fair coin $b \in \{0, 1\}$, generates a signature σ_b on message $\mathbf{M}$ and attribute set A using the signing key $\text{PSK}_{\mathbb{A}_b f_{u_b}}$, and then returns σ_b to $\mathscr{A}$.

(iv) *Phase 2*. As in Phase 1, the adversary can continue to request OSK-Oracle, SK-Oracle, and Sign-Oracle for any access structure (including $\mathbb{A}_0$ and $\mathbb{A}_1$) or message he/she chooses.

(v) *Guess*. $\mathscr{A}$ outputs a bit $b' \in \{0, 1\}$.

The advantage of $\mathscr{A}$ is defined as

$$\text{Adv}_{\text{OABS},\mathscr{A}}^{\text{PerAnon}}\left(1^{\lambda}\right) = \left| Pr\left[b' = b\right] - \frac{1}{2} \right|. \quad (13)$$

Definition 3 (perfect anonymity). *An OABS scheme is perfect anonymous, if for any adversary $\mathscr{A}$ the advantage $\text{Adv}_{\text{OABS},\mathscr{A}}^{\text{PerAnon}}(1^{\lambda})$ is negligible for the security parameter λ.*

5. Improvement

In this section, we propose a simple but significant improvement to fix our attacks. The ideas behind our improvement are as follows.

In Mo et al.'s scheme, the outsourced key and private signing key are independently generated with secret values a and b. Using such two keys to generate a signature, the public key $Y = e(g, g)^{a+b}$ will be canceled out in the verification equation. Since the outsourced key and the private signing key are independent of each other, using the outsourced key of Alice and the private signing key of Bob, one can also generate a correct signature, and the public key can also be canceled out in the verification equation. Our improvement fixes this shortcoming. We set $\alpha \in_R \mathbb{Z}_p$ as the master private key and $Y = e(g, g)^{\alpha}$ as the public key and then use $a \in_R \mathbb{Z}_p$ and $b = \alpha - a$ to generate the outsourced key and private signing key, respectively. In this way, everyone's outsourced key and private signing key are associated. The outsourced key and the private signing key of different users cannot be combined to generate a correct signature. If Alice's outsourced key and Bob's private signing key are combined to generate a signature, then $e(g, g)^{a_A+\alpha-a_B}$ will appear in the verification equation, which is not equal to the public key $e(g, g)^{\alpha}$. The signature will not be accepted as a valid signature.

In Mo et al.'s scheme, σ_3' is not blinded but directly used as a component of the final signature. This allows the adversary to track the access structure used to generate the signature. To ensure anonymity, the outsourced signature must be blinded. But the computation cost of blinding σ_3' is the same as that of computing σ_3'. Therefore, in our improved scheme, the user computes σ_4 by himself, and the server no longer computes σ_3'. σ_4 in our improvement is equivalent to σ_3 in Mo et al.'s scheme.

We split σ_1' into σ_{11}' and σ_{12}', and σ_2 into σ_2 and σ_3, all for the signature to satisfy the verification equation.

5.1. Improved Scheme

(i) Setup: it is the same as Mo et al.'s EOABS scheme, except that $Y = e(g, g)^{\alpha}$ and $\text{MSK} = \alpha \in_R \mathbb{Z}_p$.

(ii) KeyGen: it is the same as Mo et al.'s EOABS scheme but chooses $a \in_R \mathbb{Z}_p$ and sets $b = \alpha - a$.

(iii) OutSign: it takes as the inputs an attribute set A, an outsourced key $\text{OSK}_{\mathbb{A} f_u}$ with matrix $\mathbf{M}_{\mathbb{A}}$, and a flag f_u.

  (i) If $A \in \mathbb{A}$, the S-CSP finds $\{w_i : i \in I_{\mathbb{A}} = \{i \in [l_{\mathbb{A}}] : \pi(i) \in A\}\}$ such that $\sum_{i \in I_{\mathbb{A}}} w_i \mathbf{M}_{\mathbb{A}i} = (1, 0, \ldots, 0)$.

  (ii) computes

$$\sigma_{11}' = T_0 \prod_{u \in A} T_u,$$
$$\sigma_{12}' = \prod_{i \in I_{\mathbb{A}}} \left(d_i \prod_{u \in A, u \neq \pi(i)} d_{iu}'' \right)^{w_i}, \quad (14)$$
$$\sigma_2' = \prod_{i \in I_{\mathbb{A}}} d_i'^{w_i}.$$

The outsourced signature $\sigma_{\text{out}} = (\sigma_{11}', \sigma_{12}', \sigma_2')$.

(iv) Sign: with a private signing key $\text{PSK}_{\mathbb{A} f_u}$, a message $\mathbf{M} = m_1 m_2 \ldots m_m$, and an outsourced signature σ_{out}, the signer selects $r, s, s_\delta \in_R \mathbb{Z}_p$ and computes

$$\sigma_1 = d_\delta (T_0 T_\delta)^{s_\delta} \sigma'^{r}_{11} \sigma_{12}' \left(u_0 \prod_{i=1}^{m} u_i^{m_i} \right)^{s},$$
$$\sigma_2 = g^r \sigma_2', \quad (15)$$
$$\sigma_3 = d_\delta' g^{s_\delta},$$
$$\sigma_4 = g^s.$$

The final signature on $(\mathbf{M}, A)$ is $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.

(v) Verify: with $(\text{PP}, \sigma, \mathbf{M}, A)$, the verifier checks whether

$$e(g,\sigma_1) = Ye\left(\sigma_2, T_0\prod_{u\in A}T_u\right)e(\sigma_3, T_0T_\delta)e\left(\sigma_4, u_0\prod_{i=1}^{m}u_i^{m_i}\right). \tag{16}$$

If the equation holds, the verifier outputs 1. Otherwise, it outputs 0.

5.2. Proofs of Security

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

Proof. When $A \in \mathbb{A}$, we can find $\{w_i : i \in I_{\mathbb{A}}\}$ such that

$$\sum_{i\in I_{\mathbb{A}}} w_i\mathbf{M}_{\mathbb{A}i} = (1, 0, \ldots, 0), \tag{17}$$

and then

$$\sum_{i\in I_{\mathbb{A}}} w_i\lambda_i = \sum_{i\in I_{\mathbb{A}}} w_i\mathbf{M}_{\mathbb{A}i}\mathbf{v} = a. \tag{18}$$

So

$$\begin{aligned}
e(g,\sigma_1) &= e\left(g, g^b(T_0T_\delta)^{r_\delta}(T_0T_\delta)^{s_\delta}\left(T_0\prod_{u\in A}T_u\right)^r\prod_{i\in I_{\mathbb{A}}}\left(d_i\prod_{u\in A,u\neq\pi(i)}d''_{iu}\right)^{w_i}\left(u_0\prod_{i=1}^{m}u_i^{m_i})^s\right),\\
&= e\left(g, g^{\alpha-a}(T_0T_\delta)^{r_\delta+s_\delta}\left(T_0\prod_{u\in A}T_u\right)^r\prod_{i\in I_{\mathbb{A}}}\left(g^{\lambda_i}\left(T_0T_{\pi(i)}\right)^{r_i}\prod_{u\in A,u\neq\pi(i)}T_u^{r_i}\right)^{w_i}\left(u_0\prod_{i=1}^{m}u_i^{m_i}\right)^s\right),\\
&= e\left(g, g^{\alpha-a}g^{\sum_{i\in I_{\mathbb{A}}}w_i\lambda_i}(T_0T_\delta)^{r_\delta+s_\delta}\left(T_0\prod_{u\in A}T_u\right)^{r+\sum_{i\in I_{\mathbb{A}}}r_iw_i}\left(u_0\prod_{i=1}^{m}u_i^{m_i}\right)^s\right),\\
&= e(g,g^\alpha)e\left(g,(T_0T_\delta)^{r_\delta+s_\delta}\right)e\left(g,\left(T_0\prod_{u\in A}T_u\right)^{r+\sum_{i\in I_{\mathbb{A}}}r_iw_i}\right)e\left(g,\left(u_0\prod_{i=1}^{m}u_i^{m_i}\right)^s\right),\\
&= Ye(\sigma_3,T_0T_\delta)e\left(g^{r+\sum_{i\in I_{\mathbb{A}}}r_iw_i}, T_0\prod_{u\in A}T_u\right)e\left(\sigma_4,u_0\prod_{i=1}^{m}u_i^{m_i}\right),\\
&= Ye(\sigma_3,T_0T_\delta)e\left(g^r\prod_{i\in I_{\mathbb{A}}}(g^{r_i})^{w_i}, T_0\prod_{u\in A}T_u\right)e\left(\sigma_4,u_0\prod_{i=1}^{m}u_i^{m_i}\right),\\
&= Ye(\sigma_3,T_0T_\delta)e\left(\sigma_2, T_0\prod_{u\in A}T_u\right)e\left(\sigma_4,u_0\prod_{i=1}^{m}u_i^{m_i}\right).
\end{aligned} \tag{19}$$

The verification equation holds, and the improved scheme is correct. □

Theorem 2 (unforgeability). *The improved scheme is existentially unforgeable. If an adversary $\mathscr{A}$ can win GAME 1 with advantage ϵ, then there exists an algorithm $\mathscr{B}$ that solves the CDHE problem with probability $\epsilon' \geq (\epsilon/8q_s(m+1))$, where q_s is the maximum number of Sign-Oracle queries and m is the length of the message.*

Proof. In the following, $\mathscr{A}$ is an adversary with advantage ϵ, and $\mathscr{C}$ is the challenger to the CDHE problem. We build $\mathscr{B}$ as follows, which uses $\mathscr{A}$ to solve the CDHE problem.

Without loss of generality, we assume the attribute universe $U = \{1, 2, \ldots, n\}$. $\mathscr{B}$ maintains an initially empty list L_{key}.

(i) *CDHE Problem Gen.*

  (i) $\mathscr{C}$ chooses two prime order p multiplicative cyclic groups $\mathbb{G}$, $\mathbb{G}_T$ and a bilinear map $e: \mathbb{G}\times\mathbb{G} \longrightarrow \mathbb{G}_T$.

  (ii) chooses a generator $g \in \mathbb{G}$ and $a \in_R \mathbb{Z}_p$ and computes $(g, g^a, g^{a^2}, \ldots, g^{a^n}, g^{a^{n+2}}, \ldots, g^{a^{2n}})$.

  (iii) sends $(p, \mathbb{G}, \mathbb{G}_T, e, g, \{g_i = g^{a^i}\}_{i=1,i\neq n+1}^{2n})$ to $\mathscr{B}$.

(ii) *Init Phase.* $\mathscr{A}$ chooses and sends A^* to $\mathscr{B}$.

(iii) *Setup.*

(i) $\mathscr{B}$ chooses $a', t_0 \in_R \mathbb{Z}_p$ and $t_u \in_R \mathbb{Z}_p$ for all $u \in U$, and computes

$$\begin{aligned}&\left\{T_u = g^{t_u} g_{n+1-u}\colon u \in U\right\},\\ &T_0 = g^{t_0} \prod_{u\in A^*} T_u^{-1},\\ &T_\delta = g^{t_0} T_0^{-1}.\end{aligned} \tag{20}$$

(ii) computes $Y = e(g,g)^{a'} e(g_1, g_n)$ (i.e., it sets the master secret key $\alpha = a' + a^{n+1}$ implicitly).

(iii) Let $l_{\mathbf{M}} = 4q_s$, choosing $k_{\mathbf{M}} \in_R [m]$, picking $x' \in_R \mathbb{Z}_{l_{\mathbf{M}}}$, $y' \in_R \mathbb{Z}_p$, $\{\widehat{x}_i \in_R \mathbb{Z}_{l_{\mathbf{M}}}\colon i \in [m]\}$, $\{\widehat{y}_i \in_R \mathbb{Z}_p\colon i \in [m]\}$, and computing

$$\begin{aligned}&u_0 = g_n^{p-l_{\mathbf{M}}k_{\mathbf{M}}+x'} g^{y'},\\ &\left\{u_i = g_n^{\widehat{x}_i} g^{\widehat{y}_i}\colon i \in [m]\right\}.\end{aligned} \tag{21}$$

(iv) sends the public parameters $pp = (p, g, \mathbb{G}, \mathbb{G}_T, Y, e, T_0, \{T_u\colon u \in U \cup \{\delta\}\}, u_0, u_1, \ldots, u_m)$ to $\mathscr{A}$.

(iv) *OSK-Oracle*. Assume $\mathscr{A}$ queries an outsourced key on access structure $\mathbb{A}$ with the matrix $\mathbf{M}_{\mathbb{A}}$ of size $l_{\mathbb{A}} \times k_{\mathbb{A}}$ and flag f_u. If $(\mathbb{A}, f_u)$ in L_{key}, it returns the corresponding outsourced key $\text{OSK}_{\mathbb{A}f_u}$ to $\mathscr{A}$. Otherwise, we compute the keys as follows:

(i) If $A^* \in \mathbb{A}$:

(i) sets $\text{PSK}_{\mathbb{A}f_u} = \perp$ ($\mathscr{A}$ cannot query any private signing key for the access structure $\mathbb{A}$ satisfied by A^*).

(ii) runs KeyGen to get $\text{OSK}_{\mathbb{A}f_u}$.

(iii) returns $\text{OSK}_{\mathbb{A}f_u}$ to $\mathscr{A}$.

(iv) adds $(\text{OSK}_{\mathbb{A}f_u}, \perp, \mathbb{A}, f_u)$ into the key list L_{key}.

(ii) If $A^* \notin \mathbb{A}$:

(i) finds a vector $\mathbf{w} = (-1, w_2, \ldots, w_{k_{\mathbb{A}}}) \in \mathbb{Z}_p^{k_{\mathbb{A}}}$ such that $\mathbf{M}_{\mathbb{A}_i}\mathbf{w} = 0$ for each $i\colon \pi(i) \in A^*$, where $\mathbf{M}_{\mathbb{A}_i}$ is the i th row of $\mathbf{M}_{\mathbb{A}}$ (Lemma 1).

(ii) chooses $v_1, v_2, \ldots, v_{k_{\mathbb{A}}} \in_R \mathbb{Z}_p$, and sets $\mathbf{v}' = (0, v_2, \ldots, v_{k_{\mathbb{A}}})$.

(iii) For all $i \in [l_{\mathbb{A}}]$,

(i) if $\pi(i) \in A^*$, choose $r_i \in_R \mathbb{Z}_p$ and compute

$$\begin{aligned}&d_i = g^{\mathbf{M}_{\mathbb{A}_i}v'} \left(T_0 T_{\pi(i)}\right)^{r_i},\\ &d_i' = g^{r_i},\\ &\{d_{iu}'' = T_u^{r_i}\colon u \in U/\pi(i)\},\end{aligned} \tag{22}$$

(ii) if $\pi(i) \notin A^*$, choose $r_i' \in_R \mathbb{Z}_p$ and compute

$$\begin{aligned}d_i &= g^{\mathbf{M}_{\mathbb{A}_i}(v'-v_1 w)} \left(T_0 T_{\pi(i)}\right)^{r_i'} \left(g_{\pi(i)}^{-(t_0+t_{\pi(i)})} \prod_{u\in A^*} \left(g_{\pi(i)}^{t_u} g_{n+1-u+\pi(i)}\right)\right)^{-\mathbf{M}_{\mathbb{A}_i}w}\\ d_i' &= g^{r_i'} g_{\pi(i)}^{-\mathbf{M}_{\mathbb{A}_i}w}\\ d_{iu}'' &= T_u^{r_i'} \left(g_{\pi(i)}^{t_u} g_{n+1-u+\pi(i)}\right)^{\mathbf{M}_{\mathbb{A}_i}w},\\ &\quad u \in U/\{\pi(i)\}.\end{aligned} \tag{23}$$

(iv) selects $r_\delta \in_R \mathbb{Z}_p$ and computes $d_\delta = g^{a'-v_1} (T_0 T_\delta)^{r_\delta}$, $d_\delta' = g^{r_\delta}$.

(v) returns $\text{OSK}_{\mathbb{A}f_u} = (d_i, d_i', d_{iu}'')_{i\in[l_{\mathbb{A}}],(u\in U\setminus\pi(i))}$ to $\mathscr{A}$.

(vi) adds $(\text{OSK}_{\mathbb{A}f_u}, \text{PSK}_{\mathbb{A}f_u} = (d_\delta, d_\delta'), \mathbb{A}, f_u)$ into the key list L_{key}.

(iii) SK-Oracle. It is the same as the OSK-Oracle above, except that it returns a private signing key $\text{PSK}_{\mathbb{A}f_u}$ to $\mathscr{A}$. □

Claim 1. The keys simulated above are correct.

Proof. If $A^* \in \mathbb{A}$, and $\text{OSK}_{\mathbb{A}f_u}$ is generated by KeyGen, it is correct certainly.

If $A^* \notin \mathbb{A}$, according to Lemma 1, we can find a vector $\mathbf{w} = (-1, w_2, \ldots, w_{k_{\mathbb{A}}}) \in \mathbb{Z}_p^{k_{\mathbb{A}}}$ such that $\mathbf{M}_{\mathbb{A}i}\mathbf{w} = 0$ for each $i\colon \pi(i) \in A^*$. We prove $\text{OSK}_{\mathbb{A}f_u}$ is a correct outsourced key with $v = -(v_1 + a^{n+1})\mathbf{w} + \mathbf{v}'$ as follows:

(i) When $\pi(i) \in A^*$, $\mathbf{M}_{\mathbb{A}i}w = 0$, and $\lambda_i = \mathbf{M}_{\mathbb{A}i}v = -(v_1 + a^{n+1})\mathbf{M}_{\mathbb{A}i}w + \mathbf{M}_{\mathbb{A}i}v' = \mathbf{M}_{\mathbb{A}i}v'$, we have

$$\begin{aligned}&d_i = g^{\mathbf{M}_{\mathbb{A}i}v'} \left(T_0 T_{\pi(i)}\right)^{r_i} = g^{\lambda_i} \left(T_0 T_{\pi(i)}\right)^{r_i},\\ &d_i' = g^{r_i},\\ &\{d_{iu}'' = T_u^{r_i}\colon u \in U/\{\pi(i)\}\}.\end{aligned} \tag{24}$$

(ii) When $\pi(i) \notin A^*$, we have $\lambda_i = \mathbf{M}_{\mathbb{A}i}v = \mathbf{M}_{\mathbb{A}i}(v' - v_1 w) - a^{n+1}\mathbf{M}_{\mathbb{A}i}w$, and

$$\begin{aligned}
d_i &= g^{\mathbf{M}_{\mathbb{A}i}(\mathbf{v}'-v_1\mathbf{w})}\left(T_0T_{\pi(i)}\right)^{r_i'}\left(g_{\pi(\mathrm{i})}^{-(t_0+t\pi(i))}\prod_{u\in A^*}\left(g_{\pi(\mathrm{i})}^{t_u}g_{n+1-u+\pi(i)}\right)\right)^{-\mathbf{M}_{\mathbb{A}i}\mathbf{w}},\\
&= g^{\mathbf{M}_{\mathbb{A}i}(\mathbf{v}'-v_1\mathbf{w})}\left(T_0T_{\pi(i)}\right)^{r_i'}g^{a^{\pi(i)}t_0\mathbf{M}_{\mathbb{A}i}\mathbf{w}}g^{a^{\pi(i)}t_{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}},\\
&\quad\prod_{u\in A^*}\left(g^{a^{\pi(i)}t_u}g_{n+1-u}^{a^{\pi(i)}}\right)^{-\mathbf{M}_{\mathbb{A}i}\mathbf{w}}g^{a^{n+1}\mathbf{M}_{\mathbb{A}i}\mathbf{w}}g^{-a^{n+1}\mathbf{M}_{\mathbb{A}i}\mathbf{w}},\\
&= g^{\mathbf{M}_{\mathbb{A}i}(\mathbf{v}'-v_1\mathbf{w})-a^{n+1}\mathbf{M}_{\mathbb{A}i}\mathbf{w}}\left(T_0T_{\pi(i)}\right)^{r_i'}\left(g^{t_0}\prod_{u\in A^*}\left(g^{t_u}g_{n+1-u}\right)^{-1}\right)^{a^{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}},\\
&\quad g^{a^{\pi(i)}t_{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}}g^{a^{n+1}a^{-\pi(i)}a^{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}},\\
&= g^{\lambda_i}\left(T_0T_{\pi(i)}\right)^{r_i'}T_0^{a^{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}}\left(g^{t_{\pi(i)}}g_{n+1-\pi(i)}\right)^{a^{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}},\\
&= g^{\lambda_i}\left(T_0T_{\pi(i)}\right)^{r_i'}\left(T_0T_{\pi(i)}\right)^{a^{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}},\\
&= g^{\lambda_i}\left(T_0T_{\pi(i)}\right)^{r_i},\\
d_i' &= g^{r_i'}g_{\pi(i)}^{\mathbf{M}_{\mathbb{A}i}\mathbf{w}} = g^{r_i},\\
d_{iu}'' &= T_u^{r_i'}\left(g_{\pi(\mathrm{i})}^{t_u}g_{n+1-u+\pi(i)}\right)^{\mathbf{M}_{\mathbb{A}i}\mathbf{w}} = T_u^{r_i'}\left(g^{t_u}g_{n+1-u}\right)^{a^{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}} = T_u^{r_i},
\end{aligned}\tag{25}$$

where $r_i = r_i' + a^{\pi(i)}\mathbf{M}_{\mathbb{A}i}\mathbf{w}$.

This concludes that $\mathrm{OSK}_{\mathbb{A}f_u} = (d_i, d_i', d_{iu}'')_{i\in[l_{\mathbb{A}}],u\in U/\pi(i)}$ is a correct outsourced key.

Since the first component of $\mathbf{v}$ is $v_1 + a^{n+1}$, then

$$\alpha - \left(v_1 + a^{n+1}\right) = \left(a' + a^{n+1}\right) - \left(v_1 + a^{n+1}\right) = a' - v_1. \tag{26}$$

Thus $\mathrm{PSK}_{\mathbb{A}f_u} = (d_\delta = g^{a'-v_1}(T_0T_\delta)^{r_\delta}, d_\delta' = g^{r_\delta})$ is a correct private signing key.

(vi) *Sign-Oracle*. It takes $(\mathbf{M}, A, \sigma_{\mathrm{out}}, f_u)$ as inputs. Define functions

$$\begin{aligned}
F(\mathbf{M}) &= p - l_{\mathbf{M}}K_{\mathbf{M}} + x' + \sum_{i=1}^{m}\widehat{x}_i m_i,\\
J(\mathbf{M}) &= y' + \sum_{i=1}^{m}\widehat{y}_i m_i,\\
K(\mathbf{M}) &= \begin{cases} 0, & if\ x' + \sum_{i=1}^{m}\widehat{x}_i \equiv 0 \pmod{l_{\mathbf{M}}},\\ 1, & \text{otherwise.}\end{cases}
\end{aligned}\tag{27}$$

If $K(\mathbf{M}) = 0$, it aborts. Otherwise, $\mathscr{B}$ chooses $\mu_\delta, s', r' \in \mathbb{Z}_p$, and computes and returns

$$\begin{aligned}
\sigma_1 &= (T_0T_\delta)^{r_\delta+\mu_\delta}g^{a'}\left(T_0\prod_{u\in A}T_u\right)^{r'}\left(g_n^{F(\mathbf{M})}g^{J(\mathbf{M})}\right)^{s'}g_1^{-J(\mathbf{M})/F(\mathbf{M})},\\
\sigma_2 &= g^{r'},\\
\sigma_3 &= g^{\mu_\delta+r_\delta},\\
\sigma_4 &= g^{s'}g_1^{-1/F(\mathbf{M})}.
\end{aligned}\tag{28}$$

□

Claim 2. The simulated signatures are correct.

Proof. By simple calculating, we have $u_0\prod_{i=1}^{m}u_i^{m_i} = g_n^{F(\mathbf{M})}g^{J(\mathbf{M})}$. If $K(\mathbf{M}) \neq 0$, then $F(\mathbf{M}) \neq 0 \pmod{p}$, because we can assume $l_{\mathbf{M}}(m+1) < p$ for any reasonable values of p, m, and $l_{\mathbf{M}}$. Then, we have

$$\begin{aligned}
\sigma_1 &= (T_0T_\delta)^{r_\delta+\mu_\delta} g^{a'} \left(T_0 \prod_{u\in A} T_u\right)^{r'} \left(g_n^{F(\mathbf{M})} g^{J(\mathbf{M})}\right)^{s'} g_1^{-J(\mathbf{M})/F(\mathbf{M})}, \\
&= (T_0T_\delta)^{r_\delta+\mu_\delta} g^{a\prime} g^{a^{n+1}} g^{-a^{n+1}} \left(T_0 \prod_{u\in A} T_u\right)^{r'} \left(g_n^{F(\mathbf{M})} g^{J(\mathbf{M})}\right)^{s'} g_1^{-J(\mathbf{M})/F(\mathbf{M})}, \\
&= (T_0T_\delta)^{r_\delta+\mu_\delta} g^{a\prime+a^{n+1}} \left(T_0 \prod_{u\in A} T_u\right)^{r'} \left(g_n^{F(\mathbf{M})} g^{J(\mathbf{M})}\right)^{s'} g_n^{-aF(\mathbf{M})/F(\mathbf{M})} g^{-aF(\mathbf{M})/F(\mathbf{M})}, \\
&= (T_0T_\delta)^{r_\delta+\mu_\delta} g^{\alpha} \left(T_0 \prod_{u\in A} T_u\right)^{r'} \left(g_n^{F(\mathbf{M})} g^{J(\mathbf{M})}\right)^{s'} \left(g_n^{F(\mathbf{M})} g^{J(\mathbf{M})}\right)^{-a/F(\mathbf{M})}, \\
&= g^{\alpha} \left(T_0 \prod_{u\in A} T_u\right)^{r'} (T_0T_\delta)^{r_\delta+\mu_\delta} \left(u_0 \prod_{i=1}^{m} u_i^{m_i}\right)^{s'-a/F(\mathbf{M})}, \\
\sigma_2 &= g^{r'}, \\
\sigma_3 &= g^{\mu_\delta+r_\delta}, \\
\sigma_4 &= g^{s'} g_1^{-J(\mathbf{M})/F(\mathbf{M})} = g^{s'-a/F(\mathbf{M})}.
\end{aligned} \tag{29}$$

Then have

$$\begin{aligned}
e(g,\sigma_1) &= e\left(g, g^{\alpha}\left(T_0 \prod_{u\in A} T_u\right)^{r'} (T_0T_\delta)^{r_\delta+\mu_\delta} \left(u_0 \prod_{i=1}^{m} u_i^{m_i}\right)^{s'-a/F(\mathbf{M})}\right), \\
&= Ye\left(\sigma_2, T_0 \prod_{u\in A} T_u\right) e(\sigma_3, T_0T_\delta) e\left(\sigma_4, u_0 \prod_{i=1}^{m} u_i^{m_i}\right).
\end{aligned} \tag{30}$$

The verification equation holds. Thus the simulated signature is correct.

(vii) *Forgery*. $\mathscr{A}$ outputs a signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*)$ on $(\mathbf{M}^*, A^*)$.

(viii) *Output*. If $F(\mathbf{M}^*) \neq 0 \pmod p$, it aborts. Otherwise, $\mathscr{B}$ computes and outputs

$$\frac{\sigma_1^*}{g^{a'} \sigma_4^{*\,J(\mathbf{M}^*)} (\sigma_2^* \sigma_3^*)^{t_0}}. \tag{31}$$

□

Claim 3. The output of $\mathscr{B}$ is $g^{a^{n+1}}$.

Proof. Because $F(\mathbf{M}^*) = 0\, mo\ d\ p$, so $u_0 \prod_{i=0}^{m} u_i^{m_i^*} = g_n^{F(\mathbf{M}^*)} g^{J(\mathbf{M}^*)} = g^{J(\mathbf{M}^*)}$.

σ^* is a valid signature on message $\mathbf{M}^*$ for A^*, so we have

$$\begin{aligned}
e(g,\sigma_1^*) &= Ye\left(\sigma_2^*, T_0 \prod_{u\in A^*} T_u\right) e(\sigma_3^*, T_0T_\delta) e\left(\sigma_4^*, u_0 \prod_{i=1}^{m} u_i^{m_i^*}\right), \\
&= e(g, g^{\alpha}) e\left(\sigma_2^*, g^{t_0}\right) e\left(\sigma_3^*, g^{t_0}\right) e\left(\sigma_4^*, g^{J(\mathbf{M}^*)}\right), \\
&= e\left(g, g^{a'+a^{n+1}} (\sigma_2^*\sigma_3^*)^{t_0} \sigma_4^{*\,J(\mathbf{M}^*)}\right),
\end{aligned} \tag{32}$$

and then

$$\frac{\sigma_1^*}{g^{a'}\sigma_4^{*\,J(\mathbf{M}^*)}\left(\sigma_2^*\sigma_3^*\right)^{t_0}} = g^{a^{n+1}}. \tag{33}$$

□

Claim 4. The probability that the simulation is not aborted is $1/8q_s(m+1)$.

Proof. The same as Claim 2 of Waters [31].

Claims 1 and 2 show that the simulation above is correct. Thus, by Claim 3 and Claim 4, $\mathscr{B}$ can compute $g^{a^{n+1}}$ with probability $\epsilon' \geq \epsilon/8q_s(m+1)$. □

Theorem 3 (perfect anonymity). *The improved scheme is perfect anonymous.*

Proof. Challenger $\mathscr{C}$ executes the Setup algorithm to set up the system and responds to the oracle requests by running the corresponding algorithm.

Receiving $(\mathbf{M}, A, \mathbb{A}_0, \mathbb{A}_1, \sigma_{out}^0, \sigma_{out}^1, f_{u_0}, f_{u_1})$, $\mathscr{C}$ flips a fair coin $b \in \{0,1\}$, chooses $r_b, s_b, s_{b\delta} \in_R \mathbb{Z}_p$, and computes and returns a signature $\sigma_b = (\sigma_{b1}, \sigma_{b2}, \sigma_{b3}, \sigma_{b4})$ from σ_{out}^b using $OSK_{\mathbb{A}_b f_{u_b}}$.

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

Since $\sigma_{out}^b = ({\sigma'}_{11}^b, {\sigma'}_{12}^b, {\sigma'}_2^b)$ is an outsourced signature on A using $OSK_{\mathbb{A}_b f_{u_b}}$, we have

$$\begin{aligned}
{\sigma'}_{11}^b &= T_0 \prod_{u\in A} T_u, \\
{\sigma'}_{12}^b &= \prod_{i\in I_{\mathbb{A}b}} \left(d_{bi} \prod_{u\in A, u\notin\pi(i)} d''_{biu} \right)^{w_{bi}}, \\
{\sigma'}_2^b &= \prod_{i\in I_{\mathbb{A}b}} d_{bi}^{\prime\, w_{bi}}.
\end{aligned} \tag{34}$$

And $\sigma_b = (\sigma_{b1}, \sigma_{b2}, \sigma_{b3}, \sigma_{b4})$ is a signature calculated from σ_{out}^b, so we have

$$\begin{aligned}
\sigma_{b1} &= d_{b\delta}\left(T_0 T_\delta\right)^{s_{b\delta}} \left({\sigma'}_{11}^b\right)^{r_b} {\sigma'}_{12}^b \left(u_0 \prod_{i=1}^m u_i^{m_i} \right)^{s_b} \\
&= g^a \left(T_0 T_\delta\right)^{r_{b\delta}+s_{b\delta}} \left(T_0 \prod_{u\in A} T_u \right)^{r_b + \sum_{i\in I_{\mathbb{A}b}} r_{bi} w_{bi}} \left(u_0 \prod_{i=1}^m u_i^{m_i} \right)^{s_b}, \\
\sigma_{b2} &= g^{r_b} {\sigma'}_2^b = g^{r_b + \sum_{i\in I_{\mathbb{A}b}} r_{bi} w_{bi}}, \\
\sigma_{b3} &= d'_{b\delta} g^{s_{b\delta}} = g^{r_{b\delta}+s_{b\delta}}, \\
\sigma_{b4} &= g^{s_b}.
\end{aligned} \tag{35}$$

We can rewrite σ_b as

$$\begin{aligned}
\sigma_{b1} &= g^a \left(T_0 T_\delta\right)^{r_{\bar{b}\delta} + \left(r_{b\delta} - r_{\bar{b}\delta} + s_{b\delta}\right)} \left(T_0 \prod_{u\in A} T_u \right)^{\left(r_b + \sum_{i\in I_{\mathbb{A}b}} r_{bi} w_{bi} - \sum_{i\in I_{\mathbb{A}\bar{b}}} r_{\bar{b}i} w_{\bar{b}i} \right) + \sum_{i\in I_{\mathbb{A}\bar{b}}} r_{\bar{b}i} w_{\bar{b}i}} \left(u_0 \prod_{i=1}^m u_i^{m_i} \right)^{s_b} \\
&= g^a \left(T_0 T_\delta\right)^{r_{\bar{b}\delta} + s_{\bar{b}\delta}} \left(T_0 \prod_{u\in A} T_u \right)^{r_{\bar{b}} + \sum_{i\in I_{\mathbb{A}\bar{b}}} r_{\bar{b}i} w_{\bar{b}i}} \left(u_0 \prod_{i=1}^m u_i^{m_i} \right)^{s_{\bar{b}}} \\
&= d_{\bar{b}\delta} \left(T_0 T_\delta\right)^{s_{\bar{b}\delta}} \left({\sigma'}_{11}^{\bar{b}} \right)^{r_{\bar{b}}} {\sigma'}_{12}^{\bar{b}} \left(u_0 \prod_{i=1}^m u_i^{m_i} \right)^{s_{\bar{b}}}, \\
\sigma_{b2} &= g^{\left(r_b + \sum_{i\in I_{\mathbb{A}b}} r_{bi} w_{bi} - \sum_{i\in I_{\mathbb{A}\bar{b}}} r_{\bar{b}i} w_{\bar{b}i} \right) + \sum_{i\in I_{\mathbb{A}\bar{b}}} r_{\bar{b}i} w_{\bar{b}i}} \\
&= g^{r_{\bar{b}} + \sum_{i\in I_{\mathbb{A}\bar{b}}} r_{\bar{b}i} w_{\bar{b}i}} = g^{r_{\bar{b}}} \sigma_2^{\bar{b}}, \\
\sigma_{b3} &= g^{r_{b\delta}+s_{b\delta}} = g^{r_{\bar{b}\delta}+s_{\bar{b}\delta}} = d'_{\bar{b}\delta} g^{s_{\bar{b}\delta}}, \\
\sigma_{b4} &= g^{s_b} = g^{s_{\bar{b}}},
\end{aligned} \tag{36}$$

Table 2: Comparison of performance.

Scheme	EOABS [17]	Improved scheme
MPK.Size	$(n+m+4)\mathbb{G}$	$(n+m+4)\mathbb{G}$
MSK.Size	$2\mathbb{Z}_p$	$1\mathbb{Z}_p$
Out.Key.Size	$(n+1)l\ \mathbb{G}$	$(n+1)l\ \mathbb{G}$
Sig.Key.Size	$2\mathbb{G}$	$2\mathbb{G}$
Out.Sig.Size	$2\mathbb{G}$	$3\mathbb{G}$
Sig.Size	$3\mathbb{G}$	$4\mathbb{G}$
Key.Gen	$((n+2)l+3)E+(2l+1)M+lI$	$((n+2)l+3)E+(2l+1)M+lI$
Out.Sig.Gen	$(2l_r+1)E+(n+1)l_aM$	$2l_rE+((n+1)l_a-1)M$
Sig.Gen	$(m+4)E+(m+6)M$	$(m+6)E+(m+7)M$
Verify	$mE+3P+(l_a+m+1)M$	$mE+4P+(l_a+m+1)M$
Unforgeable	No	Yes
Perfect anonymity	No	Yes
Correctness	No	Yes

where $r_{\overline{b}} = r_b + \sum_{i\in I_{\mathbb{A}b}} r_{bi}w_{bi} - \sum_{i\in I_{\mathbb{A}\overline{b}}} r_{\overline{b}i}w_{\overline{b}i} \in \mathbb{Z}_p$, $s_{\overline{b}} = s_b$, $s_{\overline{b}\delta} = r_{b\delta} - r_{\overline{b}\delta} + s_{b\delta}$, $\overline{b} = b\oplus 1$.

This concludes that σ_b is also a signature calculated from $\sigma_{out}^{\overline{b}}$ out using $(r_{\overline{b}}, s_{\overline{b}}, s_{\overline{b}\delta})$ and $PSK_{\mathbb{A}_{\overline{b}}, f_{u_{\overline{b}}}}$. Because r, s, s_δ are randomly selected from $\mathbb{Z}_p$, the probability of selecting $(r_b, s_b, s_{b\delta})$ is the same as that of $(r_{\overline{b}}, s_{\overline{b}}, s_{\overline{b}\delta})$, and both are p^{-3}. Therefore, even if the adversary has an unlimited capability, it is impossible to distinguish which access structure was used to generate the signature.

On the other hand, the adversary may generate signatures by him/herself. Assuming that the random integer selected by the adversary is (r, s, s_δ), then the probabilities of $(r, s, s_\delta) = (r_b, s_b, s_{b\delta})$ and $(r, s, s_\delta) = (r_{\overline{b}}, s_{\overline{b}}, s_{\overline{b}\delta})$ are the same p^{-3}. So, even if the adversary possesses all the private signing keys and outsourced keys, it is impossible to determine which access structure was used to generate the signature.

In summary, adversary $\mathscr{A}$'s advantage $Adv_{OABS,\mathscr{A}}^{PerAnon}(1^\lambda)$ is 0, and the improved scheme achieves perfect anonymity. □

5.3. Performance Analysis. Denote by $\mathbb{G}$ an element of $\mathbb{G}$, by $\mathbb{Z}_p$ an element of $\mathbb{Z}_p$, by E an exponentiation in $\mathbb{G}$, by $\mathbf{M}$ a multiplication in $\mathbb{G}$, by P a computation of the pairing, and by I an inner product operation. Let n be the size of the attribute universe U, m be the length of the message, l be the number of rows of $\mathbf{M}_{\mathbb{A}}$, l_r be the number of rows whose attribute belongs to the attribute set, i.e., $l_r = |I_A|$, and l_a be the size of the attribute set, i.e., $l_a = |A|$. We compare our scheme to Mo et al.'s scheme in Table 2.

In terms of data size, our scheme has one less integer in the master private key and one more group element in the final signature. The other items are the same size. There is not much difference between the two schemes.

In terms of computational overhead, our scheme has an extra $2E + 1\mathbf{M}$ in signature generation. Estimated with the message length $m = 160$, this is an increase of about 0.7%.

Although our scheme is slightly inferior to Mo et al.'s schemes in terms of data length and computational overhead, our scheme has an overwhelming advantage in terms of security. Our scheme achieves correctness, unforgeability, and perfect anonymity, while their scheme does not achieve any of these three properties. It shows that our improvement is meaningful.

6. Conclusion

OABS was introduced to solve the problem that ABS is not suitable for scenarios with limited computing power. Recently, Mo et al. proposed an expressive outsourced attribute-based signature scheme. In this paper, we analyze the security of Mo et al.'s EOABS scheme. We show that it does not achieve the correctness, unforgeability, and anonymity that they claimed. We present more accurate security models for OABS and propose an improved OABS scheme to fix our attacks. Our scheme is provably secure in the standard model.

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 National Social Science Fund of China (No. 21XTQ015), the Natural Science Foundation of Fujian Province of China (Nos. 2019J01750, 2019J01752, and 2020J01814), and the Fujian Province Young and Middle-Aged Teacher Education Research Project (No. JAT200293).

References

[1] S. Amit and B. Waters, “Fuzzy identity-based encryption,” in *Proceedings of Advances in Cryptology-EUROCRYPT 2005 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, R. Cramer, Ed., vol. 3494, pp. 457–473, Springer, Aarhus, Denmark, May 2005.

[2] K. Hemanta, Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Proceedings of Topics in Cryptology 2011 (CT-RSA 2011)-The Cryptographers Track at the RSA Conference 2011*, A. Kiayias, Ed., vol. 6558, pp. 376–392, Springer, San Francisco, CA, USA, February 2011.

[3] 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. 5580, pp. 198–216, Springer, Africa, Gammarth, Tunisia, June 2009.

[4] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Two-factor authentication with end-to-end password security," in *Proceedings of Public-Key Cryptography – PKC 2018*, M. Abdalla and R. Dahab, Eds., vol. 10770, pp. 431–461, Springer, Rio de Janeiro, Brazil, March 2018.

[5] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.

[6] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[7] P. Datta, T. Okamoto, and K. Takashima, "Efficient attribute-based signatures for unbounded arithmetic branching programs," in *Proceedings of Public-Key Cryptography–PKC 2019*, D. Lin and K. Sako, Eds., vol. 11442, pp. 127–158, Springer, Beijing, China, April 2019.

[8] A. El Kaafarani and S. Katsumata, "Attribute-based signatures for unbounded circuits in the rom and efficient instantiations from lattices,"vol. 10770, pp. 89–119, in *Proceedings of Public-Key Cryptography – PKC 2018*, vol. 10770, , Springer, Rio de Janeiro, Brazil, March 2018.

[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] G. Lin, Y. Xia, Y. Chun, and Z. Sun, "F2p-abs: a fast and secure attribute-based signature for mobile platforms," *Security And Communication Networks*, vol. 2019, Article ID 5380710, 12 pages, 2019.

[11] T. Okamoto and K. Takashima, "Efficient attribute-based signatures for non-monotone predicates in the standard model," in *Proceedings of Public Key Cryptography – PKC 2011*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., vol. 6571, pp. 35–52, Springer, Taormina, Italy, March 2011.

[12] T. Okamoto and K. Takashima, "Decentralized attribute-based encryption and signatures," *IEICE Trans Fundamentals*, vol. E103-A, pp. 41–73, 2020.

[13] C. Sun, Y. Liu, X. Zeng, and H. Si, "Provable secure attribute-based proxy signature," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 1, pp. 337–343, 2020.

[14] F. Tang, H. Li, and B. Liang, "Attribute-based signatures for circuits from multilinear maps," in *Information Security*, S. S. M. Chow, C. Jan, L. C. K. Hui, and M. Y. Siu, Eds., vol. 8783, pp. 54–71, Springer, Cham, 2014.

[15] Y. Zhang, X. Liu, Y. Hu, Q. Zhang, and H. Jia, "Attribute-based signatures for Inner-Product predicate from lattices," in *Proceedings of Cyberspace Safety And Security*, J. Vaidya, X. Zhang, and L. Jin, Eds., vol. 11982, pp. 173–185, Springer, Guangzhou, China, January 2019.

[16] X. Chen, J. Li, X. Huang, J. Li, Y. Xiang, and D. S. Wong, "Secure outsourced attribute-based signatures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3285–3294, 2014.

[17] R. Mo, J. Ma, X. Liu, and H. Liu, "Eoabs: expressive outsourced attribute-based signature," *Peer-to-Peer Networking and Applications*, vol. 11, pp. 979–988, 2017.

[18] J. Sun, J. Qin, and J. Ma, "Securely outsourcing decentralized multi-authority attribute based signature," in *Proceedings of Cyberspace Safety And Security*, S. Wen, W. Wu, and A. Castiglione, Eds., vol. 10581, pp. 86–102, Springer, Xi'an, China, October 2017.

[19] 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, 2019.

[20] 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, p. e6173, 01 2021.

[21] Z. Liu, H. Yan, and Z. Li, "Server-aided anonymous attribute-based authentication in cloud computing," *Future Generation Computer Systems*, vol. 52, pp. 61–66, 2015.

[22] Y. Ren and T. Jiang, "Verifiable outsourced attribute-based signature scheme," *Multimedia Tools and Applications*, vol. 77, pp. 18105–18115, 2018.

[23] O. Uzunkol, "Comments on 'verifiable outsourced attribute-based signature scheme," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11735–11742, 2019.

[24] H. Cui, R. H. Deng, J. K. Liu, X. Yi, and Y. Li, "Server-aided attribute-based signature with revocation for resource-constrained industrial-internet-of-things devices," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3724–3732, 2018.

[25] X. Hu, Y. Bao, X. Nie, and Y. I. Asoor, "Server-aided attribute-based signature supporting expressive access structures for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1013–1023, 2020.

[26] Z. Wang, R. Xie, and S. Wang, "Attribute-based server-aided verification signature," *Applied Mathematics and Information Sciences*, vol. 8, no. 6, pp. 3183–3190, 2014.

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

[28] Y. Sreenivasa Rao and R. Dutta, "Efficient attribute-based signature and signcryption realizing expressive access structures," *International Journal of Information Security*, vol. 15, no. 1, pp. 81–109, 2015.

[29] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph. D. thesis, Israel Institute of Technology, Technion, Israel, 1996.

[30] H. Cui, R. H. Deng, B. Qin, and J. Weng, "Key regeneration-free ciphertext-policy attribute-based encryption and its application," *Information Sciences*, vol. 517, pp. 217–229, 2020.

[31] B. Waters, "Efficient identity-based encryption without random oracles," in *Proceedings of In Advances In Cryptology–EUROCRYPT 2005*, R. Cramer, Ed., pp. 114–127, Springer, Aarhus, Denmark, May 2005.