mathematics

Article

A Delegation Attack Method on Attribute-Based Signatures and Probable Solutions

Jialu Hao [1,2,*], Wei Wu [3,4,*], Shuo Wang [1], Xiaoge Zhong [1], Guang Chu [1] and Feng Shao [1]

[1] Xi'an Satellite Control Center, Xi'an 710043, China
[2] School of Electronic Science, National University of Defense Technology, Changsha 410073, China
[3] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China
[4] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
[*] Correspondence: haojialu13@nudt.edu.cn (J.H.); goodwuwei18@163.com (W.W.)

Abstract: Attribute-based signature (ABS) assures the verifier that the message is endorsed by a signer whose attributes satisfy the claimed attribute policy (predicate); thus, it can provide identity authentication with privacy preservation in scenarios like anonymous communication and access control. However, we have found that the inherent delegatibility of attribute-based cryptography, which enables the utilization of relationship between policies, could make most of the existing ABS constructions not satisfy the unforgeability requirement under the common security model. In this paper, we dig into the delegatibility property of ABS for the first time and propose the potential *delegation attack* to break the unforgeability of the existing ABS constructions under the common security model. We also give two attack instances on a typical ABS construction to demonstrate the feasibility of the proposed delegation attack. Finally, we present two solutions to improve the above issue and give a further discussion about the delegatibility property of ABS.

Keywords: attribute-based cryptography; attribute-based signature; authentication; delegation; unforgeability

MSC: 94A60; 94A62; 68P25; 68P27; 14G50

check for updates

Citation: Hao, J.; Wu, W.; Wang, S.; Zhong, X.; Chu, G.; Shao, F. A Delegation Attack Method on Attribute-Based Signatures and Probable Solutions. *Mathematics* **2023**, *11*, 29. https://doi.org/10.3390/math11010029

Academic Editors: Ding Wang, Qi Jiang and Chunhua Su

Received: 3 November 2022
Revised: 7 December 2022
Accepted: 7 December 2022
Published: 21 December 2022

1. Introduction

Among the popular primitives of public-key cryptography, digital signature effectively guarantees authenticity, integrity, and non-repudiation. In its basic form, each user is uniquely identified by his public key, and a corresponding secret key is used to demonstrate the message is signed by a certain user. Usually, a public-key infrastructure (PKI) is required to certify the relationship between the public keys and real-world user identifies. To reduce the PKI and certificate dependencies, the notion of identity-based signature (IBS) was first introduced by Shamir [1] in 1984, in which the public key of a user is simplified as his identity (e.g., name, email address, or phone number). This opens the way to more flexible and efficient digital signature schemes [2–7].

With the emerging concept of attribute-based cryptography [8–15], lots of research works have been conducted in the area of attribute-based signature (ABS) [16]. Instead of describing the signer with a unique identity as in IBS, a set of attributes are assigned to the signer by an attribute authority in ABS. A valid signature assures the verifier that the signature is generated from a signer whose attributes satisfy the claimed attribute policy (predicate) (in our context, predicate and policy have the same meaning). At the same time, the signature reveals no information about the signer identity or attributes except the claim that his attributes satisfy the public policy. With the advantages of flexibility and privacy, ABS has a good application prospect in many scenarios, such as anonymous communication, access control, message delivery and trust negotiations [17].

However, we have found that the inherent delegatibility property of attribute-based cryptography makes most of the existing ABS constructions not secure under the common unforgeability security model. The (key) delegation technique of attribute-based cryptography was first introduced by Goyal et al. [18] in their key policy attribute-based encryption (KP-ABE) scheme, with which a user is able to use a secret key corresponding to an access policy to compute a new secret key corresponding to any access policy which is more restrictive than the original one. For example, assume the original secret key is associated with the access policy "A and B", then it can be used to compute a new secret key related to the policy "A and B and C" without relying on the system key generator. Similarly, in the ciphertext policy attribute-based encryption (CP-ABE) scheme [19], the (ciphertext) delegation technique was applied on a ciphertext encrypted under a certain policy to re-encrypt it to a more restrictive policy with only public information. Thus, with the help of the delegation technique, the relationship between the policies can be exploited to derive some new and valuable components in ABE without relying on the trusted third party.

Obviously, the same case goes for ABS, in which the signature is associated with a policy. The difference is that, usually, a more restrictive policy is delegated in ABE, but it happens in the opposite direction in ABS. That is, a signature could be used to derive a new signature whose associated policy may be looser than the original one. For example, assume a signature is associated with a policy "A and B and C", which means that the signature is endorsed by a signer with at least the three attributes A, B, and C. With the delegatibility property, it seems possible to derive a new signature related to a looser policy (e.g., "A and B") from the original one.

Based on the observation that almost all of the existing ABS constructions do not consider the delegation issue in their unforgeability security model and only require a signature for a new pair of message and policy in the forge phase, some malicious attacks utilizing the delegation technique, which we call *delegation attack*, could be conducted to break the unforgeability in the ABS constructions.

In this paper, we review the security model and construction of ABS, and focus on the delegatibility issue in the ABS schemes for the first time. Our contribution can be summarized as follows.

- We first analyze the potential vulnerability related to the delegatibility property of ABS under the common unforgeability security model, and propose the delegation attack method to break the unforgeability of the existing ABS schemes.
- We then give two attack instances on a classical ABS construction [20] to show the feasibility of the delegation attack method.
- Finally, we propose two solutions to improve the above issue and give a further discussion about the delegatibility property of ABS.

The remainder of this paper is organized as follows. We first review some related preliminaries and definitions in Sections 3 and 4, respectively. In Section 5, we analyze the potential vulnerability of the existing ABS constructions, and give two concrete attack instances. Then, two improvement solutions with some discussions are proposed in Section 6. Some related works are discussed in Section 2. Finally, we draw the conclusion in Section 7.

2. Related Work

2.1. Attribute-Based Signature

Maji et al. [17] first formally defined the notion and security requirements of ABS in 2008, which offers the guarantees of strong unforgeability and privacy. An efficient construction with expressive boolean policy is proposed based on groups with bilinear pairings and proved selectively secure under the generic group model. Moreover, several potential scenarios of ABS are discussed in their scheme. Almost in the same period, Guo and Zeng [21] also proposed an ABS scheme but without the consideration of the signer privacy.

Later, Li and Kim [22] presented an ABS scheme under the standard assumption, but only an (n, n) threshold gate is supported in the policy. As an extension of [22], Shahandashti and Safavi-Naini [23] proposed the t-ABS supporting (t, n) threshold, and apply it in the anonymous credential systems. To obtain better efficiency of computation cost and signature size, Li et al. [20] proposed a new ABS construction supporting flexible threshold policy by introducing some dummy attributes. In addition, a multi-authority ABS construction is given in their scheme to further reduce the trust on attribute authority. They also constructed a non-transferable access control system as an illustrative application of ABS.

Okamoto and Takashima [24] proposed a more general non-monotone ABS scheme support not noly the traditional AND, OR, and threshold gates but also the NOT gate. Their scheme also achieves fully secure under the standard model. They also proposed a decentralized multi-authorized ABS scheme [25], in which no central authority and no trusted setup are required.

Motivated by the work of [24,25], Ge et al. [26] proposed a new general ABS construction based on the Water's CP-ABE schemes [27], with better computation efficiency and shorter signature size. Herranz et al. [28] proposed ABS with constant size signatures, but only enabling threshold predicates. Gagné et al. [29] proposed a short pairing-efficient threshold ABS, in which only three pairing operations are required in the verification algorithm and the size of the signature is independent of the number of attributes. They also discussed how to achieve shorter public parameters based on the intractability of computational Diffie-Hellman assumption in the random oracle model.

Since then, many ABS schemes with practical advantages have been proposed. In [30], a revocable ABS scheme with adaptive security in the standard model is given, which enables an external judge to break the anonymity of a signature when necessary. Similarly, Ding et al. [31] proposed a new structure and syntax for traceable ABS achieving a good balance between privacy and traceability. Given a malicious signature, the identity of the signer can be traced via the collaboration of the two trusted parties in their system. Kaafarani et al. [32] proposed decentralized traceable ABS to protect against a fully corrupted tracing authority.

To overcome the challenge for resource-limited devices to perform heavy ABS computations, Chen et al. [33] formalized a new paradigm called Outsourced ABS, in which the computational cost of the signer is greatly reduced through outsourcing intensive computations to an untrusted signing-cloud service provider. Different from that in [33], in [34], an verifying server is introduced to help the verifier to verify the signatures and reduce the computation burden.

To reduce the computational overheads for both signers and verifiers in the Internet of Things (IoT) scenario, Cui et al. [35] proposed server-aided ABS with outsourced signature generation and verification. Moreover, user revocation is achieved through enabling the server stop generating signatures for revoke users. However, Xiong et al. in [36] pointed out that the verification algorithm in [35] was insecure against the collusion attack. They also proposed a new ABS scheme, in which not only the computation in both signature generation and verification is outsourced to the server, but also the potential collusion attack is resisted. Sun et al. [37] presented an outsourced decentralized multi-authority ABS scheme, which achieves stronger multi-authority resistance and lower cost of signature generation.

On the other hand, Zhang et al. [38] introduced the online/offline technique into ABS, which splits the computation of algorithms into two phases. In the offline phase, the majority of the singing operations are executed before knowing the message and the predicate to be endorsed. In the online phase, the final signature is assembled with little cost. As a result, their scheme is more suitable for resource-constrained devices. There were also some other lightweight ABS schemes [39,40] for resource-limited devices in the mobile platform or IoT environment.

Note that all the proposed schemes does not concern the policy delegation issue, which means that they may be vulnerable (i.e, the unforgeability is broken) against our proposed delegation attack. Table 1 gives the conclusion of some related works.

In addition, from the function perspective, the attribute-based signature primitive could be combined with some blockchain-based solutions [41–44] to realize fine-grained access control and enhanced privacy preservation for the blockchain environment.

Table 1. A summary of some related works.

References	Year	Contributions
[17]	2008	first formalize ABS
[20]	2010	flexible threshold policy
[22]	2010	limited (n, n) threshold policy
[24]	2011	non-monotone policy
[25]	2013	decentralized multi-authority
[28]	2012	constant size signatures
[29]	2012	short pairing-efficient
[30]	2011	revocable
[31]	2014	traceable
[32]	2014	decentralized traceable
[33]	2014	outsourced
[34]	2021	server-aided verification
[35]	2018	server-aided verification
[37]	2019	outsourced decentralized multi-authority
[38]	2014	online/offline
[39]	2019	lightweighted
[40]	2020	lightweighted

2.2. Delegation in Attribute-Based Cryptography

Goyal et al. [18] first provided a key delegation mechanism for their KP-ABE construction, which enables individual users generate new secret keys using their secret keys and delegate them to other users. Concretely, a user with a secret key corresponding to an access policy $\mathbb{T}$ can compute a new secret key corresponding to any access policy $\mathbb{T}'$ which is more restrictive than $\mathbb{T}$ (i.e., $\mathbb{T}' \subseteq \mathbb{T}$). In this way, the users can act as a local key authority to distribute secret keys to others.

Sahai et al. [19] applied ciphertext delegation on the ABE ciphertext encrypted under a certain access policy to re-encrypt it to a more restrictive policy only with public information. A full analysis of the types of delegation in the existing ABE constructions is given in their scheme.

Blömer and Bobolz [45] introduced the notion of delegatable attribute-based anonymous credentials, which offers fine-grained anonymous access control and enables the credential holder to issue more restricted credentials to others.

Pussewalage and Oleshchuk [46] proposed a novel delegatable attribute based encryption scheme for a collaborative e-health cloud, which can enforce multi-level, controlled access delegation and be deployed in an e-health environment to securely share outsourced electronic health records (EHRs) of patients.

Joshi et al. [47] presented a novel, centralized, attribute-based authorization mechanism that uses Attribute Based Encryption (ABE) and allows for delegated secure access of patient records. Their mechanism transfers the service management overhead from the patient to the medical organization and allows easy delegation of cloud-based EHRs access authority to medical providers.

Hao et al. [48] utilized the key delegation technique in their attribute-based access control with authorized search scheme, which enables data users to customize search policies based on their access policies, and generate the corresponding trapdoor only using the secret key granted by the data owner to retrieve their interested data. In addition, the key delegation technique is also used in the self-controlled outsourced data deletion scheme [49], which enables the key update operations based on more restrictive policies such that the target data to be deleted can not be decrypted anymore.

On the other hand, although the delegation enables more flexible access control, it may also lead to the key abuse issue. Jiang et al. [50] first introduced a new mechanism to enhance CP-ABE schemes that provide protections against this key-delegation abuse issue, in which the users who have leaked their keys could be traced.

Therefore, as a unique property of attribute-based cryptography, delegatibility can be positively utilized to derive new components to enhance flexibility, and can also be adversarially abused to break security or prevent traceability. We believe that more interesting researches or applications could be conducted in this area.

3. Preliminaries

In this section, we briefly review some closely related technical preliminaries to make it easy to follow.

3.1. Access Structure

An access structure on an attribute universe U is a collection $\mathbb{C}$ of non-empty sets of attributes, i.e., $\mathbb{C} \subseteq 2^U \backslash \{0\}$. The sets in $\mathbb{C}$ are called the authorized sets, and the sets out of $\mathbb{C}$ are called the unauthorized sets. Note that an access structure is called monotone if $A \in \mathbb{C}$ and $A \subseteq B$, then $B \in \mathbb{A}$. In addition, we said that an access structure $\mathbb{C}_1$ is more restrictive than $\mathbb{C}_2$, if $\mathbb{C}_1 \subseteq \mathbb{C}_2$. That is, $\forall A \in \mathbb{C}_1$, we have $A \in \mathbb{C}_2$.

Threshold policy is a certain kind of access policy (in our context, the access structure is also referred to as access policy, and only the monotone access structures are considered), which is defined by a threshold value t and a set S of n attributes. An attribute set W satisfies a threshold policy $\mathbb{T}$ represented as (t, S) (*i.e.,* $\mathbb{T}(W) = 1$), if and only if the number of the overlapping attributes between W and S is not less than t. Alternatively, a threshold policy can be represented as an attribute S and a threshold gate (t, n), where $n = |S|$.

For example, with an attribute universe $U = [A, B, C, D, E]$, there exists a threshold policy $\mathbb{T}_1$ with $t = 3$ and $S = [A, B, C, D]$. For the attribute sets $W_1 = [A, B, C]$ and $W_2 = [A, B, E]$, we have $\mathbb{T}_1(W_1) = 1$ but $\mathbb{T}_1(W_2) = 0$. In addition, $\mathbb{T}_1$ is more restrictive than a threshold policy $\mathbb{T}_2$ with a smaller threshold value $t = 2$ and the same S, such that any attribute set satisfying $\mathbb{T}_1$ will certainly satisfy $\mathbb{T}_2$.

3.2. Bilinear Pairing

Let (p, g, G, G_T, e) be a tuple with two multiplicative cyclic groups G and G_T of a prime order p and a generator g of G. In addition, e is a bilinear pairing with the following properties:

1. **Bilinearity**: $\forall g_1, g_2 \in G, \forall x, y \in Z_p, e(g_1^x, g_2^y) = e(g_1, g_2)^{xy}$.
2. **Non-degeneracy**: $e(g, g) \neq 1$.
3. **Computability**: $\forall g_1, g_2 \in G$, the computation of $e(g_1, g_2)$ is efficient.

3.3. Lagrange Polynomial Interpolation

Assume there exists a polynomial $q(x)$ with degree $d - 1$. Given a set S of d distinct values with $q(i)$ for $i \in S$, the polynomial $q(x)$ can be represented with the Lagrange interpolating form as follows:

$$q(x) = \sum_{i \in S} q(i) \Delta_{i,S}(x),$$

where the Lagrange coefficient $\Delta_{i,S}(x)$ is:

$$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x - j}{i - j}.$$

4. Definitions

4.1. System Roles and Algorithms of ABS

As shown in Figure 1, an ABS scheme usually consists of the following four algorithms.

- $Setup(\xi, U) \rightarrow (PK, MSK)$: This is a probabilistic algorithm run by the attribute authority. It takes as input a security parameter ξ and a system attribute universe U, and outputs the system public key PK and the master secret key MSK.
- $KeyGen(MSK, W) \rightarrow SK_W$: This is a probabilistic algorithm run by the attribute authority. With the input of the master secret key MSK and an attribute set W, it generates a signing key SK_W related to W.

- $Sign(PK, M, \mathbb{T}, SK_W) \rightarrow \sigma$: This is a probabilistic algorithm run by the signer. It takes as input the system public key PK, a message M to be signed, a target policy $\mathbb{T}$ and a sining key SK_W. If $\mathbb{T}(W) = 1$, it outputs the signature σ of M under the target policy $\mathbb{T}$. Note that M and $\mathbb{T}$ are implicitly included in the signature.
- $Verify(PK, \sigma, M, \mathbb{T}) \rightarrow 1/0$: This is a deterministic algorithm run by the verifier. On input the system public key PK, the signature σ, the original message M and the target policy $\mathbb{T}$, it outputs 1 if the signature is valid, i.e., the signature is generated based on the message M and the policy $\mathbb{T}$ with a signing key whose associating attributes satisfy $\mathbb{T}$. Otherwise, it outputs 0.

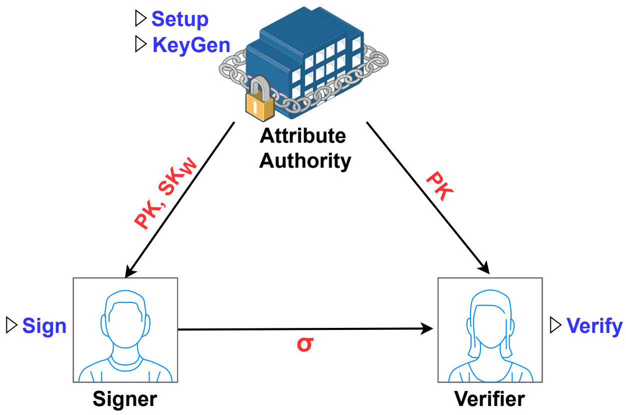

Figure 1. System roles and algorithms of ABS.

4.2. Security Model

The security requirements of ABS are mainly described as unforgeability and perfect privacy. The first one requires that a signer is not able to generate a signature out of his permission scope. The second one guarantees that the signature reveals nothing about the attributes of the signer except the claim that his attributes satisfy the policy associated with the signature.

4.2.1. Unforgeability

A common unforgeability security model which is used in almost all of the existing ABS schemes is formalized through the following game between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$:

- **Setup Phase.** With the input of a security parameter ξ and the system attribute universe U, the challenger runs the $Setup$ algorithm to generate PK and MSK. Then, it sends PK to the forger, but holds MSK itself.
- **Query Phase.** The forger $\mathcal{F}$ is allowed adaptively issue a polynomial number of queries to the $KeyGen$ and $Sign$ oracles run by the challenger. The $KeyGen$ will return the sining key related to the submitted attribute set S, and the $Sign$ oracle will return a valid signature σ based on a pair $(M, \mathbb{T})$ of the submitted message and the policy.
- **Forgery Phase.** The forger $\mathcal{F}$ outputs a signature σ^* of the message M^* with respect to a target policy $\mathbb{T}^*$.

We say that the forger $\mathcal{F}$ wins the above game with the following conditions:

- σ^* is a valid signature of the message M^* with respect to the target policy $\mathbb{T}^*$.
- Any attribute set W^* with $\mathbb{T}^*(W^*) = 1$ has not been submitted to the $KeyGen$ oracle.
- The pair of $(M^*, \mathbb{T}^*)$ has not been submitted to the $Sign$ oracle.

Note that, by enabling the forger to define the challenge policy $\mathbb{T}^*$ before the setup phase, the above full security model can be transferred into a selective one. Nevertheless, it has no effect on the proposed attacks against this unforgeability model.

- **Init Phase.** The forger selects and publishes a challenge policy $\mathbb{T}^*$ which will be included in the forgery signature.

Generally, the following two kinds of forgeries are considered for signature schemes.

Weak existential forgery. The forger can generate a minimum of one valid signature for a message without a signature (the forger usually has no control over selecting this forged message).

Strong existential forgery. The adversary can generate a valid signature, unlike any signature he has seen. Conversely to weak existential unforgeability, the corresponding message to the forged signature may have been signed already.

In our context, we mainly focus on the strong existential forgery, i.e., the forger will try to generate a signature for a message which has been queried, but associating with a different policy. If successful, the forger will win the above game.

4.2.2. Perfect Privacy

Perfect privacy of ABS means that, given any two attribute set W_1, W_2, a message M, a signature σ on the policy $\mathbb{T}$ with $\mathbb{T}(W_1) = \mathbb{T}(W_2) = 1$, any adversary $\mathcal{A}$, even with unbounded computational power and the access to the singer's secret keys, cannot distinguish which attribute set W_1 or W_2 is used to generate the signature σ with probability better than random guessing. In other words, the signature is independent of everything except the message and the policy.

5. Vulnerability Analysis and Attack Instances

In this section, we first analyze the vulnerability of the above unforgeability model for ABS and propose the potential delegation attack method to break the unforgeability. Then, we take Li's ABS construction [20] as an example and give two attack instances to demonstrate the feasibility of the proposed attack method.

5.1. Vulnerability Analysis

The key observation is that, in the above unforgeability security model, only the pair of $(m^*, \mathbb{T}^*)$ related to the final output signature is not allowed to query to the *Sign* oracle by the forger. In other words, the forger is able to request a signature for any pair of $(m^*, \mathbb{T}')$, in which the requested policy $\mathbb{T}'$ is different from the target policy $\mathbb{T}^*$.

More specifically, the forger could query to the *Sign* oracle with the target message m^* and a policy $\mathbb{T}'$ which is more restrictive than the target policy $\mathbb{T}^*$ (i.e., $\mathbb{T}' \subset \mathbb{T}^*$, but $\mathbb{T}' \neq \mathbb{T}^*$). Then, with the delegatibility property, it is possible to construct a new valid signature σ^* on $(m^*, \mathbb{T}^*)$ based on the signature σ' related to the pair of $(m^*, \mathbb{T}')$. Figure 2 shows the general steps of our proposed delegation attack method.

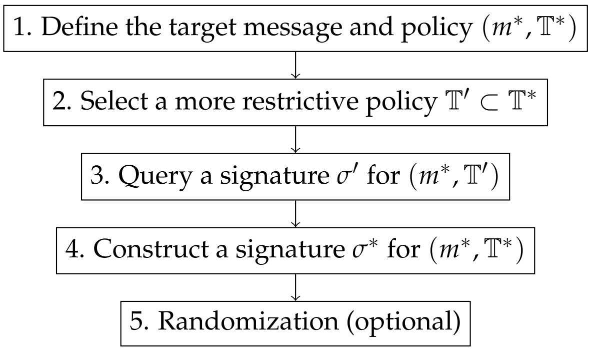
1. Define the target message and policy $(m^*, \mathbb{T}^*)$
2. Select a more restrictive policy $\mathbb{T}' \subset \mathbb{T}^*$
3. Query a signature σ' for $(m^*, \mathbb{T}')$
4. Construct a signature σ^* for $(m^*, \mathbb{T}^*)$
5. Randomization (optional)

Figure 2. General steps of delegation attack.

Considering it is relatively easier to analyze the relationship between the threshold policies, in the following we instantiate our delegation attacks on an ABS construction with threshold policies.

5.2. Review of Li's ABS Construction

In [20], Li et al. introduced $d-1$ dummy attributes and proposed an efficient ABS construction supporting flexible threshold policy with threshold value form 1 to d. Their proposed ABS construction, which was claimed to satisfy selective unforgeability and perfect privacy under the above security model, is reviewed as follows.

- $Setup(\xi, U) \rightarrow (PK, MSK)$.

It first generates the bilinear pairing (p, g, G, G_T, e) based on the security parameter ξ. Among the attributes in U defined as elements in Z_p, it selects a set Ω of $d-1$ dummy attributes. Then, it randomly picks $\alpha \in Z_p^*$ and $h \in G$, and computes $Z = e(g,h)^\alpha$. In addition, two hash functions H_1 and H_2 are selected to map the bit strings to elements in G. Finally, the public key is $PK = \langle g, h, d, Z, H_1, H_2 \rangle$, and the master secret key is $MSK = \alpha$.

- $KeyGen(MSK, W) \rightarrow SK_W$.

It first defines a random $d-1$ degree polynomial $q(x)$ with $q(0) = \alpha$. Then, for each attribute $i \in \hat{W} = W \cup \Omega$, it randomly selects $r_i \in Z_p^*$, and computes $d_{i,1} = h^{q(i)} H_1(i)^{r_i}$ and $d_{i,2} = g^{r_i}$. The signing key is generated as $SK_W = \{d_{i,1}, d_{i,2}\}_{i \in \hat{W}}$.

- $Sign(PK, M, \mathbb{T}, SK_W) \rightarrow \sigma$.

Suppose $\mathbb{T}$ is represented with a threshold t and a set S of n attributes, and $\mathbb{T}(W) = 1$, i.e., $|W \cap S| \geq t$. It selects a t-element subset $\tilde{W}$ from $W \cap S$ as the overlapping attribute set, and a subset $\tilde{\Omega} \subset \Omega$ with $d-t$ elements as the auxiliary attribute set. Note that we call the attribute set $S \setminus \tilde{W}$ as the remaining attribute set.

All the attributes used in this phase is included in the set $\tilde{S} = S \cup \tilde{\Omega}$, where $|\tilde{S}| = n + d - t$. It first selects a random value $s \in Z_p^*$, and computes $\hat{\sigma} = g^s$. Then, for each attribute $i \in \tilde{S}$, it randomly picks $s_i \in Z_p^*$, and computes

$$\sigma_0 = H_2(M)^s \prod_{i \in \tilde{W} \cup \tilde{\Omega}} d_{i,1}^{\Delta_{i, \tilde{W} \cup \tilde{\Omega}}(0)} \prod_{i \in \tilde{S}} H_1(i)^{s_i},$$

$$\sigma_i = \begin{cases} d_{i,2}^{\Delta_{i, \tilde{W} \cup \tilde{\Omega}}(0)} g^{s_i} & i \in \tilde{W} \cup \tilde{\Omega} \\ g^{s_i} & \text{others} \end{cases}.$$

Finally, the signature of the message M under the policy $\mathbb{T}$ is generated as $\sigma = \langle \sigma_0, \{\sigma_i\}_{i \in \tilde{S}}, \hat{\sigma} \rangle$.

- $Verify(PK, \sigma, M, \mathbb{T}) \rightarrow 1/0$.

With the signature $\sigma = \langle \sigma_0, \{\sigma_i\}_{i \in \tilde{S}}, \hat{\sigma}, \rangle$, it checks if the following equation holds:

$$\frac{e(g, \sigma_0)}{e(\hat{\sigma}, H_2(M)) \prod_{i \in \tilde{S}} e(\sigma_i, H_1(i))} \stackrel{?}{=} Z.$$

If it holds, it outputs 1, which means that the signature is generated from a signer whose attributes satisfy the target policy $\mathbb{T}$. Otherwise, it outputs 0 to indicate that the signature is invalid.

5.3. Attack Instances on Li's ABS Construction

We assume that the pair of the message and policy related to the final output signature in the forge phase of the unforgeability security model is $(m^*, \mathbb{T}^*)$, where $\mathbb{T}^* = (t^*, S^*)$.

In our context, we call the attributes used for Language Interpolation as the overlapping attributes, and the remaining attributes in the policy as the remaining attributes. The corresponding attribute sets are called the overlapping attribute set and the remaining attribute set, respectively.

We give the following two attack instances (note that the methods in the following attack instances could also be applied for other ABS schemes) with the different cases that the policies queried by the forger are more restrictive than the target policy $\mathbb{T}^*$.

1. $\mathbb{T}'_1 = (t^*, S^* \setminus x)$.

The forger queries to the *Sign* oracle with $(M^*, \mathbb{T}'_1)$, where $\mathbb{T}'_1 = (t^*, S')$ and $S' = S^* \setminus x$. As a response, the forger would get a valid signature $\sigma = \langle \sigma_0, \hat{\sigma}, \{\sigma_i\}_{i\in S'\cup\tilde{\Omega}} \rangle$ of $(M^*, \mathbb{T}'_1)$ as follows, in which $\tilde{W} \subset S'$ is a set of t^* attributes, $\tilde{\Omega} \subset \Omega$ is a set of $d - t^*$ dummy attributes.

$$\hat{\sigma} = g^s,$$

$$\sigma_0 = H_2(M^*)^s \prod_{i\in\tilde{W}\cup\tilde{\Omega}} d_{i,1}^{\Delta_{i,\tilde{W}\cup\tilde{\Omega}}(0)} \prod_{i\in S'\cup\tilde{\Omega}} H_1(i)^{s_i}.$$

For $i \in S' \cup \tilde{\Omega}$,

$$\sigma_i = \begin{cases} d_{i,2}^{\Delta_{i,\tilde{W}\cup\tilde{\Omega}}(0)} g^{s_i} & i \in \tilde{W}\cup\tilde{\Omega} \\ g^{s_i} & \text{others} \end{cases}.$$

To construct a signature σ^* for $(m^*, \mathbb{T}^*)$ with $\mathbb{T}^* = (t^*, S^*)$ based on σ', the forger only needs to add the components associated with the extra attribute x. Specifically, it first randomly chooses $s_x \in Z_p^*$ related to the attribute x, and computes $\sigma'_0 = \sigma_0 H_1(x)^{s_x}$. In addition, it adds $\sigma_x = g^{s_x}$ for the attribute x. Finally, the signature is constructed as $\sigma^* = \langle \sigma'_0, \hat{\sigma}, \{\sigma_i\}_{i\in S^*\cup\tilde{\Omega}} \rangle$.

Note that a randomization process is optional to make the challenger unable to identify that the signature σ^* is derived from σ. Concretely, it randomly selects s' and $\{s'_i\}_{i\in S'\cup\tilde{\Omega}}$, and lets

$$\hat{\sigma}' = \hat{\sigma} g^{s'},$$

$$\sigma''_0 = \sigma'_0 H_2(M^*)^{s'} \prod_{i\in S'\cup\tilde{\Omega}} H_1(i)^{s'_i}.$$

For $i \in S' \cup \tilde{\Omega}$,

$$\sigma'_i = \begin{cases} d_{i,2}^{\Delta_{i,\tilde{W}\cup\tilde{\Omega}}(0)} g^{s_i} g^{s'_i} & i \in \tilde{W}\cup\tilde{\Omega} \\ g^{s_i} g^{s'_i} & \text{others} \end{cases}.$$

During the verification, the pairing result of $H_1(x)^{s_x}$ and g in the numerator is equal to the pairing result of $\sigma_x = g^{s_x}$ and $H_1(x)$ in the denominator, which will not affect the verification computation result. Thus, the forger successfully constructs a valid signature σ^* for $(M^*, \mathbb{T}^*)$ based on a previously queried signature for a different pair of $(M^*, \mathbb{T}'_1)$.

Figure 3 shows an example of the relationship between the query policy and the target policy in this case. Obviously, if the attributes of a signer satisfy the query policy, it must also satisfy the target policy. With our proposed method, for the same message m, a new signature related to the target policy can be easily derived from an existing signature related to the query policy.

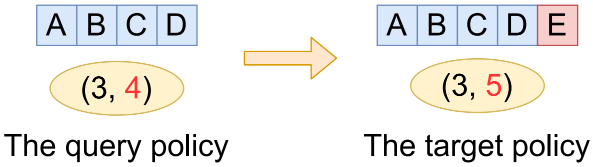

Figure 3. An example of the first case.

Note that the first case corresponds to transferring a $(t^*, n^* - 1)$ threshold gate to a (t^*, n^*) threshold gate.

2. $\mathbb{T}'_2 = (t^* + 1, S^*)$.

The forger queries to the *Sign* oracle with $(M^*, \mathbb{T}'_2)$, where $\mathbb{T}'_2 = (t^* + 1, S^*)$. As a response, the forger would get a valid signature $\sigma' = \langle \sigma_0, \hat{\sigma}, \{\sigma_i\}_{i\in S^*\cup\tilde{\Omega}} \rangle$ of $(M^*, \mathbb{T}'_2)$ as follows, in which $\tilde{W} \subset S^*$ is a set of $t^* + 1$ attributes, $\tilde{\Omega} \subset \Omega$ is a set of $d - t^* - 1$ dummy attributes.

$$\hat{\sigma} = g^s,$$

$$\sigma_0 = H_2(M^*)^s \prod_{i \in \tilde{W} \cup \tilde{\Omega}} d_{i,1}^{\Delta_{i,\tilde{W}\cup\tilde{\Omega}}(0)} \prod_{i \in S^* \cup \tilde{\Omega}} H_1(i)^{s_i}.$$

For $i \in S^* \cup \tilde{\Omega}$,

$$\sigma_i = \begin{cases} d_{i,2}^{\Delta_{i,\tilde{W}\cup\tilde{\Omega}}(0)} g^{s_i} & i \in \tilde{W} \cup \tilde{\Omega} \\ g^{s_i} & \text{others} \end{cases}.$$

Intuitively, to construct a signature σ^* for $(M^*, \mathbb{T}^*)$ with $\mathbb{T}^* = (t^*, S^*)$ based on the above signature σ' for $(M^*, \mathbb{T}_2')$, the forger needs to remove an attribute x from $\tilde{W}$ and add a dummy attribute Ω_x to $\tilde{\Omega}$. However, it seems impossible to perform the operations without the related signing key components $d_{x,1}$ and $d_{\Omega_x,1}$.

Based on our observation, even though the attribute x needs to be removed from $\tilde{W}$, it should also be included in the signature as a remaining attribute. Considering that the perfect privacy property makes it impossible for the verifier to distinguish whether an attribute belongs to the overlapping attribute set or the remaining attribute set, the forger can retain the signature component σ_x related to x and implicitly treat the additional dummy attribute Ω_x as a remaining attribute. Concretely, it first selects a random value $s_{\Omega_x} \in Z_p^*$ and computes $\sigma_0' = \sigma_0 H_1(\Omega_x)^{s_{\Omega_x}}$. Then, it adds $\sigma_{\Omega_x} = g^{s_{\Omega_x}}$ for the attribute Ω_x. Finally, the signature is constructed as $\sigma^* = \langle \sigma_0', \hat{\sigma}, \{\sigma_i\}_{i \in S^* \cup \tilde{\Omega} \cup \Omega_x} \rangle$ (a similar randomization process can also be conducted as in the first case).

During the verification, the pairing result of $H_1(\Omega_x)^{s_{\Omega_x}}$ with g in the numerator is equal to the pairing result of $\sigma_{\Omega_x} = g^{s_{\Omega_x}}$ with $H_1(\Omega_x)$ in the denominator, which will not affect the verification computation result. Thus, the forger successfully constructs a valid signature σ^* for $(M^*, \mathbb{T}^*)$ based on a previously queried signature for a different pair of $(M^*, \mathbb{T}_2')$.

Figure 4 shows an example of the relationship between the query policy and the target policy in this case.

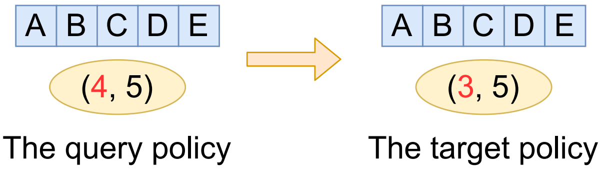

Figure 4. An example of the second case.

Note that the second case corresponds to transferring a $(t^* + 1, n^*)$ threshold gate to a (t^*, n^*) threshold gate.

6. Probable Solutions

To deal with the above proposed vulnerability, we present the following two solutions to improve the ABS scheme.

6.1. The First Solution

The first solution is binding the policy $\mathbb{T}$ with the message M tightly. Instead of only adding a explicit description of $\mathbb{T}$ in the signature as in the existing schemes, we embed $\mathbb{T}$ into the hash of the message $H(M)$.

For example, in Li's ABS construction, let

$$\sigma_0 = H_2(M||\mathbb{T})^s \prod_{i \in \tilde{W} \cup \tilde{\Omega}} d_{i,1}^{\Delta_{i,\tilde{W}\cup\tilde{\Omega}}(0)} \prod_{i \in \tilde{S}} H_1(i)^{s_i}.$$

where the notation $M||\mathbb{T}$ means the concatenation of the message M and the policy $\mathbb{T}$. Since the forger is not able to modify the component $H_2(M||\mathbb{T})^s$ without the secret s, the delegatibility property cannot be exploited again.

Note that the hash function H_2 is usually simulated via a random oracle. In the previous scheme, only the message M is sent for query of H_2, and the responses are distinguished based on whether the queried message is the target one. While in the above solution, a concatenation of the message M and the policy $\mathbb{T}$ will be queried by the adversary. In this case, the responses will be distinguished based on whether both the queried message and policy are exactly the target one. Except for this, the proof procedure is the same with the original one. Please refer to [20] for more details of the proof.

6.2. The Second Solution

The second solution is modifying the unforgeability security model, such that the forger is not allowed to query to the *Sign* oracle with the pair of $(M^*, \mathbb{T}')$, where $\mathbb{T}'$ is a more restrictive policy than the target policy $\mathbb{T}^*$. Since the capability of the forger is more limited in this modified model compared with that in the original model, the modified unforgeability security model would be a little weaker than the original one. However, based on the modified model, a new notion of delegatable Attribute-based Signature could be derived with the following *Delegate* algorithm.

- $Delegate(PK, \sigma, \mathbb{T}, \mathbb{T}') \rightarrow \sigma'$: This is a probabilistic algorithm run by anyone. It takes as input the system public key PK, an original signature σ and its related policy $\mathbb{T}$, as well as a new policy $\mathbb{T}' \supset \mathbb{T}$ (i.e., $\mathbb{T}'$ is looser than $\mathbb{T}$), and outputs a new signature related the new policy $\mathbb{T}'$ and the original message.

In delegatable ABS, deriving a new signature based on an existing signature is legal and reasonable. We believe that it can be used in some practical applications. For example, assume that a signer has already generated a signature σ for a message m with a policy $\mathbb{T}$. If he would like to generate a signature for the same message m but with a looser policy $\mathbb{T}'$, he can derive the new signature σ' from σ (with little cost) without fully computing a new one.

Note that the delegate attack method could be utilized to construct the *Delegate* algorithm in ABS. However, only two types of delegation are considered in the attack instances, i.e., from a $(t, n-1)$ threshold gate to (t, n) and from a $(t+1, n)$ threshold gate to (t, n).

Currently, for Li's construction, it seems impossible to derive a new signature related to a (t, n) threshold gate based on a signature related to a $(t+1, n+1)$ gate, which is also more restrictive than the original one. We leave this as an open problem for constructing a delegatable ABS supporting full signature delegation.

7. Conclusions

In this paper, we have focused on the delegatibility issue of attribute-based signature (ABS) for the first time. We have reviewed the security model and construction of ABS, and pointed out the potential vulnerability related to the delegatibility property of ABS under the common security model. In addition, we have proposed the delegation attack method and given two attack instances. It has been demonstrated that our proposed delegation attack method can be successfully utilized to forge new signatures based on existing signatures, such that the unforgeability requirement in the common security model cannot be satisfied by most of the existing ABS constructions. Finally, we have also presented two solutions to improve the above issue in the existing schemes and derived a new notion of delegatable attribute-based signature with independent interest.

The future directions of our work mainly includes two parts. On one hand, we will try to propose a new delegatable ABS construction supporting full signature delegation from an original policy to a looser policy, give a practical implementation, and compare it with the existing schemes in terms of not only security but also efficiency and functionality. On the other hand, we will try to combine the advantages of ABS with blockchain to realize fine-grained access control and enhanced privacy preservation for the blockchain environment.

Author Contributions: Conceptualization, J.H. and S.W.; formal analysis, X.Z. and G.C.; writing—original draft preparation, J.H. and S.W.; writing—review and editing, X.Z. and G.C.; supervision, W.W. and F.S.; funding acquisition, W.W. and F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Henan Key Laboratory of Network Cryptography Technology with grant number LNCT2022-A16. It was also funded by National Natural Science Foundation of China with grant number 62102447.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers for their careful reading of the manuscript and their insightful comments and suggestions. They would also like to thank F. Zhang and Q. He from Xi'an Satellite Control Center for their help.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Santa Barbara, CA, USA, 19–22 August 1984; Springer: Berlin/Heidelberg, Germany, 1984; pp. 47–53.
2. Gu, Y.; Shen, L.; Zhang, F.; Xiong, J. Provably Secure Linearly Homomorphic Aggregate Signature Scheme for Electronic Healthcare System. *Mathematics* **2022**, *10*, 2588. [CrossRef]
3. Choon, J.C.; Hee Cheon, J. An identity-based signature from gap Diffie-Hellman groups. In Proceedings of the International Workshop on Public Key Cryptography, Miami, FL, USA, 6–8 January 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 18–30.
4. Galindo, D.; Garcia, F.D. A Schnorr-like lightweight identity-based signature scheme. In Proceedings of the International Conference on Cryptology in Africa, Gammarth, Tunisia, 21–25 June 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 135–148.
5. Kóczy, L.T.; Susniene, D.; Purvinis, O.; Konczosné Szombathelyi, M. A New Similarity Measure of Fuzzy Signatures with a Case Study Based on the Statistical Evaluation of Questionnaires Comparing the Influential Factors of Hungarian and Lithuanian Employee Engagement. *Mathematics* **2022**, *10*, 2923. [CrossRef]
6. Yang, P.; Cao, Z.; Dong, X. Fuzzy identity based signature with applications to biometric authentication. *Comput. Electr. Eng.* **2011**, *37*, 532–540. [CrossRef]
7. Galindo, D.; Herranz, J.; Kiltz, E. On the generic construction of identity-based signatures with additional properties. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, 3–7 December 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 178–193.
8. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 457–473.
9. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 321–334.
10. Hao, J.; Huang, C.; Ni, J.; Rong, H.; Xian, M.; Shen, X.S. Fine-grained data access control with attribute-hiding policy for cloud-based IoT. *Comput. Netw.* **2019**, *153*, 1–10. [CrossRef]
11. Garcia-Grau, F.; Herrera-Joancomartí, J.; Dorca Josa, A. Attribute Based Pseudonyms: Anonymous and Linkable Scoped Credentials. *Mathematics* **2022**, *10*, 2548. [CrossRef]
12. Chinnasamy, P.; Deepalakshmi, P.; Dutta, A.K.; You, J.; Joshi, G.P. Ciphertext-Policy Attribute-Based Encryption for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System. *Mathematics* **2021**, *10*, 68. [CrossRef]
13. Hao, J.; Tang, W.; Huang, C.; Liu, J.; Wang, H.; Xian, M. Secure data sharing with flexible user access privilege update in cloud-assisted IoMT. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 933–947. [CrossRef]
14. Yang, E.; Parvathy, V.S.; Selvi, P.P.; Shankar, K.; Seo, C.; Joshi, G.P.; Yi, O. Privacy Preservation in Edge Consumer Electronics by Combining Anomaly Detection with Dynamic Attribute-Based Re-Encryption. *Mathematics* **2020**, *8*, 1871. [CrossRef]
15. Hao, J.; Huang, C.; Liu, J.; Xian, M.; Shen, X. Efficient outsourced data access control with user revocation for cloud-based IoT. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
16. Oberko, P.S.K.; Obeng, V.H.K.S.; Xiong, H.; Kumari, S. A survey on Attribute-Based Signatures. *J. Syst. Archit.* **2022**, *124*, 102396. [CrossRef]
17. Maji, H.; Prabhakaran, M.; Rosulek, M. Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance. Cryptology ePrint Archive. 2008. Available online: https://eprint.iacr.org/2008/328.pdf?origin%3Dpublication_detail (accessed on 30 October 2022).

18. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98.
19. Sahai, A.; Seyalioglu, H.; Waters, B. Dynamic credentials and ciphertext delegation for attribute-based encryption. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 199–217.
20. Li, J.; Au, M.H.; Susilo, W.; Xie, D.; Ren, K. Attribute-based signature and its applications. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, Beijing, China, 13–16 April 2010; pp. 60–69.
21. Shanqing, G.; Yingpei, Z. Attribute-based signature scheme. In Proceedings of the 2008 International Conference on Information Security and Assurance (ISA 2008), Busan, Republic of Korea, 24–26 April 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 509–511.
22. Li, J.; Kim, K. Hidden attribute-based signatures without anonymity revocation. *Inf. Sci.* **2010**, *180*, 1681–1689. [CrossRef]
23. Shahandashti, S.F.; Safavi-Naini, R. Threshold attribute-based signatures and their application to anonymous credential systems. In Proceedings of the International Conference on Cryptology in Africa, Gammarth, Tunisia, 21–25 June 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 198–216.
24. Okamoto, T.; Takashima, K. Efficient Attribute-Based Signatures for Non-monotone Predicates in the Standard Model. In Proceedings of the International Workshop on Public Key Cryptography, Taormina, Italy, 6–9 March 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 35–52.
25. Okamoto, T.; Takashima, K. Decentralized attribute-based signatures. In Proceedings of the International Workshop on Public Key Cryptography, Nara, Japan, 26 February–1 March 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 125–142.
26. Ge, A.; Chen, C.; Ma, C.; Zhang, Z. Short and Efficient Expressive Attribute-Based Signature in the Standard Model. Cryptology ePrint Archive. 2012. Available online: https://eprint.iacr.org/2012/125 (accessed on 30 October 2022).
27. Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Proceedings of the International Workshop on Public Key Cryptography, Taormina, Italy, 6–9 March 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 53–70.
28. Herranz, J.; Laguillaumie, F.; Libert, B.; Rafols, C. Short attribute-based signatures for threshold predicates. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 27 February–2 March 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 51–67.
29. Gagné, M.; Narayan, S.; Safavi-Naini, R. Short pairing-efficient threshold-attribute-based signature. In Proceedings of the International Conference on Pairing-Based Cryptography, Cologne, Germany, 16–18 May 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 295–313.
30. Escala, A.; Herranz, J.; Morillo, P. Revocable attribute-based signatures with adaptive security in the standard model. In Proceedings of the International Conference on Cryptology in Africa, Dakar, Senegal, 5–7 July 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 224–241.
31. Ding, S.; Zhao, Y.; Liu, Y. Efficient traceable attribute-based signature. In Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, Beijing, China, 24–26 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 582–589.
32. Kaafarani, A.E.; Ghadafi, E.; Khader, D. Decentralized traceable attribute-based signatures. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 25–28 February 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 327–348.
33. Chen, X.; Li, J.; Huang, X.; Li, J.; Xiang, Y.; Wong, D.S. Secure outsourced attribute-based signatures. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 3285–3294. [CrossRef]
34. Chen, Y.; Li, J.; Liu, C.; Han, J.; Zhang, Y.; Yi, P. Efficient attribute based server-aided verification signature. *IEEE Trans. Serv. Comput.* **2021**, *6*, 3224–3232. [CrossRef]
35. Cui, H.; Deng, R.H.; Liu, J.K.; Yi, X.; Li, Y. Server-aided attribute-based signature with revocation for resource-constrained industrial-internet-of-things devices. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3724–3732. [CrossRef]
36. Xiong, H.; Bao, Y.; Nie, X.; Asoor, Y.I. Server-aided attribute-based signature supporting expressive access structures for industrial internet of things. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1013–1023. [CrossRef]
37. Sun, J.; Su, Y.; Qin, J.; Hu, J.; Ma, J. Outsourced decentralized multi-authority attribute based signature and its application in IoT. *IEEE Trans. Cloud Comput.* **2019**, *9*, 1195–1209. [CrossRef]
38. Zhang, S.; Chen, P.; Wang, J. Online/offline attribute based signature. In Proceedings of the 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, Guangdong, China, 8–10 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 566–571.
39. Lin, G.; Xia, Y.; Ying, C.; Sun, Z. F2p-abs: A fast and secure attribute-based signature for mobile platforms. *Secur. Commun. Netw.* **2019**, *2019*, 5380710. [CrossRef]
40. Yu, J.; Liu, S.; Wang, S.; Xiao, Y.; Yan, B. LH-ABSC: A lightweight hybrid attribute-based signcryption scheme for cloud-fog-assisted IoT. *IEEE Internet Things J.* **2020**, *7*, 7949–7966. [CrossRef]
41. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R.; Srivastava, G. P2tif: A blockchain and deep learning framework for privacy-preserved threat intelligence in industrial iot. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6358–6367. [CrossRef]
42. Kumar, R.; Kumar, P.; Aljuhani, A.; Islam, A.N.; Jolfaei, A.; Garg, S. Deep learning and smart contract-assisted secure data sharing for IoT-based intelligent agriculture. *IEEE Intell. Syst.* **2022**, 1–8. [CrossRef]

43. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R.; Jolfaei, A.; Islam, A.N. A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system. *J. Parallel Distrib. Comput.* **2023**, *172*, 69–83. [CrossRef]
44. Kumar, P.; Kumar, R.; Kumar, A.; Franklin, A.A.; Garg, S.; Singh, S. Blockchain and Deep Learning for Secure Communication in Digital Twin Empowered Industrial IoT Network. *IEEE Trans. Netw. Sci. Eng.* **2022**, 1–13. [CrossRef]
45. Blömer, J.; Bobolz, J. Delegatable attribute-based anonymous credentials from dynamically malleable signatures. In Proceedings of the International Conference on Applied Cryptography and Network Security, Leuven, Belgium, 2–4 July 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 221–239.
46. Pussewalage, H.S.G.; Oleshchuk, V. A Delegatable Attribute Based Encryption Scheme for a Collaborative E-health Cloud. *IEEE Trans. Serv. Comput.* **2022**, 1. [CrossRef]
47. Joshi, M.; Joshi, K.P.; Finin, T. Delegated authorization framework for EHR services using attribute based encryption. *IEEE Trans. Serv. Comput.* **2019**, *14*, 1612–1623. [CrossRef]
48. Hao, J.; Liu, J.; Wang, H.; Liu, L.; Xian, M.; Shen, X. Efficient attribute-based access control with authorized search in cloud storage. *IEEE Access* **2019**, *7*, 182772–182783. [CrossRef]
49. Hao, J.; Liu, J.; Wu, W.; Tang, F.; Xian, M. Secure and fine-grained self-controlled outsourced data deletion in cloud-based IoT. *IEEE Internet Things J.* **2019**, *7*, 1140–1153. [CrossRef]
50. Jiang, Y.; Susilo, W.; Mu, Y.; Guo, F. Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing. *Future Gener. Comput. Syst.* **2018**, *78*, 720–729. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.