Information Sciences 180 (2010) 1681–1689

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Hidden attribute-based signatures without anonymity revocation

Jin Li *, Kwangjo Kim

Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), 119 Munjiro, Yusong-ku, Daejeon 305-714, Republic of Korea

ARTICLE INFO

Article history:
Received 16 September 2008
Received in revised form 7 December 2009
Accepted 5 January 2010

Keywords:
Signature
Attribute-based
Anonymity
Computational Diffie–Hellman assumption

ABSTRACT

We propose a new notion called hidden attribute-based signature, which is inspired by the recent developments in attribute-based cryptosystem. With this technique, users are able to sign messages with any subset of their attributes issued from an attribute center. In this notion, a signature attests not to the identity of the individual who endorsed a message, but instead to a claim regarding the attributes the underlying signer possesses. Users cannot forge signature with attributes which they have not been issued. Furthermore, signer remains anonymous without the fear of revocation, among all users with the attributes purported in the signature.

After formalizing the security model, we propose two constructions of hidden attribute-based signature from pairings. The first construction supports a large universe of attributes and its security proof relies on the random oracle assumption, which can be removed in the second construction. Both constructions have proven to be secure under the standard computational Diffie–Hellman assumption.

1. Introduction

Identity-based cryptosystem is a public key cryptosystem where the public key can be an arbitrary string such as an email address or a registration number, etc. The concept was proposed by Shamir [22] to simplify the key management procedures of the certificate-based public key infrastructures. A private key generator, which is believed to be a highly trusted third party, utilizes his master secret key to issue a private key to every identity who needs the private key of her own. To encrypt a message, the user only needs to know the identity of the receiver. Upon receiving the ciphertext, the receiver can decrypt and get the message with his private key. Fuzzy identity-based encryption (IBE) [20], as a related notion to identity-based cryptosystem, was proposed by Sahai and Waters. Before decryption, users need to get their attributes from an attribute center. In this system, the user with a private key for attribute identity ω can decrypt a ciphertext encrypted with attribute set ω' if and only if ω and ω' are within a certain distance of each other as judged by some metric. Recently, interest has grown in the attribute-based cryptosystem, which is an important application of fuzzy IBE. With this technique, a message can be encrypted to specified users possessing a certain set of attributes. For example, suppose a message is encrypted to attributes "University A", "Faculty", and "Department of Computer Science" with the policy that any user possessing two of these attributes can decrypt. As a result, three kinds of users can decrypt, i.e., faculty in University A, faculty in department of computer science, or faculty of department of computer science in University A.

1.1. Our contributions

Inspired by the recent developments in attribute-based cryptosystem [20], we introduce a new signature notion which is called hidden attribute-based signature. There are two entities in this system, i.e., attribute center and users. The hidden

* Corresponding author.
E-mail address: jjl@icu.ac.kr (J. Li).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.01.008

attribute-based signature is designed for the following situation: A user gets a certificate for the set of attributes ω from the attribute center. To claim that a message is endorsed by $\omega' \subseteq \omega$, the user signs the message with his secret key of attributes. The security requirements include anonymity and unforgeability. Anonymity means that no one can tell who generates the signature among the users with the attribute set ω', even if many signatures from the same signer using the same subset of attributes are provided. On the other hand, unforgeability requires that no one can forge the signature with the attribute set ω' if he has not been issued the certificate for ω'.

Let's consider the following application: Suppose Bob has the set of attributes {"University A", "Faculty", "Department of Computer Science"}. Bob wishes to complain or give some suggestions to an administrator of University A in such a way that Bob remains anonymous. However, the administrator believes that such complaints or suggestions are indeed from some faculty member of University A. To achieve this, Bob could sign the complaints or suggestions with attributes "University A" and "Faculty" by using the hidden attribute-based signature. The validity of the signature can be verified such that the administrator will be convinced that it is indeed from some faculty of University A, without knowing the signer's identity.

In this paper, the formal definitions and the security model of hidden attribute-based signatures are suggested. We also propose two constructions based on pairings. The first construction has proven to be secure under the random oracle assumption, while the second one is secure without the random oracle by utilizing the technique of [24].

1.2. Related work

1.2.1. Attribute-based encryption

Inspired by the significant work of attribute-based encryption [20], many improvements and extensions [1,3,7,14] were presented in the open literature. Baek et al. [1] showed how to shorten the public parameters of [20], but the scheme could only be proven to be secure in the random oracle model. Attribute-based encryption can be utilized to realize flexible and scalable access control systems [11,20]. In order to get fine-grained access control, Goyal et al. [14] proposed the notion of key-policy attribute-based encryption. In this system, each private key is associated with a more flexible access structure that specifies which type of ciphertexts the key can decrypt. Instead of determining the decrypting policy in private key, Bethencourt et al. [3] formalized the notion of ciphertext-policy attribute-based encryption and provided a construction. In ciphertext-policy attribute-based encryption, the encryptor can specify an associated access structure such that only the users with attributes that satisfy this access structure can decrypt the ciphertext. Actually, the notion of ciphertext-policy attribute-based encryption was first mentioned by Goyal et al. in [14]. Later, Chase [7] proposed a multi-authority attribute-based encryption scheme to reduce trust on attribute authority, where each authority issues only a part of the attributes.

1.2.2. Attribute-based signature

Recently, there have been several attempts to construct attribute-based signatures. As a similar notion, fuzzy identity-based signature was proposed and formalized in [6,25], which allows a user with attribute identity ω to sign with part of his attributes. The verifier can check if the signature is signed by some user with these attributes. To achieve the same goal as the fuzzy identity-based signature, the notion of attribute-based signature was given in [13]. However, these kinds of signatures do not take the anonymity of the signer into account. Given the relation between the attribute-based cryptosystem and the identity-based cryptosystem, such kind of signature scheme could be trivial to construct by using the method given by Galindo et al. [12]. Another work on attribute-based signature was [18], in which they presented an attribute signature scheme that achieves signer's privacy. In this signature scheme, the user can generate a signature with a flexible number of attributes issued from the attribute center, while keeping the signer anonymous. However, the security can only be proved in a non-standard hardness assumption and the generic group model.

1.2.3. Attribute-based group signature

Khader [15] proposed a notion called attribute-based group signature. It allows a verifier to request a signature from a member of a group who possesses certain attributes, and the signature can prove the ownership of certain attributes. When necessary, the identity of the signer could be revealed by a designated manager.

1.2.4. Hidden identity-based signature

Another related notion is hidden identity-based signature, which was proposed by Kiayias and Zhou [16]. It is similar to group signature which can achieve revocable anonymity for group members, through an opening authority. In hidden identity-based signature, the group membership list is not published and the opening authority is independent of group manager for anonymity revocation, which is different from the ordinary group signature. In our hidden attribute-based signature, the user list is not public. However, in the proposed notion of hidden attribute-based signature, there is no opening authority who can reveal the hidden identity, that is, no anonymity revocation exists.

1.2.5. Ring signature

Ring signature [19] allows the user to sign messages on behalf of a "ring" of legitimate signers without revealing the signer's identity. Different from the group signature (for examples, [2,8]), the ring signature has spontaneous group formation, in which there is no group manager to revoke the identity of the signer. Under the assumption that each user is previously

associated with a public key, the signer can choose a group arbitrarily by simply collecting the public keys of all the "ring" members, including his own. These diverse members can be totally unaware of being included in the group. Ring signature schemes can be used for whistle blowing and anonymous membership authentication [19] in order to keep the anonymity of the signer. Since ring signature was first formalized by Rivest et al. [19], many practical ring signature schemes and their variants have been proposed, such as threshold ring signature [5], identity-based ring signature [10], ring signature with signer-admission [23] and proxy ring signature [17]. The first efficient ring signature scheme based on standard assumptions without random oracles was proposed by Shacham and Waters [21]. In their ring signature scheme, it requires setup assumption and provides only computational anonymity.

A major feature of ring signature is that the user can generate a signature corresponding to "something" that he does not know. For example, in identity-based ring signature, the signer with identity ID can give a signature that convinces the verifier that the signature either comes from ID or ID′. In the proposed hidden attribute-based signature, the private key of ID′ is unknown and the signer can only generate a signature with respect to the attributes that he has. Another important difference is, for example, if there is only one CEO in a company who wants to issue a hidden attribute-based signature, his identity can only be hidden based on the "condition" that there can be more than one (co-)CEO in a company. In ring signature, a CEO can choose to involve other non-CEOs in a signature to show that a certain document is possibly signed by a CEO. In the proposed hidden attribute-based signature, the CEO either gives a signature with the CEO's attribute, or chooses to hide this, which makes the signature have nothing to do with the attribute "CEO". In the previous ring signatures, the signer and verifier know who is included in the signature. This is different with the proposed hidden attribute-based signatures.

2. Preliminaries

In this paper, bilinear pairings are used on elliptic curves. A brief review is given here on the property of pairings and some candidate hard problems from pairings. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic groups of prime order p with the multiplicative group action. Also, g is a generator of $\mathbb{G}_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a map with the following properties:

1. Bilinearity: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1, g_2 \in \mathbb{G}_1$, and $a, b \in_R \mathbb{Z}_p$;
2. Non-degeneracy: There exist $g_1, g_2 \in \mathbb{G}_1$ such that $e(g_1, g_2) \neq 1$. In other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$;
3. Computability: There is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}_1$.

Definition 1 (*CDH Problem*). The Computational Diffie–Hellman problem is that, given $g, g^x, g^y \in \mathbb{G}_1$ for unknown $x, y \in \mathbb{Z}_p^*$, to compute g^{xy}.

We say that the (t, ϵ)-CDH assumption holds in $\mathbb{G}_1$ if no t-time algorithm has the non-negligible probability ϵ in solving the CDH problems

3. Hidden attribute-based signature scheme

In this section, we formalize the definition and security model of hidden attribute-based signature. Then, we propose a construction with security proof under the given model. In the hidden attribute-based signature scheme, there is an attribute center to issue private keys to every user who requests them.

3.1. Syntax

The hidden attribute-based signature scheme consists of four algorithms, namely, Setup, Extract, Sign, and Verify, which are defined as follows.

- Setup The setup algorithm, on input 1^λ, where λ is the security parameter, outputs public parameter params and sk as the master secret key for the attribute center.
- Extract The private key extraction algorithm, on input attributes ω and the master key sk, outputs a private key sk_ω.
- Sign The signing algorithm, to obtain a signature on a message m with respect to attributes $\omega' \subseteq \omega$, takes as input private key sk_ω for attributes ω and outputs signature σ.
- Verify The verification algorithm, given an alleged signature σ for m with respect to attributes ω', and params, checks if it is a valid signature. If it is valid, outputs 1. Otherwise, outputs 0.

3.2. Security requirements

There are two security requirements for hidden attribute-based signatures, i.e., unforgeability and anonymity. Their definitions are described as follows.

3.2.1. Unforgeability

The definition for unforgeability requires that no user can endorse a message with an attribute set ω^* if he does not have an attribute private key for ω such that $\omega^* \subseteq \omega$. That is, the signature satisfies unforgeability if it is existentially unforgeable against adaptive-attribute and chosen message attacks. The definition for unforgeability also implies its security against collusion attacks, in which a group of users try to combine their private keys and sign a message with some attributes they could not do individually. In this work, we consider a weaker security notion under selective-attribute attack. Specifically, the adversary should select the challenge attributes before setup. This attack model has also been used in many protocols such as [4,9,10,17]. We propose the definition of unforgeability that is existentially unforgeable against selective-attribute and chosen message attacks (EUF-sA-CMA). Two oracles are provided to the adversary: (1) Private Key Extraction Oracle. Given an attribute set ω, output corresponding private key sk_ω; (2) Signing Oracle. Given a message m and ω, output signature σ. If the security is proved in the random oracle model, another oracle should also be provided to the adversary: (3) Random Oracle. Given m, output a random value r.

As explained, the challenge attributes are fixed for some predefined maximum number, for example, d. The formal definition of unforgeability is based on the following EUF-sA-CMA game involving a challenger $\mathscr{C}$ and an adversary $\mathscr{F}$: At first, $\mathscr{F}$ outputs its challenge set of attributes $|\omega^*| \leqslant d$ for some predefined number d. Then, $\mathscr{C}$ chooses a sufficiently large security parameter 1^λ and runs Setup. $\mathscr{C}$ retains private key sk and sends params generated from Setup to $\mathscr{F}$. $\mathscr{F}$ can perform a polynomially bounded number of queries m, ω, and (m', ω') with $|\omega'| \leqslant d$ to random oracle, private key extraction oracle and signing oracle, respectively. The restriction of the private key extraction query on ω should satisfy $\omega^* \not\subseteq \omega$. $\mathscr{F}$ outputs a signature σ^* on messages m^* with respect to the set of attributes ω^*. We say that the adversary wins the game if σ^* is a valid signature on message m^* with respect to ω^*, and (m^*, ω^*) has not been queried to the signing oracle. The advantage $\mathbf{Adv}^{euf}_{HAS,\mathscr{F}}(1^\lambda)$ of $\mathscr{F}$ is defined as the probability that it wins the game.

Definition 2 (*Unforgeability*). A forger $\mathscr{F}(t, q_K, q_S, q_H, \epsilon)$-breaks a hidden attribute-based signature scheme if $\mathscr{F}$ runs in time at most t, and makes at most q_K private key extraction queries, q_S signature queries and q_H hash queries, while $\mathbf{Adv}^{euf}_{HAS,\mathscr{F}}(1^\lambda)$ is at least ϵ. A hidden attribute-based signature scheme is $(t, q_K, q_S, q_H, \epsilon)$-existentially unforgeable under selective-attribute and adaptive chosen message attacks if there exists no forger that can $(t, q_K, q_S, q_H, \epsilon)$-break it.

3.2.2. Anonymity

For anonymity, we require that the signer must be anonymous among the users with the same attributes purported in the signature. Moreover, even the attribute center cannot reveal the signer's identity from the signature. The adversary can query signatures on messages with respect to certain attributes belonging to two attribute sets. As both attribute sets could generate a signature with the same attribute subset, the adversary has to guess which one signs the message, even if the adversary has the private key for both attribute sets. Its formal definition is based on the following game between a challenger $\mathscr{C}$ and an adversary $\mathscr{F}$.

The challenger $\mathscr{C}$ chooses a sufficiently large security parameter 1^λ and runs Setup to get a master key sk and public parameters params. $\mathscr{C}$ sends sk and params to $\mathscr{F}$. With the private key sk, $\mathscr{F}$ can generate private keys and signatures by himself. $\mathscr{F}$ outputs a message m^*, two attribute sets ω_1^*, ω_2^*, and challenged attribute ω^* for signature query, where $\bar{\omega}^* = \omega_1^* \cap \omega_2^*$ and, $\omega^* \subseteq \bar{\omega}^*$ such that $|\omega^*| \leqslant d$. Assume that $\mathscr{F}$ has queried private key extractions to two sets of attributes ω_1^* and ω_2^*. The private keys for ω_1^* and ω_2^* are $sk_{\omega_1^*}$ and $sk_{\omega_2^*}$, respectively. $\mathscr{C}$ chooses randomly $b \in \{0, 1\}$, computes the challenged signature $\sigma^* = \mathsf{Sign}(m^*, \omega^*, sk_{\omega_b^*})$ and provides σ^* to $\mathscr{F}$. With σ^*, $\mathscr{F}$ guesses whether the signature is generated from ω_1^* or ω_2^*. Finally, $\mathscr{F}$ outputs a bit b' as his guess. We say $\mathscr{F}$ wins the game if $b' = b$. Define $\mathbf{Adv}^{anony}_{HAS,\mathscr{F}}(1^\lambda)$ to be the advantage over 1/2 of $\mathscr{F}$ in the above game. The master key of attribute center is also given to the adversary. This means that the signer's anonymity holds even to the attribute center.

Definition 3 (*Anonymity*). A hidden attribute-based signature scheme satisfies the anonymity requirement if no $\mathscr{F}$ can win the above game with non-negligible advantage $\mathbf{Adv}^{anony}_{HAS,\mathscr{F}}(1^\lambda)$.

At first glance, it seems trivial to construct such a protocol just by preparing one private key for each signing set of attributes ω' (Only preparing one private key for each attribute $i \in \mathbb{Z}_p$, instead of a set of attributes ω', will not provide security against collusion attacks in which a group of users could combine their private keys and break the security requirement of unforgeability defined.). However, if the number of attributes in universe is ℓ, we can calculate that the number of all the possible attribute subsets for signing is at least $\binom{\ell}{d}$. As a result, the attribute center has to publish at least $\binom{\ell}{d}$ public keys. Obviously, it cannot be realized in a polynomial time in case the universe of attributes are chosen from $\mathbb{Z}_p$. However, our construction requires only to publish $4 + O(d)$ elements. The number $\binom{\ell}{d}$ is huge even for small ℓ and d, for example, when the number of attributes is $\ell = 50$ and $d = 10$, it will be approximately 10^{10}, instead of $\ell + O(d)$ in the second construction.

3.3. Our hidden attribute-based signature construction

In our construction, the signer can efficiently generate a signature with part of his attributes. A predefined number d is given before the setup algorithm. The number d should be large enough for the practical applications because in our system,

the user can sign a message with the number of attributes from 1 to d. To achieve this, the technique of dummy attributes was utilized, which was also used in [7,20].

Some preliminaries on Lagrange interpolation are also given here before the construction. Recall that, given d points $q(1), q(2), \ldots, q(d)$ on a $d-1$ degree polynomial, Lagrange interpolation can be used to compute $q(i)$ for any $i \in \mathbb{Z}_p$. Let S be a d-element set. The Lagrange coefficient is defined as $\Delta_{j,S}(i)$ of $q(j)$ in the computation of $q(i)$ as $\Delta_{j,S}(i) = \prod_{\eta \in S, \eta \neq j} \frac{i-\eta}{j-\eta}$.

Setup (d) First, define the universe of attributes U as $\mathbb{Z}_p$. A $d-1$ default set of attributes from $\mathbb{Z}_p$, $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{d-1}\}$, is also given. Then, select a random generator $g \in \mathbb{G}_1$ and a random $x \in \mathbb{Z}_p^*$. Let $g_1 = g^x$. Next, pick a random element $g_2 \in \mathbb{G}_1$ and compute $Z = e(g_1, g_2)$. Two hash functions are also chosen such that $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$. The public parameters are $\text{params} = (g, g_1, g_2, Z, d, H_1, H_2)$ and the master key is x.

Extract To generate a private key for set of attributes ω, the following steps are taken:

- First, choose a $d-1$ degree polynomial q such that $q(0) = x$;
- Generate a new attribute set $\hat{\omega} = \omega \cup \Omega$. For each $i \in \hat{\omega}$, choose $r_i \in_R \mathbb{Z}_p$;
- Then, compute $d_{i0} = g_2^{q(i)} \cdot (H_1(i))^{r_i}$ and $d_{i1} = g^{r_i}$;
- Finally, output the private key $D_i = (d_{i0}, d_{i1})$ for each $i \in \hat{\omega}$.[1]

Sign Suppose that one has a private key for the attribute set ω. To sign a message m with attribute set $\omega' = \{i_1, i_2, \ldots, i_k\} \subseteq \omega$, where $1 \leqslant k \leqslant d$, proceed as follows:

- Select a $d-k$ default attribute subset $\Omega' = \{i_{k+1}, i_{k+2}, \ldots, i_d\} \subseteq \Omega$. Then, choose $r'_1, r'_2, \ldots, r'_d, s_1, s_2, \ldots, s_d \in \mathbb{Z}_p$ and a $d-1$ degree polynomial function $q'(x)$ such that $q'(0) = 0$;
- For $1 \leqslant v \leqslant d$, compute $\sigma_{v1} = d_{i_v 0} H_1(i_v)^{r'_v} g_2^{q'(i_v)} H_2(m)^{s_v}$, $\sigma_{v2} = d_{i_v 1} g^{r'_v}$, and $\sigma_{v3} = g^{s_v}$;
- Finally, output the signature $\sigma = \{(\sigma_{v1}, \sigma_{v2}, \sigma_{v3})\}_{1 \leqslant v \leqslant d}$.

Verify To verify the signature $\sigma = \{(\sigma_{v1}, \sigma_{v2}, \sigma_{v3})\}_{1 \leqslant v \leqslant d}$ on message m for attributes $\omega' = (i_1, \ldots, i_k)$ with default attributes Ω', check the following equation: $\prod_{v=1}^{d} \left(\frac{e(g, \sigma_{v1})}{e(H_1(i_v), \sigma_{v2}) e(H_2(m), \sigma_{v3})} \right)^{\Delta_{i_v, S}(0)} = Z$. If it holds, then output 1; Otherwise, output 0.

3.4. Correctness and efficiency analysis

The correctness of verification is justified by the following equation: For $1 \leqslant v \leqslant d$, we have

$$\frac{e(\sigma_{v1}, g)}{e(H_1(i_v), \sigma_{v2}) e(H_2(m), \sigma_{v3})} = \frac{e(g_2^{q(i_v)+q'(i_v)} \cdot H_1(i_v)^{(r_{i_v}+r'_v)} H_2(m)^{s_v}, g)}{e(H_1(i_v), g^{(r_{i_v}+r'_v)}) e(H_2(m), g^{s_v})} = \frac{e(g_2^{q(i_v)+q'(i_v)}, g) e(H_1(i_v)^{r_{i_v}+r'_v}, g) e(H_2(m)^{s_v}, g)}{e(H_1(i_v), g^{(r_{i_v}+r'_v)}) e(H_2(m), g^{s_v})}$$

$$= e(g_2, g)^{q(i_v)+q'(i_v)} = e(g^{xy}, g)$$

Therefore, we have $[e(g_2, g)^{q_{i_1}+q'(i_1)}]^{\Delta_{i_1,S}(0)} \cdots [e(g_2, g)^{q_{i_d}+q'(i_d)}]^{\Delta_{i_d,S}(0)} = Z$.

Actually, the key structure of the above construction is similar to the one in [14]. To generate a signature, d exponentiations and $2d$ multi-exponentiation computations in group $\mathbb{G}_1$ are required. The signature consists of $3d$ group elements, no matter what the size of the attribute set ($k \leqslant d$) is. Here d also denotes the number of default attributes. In verification algorithm, $3d$ pairings and one multi-exponentiation computations in $\mathbb{G}_1$ are required.

3.5. Security results

Theorem 1. *The proposed hidden attribute-based signature scheme satisfies anonymity.*

Proof. First, the challenger runs Setup to get the public parameters params and the master key x. The challenger also gives the adversary params and x. After these interactions, the adversary outputs two attributes, ω_1^* and ω_2^*, where $\bar{\omega}^* = \omega_1^* \cap \omega_2^*$. Notice that the private key for each user should include the $d-1$ default attribute set Ω. Let $\widehat{\omega_b^*} = \omega_b^* \cup \Omega$ for $b \in \{1,2\}$. Assume that the challenger or adversary has generated the private keys as $sk_{\widehat{\omega_1^*}} = \left(d_{i0}^1, d_{i1}^1\right)_{i \in \widehat{\omega_1^*}}$ and $sk_{\widehat{\omega_2^*}} = \left(d_{i0}^2, d_{i1}^2\right)_{i \in \widehat{\omega_2^*}}$ for ω_1^* and ω_2^*, respectively. Let $d_{i0}^\theta = g_2^{q_\theta(i)} H_1(i)^{r_i^\theta}, d_{i1}^\theta = g^{r_i^\theta}$ for each $i \in \widehat{\omega_\theta^*}$, where $\theta \in \{1,2\}$, $r_i^\theta \in \mathbb{Z}_p$, and q_θ is $d-1$ degree polynomial function with $q_\theta(0) = x$.

Then, the adversary outputs a message m^* and a k-element subset $\omega^* = \{i_1, i_2, \ldots, i_k\} \subseteq \bar{\omega}^*$, where $|\omega^*| \leqslant d$. It asks the challenger to generate a signature on message m^* with respect to ω^* from either $sk_{\omega_1^*}$ or $sk_{\omega_2^*}$. The challenger chooses a random bit $b \in \{1,2\}$, a $(d-k)$-element subset $\Omega' = \{i_{k+1}, i_{k+2}, \ldots, i_d\} \subseteq \Omega$, and outputs a signature $\sigma^* = (d_{i0}^b H_1(i))^{r'_i}$

[1] This means that for each user the default set of attributes Ω is included in his private key. The default set of attributes is utilized to make the number of attributes used in the signing algorithm flexible from 1 to d.

$g_2^{q'(i)}H_2(m^*)^{s_i}, d_{i1}^b g^{r'_i}, g^{s_i})_{i\in\omega^*\cup\Omega'}$) by running algorithm Sign with the private key $sk_{\widehat{\omega_b^*}} = \left(d_{i0}^b, d_{i1}^b\right)_{i\in\widehat{\omega_b^*}}$, where $r'_i, s_i \in \mathbb{Z}_p$ and q' is a $d-1$ degree polynomial function with $q'(0)=0$.

From this, the signature could be generated from either $sk_{\widehat{\omega_1^*}}$ or $sk_{\widehat{\omega_2^*}}$. If $b=1$, we prove that it could be generated from $sk_{\widehat{\omega_2^*}}$ as follows:

Because $\sigma^* = \left\{\left(d_{i0}^1 H_1(i)^{r'_i} g_2^{q'(i)} H_2(m^*)^{s_i}, d_{i1}^1 g^{r'_i}, g^{s_i}\right)_{i\in\omega^*\cup\Omega'}, \left(d_{i0}^2 \frac{d_{i0}^1}{d_{i0}^2} H_1(i)^{r'_i} g_2^{q'(i)} H_2(m^*)^{s_i}, d_{i1}^2 \frac{d_{i1}^1}{d_{i1}^2} g^{r'_i}, g^{s_i}\right)_{i\in\omega^*\cup\Omega'}\right\}$, we have $\frac{d_{i0}^1}{d_{i0}^2} = \frac{g_2^{q_1(i)} H_1(i)^{r_i^1}}{g_2^{q_2(i)} H_1(i)^{r_i^2}} = g_2^{q_1(i)-q_2(i)} H_1(i)^{r_i^1-r_i^2}$, $\frac{d_{i1}^1}{d_{i1}^2} = \frac{H_1(i)^{r_i^1}}{H_1(i)^{r_i^2}}$. Define a new $d-1$ polynomial function $\bar{q}(x) = q_1(x) - q_2(x)$. We have $\bar{q}(0)=0$. Thus, $\sigma^* = (d_{i0}^2 g_2^{\bar{q}(i)} H_1(i)^{r_i^1-r_i^2} H_1(i)^{r'_i} g_2^{q'(i)} H_2(m^*)^{s_i}, d_{i1}^2 H_1(i)^{r_i^1-r_i^2} g^{r'_i}, g^{s_i})_{i\in\omega^*\cup\Omega'} = (d_{i0}^2 g_2^{\bar{q}(i)+q'(i)} H_1(i)^{r_i^1-r_i^2+r'_i} H_2(m^*)^{s_i}, d_{i1}^2 g^{r_i^1-r_i^2+r'_i}, g^{s_i})_{i\in\omega^*\cup\Omega'}$. Define another $d-1$ polynomial function $q''(x) = \bar{q}(x) + q'(x)$. We have $q''(0)=0$ and $q''(i) = \bar{q}(i) + q'(i)$. Let $r''_i = r_i^1 - r_i^2 + r'_i$. Then, σ^* could be rewritten as $\sigma^* = \left(d_{i0}^2 g_2^{q''(i)} H_1(i)^{r''_i} H_2(m^*)^{s_i}, d_{i1}^2 g^{r''_i}, g^{s_i}\right)_{i\in\omega^*\cup\Omega'}$, which is a valid signature generated from $sk_{\widehat{\omega_2^*}}$.

Therefore, it has been proven that the signature could also be generated from the private key $sk_{\widehat{\omega_2^*}}$ for attribute set ω_2^*. By using the similar proof as above, one can also get the following result: If a signature is generated by the private key $sk_{\widehat{\omega_2^*}}$ for attributes ω_2^*, then it could also be generated from private key $sk_{\widehat{\omega_1^*}}$ for attributes ω_1^*. From the proof, it has been shown that the hidden attribute-based signature scheme satisfies unconditional anonymity. □

Theorem 2. *Suppose the (t',ϵ')-CDH assumption holds in $\mathbb{G}_1$ and the adversary makes at most q_{H_1}, q_{H_2}, q_K and q_S times queries to random oracle H_1, H_2, private key extraction and signature queries, respectively. Then, the hidden attribute-based signature scheme is $(t, q_{H_1}, q_{H_2}, q_K, q_S, \epsilon)$-EUF-sA-CMA, where $t' < t + (q_{H_1} + q_{H_2} + 2q_K + 3q_S d)t_{exp}$, t_{exp} is the maximum time for an exponentiation in $\mathbb{G}_1$, and $\epsilon' \approx \epsilon / \left(q_{H_2}\binom{d-k}{d-1}\right)$.*

Proof. The above construction utilizes the technique from [4], which can only be proven in the selective-attribute security model. Meanwhile, the signing algorithm uses the implicit chameleon hash function [24] on a message m. Suppose that an adversary $\mathscr{F}$ has an advantage ϵ in attacking the scheme, we build an algorithm $\mathscr{A}$ that uses $\mathscr{F}$ to solve the CDH problem. Algorithm $\mathscr{A}$ is given a random $(g, X=g^x, Y=g^y)$ and asked to compute g^{xy}. Let the default set of attributes be $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{d-1}\}$ for some predefined integer d. First, $\mathscr{F}$ outputs the challenge attribute identity ω^* with the condition $|\omega^*| = k \leqslant d$. Then, $\mathscr{A}$ selects randomly a subset $\Omega^* \subseteq \Omega$ with $|\Omega^*| = d-k$.

`Simulation of Setup` $\mathscr{A}$ sets $g_1 = X$ and $g_2 = Y$.

`Simulation of Random Oracle` Assume that $\mathscr{F}$ makes at most q_{H_1} times to H_1-oracle and q_{H_2} times to H_2-oracle, respectively. $\mathscr{C}$ maintains list $\mathscr{L}_1$ and $\mathscr{L}_2$ to store the answers of H_1-oracle and H_2-oracle, respectively. Meanwhile, it selects a random integer $\delta \in [1, q_{H_2}]$ and a subset $\Omega^* \subseteq \Omega$ with $|\Omega^*| = d-k$. If i is sent for query of H_1, $\mathscr{A}$ checks the list $\mathscr{L}_1$. This works as follows: If an entry for the query is found in $\mathscr{L}_1$, the same answer will be returned to $\mathscr{F}$. Otherwise, it simulates as follows: If $i \in \omega^* \cup \Omega^*$, it chooses random $\beta_i \in \mathbb{Z}_p$ and answers $H_1(i) = g^{\beta_i}$. If $i \notin \omega^* \cup \Omega^*$, it chooses random $\beta_i, \gamma_i \in \mathbb{Z}_p$ and answers $H_1(i) = g_1^{-\beta_i} g^{\gamma_i}$. If m_i is sent for query of H_2, $\mathscr{A}$ checks the list $\mathscr{L}_2$. This works as follows: If an entry for the query is found in $\mathscr{L}_2$, the same answer will be returned. Otherwise, it simulates as follows: If $i \neq \delta$, it chooses random $\alpha_i, \beta_i \in \mathbb{Z}_p$ and answers $H_2(m_i) = g_1^{\alpha_i} g^{\beta_i}$. Otherwise, if $i = \delta$, it chooses random $\beta_i \in \mathbb{Z}_p$ and answer $H_2(m_i) = g^{\beta_i}$.

`Simulation of Private Key Extraction Oracle` Assume that $\mathscr{F}$ makes at most q_K private key extraction queries. $\mathscr{F}$ can make requests for private keys on ω such that $\omega^* \not\subseteq \omega$. We show that how $\mathscr{A}$ simulates a private key on ω on request. We first define three sets Γ, Γ', and S in the following manner: $\Gamma = (\omega \cap \omega^*) \cup \Omega^*$, and Γ' such that $\Gamma \subseteq \Gamma' \subseteq S$ and $|\Gamma'| = d-1$. Let $S = \Gamma' \cup \{0\}$. Next, simulate the decryption key components D_i as follows: For $i \in \Gamma' : D_i = \left(g_2^{\tau_i} H_1(i)^{r_i}, g^{r_i}\right)$, where τ_i, r_i are randomly chosen from $\mathbb{Z}_p$. The intuition behind these assignments is that we are implicitly choosing a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly such that $q(i) = \tau_i$, in addition to having $q(0) = x$. For $i \notin \Gamma'$, compute $D_i = \left(g_2^{\frac{\Delta_{0,S}(i)\gamma_i}{\beta_i} + \sum_{j\in\Gamma'} \Delta_{j,S}(i) q(j)} \left(g_1^{-\beta_i} g^{\gamma_i}\right)^{r'_i}, g_2^{\frac{\Delta_{0,S}(i)}{\beta_i}} g^{r'_i}\right)$. It is the correct simulation key because we let $r_i = \frac{\Delta_{0,S}(i)}{\beta_i} y + r'_i$. As we know, $q(i) = \sum_{j\in\Gamma'} \Delta_{j,S}(i)(q(j) + \Delta_{0,S}(i) q(0))$. Thus, we have $g_2^{q(i)} H_1(i)^{r_i} = g_2^{\frac{\Delta_{0,S}(i)}{\beta_i} + \sum_{j\in\Gamma'} \Delta_{j,S}(i) q(j)} H_1(i)^{r'_i}$, and $g^{r_i} = g_2^{\frac{\Delta_{0,S}(i)}{\beta_i}} g^{r'_i}$.

`Simulation of Signing Oracle` $\mathscr{F}$ also makes requests for signature query on message m for any attributes ω. If $\omega^* \not\subseteq \omega$, then, $\mathscr{A}$ can generate a simulated private key for ω as in the private key simulation and get a signature for ω on message m normally. If $\omega^* \subseteq \omega$, $\mathscr{A}$ selects a random $(d - |\omega|)$-element subset Ω' from Ω. If $H_2(m) \neq g^{\beta_i}$, $\mathscr{A}$ can simulate the signature as follows: Assume that $\omega \cup \Omega' = \{i_1, i_2, \ldots, i_d\}$. First, it chooses $d-1$ values $\tau_{i_k} \in \mathbb{Z}_p$ and lets $q(i_k) = \tau_{i_k}$ for $1 \leqslant k \leqslant d-1$. For these points, $\left(g_2^{q(i_k)} H_1(i_k)^{r_{i_k}} H_2(m)^{s_k}, g^{r_{i_k}}, g^{s_k}\right)$ could be simulated by choosing $s_k \in \mathbb{Z}_p$. The dth point $q(i_d)$ is also determined because $q(0) = x$, which could be denoted by $q(i_d) = \sum_{k=1}^{d-1} \Delta_{i_k,S}(i_d) q(i_k) + \Delta_{0,S}(i_d) q(0)$. Thus, in order to

simulate $\left(g_2^{q(i_d)}(H_1(i_d))^{r_d}(H_2(m))^{s_d}, g^{r_d}, g^{s_d}\right)$, choose $s'_d, r_{i_d} \in \mathbb{Z}_p$ and let $s_d = -\frac{\Delta_{0,S}(i_d)}{\alpha_{i_d}} y + s'_d$. Then, $g_2^{q(i_d)}(H_1(i_d))^{r_d}(H_2(m))^{s_d} = g_2^{\sum_{k=1}^{d-1}\Delta_{i_k,S}(i_d)(q(i_k)} g_2^{\frac{-\Delta_{0,S}(i_d)\beta_i}{\alpha_{i_d}}} g_1^{s'_d\alpha_i} g^{s'_d\beta_i} H_1(i_d)^{r_{i_d}}$, and $g^{s_d} = g_2^{\frac{-\Delta_{0,S}(i_d)}{\alpha_{i_d}}}$.

Forgery Finally, the adversary outputs a forged signature $\sigma^* = \{(\sigma^*_{v1}, \sigma^*_{v2}, \sigma^*_{v3})\}$ for $1 \leqslant v \leqslant d$ on message m^* for attributes ω^* with default attributes $\overline{\Omega^*}$. If $H_2(m^*) \neq g^{\beta_\delta}$ or $\overline{\Omega^*} \neq \Omega^*$, $\mathscr{A}$ will abort. Otherwise, the following verification holds: $\prod_{v=1}^{d}\left(\frac{e(\sigma^*_{v1}, g)}{e(H_1(i_v), \sigma^*_{v2})e(H_2(m^*), \sigma^*_{v3})}\right)^{\Delta_{i_v,S}(0)} = Z$. Because $H_1(i) = g^{\beta_i}$ for $i \in \omega^* \cup \Omega^*$, and $H_2(m^*) = g^{\beta_\delta}$, we have

$$\prod_{v=1}^{d}\left(\frac{e(\sigma^*_{v1}, g)}{e(H_1(i_v), \sigma^*_{v2})e(H_2(m^*), \sigma^*_{v3})}\right)^{\Delta_{i_v,S}(0)} = \prod_{v=1}^{d}\left(\frac{e(\sigma^*_{v1}, g)}{e(g, \sigma^{*\gamma_{i_v}}_{v2})e(g, \sigma^{*\beta_\delta}_{v3})}\right)^{\Delta_{i_v,S}(0)} = \prod_{v=1}^{d}\left(e(\sigma^*_{v1}/[\sigma^{*\gamma_{i_v}}_{v2}\sigma^{*\beta_\delta}_{v3}], g)\right)^{\Delta_{i_v,S}(0)}$$
$$= \prod_{v=1}^{d} e\left((\sigma^*_{v1})/[\sigma^{*\gamma_{i_v}}_{v2}\sigma^{\beta_\delta}_{v3}]\right)^{\Delta_{i_v,S}(0)} = e(g_1, g_2) = e(g^{xy}, g).$$

Thus, $\mathscr{A}$ will compute $g^{xy} = \prod_{v=1}^{d}\left(\sigma^*_{v1}/\left[(\sigma^*_{v2})^{\gamma_{i_v}}(\sigma^*_{v3})^{\beta_\delta}\right]\right)^{\Delta_{i_v,S}(0)}$.

For the success of $\mathscr{A}$, we require that forgery signature on message m^* such that $H_2(m^*) = g^{\beta_\delta}$ and $\overline{\Omega^*} = \Omega^*$. For the correct guess of $d-k$ elements subset Ω^* from a $d-1$-element set Ω, the probability is $1/\binom{d-k}{d-1}$. Therefore, we can get the probability of solving CDH problem as $\epsilon' \approx \epsilon/\left(q_{H_2}\binom{d-k}{d-1}\right)$, if the adversary succeeds with probability ϵ. □

4. The hidden attribute-based signature construction without random oracle

In this construction, we assume that there are ℓ attributes in universe denoted by the set U. Associate each element in U with a unique integer in $\mathbb{Z}_p$. A $d-1$ default set of attributes $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{d-1}\}$ is also given. In this system, the size of messages is n bits, which is a separate parameter unrelated to p. The messages could be bit strings of an arbitrary length and n be the output of a collision-resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$. Our construction works as follows:

Setup (ℓ, d) First, define the universe of attributes as U. For simplicity, let $\ell = |U|$ and we can take the first ℓ elements of $\mathbb{Z}_p$ as the universe of attributes. Namely, the integers $1, 2, \ldots, \ell \pmod p$. Let the $d-1$ default set of attributes be $\Omega = \{\ell+1, \ell+2, \ldots, \ell+d-1\}$. Select a random generator $g \in \mathbb{G}_1$, a random $x \in \mathbb{Z}_p^*$, and set $g_1 = g^x$. Next, pick two random elements $g_2, u' \in \mathbb{G}_1$, a random $(\ell+d-1)$-length vector $\mathsf{H} = (h_i)$, and a random n-length vector $\mathsf{U} = (u_i)$, whose elements are chosen from $\mathbb{G}_1$. Define $Z = e(g_1, g_2)$. The public parameters are $\mathsf{params} = (g, g_1, g_2, Z, \mathsf{H}, \mathsf{U})$, the master key is x.

Extract To generate private key for the attribute set ω, proceed as follows:

- Choose a $d-1$ degree polynomial q such that $q(0) = x$;
- Generate a new set of attributes $\hat{\omega} = \omega \cup \Omega$. For each $i \in \hat{\omega}$, choose $r_i \in_R \mathbb{Z}_p$ and compute $d_{i0} = g_2^{q(i)} \cdot (g_1 h_i)^{r_i}$ and $d_{i1} = g^{r_i}$;
- Finally, output the private key $D_i = \{(d_{i0}, d_{i1})\}$ for each $i \in \hat{\omega}$.

Sign Suppose that one has a private key $(D_i)_{i\in\hat{\omega}} = (d_{i0}, d_{i1})$ for attributes set ω. To generate a signature on message $m = (\mu_1, \mu_2, \ldots, \mu_n) \in \{0,1\}^n$ with respect to attributes $\omega' = \{i_1, i_2, \ldots, i_k\} \subseteq \omega$ where $1 \leqslant k \leqslant d$, proceed as follows:

- Choose a $d-k$ default attributes subset $\Omega' = \{i_{k+1}, i_{k+2}, \ldots, i_d\} \subseteq \Omega$. Then, choose $r'_1, r'_2, \ldots, r'_d, s_1, s_2, \ldots, s_d \in \mathbb{Z}_p$ and a $d-1$ degree polynomial function $q'(x)$ such that $q'(0) = 0$;
- For $1 \leqslant v \leqslant d$, compute $\sigma_{v1} = d_{i_v 0} g_2^{q'(i_v)} (g_1 h_{i_v})^{r'_v}\left(u'\prod_{j=1}^{n} u_j^{\mu_j}\right)^{s_v}$, $\sigma_{v2} = d_{i_v 1} g^{r'_v}$, and $\sigma_{v3} = g^{s_v}$;
- Finally, output the signature $\sigma = \{(\sigma_{v1}, \sigma_{v2}, \sigma_{v3})\}_{1\leqslant v\leqslant d}$.

Verify Take as input the signature $\sigma = \{(\sigma_{v1}, \sigma_{v2}, \sigma_{v3})\}_{1\leqslant v\leqslant d}$ on message $m = (\mu_1, \mu_2, \ldots, \mu_n) \in \{0,1\}^n$ for attributes $\omega' = (i_1, i_2, \ldots, i_k)$ with default attributes subset Ω'. The signature is valid if the following equation holds:

$$\prod_{v=1}^{d}\left(\frac{e(g, \sigma_{v1})}{e(g_1 h_{i_v}, \sigma_{v2})e\left(u'\prod_{j=1}^{n} u_j^{\mu_j}, \sigma_{v3}\right)}\right)^{\Delta_{i_v,S}(0)} = Z.$$

Correctness can be verified similarly with the hidden attribute-based signature scheme in Section 2. The computational cost in the signing and verification algorithms is almost the same with the construction in Section 2.

4.1. Security results

Theorem 3. *The proposed hidden attribute-based signature scheme satisfies anonymity.*

Proof. The proof is very similar to the proof of Theorem 1. So, we omit it here. □

Theorem 4. *Assume that the adversary makes at most q_K and q_S time queries to private key extraction and signature queries, respectively. The hidden attribute-based signature scheme is (t, q_K, q_S, ϵ)-EUF-sA-CMA if the (t', ϵ')-CDH assumption holds in $\mathbb{G}_1$, where $t' < t + (2q_K + 3q_S d)t_{exp}$ and t_{exp} is the maximum time for an exponentiation in $\mathbb{G}_1$, $\epsilon' = \epsilon/(16q_S(n+1)\binom{d-k}{d-1})$.*

Proof. Suppose that an adversary $\mathscr{F}$ has an advantage ϵ in attacking the scheme. An algorithm $\mathscr{A}$ that uses $\mathscr{F}$ to solve the CDH problem can be built. The algorithm $\mathscr{A}$ is given a random $(g, X = g^x, Y = g^y)$ and asked to compute g^{xy}.

First, define the universe U of ℓ elements as $\{1, 2, \ldots, \ell\}$. Then, let the $d-1$ default set of attributes be $\Omega = \{\ell+1, \ell+2, \ldots, \ell+d-1\}$ for simplicity. $\mathscr{F}$ outputs the challenge attribute identity ω^* satisfying $|\omega^*| = k \leqslant d$.

`Simulation of Setup` $\mathscr{A}$ sets $g_1 = X$ and $g_2 = Y$. $\mathscr{A}$ also selects a random subset $\Omega^* \subseteq \Omega$ with $|\Omega^*| = d - k$. For each $i \in \omega^* \cup \Omega^*$, $\mathscr{A}$ chooses $\beta_i \in \mathbb{Z}_p$ and sets $h_i = g_1^{-1} g^{\beta_i}$. For each $i \notin \omega^* \cup \Omega^*$, $\mathscr{A}$ chooses $\beta_i \in \mathbb{Z}_p$ and sets $h_i = g^{\beta_i}$. Then $\mathscr{A}$ sets an integer, $t = 4q_S$ and chooses an integer k' uniformly at random between 0 and n. It then chooses a random n-length vector, $\vec{a} = (a_i)$ where the elements of $\vec{a}$ are chosen uniformly at random between 0 and $t-1$. Additionally, the simulator chooses a random $b' \in \mathbb{Z}_p$ and an n-length vector $\vec{b} = (b_i)$, where the elements of $\vec{b}$ are chosen at random in $\mathbb{Z}_p$. These values are all kept internally to itself. $\mathscr{A}$ then assigns $u' = g_1^{p-kt+a'} g^{b'}$ and $u_i = g_1^{a_i} g^{b_i}$ for $1 \leqslant i \leqslant n$. The system parameters $\mathsf{params} = (g, g_1, \mathsf{H} = (h_i), u', \mathsf{U} = (u_i))$ and params is sent to $\mathscr{F}$. To make the notation easier to understand, the following two pairs of functions $F(m)$ and $J(m)$ are defined for a message $m = \{\mu_1, \mu_2, \ldots, \mu_n\} \in \{0,1\}^n$: $F(m) = (p - tk) + a' + \sum_{i=1}^n a_i^{\mu_i}$ and $J(m) = b' + \sum_{i=1}^n b_i^{\mu_i}$. Finally, a binary function $K(m)$ is defined as $K(m) = \begin{cases} 0, & \text{if } a' + \sum_{i=1}^n a_i^{\mu_i} \equiv 0 (mod\ t); \\ 1, & \text{otherwise.} \end{cases}$.

`Simulation of Private Key Extraction Oracle` Assume that $\mathscr{F}$ makes at most q_K private key extraction queries on ω with the condition $\omega^* \not\subseteq \omega$. Define three sets Γ, Γ', and S in the following manner: $\Gamma = (\omega \cap \omega^*) \cup \Omega^*$, and Γ' such that $\Gamma \subseteq \Gamma' \subseteq S$ and $|\Gamma'| = d - 1$. Let $S = \Gamma' \cup \{0\}$. The private key components D_i can be computed as follows: A random $d-1$ degree polynomial $q(x)$ is chosen by assigning its value for the $d-1$ points randomly in addition to having $q(0) = x$. For $i \in \Gamma'$, choose $s_i, r_i \in \mathbb{Z}_p$ and let $q(i) = s_i$. Then output $D_i = \left(g_2^{s_i}(g_1 h_i)^{r_i}, g^{r_i}\right)$. For $i \notin \Gamma'$, $\mathscr{A}$ can calculate the simulated private key as $D_i = \left(g_2^{\sum_{j\in\Gamma'} \Delta_{j,S}(i) q(j)} g_2^{\beta_i \Delta_{0,S}(i)} (g_1 h_i)^{r_i'}, g_2^{\Delta_{0,S}(i)} g^{r_i'}\right)$. It is easy to verify that this simulated signature is valid: *i.e.*, we show that $D_i = \left(g_2^{q(i)}(g_1 h_i)^{r_i}, g^{r_i}\right) = \left(g_2^{\sum_{j\in\Gamma'} \Delta_{j,S}(i) q(j)} g_2^{\beta_i \Delta_{0,S}(i)} (g_1 h_i)^{r_i'}, g_2^{\Delta_{0,S}(i)} g^{r_i'}\right)$. By using interpolation, for $i \notin \Gamma', q(i) = \Sigma_{j\in\Gamma'} \Delta_{j,S}(i) q(j) + \Delta_{0,S}(i) q(0)$ and $q(x)$ was implicitly defined by the random assignment of the other $d-1$ variables and the variable g_1. Let $r_i = -\Delta_{0,S}(i) y + r_i'$ (In fact, $\mathscr{A}$ does not know the value of r_i), then $g_2^{q(i)}(g_1 h_i)^{r_i} = g_2^{\sum_{j\in\Gamma'} \Delta_{j,S}(i) q(j)} g_2^{\beta_i \Delta_{0,S}(i)} (g_1 h_i)^{r_i'}$ and $g^{r_i} = g_2^{-\Delta_{0,S}(i)} g^{r_i'}$. Therefore, the simulator can construct a private key for the identity ω. Furthermore, the distribution of the private key for ω is identical to that of the original scheme.

`Simulation of Signing Oracle` The original technique in [24] is utilized to avoid the random oracle model. Therefore, the proof is similar to the technique used in [24]. Assume that $\mathscr{F}$ makes requests for signature query on message $m = (\mu_1, \mu_2, \ldots, \mu_n) \in \{0,1\}^n$ for any attributes ω. If $\omega^* \not\subseteq \omega$, $\mathscr{A}$ can generate a simulated private key for ω as in the private key simulation and get a signature for ω on message m normally. If $\omega^* \subseteq \omega$ and $K(m) = 0$, $\mathscr{A}$ will abort. Otherwise, $\mathscr{A}$ selects a random $(d - |\omega|)$-element subset Ω' from Ω and simulates the signature in the following way: Assume that $\omega \cup \Omega' = \{i_1, i_2, \ldots, i_d\}$. First, $\mathscr{A}$ chooses $d-1$ values μ_{i_k} and lets $q(i_k) = \tau_{i_k}$ for $1 \leqslant k \leqslant d-1$. For these points, $\left(g_2^{q(i_k)}(g_1 h_{i_k})^{r_{i_k}} \left(u' \prod_{j=1}^n u_j^{\mu_j}\right)^{s_k}, g^{r_{i_k}}, g^{s_k}\right)$ could be simulated by choosing $r_{i_k}, s_k \in Z_p$. The d-th point $q(i_d)$ is also determined because $q(0) = x$, which could be denoted by $q(i_d) = \sum_{k=1}^{d-1} \Delta_{i_k,S}(i_d) q(i_k) + \Delta_{0,S}(i_d) q(0)$. Therefore, in order to simulate $\left(g_2^{q(i_d)}(g_1 h_{i_d})^{r_{i_d}} \left(u' \prod_{j=1}^n u_j^{\mu_j}\right)^{s_d}, g^{r_{i_d}}, g^{s_d}\right)$, let $s_d = -\frac{\Delta_{0,S}(i_d)}{F(m)} y + s_d'$. Then, $(g_2^{q(i_d)}(g_1 h_{i_d})^{r_{i_d}} \left(u' \prod_{j=1}^n u_j^{\mu_j}\right) = (g_1 h_{i_d})^{r_{i_d}} g_2^{\sum_{k=1}^{d-1} \Delta_{i_k}(i_d)(q(i_k)} g_2^{\frac{-J(m)\Delta_{0,S}(i_d)}{F(m)}} (g_1^{F(m)} g^{J(m)})^{s_d'}$. We also have $g^{s_d} = g_2^{\frac{-\Delta_{0,S}(i_d)}{F(m)}}$.

`Forgery` Finally, the adversary outputs a forged signature $\sigma^* = \left\{(\sigma_{v1}^*, \sigma_{v2}^*, \sigma_{v3}^*)\right\}$ for $1 \leqslant v \leqslant d$ on message $m^* = (\mu_1^*, \mu_2^*, \ldots, \mu_n^*)$ for ω^* with default attribute subset $\overline{\Omega^*}$. If the signature is valid, then $\prod_{v=1}^d \left(\frac{e(\sigma_{v1}^*, g)}{e(g_1 h_{i_v}, \sigma_{v2}^*) e\left(u' \prod_{j=1}^n u_j^{\mu_j^*}, \sigma_{v3}^*\right)}\right)^{\Delta_{i_v,S}(0)} = e(g_1, g_2)$.

If $a' + \sum_{i=1}^n a_i^{\mu_i^*} \neq kt$ or $\overline{\Omega^*} \neq \Omega^*$, $\mathscr{A}$ will abort. Otherwise, because $K(m^*) = 0$ and $\overline{\Omega^*} = \Omega^*$, we have

$$\prod_{v=1}^d \left(\frac{e(\sigma_{v1}^*, g)}{e\left(g, (\sigma_{v2}^*)^{\beta_{i_v}}\right) e\left(g, (\sigma_{v3}^*)^{J(m^*)}\right)}\right)^{\Delta_{i_v,S}(0)} = \prod_{v=1}^d e\left(\frac{\sigma_{v1}^*}{(\sigma_{v2}^*)^{\beta_{i_v}} (\sigma_{v3}^*)^{J(m^*)}}, g\right)^{\Delta_{i_v,S}(0)} = e(g_1, g_2) = e(g^{xy}, g).$$

Therefore, CDH assumption can be solved by computing

$$g^{xy} = \prod_{v=1}^{d} \left(\sigma^*_{v1} / \left[\left(\sigma^*_{v2} \right)^{\beta_{i_v}} \left(\sigma^*_{v3} \right)^{J(m^*)} \right] \right)^{\Delta_{i_v,S}(0)}.$$

What remains is to analyze the probability of $\mathscr{A}$ not aborting. For the simulation to complete without aborting, it is required that all signature queries on m_i will have $K(m_i) \neq kt$, that forgery signature on message m^* has $K(m^*) = 0 \bmod p$ as well as $\overline{\Omega^*} = \Omega^*$. In fact, the probability analysis is very similar to [19]. Therefore, one can get the probability of solving CDH problem as $\epsilon' = \epsilon / \left(16 q_S (n+1) \binom{d-k}{d-1} \right)$ if the adversary succeeds with probability ϵ. □

5. Conclusion

The notion of hidden attribute-based signature was initially proposed in this paper. The security definitions and models were also suggested and formalized. Unforgeability and anonymity are defined for the hidden attribute-based signature. More specifically, unforgeability requires that any user without certain attributes cannot generate signature with attributes that he does not have. Anonymity allows the signer to generate a signature with part of his attributes while being anonymous among all the users with the same attributes purported in the signature. Furthermore, we proposed two constructions of hidden attribute-based signature in this paper. The first construction supports a large universe of attributes and its security proof relies on the random oracle assumption, which is removed in the second construction. Both constructions are proven to be secure under the standard computational Diffie–Hellman assumption.

References

[1] J. Baek, W. Susilo, J. Zhou, New Constructions of fuzzy identity-based encryption, ASIACCS'07, ACM, 2007, pp. 368–370.
[2] M. Bellare, D. Micciancio, B. Warinschi, Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions, EUROCRYPT'03, LNCS, vol. 2656, Springer, 2003, pp. 614–629.
[3] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: IEEE Symposium on Security and Privacy'07, IEEE, 2007, pp. 321–334.
[4] D. Boneh, X. Boyen. Efficient selective-ID secure identity based encryption without random oracles, EUROCRYPT'04, Springer, LNCS, vol. 3027, 2004, pp. 223–238.
[5] E. Bresson, J. Stern, M. Szydlo, Threshold ring signatures and applications to ad-hoc groups, CRYPTO'02, LNCS, vol. 2442, Springer, 2002, pp. 465–480.
[6] A. Burnett, A. Duffy, Tom Dowling, A biometric identity based signature scheme, Also appeared at Industry Track of ACNS 2005. Available at: <http://eprint.iacr.org/2004/176>.
[7] M. Chase, Multi-authority attribute based encryption, TCC'07, LNCS, vol. 4392, Springer, 2007, pp. 515–534.
[8] D. Chaum, E.V. Heyst, Group signatures, EUROCRYPT'91, LNCS, vol. 547, Springer, 1991, pp. 257–265.
[9] S.S.M. Chow, V.K.-W. Wei, J.K. Liu, T.H. Yuen, Ring signatures without random oracles, ASIACCS'06, ACM Press, 2006, pp. 297–302.
[10] S.S.M. Chow, S.-M. Yiu, L.C.K. Hui, Efficient identity based ring signature, ACNS'05, LNCS, vol. 3531, Springer, 2005, pp. 499–512.
[11] Y.F. Chung, H.H. Lee, F. Lai, T.S. Chen, Access control in user hierarchy based on elliptic curve cryptosystem, Information Sciences 178 (2008) 230–243.
[12] D. Galindo, J. Herranz, E. Kiltz, On the generic construction of identity-based signatures with additional properties, ASIACRYPT'06, LNCS, vol. 4284, Springer, 2006, pp. 178–193.
[13] S. Guo, Y. Zeng, Attribute-based signature scheme, in: 2008 International Conference on Information Security and Assurance (ISA 2008), 2008, pp. 509–511.
[14] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: CCS'06, ACM, 2006, pp. 89–98.
[15] D. Khader, Attribute based group signatures, 2007. Available at: <http://eprint.iacr.org/2007/159>.
[16] A. Kiayias, H.S. Zhou, Hidden identity-based signatures, FC'07, LNCS, vol. 4886, Springer, 2007, pp. 134–147.
[17] J. Li, X. Chen, T.H. Yuen, Y. Wang, Proxy ring signature: formal definitions, efficient construction and new variant, CIS'06, LNCS, vol. 4456, Springer, 2007, pp. 545–555.
[18] H. Maji, M. Prabhakaran, M. Rosulek, Attribute based signatures: achieving attribute privacy and collusion–resistance, 2008. Available at: <http://eprint.iacr.org/2008/328>.
[19] R.L. Rivest, A. Shamir, Y. Tauman, How to leak a secret, ASIACRYPT'01, LNCS, vol. 2248, Springer, 2001, pp. 552–565.
[20] A. Sahai, B. Waters, Fuzzy identity-based encryption, EUROCRYPT'05, LNCS, vol. 3494, Springer, 2005, pp. 457–473.
[21] H. Shacham, B. Waters, Efficient ring signatures without random oracles, PKC'07, LNCS, vol. 4450, Springer, 2007, pp. 166–180.
[22] A. Shamir, Identity based cryptosystems and signature schemes, CRYPTO'84, LNCS, vol. 196, Springer, 1984, pp. 47–53.
[23] C.-H. Wang, C.-Y. Liu, A new ring signature scheme with signer-admission property, Information Sciences 177 (2007) 747–754.
[24] B. Waters, Efficient identity-based encryption without random oracles, EUROCRYPT'05, LNCS, vol. 3494, Springer, 2005, pp. 114–127.
[25] P. Yang, Z. Cao, X. Dong, Fuzzy identity based signature, 2008. Available at: <http://eprint.iacr.org/2008/002>.