Attribute-Based Ring Signatures

Jin Li and Kwangjo Kim

International Research center for Information Security (IRIS)
Information and Communications University(ICU)
103-6 Munji-Dong, Yuseong-Gu, Daejeon, 305-732, Korea
{jjl,kkj}@icu.ac.kr

Abstract. Ring signature was proposed to keep signer's anonymity when it signs messages on behalf of a "ring" of possible signers. In this paper, we propose a novel notion of ring signature which is called attribute-based ring signature. In this kind of signature, it allows the signer to sign message with its attributes from attribute center. All users that possess of these attributes form a ring. The identity of signer is kept anonymous in this ring. Furthermore, anyone out of this ring could not forge the signature on behalf of the ring.
Two constructions of attribute-based ring signature are also presented in this paper. The first scheme is proved to be secure in the random oracle model, with large universal attributes. We also present another scheme in order to avoid the random oracle model. It does not rely on non-standard hardness assumption or random oracle model. Both schemes in this paper are based on standard computational Diffie-Hellman assumption.

Keywords: Ring Signature, Attribute-Based, Anonymity, Computational Diffie-Hellman Assumption

1 Introduction

Ring signatures [17] allow user to sign messages on behalf of a "ring" of legitimate signers, without revealing the signer's identity. Different from group signature (for examples, [2,8]), in ring signature, the group formation is spontaneous and there is no group manager to revoke the identity of the signer. Therefore under the assumption that each user is previously associated with a public key, a user can form a group by simply collecting the public keys of all the "ring" members including his own. These diversion members can be totally unaware of being included into the group. Ring signature schemes could be used for whistle blowing and anonymous membership authentication [17] to keep the anonymity of the signer and can be publicly verifiable.

In order to simplify the key management procedures of the certificate-based public key infrastructures, Shamir [21] introduced the concept of identity-based cryptosystem in 1984. Identity-based cryptosystem is a public key cryptosystem where the public key can be an arbitrary string such as an email address. A private key generator uses a master secret key to issue private keys to identities that request them.

As a related notion to identity-based cryptosystem, fuzzy identity-based encryption (IBE) [18] was proposed by Sahai and Waters at Eurocrypt 2005. In fuzzy IBE, the identity of user is viewed as a set of descriptive attributes. Before decryption, they get their attributes from the authorities. In this system, the user with secret key for identity ω is able to decrypt a ciphertext encrypted with identity ω' if and only if ω and ω' are within a certain distance of each other as judged by some metric. Fuzzy IBE has many important applications. For example, fuzzy IBE can be applied to enable encryption using biometric inputs as identities because the error-tolerance property of an fuzzy IBE scheme allows for the use of biometric identities. Another important application of fuzzy IBE is attribute-based encryption. In this application, an entity could encrypt a document to all users that have a certain set of attributes. For example, define the distance to be two by overlapping. A message is encrypted with respect to attributes "University A", "Faculty", and "Computer Science Department". Then, only users having two attributes of these three attributes can decrypt and get the message. In more detail, there are three kinds of users are allowed to decrypt, *i.e.*, if they are faculty in university A or computer science department, or member of computer science department in university A.

1.1 Our Contributions

In this paper, we introduce the notion of attribute-based ring signatures. Attribute-based ring signatures are designated for the following situation: An entity gets its certificate for attributes ω from an attribute center. Then, this entity signs message by using subset of its attributes $\omega' \subseteq \omega$ to prove that the message is signed by user with attributes ω'. Let all users with this attributes subset ω' be a ring. It requires that anyone can not tell who generates the signature in this ring, even if many ring signatures are available. Furthermore, anyone could not forge the signature for this ring if it is not in this ring.

Consider the following application: Bob has attributes {"University A", "Faculty", "Computer Science Department", "Driver Licence"}. And, Bob wishes to complain or give some suggestions to an administrator in the university A, in such a way that Bob remains anonymous, yet the administrator is convinced that such complaints or suggestions are indeed from some faculty in the university A. In order to do this, Bob could sign the complaints or suggestions with attributes "University A" and "Faculty", by using the attribute-based ring signature. The administrator can verify the validity of the signature and be convinced it is indeed from some faculty in the university A, without knowing its identity.

In all of the previous kinds of ring signatures, the signer and verifier know which people are included in this ring. In our proposed ring signatures, the signer could only decide the attributes for the signature. It need not know who are involved in this ring.

Then, two constructions of attribute-based ring signatures are proposed. The first scheme is proved to be secure in the random oracle model, with large universal attributes. Then, we present another construction that is provably secure, without random oracles.

1.2 Related Work

After the notion of fuzzy IBE [18] was proposed, many improvements and extension were proposed. Baek *et al.* [1] showed how to shorten the public parameters, but the scheme could only be proved to be secure in the random oracle model.

Later, there are many extensions for the notion of fuzzy IBE. Chase [7] proposed a multi-authority attribute-based encryption scheme. In this protocol, each authority controls some of the attributes. To add some restriction on the private key, Goyal *et al.* [13] proposed a key-policy attribute-based encryption. In this system, each private key is associated with an access structure that specifies which type of ciphertexts the key can decrypt. Instead of determining the decrypting policy in private key, in [4], they proposed the notion of ciphertext-policy attribute-based encryption. It allows the encryptor to specify an associated access structure such that only users, with attributes satisfying this access structure, can decrypt the ciphertext.

There are several attempts to construct attribute-based signatures. The notion of fuzzy identity-based signature was proposed in [24]. In their definition of fuzzy identity-based signature, it allows a user with identity ω to issue a signature, that is to say, the user can sign a message with some of its attributes. The verifier can check if the signature is signed by the user with these attributes. And, a similar notion to fuzzy identity-based signature, called attribute-based signature, was also proposed in [12], to achieve almost the same goal. However, these kind of signature does not consider the anonymity for signer. As the relation between attribute-based cryptosystem and identity-based cryptosystem, such kind of signature scheme could be trivial to construct by using the method given by Galindo *et al* [11]. Another work on attribute-based signature is [16]. In this work, they tried to get a signature with signer privacy, however, the security is very weak because it can only be proved in non-standard hardness assumption and generic group model.

Recently, Khader [14] proposed another notion which was 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 should prove ownership of certain properties.

Since after ring signature scheme was first formalized by Rivest, Shamir and Tauman [17], many practical ring signature schemes and its variants have been proposed, such as threshold ring signature [6], identity-based ring signature [10], and proxy ring signature [15]. Most subsequent papers holds in the random oracle model. Xu *et al.* [23] described a ring signature secure in the standard model without rigorous proof. Later, Chow *et al.* [9] proved the security of this ring signature scheme in the standard model, but based on a new strong assumption. Bender *et al.* [3] presented a stronger model and ring signatures that are secure in this standard model assuming trapdoor permutations exist. And, the first efficient ring signature scheme secure without random oracles and based on standard assumptions, was proposed by Shacham and Waters in [19]. In their ring signature scheme, setup assumption was required.

2 Preliminaries

In this paper, we use the bilinear pairings on elliptic curves. We now give a brief review on the property of pairings and some candidate hard problems from pairings that will be used later.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic groups of prime order p with the multiplicative group action. And, g is a generator of $\mathbb{G}_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ 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 exists $g_1, g_2 \in \mathbb{G}_1$ 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 problem.

3 Attribute-Based Ring Signature Scheme

In this section, we show the definition and security model of attribute-based ring signature. Then, we propose a construction. In attribute-based ring signature, one can get private key for attributes from the attributes center.

3.1 Syntax

The attribute-based ring 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 parameters params and sk as master secret key for attribute center.
- Extract. The private key extraction algorithm, on input attributes ω, 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 secret key sk_ω for attributes ω, outputs signature σ.
- Verify. The verification algorithm, given an alleged signature σ for m with respect to attributes ω', and the public parameters 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 attribute-based ring signatures: unforgeability and anonymity. The definition for unforgeability requires any user can not generate a signature for some attributes ω^* if it has not private key for ω such that $\omega^* \subseteq \omega$. The following definition for unforgeability also implies the security against collusion attacks, in which a group of users could combine their secret keys and sign message with attributes they could not do individually.

3.2.1 Unfrogeability

For unforgeability, we require that it is existentially unforgeable against selective attribute ring and chosen message attacks (EUF-sA-CMA). This kind of definition is similar to the fixed-ring attacks as in many ring signature schemes [9,10,15].

There are two oracles provided to the adversary:

1. Private Key Extraction Oracle: Given attributes set ω, output corresponding secret key sk_ω.
2. Signing Oracle: Given message m and attributes set ω, output signature σ.

If the security is proved in the random oracle model, another oracle should also be provided to the adversary additionally:
(3) Random Oracle: Given m, output a random value r.

As we explained above, the attribute ring is fixed for some predefined maximum number, for example, d. Its formal definition is based on the following EUF-sA-CMA game involving a challenger $\mathcal{C}$ and an adversary $\mathcal{F}$.

[Game EUF-sA-CMA]

Initial. $\mathcal{F}$ outputs its challenge attributes set $|\omega^*| \leq d$ for some predefined number d.
Setup(d). $\mathcal{C}$ chooses a sufficiently large security parameter 1^λ and runs Setup. $\mathcal{C}$ retains secret key sk and sends params generated from Setup to $\mathcal{F}$.
***Phase* 1**. $\mathcal{F}$ can perform a polynomially bounded number of queries m, ω, and (m', ω') with $|\omega'| \leq d$ to random oracle, private key extraction oracle and signing oracle, respectively. The restriction of the private key extraction query on ω should satisfy $\omega^* \nsubseteq \omega$.
Forgery. $\mathcal{F}$ outputs a signature σ^* on messages m^* with respect to attributes set 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 $\mathbf{Adv}^{euf}_{ARS,\mathcal{F}}(1^\lambda)$ of $\mathcal{F}$ is defined as the probability that it wins the game.

Definition 2 (Unforgeability). *A forger $\mathcal{F}$ $(t, q_K, q_S, q_H, \epsilon)$-breaks a signature scheme if $\mathcal{A}$ runs in time at most t, and makes at most q_K private key ex-*

traction queries, q_S signature queries and q_H hash queries, while $\mathbf{Adv}^{euf}_{ARS,\mathcal{F}}(1^\lambda)$ *is at least ϵ.*

A signature scheme is $(t, q_K, q_S, q_H, \epsilon)$-existentially unforgeable under an adaptive chosen message attack 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 is anonymous among the users with the same attributes for signature, even for the attribute center. The adversary asks for some signature with respect to some attributes belong to two attribute identities. Because two attribute identities could generate the signature with the same attribute subset, the adversary has to guess which one signs the message, even with the secret key for both identities. Its formal definition is based on the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{F}$.

[Game Anonymity]

Setup(d). The challenger $\mathcal{C}$ chooses a sufficiently large security parameter 1^λ and runs Setup to get master key sk and public parameter params. $\mathcal{C}$ sends sk and params to $\mathcal{F}$. With the secret key sk, $\mathcal{F}$ can generate private key and signature itself.
Challenge Phase. $\mathcal{F}$ outputs a message m^*, two attribute identities ω_1^*, ω_2^*, and challenged attribute ω^* for signature query, where $\overline{\omega}^* = \omega_1^* \cap \omega_2^*$ and, $\omega^* \subseteq \overline{\omega}^*$ such that $\mid \omega^* \mid \leq d$. Assume $\mathcal{F}$ has queried private key extractions to two attributes set ω_1^* and ω_2^*. The secret keys for ω_1^* and ω_2^* are $sk_{\omega_1^*}$ and $sk_{\omega_2^*}$, respectively. $\mathcal{C}$ chooses a bit $b \in \{0, 1\}$, computes the challenge signature $\sigma^* = \mathsf{Sign}(m^*, \omega^*, sk_{\omega_b^*})$ and provide σ^* to $\mathcal{F}$.
Guess. Algorithm $\mathcal{F}$ tries to guess the signature is generated from signer with ω_1^* or ω_2^*. Finally, it outputs a bit b' for b, and wins if $b' = b$.

We define $\mathbf{Adv}^{anony}_{ARS,\mathcal{F}}(1^\lambda)$ to be the advantage over $1/2$ of $\mathcal{F}$ in the above game.

Definition 3 (Anonymity). *An attribute-based ring signature scheme satisfies the anonymity requirement if there exists no forger $\mathcal{F}$ can win the above game with non-negligible advantage* $\mathbf{Adv}^{anony}_{ARS,\mathcal{F}}(1^\lambda)$.

In this game, the master key of attribute center is also given to the adversary. This means that signer's anonymity holds even for the attribute center.

At first glance, it seems trivial to construct such a protocol just by preparing one secret key for each signing attributes set ω' (Only preparing one secret key for each attribute $i \in \mathbb{Z}_p$, instead of signing attributes set ω', will not provide security against collusion attacks, in which a group of users could combine their secret keys and break the security requirement of unforgeability defined in Section 3.2.). However, if the number of universal attributes set U is k, we can calculate that the number of all signing attributes is at least $\binom{k}{d}$. As a result, the attribute center has to publish at least $\binom{k}{d}$ public keys. Obviously, it can not be

realized in polynomial time for the following case when the universal attributes are chosen from $\mathbb{Z}_p$. However, in our construction, it only requires to publish 4 group elements. The number $\binom{k}{d}$ will be huge even for small k and d, for example, when the attributes number is $k = 50$ and $d = 10$, it will be approximately 10^{10}, instead of k in our second construction.

3.3 Our Attribute-Based Ring Signature Construction

In our construction, the signer could generate a signature with some of its attributes. A predefined number d will be given before setup algorithm. And, in our system, the user can sign a document with the number of attributes from 1 to d.

Before giving the construction, some preliminaries on Lagrange interpolation are given here. Recall that, given d points $q(1), \cdots, q(d)$ on a $d-1$ degree polynomial, we can use Lagrange interpolation to compute $q(i)$ for any $i \in \mathbb{Z}_p$. Let S be a d-element set. We define the Lagrange coefficient $\Delta_{j,S}(i)$ of $q(j)$ in the computation of $q(i)$ as:

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

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a pairing defined in Section 2.

Setup(d). First, define the universal attributes U as $\mathbb{Z}_p$. Furthermore, a $d-1$ default attributes set from $\mathbb{Z}_p$ is also given as $\Omega = \{\Omega_1, \Omega_2, \cdots, \Omega_{d-1}\}$. Then, select a random generator $g \in \mathbb{G}_1$, a random $x \in \mathbb{Z}_p^*$, and set $g_1 = g^x$. Next, pick random element $g_2 \in \mathbb{G}_1$, 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 $params = (g, g_1, g_2, Z, H_1, H_2)$, and the master key is x.

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

- First, a $d-1$ degree polynomial q is randomly chosen such that $q(0) = x$;
- Generate a new attributes set $\widehat{\omega} = \omega \cup \Omega$. For each $i \in \widehat{\omega}$, 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 $i \in \widehat{\omega}$. [1]

Sign. Suppose one has a private key for attributes set ω. To sign a message m for attributes $\omega'{=}\{i_1, \cdots, i_k\} \subseteq \omega$, where $1 \leq k \leq d$, it proceeds as follows:

- First, select a $d-k$ default attributes subset $\Omega' = \{i_{k+1}, i_{k+2}, \cdots, i_d\} \subseteq \Omega$. Then, choose $r'_1, \cdots, r'_d, s_1, \cdots, s_d \in \mathbb{Z}_p$ and a $d-1$ degree polynomial function $q'(x)$ such that $q'(0) = 0$;

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

- For $1 \leq v \leq d$, 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 $\sigma=\{(\sigma_{v1},\sigma_{v2},\sigma_{v3})\}_{1\leq v\leq d}$.

Verify. To verify the signature $\sigma=\{(\sigma_{v1},\sigma_{v2},\sigma_{v3})\}_{1\leq v\leq d}$ on message m for attributes $\omega'=(i_1, \cdots, i_k)$ with default attributes Ω', it checks 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

The correctness of verification is justified by the following equations:

For $1 \leq v \leq d$, we have

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

So, we have

$$\begin{aligned}
&[e(g_2,g)^{q_{i_1}+q'(i_1)}]^{\Delta_{i_1,S}(0)} \cdots [e(g_2,g)^{q_{i_d}+q'(i_d)}]^{\Delta_{i_d,S}(0)} \\
&= e(g_2,g)^{q(0)+q'(0)} \\
&= e(g_2,g_1) \\
&= Z
\end{aligned}$$

3.5 Security Results

Theorem 1. *Our attribute-based ring signature scheme satisfies unconditional anonymity.*

Proof. First, the attribute center runs Setup to get the public parameters params and the master key x. It gives the adversary params and x. After these interactions, the adversary outputs two attributes ω_1^* and ω_2^*, where $\overline{\omega}^* = \omega_1^* \cap \omega_2^*$. 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 the challenger or adversary has generated the secret keys as $sk_{\widehat{\omega_1^*}} = (d_{i0}^1, d_{i1}^1)_{i\in\widehat{\omega_1^*}}$ and $sk_{\widehat{\omega_2^*}} = (d_{i0}^2, d_{i1}^2)_{i\in\widehat{\omega_2^*}}$ 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 $\omega^* = \{i_1, \cdots, i_k\} \subseteq \overline{\omega}^*$, where $\mid \omega^* \mid \leq d$. 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 $\Omega' = \{i_{k+1}, i_{k+2}, \cdots, i_d\} \subseteq \Omega$, and outputs a signature $\sigma^* = (d_{i0}^b H_1(i)^{r_i'} \ g_2^{q'(i)} H_2(m^*)^{s_i}, \ d_{i1}^b g^{r_i'}, \ g^{s_i})_{i \in \omega^* \cup \Omega'})$ by running algorithm Sign with the secret key $sk_{\widehat{\omega_b^*}} = (d_{i0}^b, d_{i1}^b)_{i \in \widehat{\omega_b^*}}$, where $r_i', s_i \in \mathbb{Z}_p$ and q' is a $d-1$ degree polynomial function with $q'(0) = 0$.

Then, we will show that this signature could be generated from either by $sk_{\widehat{\omega_1^*}}$ or $sk_{\widehat{\omega_2^*}}$. If $b = 1$, then we will show it also could be generated from $sk_{\widehat{\omega_2^*}}$ as follows:

Because

$\sigma^* = (d_{i0}^1 H_1(i)^{r_i'} \ g_2^{q'(i)} H_2(m^*)^{s_i}, \ d_{i1}^1 g^{r_i'}, \ g^{s_i})_{i \in \omega^* \cup \Omega'}$, we have

$\sigma^* = (d_{i0}^2 \frac{d_{i0}^1}{d_{i0}^2} H_1(i)^{r_i'} \ g_2^{q'(i)} H_2(m^*)^{s_i}, \ d_{i1}^2 \frac{d_{i1}^1}{d_{i1}^2} g^{r_i'}, \ g^{s_i})_{i \in \omega^* \cup \Omega'}$.

We have, $\frac{d_{i0}^1}{d_{i0}^2} = \frac{g_2^{q_1(i)} H_1(i)^{r_i^1}}{g_2^{q_2(i)} H_1(i)^{r_i^2}} = g_2^{q_1(i) - q_2(i)} H_1(i)^{r_i^1 - r_i^2}$.

And, $\frac{d_{i1}^1}{d_{i1}^2} = \frac{H_1(i)^{r_i^1}}{H_1(i)^{r_i^2}}$.

Define a new $d-1$ polynomial function $\overline{q}(x) = q_1(x) - q_2(x)$. We have $\overline{q}(0) = 0$. Thus,

$\sigma^* = (d_{i0}^2 \ g_2^{\overline{q}(i)} \ H_1(i)^{r_i^1 - r_i^2} \ H_1(i)^{r_i'} \ g_2^{q'(i)} \ H_2(m^*)^{s_i}, \ d_{i1}^2 \ H_1(i)^{r_i^1 - r_i^2} \ g^{r_i'}, \ g^{s_i})_{i \in \omega^* \cup \Omega'}$

$= (d_{i0}^2 \ g_2^{\overline{q}(i) + q'(i)} \ H_1(i)^{r_i^1 - r_i^2 + r_i'} \ H_2(m^*)^{s_i}, \ d_{i1}^2 \ g^{r_i^1 - r_i^2 + r_i'}, \ g^{s_i})_{i \in \omega^* \cup \Omega'}$

Define another $d-1$ polynomial function $q''(x) = \overline{q}(x) + q'(x)$. We have $q''(0) = 0$ and $q''(i) = \overline{q}(i) + q'(i)$. Let $r_i'' = r_i^1 - r_i^2 + r_i'$.

Then, σ^* could be rewritten as

$\sigma^* = (d_{i0}^2 g_2^{q''(i)} \ H_1(i)^{r_i''} \ H_2(m^*)^{s_i}, \ d_{i1}^2 g^{r_i''}, \ g^{s_i})_{i \in \omega^* \cup \Omega'}$, which is a valid signature generated from $sk_{\widehat{\omega_2^*}}$.

So, we have proved the signature could also be generated from the secret key $sk_{\widehat{\omega_2^*}}$ for attributes ω_2^*.

By using similar proof as above, we can also get the following result: If a signature is generated by the secret key $sk_{\widehat{\omega_2^*}}$ for attributes ω_2^*, then, it could also be generated from secret key $sk_{\widehat{\omega_1^*}}$ for attributes ω_1^*.

From the proof, we have showed that the attribute-based ring 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 attribute-based ring 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 $\epsilon' \approx \epsilon / (q_{H_2} \binom{d-k}{d-1})$.*

Proof. See Appendix A.

4 The Attribute-Based Ring Signature Construction Without Random Oracle

In our scheme, it is assumed there are ℓ attributes in universe, which are denoted by the set U. Associate each element in U with a unique integer in $\mathbb{Z}_p$. Also, a $d-1$ default attributes set $\Omega = \{\Omega_1, \Omega_2, \cdots, \Omega_{d-1}\}$ is also given. We will sign messages of n bits, a separate parameter unrelated to p. The messages could be bit strings of 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, U of elements. For simplicity, let ℓ=$|U|$ and we can take the first ℓ elements of $\mathbb{Z}_p$, to be the universe. Namely, the integers $1, \cdots, \ell \pmod p$. And, let the $d-1$ default attributes set $\Omega = \{\ell+1, \ell+2, \cdots, \ell+d-1\}$. Then, select a random generator $g \in \mathbb{G}_1$, a random $x \in \mathbb{Z}_p^*$, and set $g_1 = g^x$. Next, pick 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 $\mathbb{G}_1$. Let $Z = e(g_1, g_2)$. The public parameters are $params = (g, g_1, g_2, Z, \mathsf{H}, \mathsf{U})$, master key is x.

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

- A $d-1$ degree polynomial q is randomly chosen such that $q(0) = x$;
- Generate a new attributes set $\widehat{\omega} = \omega \cup \Omega$. For each $i \in \widehat{\omega}$, choose $r_i \in_R \mathbb{Z}_p$. Then, 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 $i \in \widehat{\omega}$.

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

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

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

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

Correctness can be verified similarly with the attribute-based ring signature scheme in Section 3.3.

4.1 Security Results

Theorem 3. *Our attribute-based ring signature scheme satisfies unconditional anonymity.*

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

Theorem 4. *Assume the adversary makes at most q_K and q_S times queries to private key extraction and signature queries, respectively. The attribute-based ring 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. See Appendix B.

5 Conclusion

We proposed the new notion of attribute-based ring signature in this paper. It allows the signer to sign message with some of its attributes while keeping the signer's anonymity in this attributes ring. Furthermore, anyone out of this ring could not forge the signature on behalf of the ring. In our proposed ring signatures, the signer could only decide the attributes for the signature, and the users with these attributes are included in this ring. Different from the previous ring signatures, the signer does not know which users are involved in this ring.

Two constructions were also proposed in this paper. The first one is proved to be secure in the random oracle model, with large universal attributes. The second construction is provably secure, without random oracles. And, both of constructions are secure based on standard assumption under the selective attribute ring security model.

References

1. J. Baek, W. Susilo, and J. Zhou. New Constructions of Fuzzy Identity-Based Encryption, ASIACCS'07, ACM, 2007, pp. 368-370.
2. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions, EUROCRYPT'03, LNCS 2656, Springer, 2003, pp. 614-629.
3. A. Bender, J. Katz, and R. Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles, TCC'06, LNCS 3876, Springer, 2006, pp. 60-79.
4. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-Policy Attribute-Based Encryption, IEEE Symposium on Security and Privacy'07, IEEE, 2007, pp. 321-334.
5. D. Boneh and X. Boyen. Efficient selective-ID Secure Identity Based Encryption Without Random Oracles, EUROCRYPT'04, LNCS 3027, Springer, 2004, pp. 223-238.

6. E. Bresson, J. Stern, and M. Szydlo. Threshold Ring Signatures and Applications to Ad-hoc Groups, CRYPTO'02, LNCS 2442, Springer, 2002, pp. 465-480.
7. M. Chase. Multi-Authority Attribute Based Encryption, TCC'07, LNCS 4392, Springer, 2007, pp. 515-534.
8. D. Chaum and E. V. Heyst. Group Signatures, EUROCRYPT'91, LNCS 547, Springer, 1991, pp. 257-265.
9. S. S. M. Chow, V. K.-W. Wei, J. K. Liu, and T. H. Yuen. Ring Signatures Without Random Oracles, ASIACCS'06, ACM Press, 2006, pp. 297-302.
10. S. S. M. Chow, S.-M. Yiu, and L. C. K. Hui. Efficient Identity Based Ring Signature, ACNS'05, LNCS 3531, Springer, 2005, pp. 499-512.
11. D. Galindo, J. Herranz, and E. Kiltz. On the Generic Construction of Identity-Based Signatures with Additional Properties. ASIACRYPT'06, LNCS 4284, Springer, 2006, pp. 178-193.
12. S. Guo and Y. Zeng. Attribute-Based Signature Scheme, 2008 International Conference on Information Security and Assurance (ISA 2008), 2008, pp. 509-511.
13. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, CCS'06, ACM, 2006, pp. 89-98.
14. D. Khader. Attribute Based Group Signatures, Available at http://eprint.iacr.org/2007/159, 2007.
15. J. Li, X. Chen, T. H. Yuen, and Y. Wang. Proxy Ring Signature: Formal Definitions, Efficient Construction and New Variant, CIS'06, LNCS 4456, Springer, 2007, pp. 545-555.
16. H. Maji, M. Prabhakaran, and M. Rosulek. Attribute Based Signatures: Achieving Attribute Privacy and Collusion-Resistance, Available at http://eprint.iacr.org/2008/328, 2008.
17. R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret, ASIACRYPT'01, LNCS 2248, Springer, 2001, pp. 552-565.
18. A. Sahai and B. Waters. Fuzzy Identity-Based Encryption, EUROCRYPT'05, LNCS 3494, Springer, 2005, pp. 457-473.
19. H. Shacham and B. Waters. Efficient Ring Signatures Without Random Oracles, PKC'07, LNCS 4450, Springer, 2007, pp. 166-180.
20. A. Shamir. How to Share a Secret, ACM, 22(1979) 612-613.
21. A. Shamir, Identity Based Cryptosystems and Signature Schemes, CRYPTO'84, LNCS 196, Springer, 1984, pp.47-53.
22. B. Waters. Efficient Identity-Based Encryption Without Random Oracles, EUROCRYPT'05, LNCS 3494, Springer, 2005, pp. 114-127.
23. J. Xu, Z. Zhang, and D. Feng. A Ring Signature Scheme Using Bilinear Pairings, WISA'04, LNCS 3325, Springer, 2004, pp. 160-169.
24. P. Yang, Z. Cao, and X. Dong. Fuzzy Identity Based Signature, Available at http://eprint.iacr.org/2008/002, 2008.

Appendix A: Proof of Theorem 2

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

Let the default attributes set be $\Omega = \{\Omega_1, \Omega_2, \cdots, \Omega_{d-1}\}$ for some predefined integer d.

First, $\mathcal{F}$ outputs the challenge attribute identity ω^* with the condition $|\omega^*| = k \leq d$. Then, $\mathcal{A}$ selects randomly a subset $\Omega^* \subseteq \Omega$ with $|\Omega^*| = d - k$.

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

Simulation of Random Oracle Assume $\mathcal{F}$ makes at most q_{H_1} times to H_1-oracle and q_{H_2} times to H_2-oracle, respectively. $\mathcal{C}$ maintains a list $\mathcal{L}_1$ and $\mathcal{L}_2$ to store the answers of H_1-oracle and H_2-oracle. 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, $\mathcal{A}$ checks the list $\mathcal{L}_1$. And it works as follows:

- If an entry for the query is found in $\mathcal{L}_1$, the same answer will be returned to $\mathcal{F}$.
- Otherwise, it simulates as follows:
  1. If $i \in \omega^* \cup \Omega^*$, it chooses a random $\beta_i \in \mathbb{Z}_p$ and answers $H_1(i) = g^{\beta_i}$.
  2. 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, $\mathcal{A}$ checks the list $\mathcal{L}_2$. And it works as follows:

- If an entry for the query is found in $\mathcal{L}_2$, the same answer will be returned to $\mathcal{F}$.
- Otherwise, it simulates as follows:
  1. If $i \neq \delta$, it chooses random $\alpha_i, \beta_i \in \mathbb{Z}_p$ and answers $H_2(m_i) = g_1^{\alpha_i} g^{\beta_i}$.
  2. If $i = \delta$, it chooses random $\beta_i \in \mathbb{Z}_p$ and answers $H_2(m_i) = g^{\beta_i}$.

Simulation of Private Key Extraction Oracle Assume $\mathcal{F}$ makes at most q_K private key extraction queries. $\mathcal{F}$ can make requests for private keys on ω such that $\omega^* \nsubseteq \omega$. We show how $\mathcal{A}$ simulate a private key on ω on request. We first define three sets Γ, Γ', 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 $i \in \Gamma'$: $D_i = (g_2^{\tau_i} H_1(i)^{r_i}, g^{r_i})$, where τ_i, r_i is 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 $i \notin \Gamma'$, D_i could also be simulated as
  $D_i = (g_2^{\frac{\Delta_{0,S}(i)\gamma_i}{\beta_i} + \Sigma_{j \in \Gamma'} \Delta_{j,S}(i) q(j)} (g_1^{-\beta_i} g^{\gamma_i})^{r_i'},\ g_2^{\frac{\Delta_{0,S}(i)}{\beta_i}} g^{r_i'})$.
  It is correct simulation key because just 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,
  $g_2^{q(i)} H_1(i)^{r_i} = g_2^{\frac{\Delta_{0,S}(i)}{\beta_i} + \Sigma_{j \in \Gamma'} \Delta_{j,S}(i) q(j)} H_1(i)^{r_i'}$, and
  $g^{r_i} = g_2^{\frac{\Delta_{0,S}(i)}{\beta_i}} g^{r_i'}$.

Simulation of Signing Oracle $\mathcal{F}$ also makes requests for signature query on message m for any attributes ω.

If $\omega^* \nsubseteq \omega$, then, $\mathcal{A}$ 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 $\omega^* \subseteq \omega$, $\mathcal{A}$ selects a random $(d-|\omega|)$-element subset Ω' from Ω. If $H_2(m) \neq g^{\beta_i}$, $\mathcal{A}$ can simulate the signature as follows:

Assume that $\omega \cup \Omega' = \{i_1, i_2, \cdots, i_d\}$. First, it chooses $d-1$ values $\tau_{i_k} \in \mathbb{Z}_p$ and lets $q(i_k) = \tau_{i_k}$ for $1 \leq k \leq d-1$. For these points, $(g_2^{q(i_k)} (H_1(i_k))^{r_{i_k}} (H_2(m))^{s_k}$, $g^{r_{i_k}}$, $g^{s_k})$ could be simulated by choosing $s_k \in \mathbb{Z}_p$. The d-th point $q(i_d)$ is also determined because $q(0) = x$, which could be denoted by $q(i_d) = \sum_{k=1}^{d-1} \Delta_{i_k,S}(i_d)$ $(q(i_k)+\Delta_{0,S}(i_d)q(0)$. Thus, in order to simulate $(g_2^{q(i_d)}(H_1(i_d))^{r_d} (H_2(m))^{s_d}, g^{r_d}$, $g^{s_d})$, 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 $\sigma^* = \{(\sigma^*_{v1},\sigma^*_{v2},\sigma^*_{v3})\}_{1\leq v\leq d}$ on message m^* for attributes ω^* with default attributes $\overline{\Omega^*}$. If $H_2(m^*) \neq g^{\beta_\delta}$ or $\overline{\Omega^*} \neq \Omega^*$, $\mathcal{A}$ 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

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

Thus, $\mathcal{A}$ will compute

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

For the success of $\mathcal{A}$, 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 $1/\binom{d-k}{d-1}$. So, we can get the probability of solving CDH problem as $\epsilon' \approx \epsilon/(q_{H_2}\binom{d-k}{d-1})$, if the adversary successes with probability ϵ. □

Appendix B: Proof of Theorem 4

Proof. Suppose an adversary $\mathcal{F}$ has an advantage ϵ in attacking the scheme, we build an algorithm $\mathcal{A}$ that uses $\mathcal{F}$ to solve the CDH problem. Algorithm $\mathcal{A}$ 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 $\{1, 2, \cdots, \ell\}$. And, let the $d-1$ default attributes set $\Omega = \{\ell+1, \ell+2, \cdots, \ell+d-1\}$ for simplicity.

$\mathcal{F}$ outputs the challenge attribute identity ω^* satisfying $|\omega^*| = k \leq d$.

Simulation of Setup $\mathcal{A}$ sets $g_1 = X$ and $g_2 = Y$. It selects randomly a subset $\Omega^* \subseteq \Omega$ with $|\Omega^*| = d - k$. For all $i \in \omega^* \cup \Omega^*$, it chooses random $\beta_i \in \mathbb{Z}_p$ and sets $h_i = g_1^{-1} g^{\beta_i}$. For all $i \notin \omega^* \cup \Omega^*$, it chooses random $\beta_i \in \mathbb{Z}_p$ and sets $h_i = g^{\beta_i}$. Then it 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, $\overrightarrow{a} = (a_i)$, where the elements of $\overrightarrow{a}$ 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, $\overrightarrow{b} = (b_i)$, where the elements of $\overrightarrow{b}$ are chosen at random in $\mathbb{Z}_p$. These values are all kept internal to the simulator. It then assigns $u' = g_1^{p-kt+a'} g^{b'}$ and $u_i = g_1^{a_i} g^{b_i}$ for $1 \leq i \leq n$. The system parameters $\mathsf{params}= (g, g_1, \mathsf{H} = (h_i), u', \mathsf{U} = (u_i))$ are sent to $\mathcal{F}$. To make the notation easier to understand, the following two pairs of functions are defined for a message $m = \{\mu_1, \cdots, \mu_n\} \in \{0,1\}^n$. We define $F(m) = (p - tk) + a' + \sum_{i=1}^{n} a_i^{\mu_i}$. Next, we define $J(m) = b' + \sum_{i=1}^{n} b_i^{\mu_i}$. Finally, we define a binary function $K(m)$ 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 $\mathcal{F}$ makes at most q_K private key extraction queries. $\mathcal{F}$ makes requests for private keys where $\omega^* \nsubseteq \omega$. We first define three sets Γ, Γ', 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, we define the private key components D_i:

For $i \in \Gamma'$: Choose $s_i, r_i \in \mathbb{Z}_p$ and let $q(i) = s_i$. Then output $D_i = (g_2^{s_i} (g_1 h_i)^{r_i}, g^{r_i})$.

We have chosen a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly in addition to having $q(0) = x$. $\mathcal{A}$ is able to calculate the simulated private key for $i \notin \Gamma'$ as $D_i = (g_2^{\Sigma_{j \in \Gamma'} \Delta_{j,S}(i) q(j)} g_2^{\beta_i \Delta_{0,S}(i)} (g_1 h_i)^{r'_i}, g_2^{\Delta_{0,S}(i)} g^{r'_i})$ to $\mathcal{F}$.

It is easy to verify this is a valid: *i.e.*, it is required to show that $D_i = (g_2^{q(i)} (g_1 h_i)^{r_i}, g^{r_i}) = (g_2^{\Sigma_{j \in \Gamma'} \Delta_{j,S}(i) q(j)}\ g_2^{\beta_i \Delta_{0,S}(i)} (g_1 h_i)^{r'_i}, g_2^{\Delta_{0,S}(i)}\ g^{r'_i})$. 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, $\mathcal{A}$ does not know the value of r_i), then $g_2^{q(i)} (g_1 h_i)^{r_i} = g_2^{\Sigma_{j \in \Gamma'} \Delta_{j,S}(i) q(j)}\ g_2^{\beta_i \Delta_{0,S}(i)}\ (g_1 h_i)^{r'_i}$ and $g^{r_i} = g_2^{-\Delta_{0,S}(i)} g^{r'_i}$. Therefore, the simulator is able to 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 $\mathcal{F}$ also makes requests for signature query on message $m = (\mu_1, \cdots, \mu_n) \in \{0,1\}^n$ for any attributes ω.

If $\omega^* \nsubseteq \omega$, then, $\mathcal{A}$ 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 $K(m) = 0$, $\mathcal{A}$ will abort. Otherwise, $\mathcal{A}$ selects a random $(d - |\omega|)$-element subset Ω' from Ω and can simulate the signature as follows:

Assume that $\omega \cup \Omega' = \{i_1, i_2, \cdots, i_d\}$. Firstly, it chooses $d-1$ values μ_{i_k} and lets $q(i_k) = \tau_{i_k}$ for $1 \leq k \leq d-1$. For these points, $(g_2^{q(i_k)} (g_1 h_{i_k})^{r_{i_k}} (u' \prod_{j=1}^{n} u_j^{\mu_j})^{s_k}, g^{r_{i_k}}, g^{s_k})$ 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 $q(i_d) = \sum_{k=1}^{d-1} \Delta_{i_k,S}(i_d)\ q(i_k) + \Delta_{0,S}(i_d) q(0)$. So, in order to simulate $(g_2^{q(i_d)} (g_1 h_{i_d})^{r_{i_d}} (u' \prod_{j=1}^{n} u_j^{\mu_j})^{s_d}, g^{r_{i_d}}, g^{s_d})$, let $s_d = -\frac{\Delta_{0,S}(i_d)}{F(m)} y + s'_d$. Then, $(g_2^{q(i_d)} (g_1 h_{i_d})^{r_{i_d}} (u' \prod_{j=1}^{n} u_j^{\mu_j}) = (g_1 h_{i_d})^{r_{i_d}} g_2^{\sum_{k=1}^{d-1} \Delta_{i_k,S}(i_d)(q(i_k)} g_2^{\frac{-J(m)\Delta_{0,S}(i_d)}{F(m)}} (g_1^{F(m)} g^{J(m)})^{s'_d}$. Meanwhile, $g^{s_d} = g_2^{\frac{-\Delta_{0,S}(i_d)}{F(m)}}$.

Forgery Finally, the adversary outputs a forged signature $\sigma^* = \{(\sigma^*_{v1}, \sigma^*_{v2}, \sigma^*_{v3})\}_{1 \leq v \leq d}$ on message $m^* = (\mu_1^*, \cdots, \mu_n^*)$ for ω^* with default attribute subset $\overline{\Omega^*}$. If it is valid, then

$$\prod_{v=1}^{d} \left(\frac{e(\sigma^*_{v1}, g)}{e(g_1 h_{i_v}, \sigma^*_{v2}) e(u' \prod_{j=1}^{n} u_j^{\mu_j^*}, \sigma^*_{v3})}\right)^{\Delta_{i_v,S}(0)} = e(g_1, g_2)$$

If $a' + \sum_{i=1}^{n} a_i^{\mu_i^*} \neq kt$ or $\overline{\Omega^*} \neq \Omega^*$, the challenger will abort.

Otherwise, because $K(m^*) = 0$ and $\overline{\Omega^*} = \Omega^*$, we have
$\prod_{v=1}^{d} \left(\frac{e(\sigma^*_{v1}, g)}{e(g, (\sigma^*_{v2})^{\beta_{i_v}}) e(g, (\sigma^*_{v3})^{J(m^*)})}\right)^{\Delta_{i_v,S}(0)}$
$= \prod_{v=1}^{d} (e(\sigma^*_{v1} / [(\sigma^*_{v2})^{\beta_{i_v}} (\sigma^*_{v3})^{J(m^*)}], g))^{\Delta_{i_v,S}(0)}$
$= \prod_{v=1}^{d} e((\sigma^*_{v1}) / [(\sigma^*_{v2})^{\beta_{i_v}} (\sigma^*_{v3})^{J(m^*)}])^{\Delta_{i_v,S}(0)}$
$= e(g_1, g_2)$
$= e(g^{xy}, g)$.
So, $\mathcal{A}$ will compute

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

It remains to analyze the probability of $\mathcal{A}$ not aborting. For the simulation to complete without aborting, we require that all signature queries on m will have $K(m) \neq kt$, that forgery signature on message m^* has $K(m^*) = 0 \bmod p$ and $\overline{\Omega^*} = \Omega^*$. In fact, the probability analysis is very similar to [22]. So, we can get the probability of solving CDH problem as $\epsilon' = \epsilon / (16 q_S (n+1) \binom{d-k}{d-1})$ if the adversary success with probability ϵ. $\square$