Mobile Information Systems 8 (2012) 255–274
DOI 10.3233/MIS-2012-0142
IOS Press

Flexible multi-authority attribute-based signature schemes for expressive policy

Dan Cao, Baokang Zhao*, Xiaofeng Wang and Jinshu Su
School of Computer, National University of Defense Technology, Changsha, Hunan, China

Abstract. Attribute-based signature (ABS) is a new cryptographic primitive, in which a signer can sign a message with his attributes, and the verifier can only known whether the signer owns attributes satisfying his policy. Moreover, the signature cannot be forged by any user not having attributes satisfying the policy. ABS has many applications, such as anonymous authentication, and attribute-based messaging systems. But many applications may require a user obtaining attributes from different authorities, which calls for multi-authority ABS schemes.

In this paper, we first propose a multi-authority ABS scheme, called TR_MABS, adopting an attribute tree to support expressive policy consisting of AND, OR, threshold gates. As TR_MABS brings in expensive cost on adding or removing attribute authorities, we present another multi-authority ABS scheme, named DNF_MABS, which uses a disjunctive normal form (DNF) to express a policy, bringing in the capability of implementing NOT gate. To prevent collusion attack, we adopt a unique global identity (GID) for a user to combine his attribute keys and identity. Moreover, we use a central authority to assure the usability of attribute keys a user getting from different attribute authorities, make the verification independent of user's identity, and allow attribute authorities' dynamic change. Our schemes fit the requirements of applications, and also distribute the trust to authorities in the system. In addition, we prove the security of our schemes under computational Diffie-Hellman assumption.

Keywords: ABS, multi-authority, trust, policy, tree, DNF, GID, central authority

1. Introduction

In attribute-based systems, users obtain multiple attributes from one or more attribute authorities, and their roles depend on the combination of attributes they possess. A user's capabilities (e.g., sending or receiving messages, access to a resource) depend on their attributes. While offering several advantages, including low cost on management and flexibility on access control, such systems also present fundamental cryptographic challenges. Recently, attribute-based encryption schemes [1–3] have emerged to tackle some of encryption challenges. And solutions for authentication have been proposed one after the other [4–7].

The requirement of authentication in an attribute-based system differs from that offered by digital signatures, in much the same way public-key encryption does not fit the bill for attribute-based encryption. An attribute-based solution requires unforgeability and signer anonymity, similar to group signatures [8], ring signatures [9], and mesh signatures [10]. A valid signature does not reveal any further information about which of the ways was actually used to generate it. However, a ring signature reveals the list of possible signers explicitly, and a group manager in the group signature can revoke the anonymity

*Corresponding author: Dr. Baokang Zhao, School of Computer, National University of Defense Technology, Changsha 410073, Hunan, China. Tel.: +86 731 84575815; E-mail: bkzhao@nudt.edu.cn.

of any signer. And mesh signatures require collusion of users, which compromises unforgeability of a signature.

To satisfy the requirements of authentication in attribute-based systems, attribute-based solutions have been proposed, such as attribute-based group signatures (ABGS) [4,11–14], attribute-based ring signatures (ABRS) [7,15,16], and attribute-based signatures (ABS) [5,6,17–26]. But ABGS has the drawback of anonymity revocation as the group signature, which cannot satisfy the requirement of signer privacy. And ABRS reveals the attributes of a signer used to generate the signature.

ABS was proposed in [5,6], where users cannot forge signatures with attributes they do not possess even through colluding. Although the ABS scheme in [5] reveals the set of attributes satisfying the policy, subsequent research of ABS offers an attribute-signer privacy guarantee for the signer, that is, a legitimate signer remains anonymous without the fear of revocation and is indistinguishable among all the users whose attributes satisfying the policy specified in the signature. ABS is useful in many important applications such as access control in attribute-based systems.

In an attribute-based signature scheme, users get their secret keys according to their attributes from an attribute authority. A signer with an attribute set can use his secret key to sign messages using any subset of his attribute set. A signature can be verified against a policy of attributes, and verification succeeds if the signer's attribute set satisfying the policy. Under this notion, a signature attests not to the identity of the individual who signed a message, but a claim regarding the attributes the underlying signer possesses.

According to the policy the scheme supporting, existing works of ABS can be divided into threshold ABS [17,18,22,24] and expressive ABS [6,19–21,23,25,26]. The formers only support a threshold policy under computational Diffie-Hellman problem, while the latter supports an expressive policy consisting of AND, OR, threshold gates. And the scheme in [21] even support NOT gate. To our knowledge, most ABS schemes cannot simultaneously implement an expressive policy and construct the security in standard model, that is, Diffie-Hellman assumptions, which are as famous as Diffie-Hellman key based methods used in [27,28] to improve the security of wireless networks, in which other techniques, such as register allocation method [29], MISP protocol [30], and SVO logic [31], have been utilized too.

So far, most of existing works build the ABS scheme with a single attribute authority, putting trust on a single authority. However, the applications in real world often require a user owning some attributes obtained from different authorities (e.g., different government agencies, different commercial services he has subscribed to, different social networks he is registered with and so on). Therefore, to reduce the trust of the single authority, and to satisfy the requirement of real applications, the research of ABS should be extended to multi-authority ABS. There are several works on multi-authority ABS [6,18,21], but the security [6], the policy supported [18], or the efficiency [21] is limited on account of the original ABS scheme.

There are three sixty-four-dollar questions in designing a multi-authority ABS scheme, as the mutually distrusting attribute authorities in the system may not trust each other, and may not even be aware of each other. The first one is how to ensure the user's attribute keys getting from distributed attribute authorities work correctly. Next, it is how to prevent collusion attacks from different users as they may aggregate their attribute keys to forge a valid signature that the individual cannot generate. Last come the cost of management, as an attribute authority may dynamically moves into or out of the system.

To solve above three questions, we use two main techniques. The first is to require that every user has a global identifier (GID), so that no user can claim another user's identifier, and all authorities can verify a user's identifier and then embed it in the user's secret keys to prevent collusion. To assure that the verification is independent of the GID, a central authority (CA) is needed, which is the second main technique we use. Each user will send his GID to the central authority and receive a corresponding key.

And the CA is trusted: it will hold the master secret for the system. As the attribute authorities should use their secret keys to issue attribute keys for a user, to prevent a user with suffcient attributes using his secret keys to reconstruct this secret for each authority, each attribute authority has a pseudorandom function (PRF) to randomize the secret key. The seed of a PRF is generated by the CA to ensure the correctness of the scheme. If and only if the signer owning attributes form different authorities satisfying the policy, the verification succeeds. Moreover, the CA and distributed attribute authorities architecture brings in the convenience of attribute authority management.

In this paper, we present two multi-authority ABS schemes, named TR_MABS and DNF_MABS respectively. TR_MABS extends the ABS scheme in [26] as it implements an expressive policy in a standard model with efficiency. Our TR_MABS is flexible in expressing a policy, as it adopts the form of an attribute tree. What a pity is that it's a little expensive on adding a new attribute authority, as each authority has to re-compute users' secret keys. Then, we construct DNF_MABS to reduce the cost of authority management. In DNF_MABS, we use a disjunctive normal from (DNF) to express a policy, requiring a signer to analyse it during signing, but brings in the capability of expressing negative attributes. The most important point is that DNF_MABS reduces the cost on adding a new attribute authority dramatically, as old attribute authorities needn't re-compute secret keys for users. In addition, we use the signing technique in [18] to enhance the efficiency.

Our contribution is follows:1) achieving expressive policy, even NOT gate in DNF_MABS; 2) building security on computational Diffie-Hellman assumption; 3) supporting authority's dynamic change, even with low cost in DNF_MABS. Both schemes achieve the requirement of real applications, and reduce the trust on one authority.

The remainder of this paper is organized as follows. In Section 2, we review related works. Then, we introduce the concepts of preliminaries in Section 3, followed by the definition of our TR_MABS and DNF_MABS schemes and its security requirements in Section 4. After that, we construct our TR_MABS, and analyze its security in Section 5. And in Section 6, we give the construction and security analysis of DNF_MABS. Finally, we compare TR_MABS and DNF_MABS in Section 7, before drawing our conclusion in Section 8.

2. Related work

Here, we introduce existing works of attribute-based encryption (ABE) and attribute-based signatures (ABS).

2.1. Attribute-based encryption

Extensive research has been done since the introduction of attribute-based encryption (ABE) [1]. In an ABE system, a ciphertext is labeled by the sender with a set of attributes. And user's private key is associated with attributes too. Only the user with attributes satisfying a threshold predicate can decrypt the ciphertext. Goyal et al. [2] extended it to key-policy ABE (KP-ABE) by allowing users' private keys to include any policy consisting of AND, OR and threshold gates. Ostrovsky et al. [32] further extended the ABE that support users' private key to include negative constraints. Bethencourt et al. [3] formalized the notion of ciphertext-policy ABE (CP-ABE) and provided a construction. In CP-ABE, 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 CP-ABE was first mentioned by Goyal et

al. [2]. Compared with ABS, ABE requires interaction between two participants to realize the access control.

There are also many other research topics in ABE, for example, to improve the efficiency of the ABE [33], utilized underlying relation-ship among the attributes and proposed a hierarchical ABE scheme. To prevent the users from sharing their attribute private keys, which are related to the privileges assigned in the access control system [34], proposed the notion of accountable ABE.

To reduce the trust of a single attribute authority, Chase [35] and Božović et al. [36] proposed multi-authority ABE schemes with a central authority to reduce trust on attribute authority, where each attribute authority issues only a part of the attributes. To reduce the trust of central authority further, Lin et al. [37] and Chase and Chow [38] presented multi-authority schemes without central authority successively. However, the structure with a central authority has the advantage of adding or removing an attribute authority, which doesn't affect the attribute keys a user obtained from other attribute authorities. Whereas the structure with mutually distributing attribute authorities spends cost on interactions, and is expensive on additions and deletions of authorities. These multi-authority works are based on the original ABE scheme in [1]. Before long, the research of multi-authority CP-ABE comes forth [39–41]. Muller et al. [39,40] presented distributed attribute-based encryption (DABE) with the architecture of one master and multiple attribute authorities. However, Lewko and Waters [41] didn't adopt a central authority, only used a hash function on the user's global identity, GID to manage collusion resistance across multiple key generations issued by different authorities, and utlized the recent dual system encryption methodology [42] to assure the security.

2.2. Attribute-based signatures

Recently, there have been several attempts to construct attribute-based signatures. As a similar notion to ABS, fuzzy identity-based signature was proposed and formalized in [43,44], which enables users to generate signatures with part of their attributes. To achieve the same goal as the fuzzy identity-based signature, the notion of attribute-based signature was given in [5], but Tan et al. [45] pointed out that this scheme is vulnerable to the partial key replacement attack. Moreover, in these works, authors do not consider any notion of privacy, resulting in leaking attributes used in producing signatures to the verifier.

To achieve weak attribute-privacy, Shahandashti and Safavi-Naini [24] presented a (d, n)-threshold ABS scheme, allowing users to sign messages with subset of attributes satisfying the fixed threshold d, and only the d attributes of the signer that are chosen by the signature holder. To support flexible, not fixed, threshold value k, Li and Kim [22] used an default set of attributes during signing, and constructed an (k, d)-threshold ABS scheme, realizing attribute-signer privacy. However, in both schemes, the size of signature components is three times of attributes used for signing. In order to enhance the efficiency, Li et al. [18] integrated all the secret attributes components into one, at the same time maintained attribute-privacy and flexible (k, d)-threshold policy. Moreover, Li et al. extended their construction without random oracles. Kumar et al. [17] did similar work to [18] Whereas, these schemes cannot provide expressive policies, that is, any policy formed by AND, OR and threshold gates.

Maji et al. [6] constructed an ABS scheme using monotone span programs [46] that supports a powerful set of policies, namely, any policy consists of AND, OR and threshold gates. It holds signer privacy against everyone, including all authorities. However, the security is weak as the construction is only proved in the generic group model. Then, they [19,20] proposed the first ABS scheme with full security proven in the standard model. However, it is much less efficient and more complicated than the scheme in [6].

Cao et al. [26] present the first ABS scheme supporting policies consisting of AND, OR, threshold gates and achieving security in random oracle model under computational Diffie-Hellman problem at the same time, while holding attribute-signer privacy and unforgeability. Moreover, the scheme cost less on signature generation and verification than the scheme in [6].

Okamoto and Takashima [21] present the first ABS scheme to support general non-monotone predicates, which can be expressed using NOT gates as well as AND, OR, and Threshold gates. It has full security under the decisional linear (DLIN) assumption and the existence of collision resistant (CR) hash functions. Although it has full security and perfect signer privacy, it is less efficient than Maji et al. [6], which cost more than Cao et al. [26] on signature generation and verification.

Escala et al. [23] proposed the first attribute-based signature scheme with revocability, that is, an external judge can break the anonymity of a signature, when necessary. It's fully secure in standard model, and supports general signing policies.

Most of above works commonly build their ABS schemes with random oracles. Up to the present, only Li et al. [18] gave a construction without random oracles based on their original ABS scheme, following Cao et al. [25] constructing an expressive ABS without random oracles according to the scheme in [26].

Moreover, few works of ABS have extended research on multi-authority ABS, except that Maji et al. [6], Li et al. [18] and Okamoto and Takashima [21] extended their original schemes to multi-authority ABS. However, as the original ABS in [6] built its security under generic group model, the multi-authority ABS Maji et al. presented also had a weak security. In addition, the multi-authority ABS in [18] only supported a threshold policy as its original scheme did. And the multi-authority ABS scheme in [21] has low efficiency.

3. Preliminaries

In this Section we briefly review the basic concepts on bilinear pairing, Lagrange interpolation, complexity assumptions, attribute tree, and disjunctive normal form (DNF), while introducing notations used in this paper.

Notation. Let q be a prime. From here on we use $\mathbb{Z}_q$ to denote the group $\{0, \ldots, q-1\}$ under addition modulo q. Let G_1 and G_2 be multiplicative groups of order q. Let g denote a generator of G_1.

Bilinear pairing. A bilinear pairing is a map $e : G_1 \times G_1 \to G_2$ with the following properties:

(a) Bilinear: Given $a, b \in \mathbb{Z}_q$, $f, h \in G_1$, we have $e(f^a, h^b) = e(f, h)^{ab}$.
(b) Non-degenerate: $e(g, g) \neq 1$ and therefore it is a generator of G_2.
(c) Computable: There is an effcient algorithm to compute $e(f, h)$ for all $f, h \in G_1$.

Lagrange interpolation. Lagrange interpolation for a polynomial $p(\cdot)$ over $\mathbb{Z}_q$ of order $d-1$ and a set $S \subset \mathbb{Z}_q$ with size $|S| = d$ is calculate as

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

where

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

Complexity assumptions. Here we describe two mathematical problems.

(a) Discrete Logarithm(DL) Assumption: Given two group elements f and h in G_1, find an integer n, such that $h = f^n$.
(b) Computational Diffie-Hellman(CDH) Assumption: Given (g, g^a, g^b) for some $a, b \in \mathbb{Z}_q^*$, compute g^{ab}.

Attribute tree. An attribute tree Γ can describe a predicate. Each interior node is a threshold gate, described by its children and a threshold value. If num_x is the number of children of a node x and k_x is its threshold value, then $0 < k_x \leqslant num_x$. This setting is expressive to represent AND ($k_x = num_x$), OR ($k_x = 1$), and threshold predicates over attributes. The children of every node are numbered from 1 to *num*. The function index (x) returns such a number associated with the node x, where the index values are uniquely assigned to nodes in the attribute tree in an arbitrary manner. Let function parent(x) denote the parent of the node x. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$. The function att(x) denotes the attribute associated with the leaf node x.

Disjunctive normal form (DNF). In proposition logic systems, a disjunctive normal form is defined as follows:

(a) If t is a proposition variant, t or $\sim t$ is a literal.
(b) The disjunction of limited literals is a clause. And the conjunction of limited clauses is a conjunctive term.
(c) The conjunction of limited clauses is called conjunctive normal form. And the disjunction of conjunctive terms is called disjunctive normal form.

4. Definitions

In this section, we introduce our multi-authority systems, and formalize the definition and security model of our proposed multi-authority attribute-based signature schemes, that is, TR_MABS and DNF_MABS, using an attribute tree or a DNF to express a verification policy respectively.

4.1. Systems

In both systems of TR_MABS and DNF_MABS schemes, there are three kinds of entities: central authority (CA), attribute authorities (AAs) and users. The CA and each AA have their own private keys to issue keys to users. Furthermore, the AAs will receive their private keys from the CA. These different AAs may not trust each other, and may not even be aware of each other. They issue attribute keys to users independently. However, the CA will be trusted by all AAs. It knows enough of secret state of AAs to reconstruct secret values, which will be used to generate attribute keys for any user, for all authorities. These architectures allow adding or deleting an AA dynamically.

To prevent collusion attack from different users, we use a unique global identity (GID) of a user to bind his attributes and identity together, thus different users cannot pool their attribute keys to imitate a valid signer. A user must present the same GID to each authority in order to receive a coherent set of keys. However, the policy only specifies some attributes to be owned by a signer. Thus, the ability to verify a signature is independent of the GID (except in that all secret keys must have been obtained for the same GID). To implement this, the CA reconstructs all AAs' secret values and generates a secret key d_{CA} for the user.

Here we assume that there're K attribute authorities. A user with a GID u obtains an attribute set $\omega_{k,u}$ from the k-th attribute authority AA_k ($k = 1, \ldots, K$). Instead of using truly random values, we let each

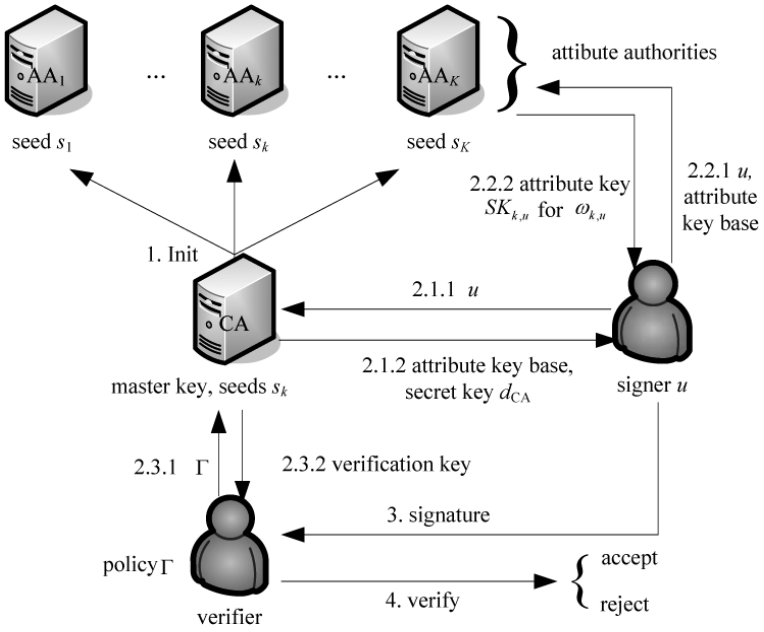

Fig. 1. Procedure of TR_MABS.

of our K authorities choose the secret value $y_{k,u}$ for user u using a pseudorandom function (PRF) f. Thus, now the CA has only to store the seeds s_k of all of the PRFs.

In TR_MABS, a user u gets his secret key and attribute key base from the CA. Then, he goes to AA_k to obtain attribute set $\omega_{k,u}$ and its corresponding key $SK_{k,u}$. After that, he generates a signature. Before verifying a signature, the verifier with a policy Γ gets a verification key *gpk* from the central authority. If and only if the signer owns attributes satisfying the policy, the check of the signature can success. We conclude the procedure of TR_MABS in Fig. 1.

The procedure of DNF_MABS is similar to that of TR_MABS, but simpler. We conclude it in Fig. 2. The difference from TR_MABS is follows:

(1) The CA needn't compute an attribute key base of each AA for the user.
(2) The CA needn't generate a verification key corresponding to a policy $\mathbf{A}$, which is a DNF over attributes.
(3) AAs can issue negative attribute to users.

4.2. Syntax

The proposed TR_MABS scheme consists of following algorithms:

TR.Setup is the algorithm run by the CA on inputs the security parameter, and outputs public parameters *params*, a private master key *MK* and seeds s_k for AA_k ($k = 1, \ldots, K$).

TR.KeyGen$_{private}$ is the algorithm run by the CA and AA_k on inputs *params*, *MK*, s_k, signer's GID u and attribute subset $\omega_{k,u}$, and outputs a secret key d_{CA} and attribute key $SK_{k,u}$ for the signer.

TR.KeyGen$_{public}$ is the algorithm run by the CA on inputs *params*, *MK* and an attribute tree Γ, and outputs a public key *gpk* for the verifier.

TR.Sign is the algorithm run by a signer on inputs *params*, policy Γ, d_{CA}, $SK_{k,u}$ and a message M, and generates a signature σ on the message.

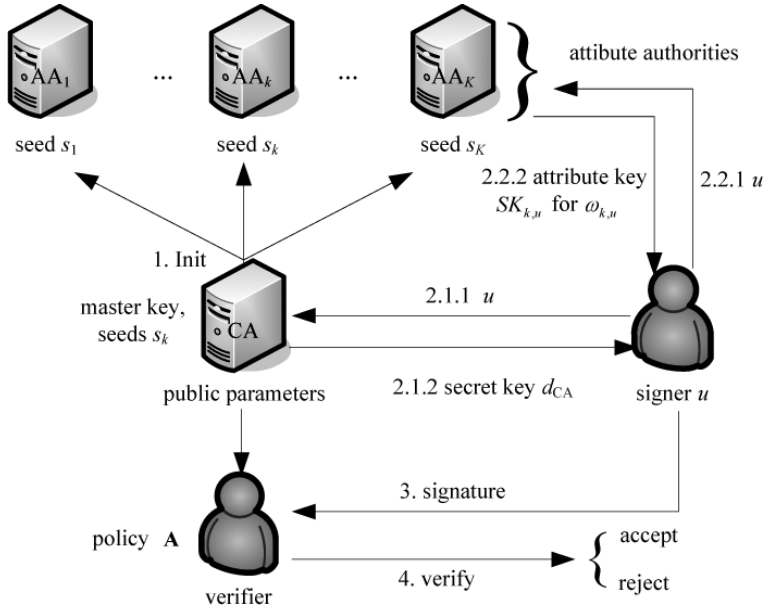

Fig. 2. Procedure of DNF_MABS.

TR.Verify is the algorithm run by a verifier on inputs *params*, a message signature pair (M, σ), and a verification key *gpk* and outputs a boolean value *accepte* if σ is a valid signature by a signer who has attributes $\omega = \bigcup_{k=1}^{K} \omega_{k,u}$ satisfies Γ, i.e., $\Gamma(\omega) = 1$.

The proposed DNF_MABS scheme consists of following algorithms:

DNF.Setup is by the CA, inputs the security parameter and a threshold value d, and outputs public parameters *params*, a private master key *MK* and seeds s_k for AA_k $(k = 1, \ldots, K)$.

DNF.KeyGen is run by the CA and AA_k, inputs *params*, *MK*, s_k, signer's GID u and attribute subset $\omega_{k,u}$, and outputs a secret key d_{CA} and attribute key $SK_{k,u}$ for the signer.

DNF.Sign is run by a signer, inputs *params*, policy $\mathbf{A}$, d_{CA}, $SK_{k,u}$ and a message M, and generates a signature σ on the message.

DNF.Verify is run by a verifier, inputs *params*, a message signature pair (M, σ), and policy $\mathbf{A}$, and outputs a boolean value *accepte* if σ is a valid signature by a signer who has attributes $\omega = \bigcup_{k=1}^{K} \omega_{k,u}$ satisfies $\mathbf{A}$, i.e., $\mathrm{A}(\omega) = 1$.

4.3. Security model

The basic requirement of our schemes is correctness. As an adversary may try to forge a signature with a policy that his attributes do not satisfy, the essential security requirement can be formally summarized as unforgeability.

Definition 1 (**Correctness**): We call the TR_MABS scheme correct if for all (*params*, s_k, *MK*) $\leftarrow$ TR.Setup, all messages M, all attribute sets $\omega = \bigcup_{k=1}^{K} \omega_{k,u}$, all secret keys $\{d_{CA}, SK_{k,u}\} \leftarrow$

TR.KeyGen$_{private}$ (*params*, *MK*, s_k, u, $\omega_{k,u}$), all verification keys *gpk* $\leftarrow$ *TR.KeyGen*$_{public}$ (*params*, *MK*, Γ), and all attribute trees Γ such that $\Gamma(\omega) = 1$,

$$\text{TR.Verify}(\textit{params}, M, \textit{gpk}, \text{TR.Sign}(\textit{params}, d_{CA}, \textit{SK}_{k,u}, M, \Gamma)) = \textit{accept}$$

with probability 1 over the randomness of all the algorithms.

We call the DNF_MABS scheme correct if for all (*params*, s_k, *MK*) $\leftarrow$ DNF.Setup(d), all messages M, all attribute sets $\omega = \bigcup_{k=1}^{K} \omega_{k,u}$, all secret keys $\{d_{CA}, \textit{SK}_{k,u}\} \leftarrow$ DNF.KeyGen(*params*, *MK*, s_k, u, $\omega_{k,u}$), and all DNFs $\mathbf{A}$ such that $\mathrm{A}(\omega) = 1$,

$$\text{DNF.Verify}(\textit{params}, M, \mathbf{A}, \text{DNF.Sign}(\textit{params}, d_{CA}, \textit{SK}_{k,u}, M, \mathbf{A})) = \textit{accept}$$

with probability 1 over the randomness of all the algorithms.

Definition 2 (**Unforgeability**): The TR_MABS scheme is unforgeable if the success probability of any polynomial-time adversary A in the following selective-policy attack, which is denoted by EUF-sP-CMA, is negligible:

- Initial Phase: A chooses and outputs a challenge policy Γ^* that will be included in the forgery signature;
- Setup Phase: After receive the challenge policy Γ^*, the challenger C chooses a suffciently large security parameter κ and runs TR.Setup algorithm to generate public parameters *params*, seeds $\{s_k\}_{k=1,\ldots,K}$ and master key *MK*, and sends *params* to A;
- Query Phase: After receive the public parameters, A can perform a polynomially bounded number of queries on $\omega = \bigcup_{k=1}^{K} \omega_{k,u}$ and (M, Γ) to private key extraction oracle and signing oracle, respectively. C answers the queries with the master key *MK*.
- Forgery: Finally, A outputs a signature σ^* on messages M^* with respect to Γ^*, which is the challenge policy sent to C in the initial phase.

We say that the adversary wins the game if σ^* is a valid signature on message M^* for Γ^*, (M^*, Γ^*) has not been queried to the signing oracle and no attribute set ω^* satisfying $\Gamma^*(\omega^*) = 1$ has been submitted to the private key extraction oracle. The success probability of adversary A is $\mathrm{Succ}^{EUF-sP-CMA}_{TR\_MABS,A}(\kappa) = \Pr[\text{TR.Verify}(\sigma^*, M^*, \Gamma^*)] = 1$.

The definition of unforgeability of DNF_MABS is similar to the above definition of that of TR_MABS, except the form of policy is DNF $\mathbf{A}$.

Collusion resistance: It is important to note that the above definition of unforgeability guarantees collusion resistance in the sense that no colluding group of users can generate a signature that is not generable by one of the colluders. This is because if a group of signers can construct a signature that none of them could individually produce, then they can build another adversary and output a forgery to win the above game.

5. Proposed TR_MABS

To satisfy the requirements of real applications, which always require a signer owning attributes from one or more attribute authorities, we extend the attribute-based signature scheme with single authority

presented by Cao et al. [26] to multiple attribute authorities, which maintains the advantages of expressive policy and provable security under standard Diffie-Hellman assumptions. In this section, we give the construction of TR_MABS scheme, following the security analysis.

5.1. Construction of TR_MABS

We now construct the TR_MABS scheme as follows:

TR.Setup: First, define the attributes in universe U as elements in $\mathbb{Z}_q$. Select a random generator $g \in G_1$, a random $\alpha \in \mathbb{Z}_q^*$, and set $g_1 = g^\alpha$. Next, pick a random element $g_2 \in G_1$ and compute $R = g_2^{1/\alpha}$. Then, compute $Z = e(g_1, g_2)$. Choose seeds $s_1, \ldots, s_K$ for all attribute authorities. Two hash functions are also chosen such that $H_1, H_2 : \{0,1\}^* \to G_1$.

The public parameters are *params* $= (q, G_1, G_2, e, g, g_1, g_2, Z, R, H_1, H_2)$. The master key is α.

TR.KeyGen$_{private}$: a user with GID u gets a secret key from the central authority as follows:

$$d_{CA} = g_2^{\alpha - \sum_{k=1}^{K} y_{k,u}} \tag{1}$$

where $y_{k,u} = f_{s_k}(u)$.

And for each attribute authority AA_k, the user gets an attribute key base

$$T_k = g_2^{\sum_{\substack{j=1 \\ j \neq k}}^{K} y_{j,u}/\alpha} \tag{2}$$

Then, user u sends his attribute key base to AA_k and gets attribute key $SK_{k,u}$ for an attribute set $\omega_{k,u}$ from AA_k:

First, compute $y_{k,u} = f_{s_k}(u)$. And for each attribute $i \in \omega_{k,u}$, choose $r_{ki} \in \mathrm{Z}_q$ and compute

$$d_{ki0} = T_k R^{y_{k,u}} H_1(i)^{r_{ki}} \tag{3}$$

$$d_{ki1} = g^{r_{ki}} \tag{4}$$

Then return $SK_{k,u} = \{d_{ki0}, d_{ki1}\}_{i \in \omega_{k,u}}$ to the user.

Finally, user's attribute key is $\{d_{i0} = \{d_{ki0}\}, d_{i1} = \{d_{ki1}\}\}_{i \in \cup \omega_{k,u}, 1 \leqslant k \leqslant K}$.

TR.KeyGen$_{public}$: To generates a verification key of a specific attribute tree Γ, choose a polynomial p_x of degree $d_x = k_x - 1$ for each node in the tree, where k_x is the threshold gate. That is done in a top-down manner. Starting from the $p_{root}(0) = \alpha$ and d_{root} other points in the polynomial will be random. The other nodes we set $p_x(0) = p_{parent(x)}$ (index(x)) and choose d_x other points randomly.

Once the polynomials have been decided, the verification key *gpk* for Γ is

$$\{D_x = g^{p_x(0)}, h_i = H_1(i)^{p_x(0)}\} \tag{5}$$

where $i = \mathrm{att}(x)$, x is a leaf node.

TR.Sign: Suppose one has a private key for the attribute set $\{\omega_{k,u}\}$ for $1 \leqslant k \leqslant K$. To sign a message M with Γ, namely, to prove $\omega = \bigcup_{k=1}^{K} \omega_{k,u}$ satisfies the tree, i.e. $\Gamma(\omega) = 1$, he proceeds as follows:

Choose a random $s \in \mathbb{Z}_q$ and compute

$$\sigma_0 = H_2(M)^s d_{CA} \tag{6}$$

$$\sigma_0' = g^s \tag{7}$$

Let ω^* denote the attribute set associated with leaves in Γ. For each $i \in \omega^*$, choose $r_i' \in \mathbb{Z}_q$ randomly and compute

$$\{\sigma_{i0} = d_{i0}H_1(i)^{r_i'}, \sigma_{i1} = d_{i1}g^{r_i'}\}_{i\in\omega\cap\omega^*} \tag{8}$$

$$\{\sigma_{i0} = H_1(i)^{r_i'}, \sigma_{i1} = g^{r_i'}\}_{i\in\omega^*/\omega\cap\omega^*} \tag{9}$$

Finally, he outputs the signature $\sigma = (\sigma_0, \{\sigma_{i0}, \sigma_{i1}\}_{i\in\omega^*}, \sigma_0')$.

TR.Verify: To verify the signature, we first define a recursive algorithm VerNode (σ, *gpk*, x) , where x is a node in the tree Γ. It outputs a group element of G_2 or $\perp$.

Let $i = \text{att}(x)$. If the node x is a leaf node then:

$$\text{VerNode}(\sigma, gpk, x) = \begin{cases} e(\sigma_{i0}, D_x)/e(\sigma_{i1}, h_i), \text{if } e(\sigma_{i0}, D_x)/e(\sigma_{i1}, h_i) \neq 1 \\ \perp, \text{otherwise} \end{cases} \tag{10}$$

Notice that if $i \in \omega \cap \omega^*$

$$\begin{aligned} e(\sigma_{i0}, D_x)/e(\sigma_{i1}, h_i) &= e(d_{i0}H_1(i)^{r_i'}, g^{p_x(0)})\Big/e(d_{i1}g^{r_i'}, H_1(i)^{p_x(0)}) \\ &= e(T_kR^{y_{k,u}}H_1(i)^{r_i+r_i'}, g^{p_x(0)})\Big/e(g^{r_i+r_i'}, H_1(i)^{p_x(0)}) \\ &= e(g_2^{\sum_{\substack{j=1 \\ j\neq k}}^{K} y_{j,u}/\alpha} g_2^{y_{k,u}/\alpha}, g^{p_x(0)})e(H_1(i)^{r_i+r_i'}, g^{p_x(0)})/e(H_1(i)^{r_i+r_i'}, g^{p_x(0)} \\ &= e(g, g_2)^{\sum_{k=1}^{K} y_{k,u}/\alpha p_x(0)} \end{aligned} \tag{11}$$

And if $i \in \omega^*/\omega \cap \omega^*$

$$e(\sigma_{i0}, D_x)/e(\sigma_{i1}, h_i) = e(H_1(i)^{r_i'}, g^{p_x(0)})\Big/e(g^{r_i'}, H_1(i)^{p_x(0)}) = 1 \tag{12}$$

For a non-leaf node x. The algorithm VerNode (σ, *gpk*, x) then proceeds as follows: For all nodes z that are children of x, it calls VerNode (σ, *gpk*, z) and stores the output as F_z. Let S_x be an arbitrary k_x-sized set of child nodes z such that $F_z \neq \perp$. If no such set exists then the node was not satisfied and the function returns $\perp$. Otherwise let $i = \text{index}(z)$, $S_x' = \{\text{index}(z) : z \in S_x\}$ and compute:

$$\begin{aligned} F_x &= \prod_{z\in S_x} F_z^{\Delta_{i,S_x'}(0)} \\ &= \prod_{z\in S_x} \left(e(g, g_2)^{\sum_{k=1}^{K} y_{k,u}/\alpha p_z(0)}\right)^{\Delta_{i,S_x'}(0)} \\ &= \prod_{z\in S_x} \left(e(g, g_2)^{\sum_{k=1}^{K} y_{k,u}/\alpha p_{\text{parent}(z)}(\text{index}(z))}\right)^{\Delta_{i,S_x'}(0)} \\ &= \prod_{z\in S_x} e(g, g_2)^{\sum_{k=1}^{K} y_{k,u}/\alpha p_x(i)\Delta_{i,S_x'}(0)} \\ &= e(g, g_2)^{\sum_{k=1}^{K} y_{k,u}/\alpha p_x(0)} \end{aligned} \tag{13}$$

To verify the signature calculate F_{root}. Then check if

$$\frac{e(g,\sigma_0)\cdot F_{root}}{e(H_2(M),\sigma_0')} \stackrel{?}{=} Z \tag{14}$$

If the equation holds then accept the signature, which indicates the signature is indeed from some user with attributes satisfying Γ. Otherwise reject it.

5.2. Security analysis

We give the security analysis of correctness and unforgeability. The correctness of verification is justified by Theorem 1, while the unforgeability can be obtained from Theorem 2.

Theorem 1: Our TR_MABS scheme is correct.

Proof. If and only if the tree is satisfied, i.e. $\Gamma(\omega)=1$, then according to Lagrange interpolation,

$$F_{root} = e(g,g_2)^{\sum_{k=1}^{K} y_{k,u}/\alpha\cdot p_{root}(0)} = e(g,g_2)^{\sum_{k=1}^{K} y_{k,u}/\alpha\cdot\alpha} = e(g,g_2)^{\sum_{k=1}^{K} y_{k,u}} \tag{15}$$

So

$$\begin{aligned}
\frac{e(g,\sigma_0)\cdot F_{root}}{e(H_2(M),\sigma_0')} &= \frac{e(g,H_2(M)^s d_{CA})e(g,g_2)^{\sum_{k=1}^{K} y_{k,u}}}{e(H_2(M),g^s)} \\
&= \frac{e(g,H_2(M)^s g_2^{\alpha-\sum_{k=1}^{K} y_{k,u}})e(g,g_2)^{\sum_{k=1}^{K} y_{k,u}}}{e(H_2(M),g)^s} \\
&= \frac{e(g,H_2(M))^s e(g,g_2)^{\alpha-\sum_{k=1}^{K} y_{k,u}} e(g,g_2)^{\sum_{k=1}^{K} y_{k,u}}}{e(g,H_2(M))^s} \\
&= e(g,g_2)^{\alpha} = e(g^{\alpha},g_2) = e(g_1,g_2) = Z
\end{aligned} \tag{16}$$

Theorem 2: Let A be an adversary that 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, and produces a successful forgery against our scheme with probability ε in time t in EUF-sP-CMA. Then there exists an algorithm B that solves the CDH problem with probability $\varepsilon' \approx \varepsilon/q_{H_2}$ in time $t' < t + (q_{H_1}+q_{H_2}+3q_K+4q_S)t_{\exp}$, t_{exp} is the maximum time for an exponentiation in G_1.

Proof. Suppose that an adversary A has an advantage ε in attacking the scheme, we build an algorithm B that uses A to solve the CDH problem. The simulator B is given an instance $(g, X=g^{\alpha}, Y=g^{\beta}) \in G_1$ of the CDH problem, and must produce $g^{\alpha\beta}$.

First, A outputs the challenge policy, namely, an attribute tree Γ^*. Let ω^* denote the set of attributes associated with leaves in Γ^*. Then, B sets $g_1 = X$ and $g_2 = Y$. Assume that A makes at most q_{H_1} times to H_1-oracle and q_{H_2} times to H_2-oracle, respectively. C maintains lists L_1 and L_2 to store the answers of H_1-oracle and H_2-oracle. In addition, C selects a random integer $\delta \in [1, q_{H_2}]$. If i is sent for query of H_1, B checks the list L_1 and works as follows:

- If an entry for the query is found in L_1, the same answer will be returned to A.
- Otherwise, it simulates as follows:

(1) If $i \in \omega^*$, it chooses a random $\beta_i \in \mathbb{Z}_q$ and answers $H_1(i) = g^{\beta_i}$.
(2) If $i \notin \omega^*$, it chooses random $\beta_i, \gamma_i \in \mathbb{Z}_q$ and answers $H_1(i) = g_1^{-\beta_i} g^{\gamma_i}$.

If M_i is sent for query of H_2, B checks the list L_2. And it works as follows:

– If an entry for the query is found in L_2, the same answer will be returned to A.
– Otherwise, it simulates as follows:

(1) If $i \neq \delta$, it chooses random $\alpha_i, \beta_i' \in \mathbb{Z}_q$ and answers $H_2(M_i) = g_1^{\alpha_i} g^{\beta_i'}$.
(2) If $i = \delta$, it chooses random $\beta_i' \in \mathbb{Z}_q$ and answers $H_2(M_i) = g^{\beta_i'}$.

Assume that A makes at most q_K private key extraction queries. A can make requests for private keys on ω such that $\Gamma^*(\omega) \neq 1$. We show how B simulates a private key on user u and his attribute set ω on request. First, B chooses random values $s_1, \ldots, s_K$ for all attribute authorities. Then simulate the private key components

$$d_{CA} = g_2^{\alpha - \sum_{k=1}^{K} s_k} \tag{17}$$

Because $\Gamma^*(\omega) \neq 1$ and $\omega^* \cap \omega \subseteq \omega$, $\Gamma^*(\omega^* \cap \omega) \neq 1$. Let S be the set of attributes satisfying the tree Γ^* and $\omega^* \cap \omega \subseteq S$, and S_k denote the subset of attributes managed by AA_k so that $S = \bigcup_{k=1}^{K} S_k$. Then simulate the private key components d_{ki0} and d_{ki1} from AA_k as follows:

– For $i \in S_k$: $d_{ki0} = g_2^{\sum_{k=1}^{K} s_k/\alpha} H_1(i)^{r_{ki}}$, $d_{ki1} = g^{r_{ki}}$, where r_{ki} is randomly chosen from Z_q.
– For $i \notin S_k$: just let $r_{ki} = \beta \Delta_{0,S_k}(i)/\beta_i + r'_{ki}$, then d_{ki0} and d_{ki1} could be simulated as follows:

$$d_{ki0} = g_2^{\sum_{k=1}^{K} s_k/\alpha} H_1(i)^{r_{ki}} = g_2^{\sum_{k=1}^{K} s_k/\alpha + \Delta_{0,S_k}(i)\gamma_i/\beta_i} g^{-\alpha\beta\Delta_{0,S_k}(i)} (g_1^{-\beta_i} g^{\gamma_i})^{r'_{ki}} \tag{18}$$

$$d_{ki1} = g^{r_{ki}} = g_2^{\Delta_{0,S_k}(i)/\beta_i} g^{r'_{ki}} \tag{19}$$

The intuition behind these assignments is that the public key of Γ^* used for verification is generated by choosing a random polynomial $p(.)$ for each node from top to down, starting from the $p_{root}(0) = \alpha$ and d_{root} other points in the polynomial will be random. The other nodes we set $p_x(0) = p_{parent(x)}(\mathrm{index}(x))$ and choose d_x other points randomly.

A also makes requests for signature query on message M for an attribute set ω. If $H_2(M) \neq g^{\beta_i}$, B can simulate the signature as follows:

In order to simulate

$$(g_2^{\alpha} g_2^{-\sum_{k=1}^{K} s_k} H_2(M)^s, \{g_2^{\sum_{k=1}^{K} s_k/\alpha} H_1(i)^{r_i}, g^{r_i}\}_{i \in \omega *}, g^s),$$

choose $s', r_i \in \mathbb{Z}_q$ and let $s = -\beta/\alpha_i + s'$. When $H_2(M) = g_1^{\alpha_i} g^{\beta_i}$,

$$g_2^{\alpha} g_2^{\sum_{k=1}^{K} s_k} H_2(M)^s = g^{-\alpha\beta} (g_1^{\alpha_i} g'^{\beta_i})^s g_2^{-\beta_i/\alpha_i} \tag{20}$$

$$g^s = g_2^{-1/\alpha_i} g'^s \tag{21}$$

Finally, the adversary outputs a forged signature σ^* on message M^* for attributes ω'. If $H_2(M^*) \neq g^{\beta'_\delta}$, the simulator B will abort. Otherwise, it satisfies the verification equation, which means that

$$\sigma^* = (\sigma_0^*, \{\sigma_{i0}^*, \sigma_{i1}^*\}_{i \in \omega^*}, \sigma'^*_0) = \left(g_2^{\alpha} g_2^{\sum_{k=1}^{K} s_k} H_2(M^*)^s, \{g_2^{\sum_{k=1}^{K} s_k/\alpha} H_1(i)^{r_i}, g^{r_i}\}_{i \in \omega *}, g^s \right) \tag{22}$$

Before simulating $g^{\alpha\beta}$, B simulates a recursive function ReNode (x,Γ^*), where x is a node in Γ^*. It outputs an element of G_1. When x is a leaf node in Γ^*, let $i = \text{att}(x)$, because $H_1(i) = g^{\beta_i}$,

$$\text{ReNode}(x, \Gamma^*) = ((\sigma_{i1}^*)^{\beta_i}/(\sigma_{i0}^*))^{p_x(0)} = g_2^{-\sum_{k=1}^{K} s_k/\alpha \cdot p_x(0)} \tag{23}$$

As ω^* denote the set of attributes associated with leaves in Γ^*, for each non-leaf node x, ReNode(x, Γ^*) can proceed as follows: For all nodes z that are children of x, it calls ReNode(z, Γ^*) and stores the output as R_z. Let S_x be an arbitrary k_x-sized set of child nodes z. Let $i = \text{index}(z)$, $S'_x = \{\text{index}(z) : z \in S_x\}$ and compute:

$$\begin{aligned} R_x &= \prod_{z \in S_x} R_z^{\Delta_{i,S'_x}(0)} = \prod_{z \in S_x} g_2^{-\sum_{k=1}^{K} s_k/\alpha p_z(0)\Delta_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} g_2^{-\sum_{k=1}^{K} s_k/\alpha p_{parent(z)}(\text{index}(z))\Delta_{i,S'_x}(0)} \\ &= \prod_{z \in S_x} g_2^{-\sum_{k=1}^{K} s_k/\alpha p_x(i)\Delta_{i,S'_x}(0)} = g_2^{-\sum_{k=1}^{K} s_k/\alpha p_x(0)} \end{aligned} \tag{24}$$

So,

$$R_{root} = g_2^{-\sum_{k=1}^{K} s_k/\alpha \cdot p_{root}(0)} = g_2^{-\sum_{k=1}^{K} s_k/\alpha \cdot \alpha} = g_2^{-\sum_{k=1}^{K} s_k} \tag{25}$$

Then, because $H_2(M^*) = g^{\beta'_\delta}$, B can compute

$$g^{\alpha\beta} = R_{root}\sigma_0^*/(\sigma_0'^*)^{\beta'_\delta} \tag{26}$$

For the success of B, we require that forgery signature on message M^* such that $H_2(M^*) = g^{\beta'_\delta}$. Therefore, if the adversary success with probability ε, we can get the probability of solving CDH problem as $\varepsilon' \approx \varepsilon/q_{H_2}$.

6. Proposed DNF_MABS

Enlightened by the form of policy in [40], we construct the DNF_MABS scheme, on CDH assumption, formulating a policy in the form of a Boolean formula over some attributes, that is, Disjunctive Normal Form (DNF). In DNF, all negations are atomic, so there should be negative attributes to make use of the full expressive power of DNF formulas. Therefore, the attribute authorities in this system can issue negative attributes to users as in [21]. But our scheme is more efficient than the multi-authority ABS scheme. in [21], as we adopt the signing technique used in [18]. Certainly, our DNF_MABS scheme supports more expressive policy than the scheme in [18], because Li et al. [18] only implement a threshold policy. This section includes the construction of the DNF_MABS scheme and the security analysis.

6.1. Construction of DNF_MABS

First of all, we add some notations. Let $\mathrm{A} = \bigvee_{j=1}^{n} (\bigwedge_{A \in X_j} A)$ denote a DNF, expressing a policy. Here n (not pairwise disjoint) sets $X_1, \ldots, X_n$ denote attributes that occur in the j-th conjunction

of A. Then, the attribute set associated with the policy is $\omega^* = \bigcup_{j=1}^{n} X_j$. Set $N = |\omega^*|$. As the number of attributes appearing in each conjunction of a DNF may be different, we set a threshold value $d = \max\{|X_j|, j = 1, \ldots, n\}$. If a user owns attributes satisfying the policy A, i.e. $\mathrm{A}(\omega) = 1$, then there exists some j, $X_j \subseteq \omega$. Let $l = |X_j|$ denote the size of X_j, then $1 \leqslant l \leqslant d \leqslant N$.

Now, we give the construction of DNF_MABS.

DNF.Setup(d): First, define the attributes in universe U as elements in $\mathbb{Z}_q$. Assume that there are K distributed attribute authorities. Define a default attribute set Ω_k of d elements for attribute authority AA_k ($k = 1, \ldots, K$). The CA chooses seeds $s_1, \ldots, s_K$ for all attribute authorities, selects a random generator $g \in G_1$, a random $\alpha \in \mathbb{Z}_q^*$, and set $g_1 = g^\alpha$. Next, it picks a random element $g_2 \in G_1$, and computes $Z = e(g_1, g_2)$. Two hash functions are also chosen by CA such that $H_1, H_2 : \{0,1\}^* \to G_1$. The public parameters are *params* = (q, G_1, G_2, e, g, g_1, g_2, Z, H_1, H_2), the master key is α. For the attribute authority AA_k, its secret key is s_k, which is assigned by the central authority. The secret key of CA is $(s_1, \ldots, s_K, \alpha)$.

DNF.KeyGen: A user with GID u gets a secret key from the CA as

$$d_{CA} = g_2^{\alpha - \sum_{k=1}^{K} y_{k,u}} \tag{27}$$

where $y_{k,u} = f_{s_k}(u)$.

Then, he gets his attribute keys from attribute authorities. The k-th attribute authority AA_k issues an attribute key for user's attribute set $\omega_{k,u}$ as follows:

First, choose a $d-1$ degree polynomial $p_k(\cdot)$ randomly such that $p_k(0) = y_{k,u} = f_{s_k}(u)$.

Generate a new attribute set $\widehat{\omega}_k = \omega_{k,u} \cup \Omega_k$. For each $i \in \widehat{\omega}_k$, choose $r_{ki} \in \mathbb{Z}_q$ and compute

$$d_{ki0} = g_2^{p_k(i)} H_1(i)^{r_{ki}} \tag{28}$$

$$d_{ki1} = g^{r_{ki}} \tag{29}$$

Then user's attribute key is $SK_{k,u} = \{d_{ki0}, d_{ki1}\}_{i \in \widehat{\omega}_k}$.

DNF.Sign: Suppose one has a private key for the attribute set $\{\omega_{k,u}\}$ for $1 \leqslant k \leqslant K$. To sign a message M with a policy $\mathrm{A} = \bigvee_{j=1}^{n} (\bigwedge_{A \in X_j} A)$, the signer u should prove that his attribute set $\omega = \bigcup_{k=1}^{K} \omega_{k,u}$ satisfies the DNF A, i.e. $\mathrm{A}(\omega) = 1$. According to the definition of DNF, if one disjunction can be satisfied, the DNF can be satisfied. Therefore, if the signer u find some j, $X_j \subseteq \omega$, the j-th disjunction can be satisfied, then $\mathrm{A}(\omega) = 1$. Maybe the j is not unique. Considering the performance, the signer only choose one j. Let $l = |X_j|$, then $1 \leqslant l \leqslant d \leqslant N$. For each k, at least l_k out of l attributes X_j are issued from the attribute authority AA_k (Notel_k could be equal to 0). The signer selects a l_k-value attribute subset$\omega'_k \subseteq X_j \cap \omega_{k,u}$, and takes following steps:

First, choose $r'_{k1}, r'_{k2}, \ldots, r'_{k,l+d-l_k} \in \mathbb{Z}_q$, and select a $d - l_k$ default attribute subset $\Omega'_k \subseteq \Omega_k$. Define $S_k = \omega'_k \cup \Omega'_k$.

Randomly choose $s \in \mathbb{Z}_q$. Compute

$$\sigma_0 = d_{CA} \prod\nolimits_{1 \leqslant k \leqslant K} \left(\prod\nolimits_{i \in S_k} d_{ki0}^{\Delta_{i,S_k}(0)} \prod\nolimits_{i \in X_j \cup \Omega'_k} H_1(i)^{r'_{ki}} \right) H_2(M)^s \tag{30}$$

$$\{\sigma_{ki} = d_{ki1}^{\Delta_{i,S_k}(0)} g^{r'_{ki}}\}_{i \in S_k} \tag{31}$$

$$\{\sigma_{ki} = g^{r'_{ki}}\}_{i \in X_j/\omega'_k} \tag{32}$$

$$\sigma'_0 = g^s \tag{33}$$

Finally, the signature is $(\sigma_0, \{\sigma_{ki}\}_{i \in X_j \cup \Omega'_k}, \sigma'_0)$.

DNF.Verify: After receiving the signature $(\sigma_0, \{\sigma_{ki}\}_{i \in X_j \cup \Omega' k}, \sigma' 0)$ with policy A, check if

$$\frac{e(g, \sigma_0)}{\prod_{1 \leqslant k \leqslant K} \prod_{i \in X_j \cup \Omega'_k} e(H_1(i), \sigma_{ki}) e(H_2(M), \sigma'_0)} \stackrel{?}{=} Z \tag{34}$$

If the equation holds, the signature is valid and the algorithm outputs *accept*. Otherwise, the algorithm outputs *reject*.

6.2. Security analysis

We give the security analysis of correctness and unforgeability by Theorem 3 and Theorem 4, respectively.

Theorem 3: Our DNF_MABS scheme is correct.

Proof. The correctness of verification is justified by the following equations:

$$\begin{aligned}
&\frac{e(g, \sigma_0)}{\prod_{1 \leqslant k \leqslant K} \prod_{i \in X_j \cup \Omega'_k} e(H_1(i), \sigma_{ki}) e(H_2(M), \sigma'_0)} \\
&= \frac{e(g, d_{CA} \prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} d_{ki0}^{\Delta_{i,S_k}(0)} \prod_{i \in X_j \cup \Omega'_k} H_1(i)^{r'_{ki}}) H_2(M)^s)}{\prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} e(H_1(i), d_{ki1}^{\Delta_{i,S_k}(0)} g^{r'_{ki}}) \prod_{i \in X_j/\omega'_k} e(H_1(i), g^{r'_{ki}})) e(H_2(M), g^s)} \\
&= \frac{e(g, d_{CA} \prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} d_{ki0}^{\Delta_{i,S_k}(0)} \prod_{i \in X_j \cup \Omega'_k} H_1(i)^{r'_{ki}})) e(g, H_2(M)^s)}{\prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} e(H_1(i), d_{ki1}^{\Delta_{i,S_k}(0)} g^{r'_{ki}}) \prod_{i \in X_j/\omega'_k} e(H_1(i), g^{r'_{ki}})) e(H_2(M), g^s)} \\
&= \frac{e(g, d_{CA}) e(g, \prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} (g_2^{p_k(i)} H_1(i)^{r_{ki}})^{\Delta_{i,S_k}(0)} \prod_{i \in X_j \cup \Omega'_k} H_1(i)^{r'_{ki}}))}{\prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} e(H_1(i), (g^{r_{ki}})^{\Delta_{i,S_k}(0)} g^{r'_{ki}}) \prod_{i \in X_j/\omega'_k} e(H_1(i), g^{r'_{ki}}))} \\
&= \frac{e(g, d_{CA}) \prod_{1 \leqslant k \leqslant K} e(g, (\prod_{i \in S_k} (g_2^{p_k(i)} H_1(i)^{r_{ki}})^{\Delta_{i,S_k}(0)} H_1(i)^{r'_{ki}} \prod_{i \in X_j/\omega'_k} H_1(i)^{r'_{ki}}))}{\prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} e(H_1(i), (g^{r_{ki}})^{\Delta_{i,S_k}(0)} g^{r'_{ki}}) \prod_{i \in X_j/\omega'_k} e(H_1(i), g^{r'_{ki}}))} \\
&= \frac{e(g, d_{CA}) \prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} e(g, (g_2^{p_k(i)} H_1(i)^{r_{ki}})^{\Delta_{i,S_k}(0)} H_1(i)^{r'_{ki}}) \prod_{i \in X_j/\omega'_k} e(g, H_1(i)^{r'_{ki}}))}{\prod_{1 \leqslant k \leqslant K} (\prod_{i \in S_k} e(H_1(i), (g^{r_{ki}})^{\Delta_{i,S_k}(0)} g^{r'_{ki}}) \prod_{i \in X_j/\omega'_k} e(H_1(i), g^{r'_{ki}}))} \\
&= e(g, d_{CA}) \prod_{1 \leqslant k \leqslant K} \prod_{i \in S_k} \frac{e(g, (g_2^{p_k(i)} H_1(i)^{r_{ki}})^{\Delta_{i,S_k}(0)} H_1(i)^{r'_{ki}})}{e(H_1(i), (g^{r_{ki}})^{\Delta_{i,S_k}(0)} g^{r'_{ki}})}
\end{aligned} \tag{35}$$

$$
\begin{aligned}
&= e(g, d_{CA}) \prod_{1 \leqslant k \leqslant K} \prod_{i \in S_k} \frac{e(g, H_1(i)^{r'_{ki} + r_{ki}\Delta_{i,S_k}(0)}) e(g, g_2^{p_k(i)\Delta_{i,S_k(0)}})}{e(H_1(i), g^{r_{ki}\Delta_{i,S_k}(0) + r'_{ki}})} \\
&= e(g, d_{CA}) \prod_{1 \leqslant k \leqslant K} \prod_{i \in S_k} e(g, g_2^{p_k(i)}\Delta_{i,S_k}(0)) \\
&= e(g, g_2^{\alpha - \sum_{k=1}^{K} y_{k,u}}) \prod_{1 \leqslant k \leqslant K} \prod_{i \in S_k} e(g, g_2)^{p_k(i)\Delta_{i,S_k}(0)} \\
&= e(g, g_2^{\alpha - \sum_{k=1}^{K} y_{k,u}}) \prod_{1 \leqslant k \leqslant K} e(g, g_2)^{\sum_{i \in S_k} p_k(i)\Delta_{i,S_k}(0)} \\
&= e(g, g_2)^{\alpha - \sum_{k=1}^{K} y_{k,u} + \sum_{k=1}^{K} p_k(0)} \\
&= e(g, g_2)^{\alpha} = Z
\end{aligned}
$$

Theorem 4: Suppose that the (t', ε')-CDH assumption holds in 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 DNF_MABS scheme is (t, q_{H_1}, q_{H_2}, q_K, q_S, ε)-EUF-sP-CMA, where $t' < t + (q_{H_1} + q_{H_2} + 3q_K + 3q_S d)t_{\exp}$, t_{exp} is the maximum time for an exponentiation in G_1, and $\varepsilon' \approx \varepsilon \Big/ (q_{H_2}(\begin{matrix} d-l \\ d-1 \end{matrix}))$.

Proof. As we adopt the signing technique of [18], the proof is similar to that in [18]. And the difference from [18] is just like that between TR_MABS and ABS in [26]. So we omit the procedure of proof here, please see it in [18].

7. Comparison

As pointed out in Section 4.1, here we don't repeat the main differences between TR_MABS and DNF_MABS. Now we compare the cost of adding or removing an AA in TR_MABS and DNF_MABS, in Tables 1 and 2, respectively. In Table 1, we assume adding a new attribute authority AA_{K+1}. In Table 2, we assume removing the attribute authority AA_k. Generally speaking, $1 \leqslant k, m \leqslant K$, and $m \neq k$. Moreover, we compare the cost in terms of the updated components by each authority in the systems.

Table 1
Comparison of adding an AA

Schemes	Updated components		
	CA	AA_k	AA_{K+1}
TR_MABS	$d_{CA}, \{T_k\}, T_{K+1}$	d_{ki0}	d_{K+Ii0}, d_{k+Ii1}
DNF_MABS	d_{CA}	–	d_{K+Ii0}, d_{K+Ii1}

Table 2
Comparison of deleting an AA

Schemes	Updated components	
	CA	AA_m
TR_MABS	$d_{CA}, \{T_m\}$	d_{mi0}
DNF_MABS	d_{CA}	–

From Tables 1 and 2, we can draw following viewpoints:

(1) In TR_MABS system, when adding or deleting an AA, a user's attribute key bases T_k for all AAs will change, and all AAs in the system should issue new attribute keys to the user, increasing computation and communication cost.

(2) In DNF_MABS system, when adding a new attribute authority AA_{K+1}, only AA_{K+1} issues attribute keys to its users, other AAs needn't update their users' attribute keys, which dramatically reduce the cost. When delete an AA, the remained AAs would do nothing.
(3) The modification of AAs will bring tasks to CA, as it should modify the secret key for each user, which is the same in both schemes.

8. Conclusion

In this paper, we have proposed two multi-authority attribute-based signature schemes, namely TR_MABS and DNF_MABS, supporting any policy consisting of AND, OR, and threshold gate over attributes. And we have proved the correctness and unforgeability of them on computational Diffie-Hellman assumption. These systems allow adding or removing attribute authorities flexibly. The main differences between them are the form of policy and the cost of adding or removing an AA.

The policy supported in TR_MABS scheme is an attribute tree, which is flexible to express AND, OR, and threshold conditions. But the cost of adding or deleting an AA in TR_MABS is larger than DNF_MABS, in which the policy is represented by a DNF, bringing in the benefit of negative attributes, with the pity of requiring a signer to analyse the DNF before generating a signature.

Although DNF_MABS scheme has shorter signature size than TR_MABS, we still need to develop good techniques to compact the signature size, in order to reduce the communication and verification cost in future.

Acknowledgments

The work described in this paper is partially supported by the grants of the National Basic Research Program of China (973 project) under Grant No. 2009CB320503, 2012CB315906; the project of National Science Foundation of China under grant No. 61070199, 61003301, 61103189, 61103194, 61103182; and supported by Program for Changjiang Scholars and Innovative Research Team in University (No. IRT1012), the Innovative Research Team in University of Hunan Province ("Network Technology", NUDT), and the Innovative Research Team of Hunan Provincial natural science Foundation (11JJ7003).

References

[1] A. Sahai and B. Waters, Fuzzy Identity-Based Encryption, in: *Advances in Cryptology-EUROCRYPT'05*, R. Cramer, ed., 2005, pp. 457–473.
[2] V. Goyal, O. Pandey, A. Sahai and B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, 2006, pp. 89–98.
[3] J. Bethencourt, A. Sahai and B. Waters, Ciphertext-Policy Attribute-Based Encryption, in: *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, Washington, DC, 2007, pp. 321–334.
[4] D. Khader, Authenticating with Attributes, http://eprint.iacr.org/2008/031.pdf, 2008.
[5] G. Shanqing and Z. Yingpei, Attribute-based Signature Scheme, in: *Proceedings of the 2008 International Conference on Information Security and Assurance* (*ISA 2008*), Washington, DC, USA, 2008, pp. 509–511.
[6] H. Maji, M. Prabhakaran and M. Rosulek, Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance, http://eprint.iacr.org/2008/328, 2008, pp. 1–23.
[7] J. Li and K. Kim, Attribute-Based Ring Signatures, http://eprint.iacr.org/2008/394, 2008.

[8] D. Chaum and E. van Heyst, Group signatures, in: *EUROCRYPT 1991*, D.W. Davies, ed., 1991, pp. 257–265.
[9] R.L. Rivest, A. Shamir and Y. Tauman, How to Leak a Secret, in: *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology-ASIACRYPT '01*, C. Boyd, ed., London, UK, 2001, pp. 552–565.
[10] X. Boyen, Mesh signatures. in: *advances in Cryptology-EUROCRYPT'07*, 2007, pp. 210–227.
[11] K. Emura, A. Miyaji and K. Omote, A Dynamic Attribute-Based Group Signature Scheme and its Application in an Anonymous Survey for the Collection of Attribute Statistics, in: *2009 International Conference on Availability, Reliability and Security*(*ARES'09*), 2009, pp. 487–492.
[12] D. Khader, Attribute Based Group Signature with Revocation, http://eprint.iacr.org/2007/241.pdf, 2007.
[13] D. Khader, Attribute Based Group Signatures, http://eprint.iacr.org/2007/159.pdf, 2007.
[14] Y. Qian and Y. Zhao, Strongly Unforgeable Attribute-Based Group Signature in the Standard Model, in: *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems* (*ICIS*), 2010, pp. 843–852.
[15] W. Wenqiang and C. Shaozhen, An Efficient Attribute-Based Ring Signature Scheme, in: *2009 International Forum on Computer Science-Technology and Applications*(*IFCSTA'09*), 2009, pp. 147–150.
[16] W. Wenqiang and C. Shaozhen, Attribute-based ring signature scheme with constant-size signature, *IET Information Security* **4** (2010), 104–110.
[17] S. Kumar, S. Agrawal, S. Balaraman and C. Rangan, Attribute Based Signatures for Bounded Multi-level Threshold Circuits, in: *Public Key Infrastructures, Services and Applications* (*EuroPKI 2010*), J. Camenisch and C. Lambrinoudakis, eds, Berlin Heidelberg, 2011, pp. 141–154.
[18] J. Li, M.H. Au, W. Susilo, D. Xie and K. Ren, Attribute-based signature and its applications, in: *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security-ASIACCS'10*, Beijing, China, 2010, pp. 60–69.
[19] H.K. Maji, M. Prabhakaran and M. Rosulek, Attribute-Based Signatures, http://eprint.iacr.org/2010/595, 2010, pp. 1–35.
[20] H. Maji, M. Prabhakaran and M. Rosulek, Attribute-Based Signatures, in: *Topics in Cryptology – CT-RSA 2011, Lecture Notes in Computer Science*, A. Kiayias, eds, Berlin Heidelberg, 2011, pp. 376–392.
[21] T. Okamoto and K. Takashima, Efficient Attribute-Based Signatures for Non-monotone Predicates in the Standard Model, in: *Public Key Cryptography – PKC 2011*, D. Catalano, N. Fazio, R. Gennaro and A. Nicolosi, eds, Berlin Heidelberg, 2011, pp. 35–52.
[22] J. Li and K. Kim, Hidden attribute-based signatures without anonymity revocation, *Information Sciences: an International Journal* **180** (2010), 1681–1689.
[23] A. Escala, J. Herranz and P. Morillo, Revocable attribute-based signatures with adaptive security in the standard model, in: *Proceedings of the 4th international conference on Progress in cryptology in Africa* (*AFRICACRYPT'11*), A. Nitaj and D. Pointcheval, eds, Berlin, Heidelberg, 2011, pp. 224–241.
[24] S. Shahandashti and R. Safavi-Naini, Threshold Attribute-Based Signatures and Their Application to Anonymous Credential Systems, in: *Progress in Cryptology – AFRICACRYPT 2009*, B. Preneel, ed., 2009, pp. 198–216.
[25] D. Cao, X. Wang, T. Wang and J. Su, An Expressive Attribute-based Signature Scheme without Random Oracles, in: *2011 International Conference on Computer Application and System Modeling* (*ICCASM 2011*), Xiamen, Fujian, China, to appear, 2011.
[26] D. Cao, B. Zhao, X. Wang, J. Su and Y. Chen, Authenticating with Attributes in Online Social Networks, in: *2th International Symposium on Frontiers in Ubiquitous Computing, Networking and Applications* (*NeoFUSION-2011*) *conjunction with 14th International Conference on Network-Based Information Systems* (*NBiS-2011*), Tirana, Albania, to appear, 2011.
[27] Y. Ciou, F. Leu, Y. Huang and K. Yim, A handover security mechanism employing the Diffie-Hellman key exchange approach for the IEEE802.16e wireless networks, *Mobile Information Systems* **7** (2011), 241–269.
[28] H. Kim and J. Lee, Diffie-Hellman key based authentication in proxy mobile IPv6, *Mobile Information Systems* **6** (2010), 107–121.
[29] F. Tang, I. You, M. Guo, S. Guo and L. Zheng, Balanced bipartite graph based register allocation for network processors in mobile and wireless networks, *Mobile Information Systems* **6** (2010), 65–83.
[30] I. You, J. Lee, Y. Hori and K. Sakurai, Enhancing MISP with Fast Mobile IPv6 Security, *Mobile Information Systems* **7** (2011), 271–283.
[31] I. You, Y. Hori and K. Sakurai, Enhancing SVO Logic for Mobile IPv6 Security Protocols, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* **2** (2011), 26–52.
[32] R. Ostrovsky, A. Sahai and B. Waters, Attribute-based encryption with non-monotonic access structures, in: *ACM Conference on Computer and Communications Security*, Alexandria, Virginia, USA, 2007, pp. 195–203.
[33] J. Li, Q. Wang, C. Wang and K. Ren, Enhancing Attribute-Based Encryption with Attribute Hierarchy, in: *Mobile Networks and Applications*, 2010, pp. 1–9.
[34] J. Li, K. Ren, B. Zhu and Z. Wan, Privacy-Aware Attribute-Based Encryption with User Accountability, in: *Information Security Conference 2009*(*ISC'09*), Italy, 2009, pp. 347–362.

[35] M. Chase, Multi-authority Attribute Based Encryption, in: *Theory of Cryptography Conference*(*TCC*), 2007, pp. 515–534.
[36] V. Božovi c, D. Socek, R. Steinwandt and V.O.I. Vill anyi, Multi-authority attribute based encryption with honest-but-curious central authority. in: *Cryptology ePrint Archive: Report 2009/083*, http://eprint.iacr.org/2009/083.pdf, 2009.
[37] H. Lin, Z. Cao, X. Liang and J. Shao, Secure Threshold Multi Authority Attribute Based Encryption without a Central Authority, in: *Cryptology in India*, Kharagpur, India, 2008, pp. 426–436.
[38] M. Chase and S.S.M. Chow, Improving privacy and security in multi-authority attribute-based encryption, in: *ACM Conference on Computer and Communications Security*, Chicago, Illinois, USA, 2009, pp. 121–130.
[39] S. Müller, S. Katzenbeisser and C. Eckert, On Multi-Authority Ciphertext-Policy Attribute-Based Encryption, *Bulletin of the Korean Mathematical Society* **46** (2009), 803–819.
[40] S.M. uller, S. Katzenbeisser and C. Eckert, Distributed Attribute-Based Encryption, in: *Information Security and Cryptology-ICISC 2008*, P.J. Lee and J.H. Cheon, eds, 2009, pp. 20–36.
[41] A. Lewko and B. Waters, Decentralizing Attribute-Based Encryption, http://eprint.iacr.org/2010/351, 2010.
[42] B. Waters, Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions, in: *Advances in Cryptology-CRYTO'09*, S. Halevi, ed., 2009, pp. 619–636.
[43] C. Wang, W. Chen and Y. Liu, A Fuzzy Identity Based Signature Scheme, in: *International Conference on E-Business and Information System Security* (*EBISS'09*), 2009, pp. 1–5.
[44] P. Yang, Z. Cao and X. Dong, Fuzzy Identity Based Signature, http://eprint.iacr.org/2008/002.pdf, 2008.
[45] S. Tan, S. Heng and B. Goi, On the Security of an Attribute-Based Signature Scheme, in: *Communications in Computer and Information Science*, D. Ślęzak, T. Kim, J. Ma, W. Fang, F.E. Sandnes, B. Kang and B. Gu, eds, 2009, pp. 161–168.
[46] M. Karchmer and A. Wigderson, On Span Programs, in: *the 8th Annual Structure in Complexity Theory*, 1993, pp. 102–111.

Dan Cao received the B.S degree in Comupter Science from National University of Defense Technology, Changsha, China, in 2006. She is currently a PhD student in School of Computer at the National University of Defense Technology, Changsha, China. Her research interests include information security and access control.

Baokang Zhao is an Assistant Professor in the School of Computer Science, National University of Defense Technology. He recevied his Ph.D. degree in Computer Science from National University of Defense Technology, in 2009. He served as a program committee member for several international conferences and a reviewer for several international journals. He serves on the editor board of Journal of Internet Sevices and Information Security(JISIS). His current research interests include security and privacy in wireless networks, algorithms and protocols in computer networks, design and optimization in embedded systems. He is a member of the ACM, IEEE and CCF.

Xiaofeng Wang is an assistant professor in School of Computer, National University of Defense Technology (NUDT), China. He completed his Ph.D at NUDT in 2009. His current research interests are in trust and security of networking systems, distributed and intelligent date processing. He has published several papers in renowned journals and conferences like IEEE/ACM CCGrid, AINA, IEEE Transactions on Services Computing and Elsevier FGCS etc.

Jinshu Su received the B. Sc degree in Mathematics from Nankai University, Tianjin, China, in 1983, the M.S. degree in Computer Science, National University of Defense Technology, changsha, China, in 1989, and the PhD degree in Computer Science, National University of Defense Technology, changsha, China, in 1999. He is a full professor at the School of Computer Science, National University of Defense Technology, and serves as head of the Institute of network and information security, NUDT. He is the academic leader of the State Innovative Research Team in University ("Network Technology" Innovative Team) awarded by the Ministry of Education, CHINA. He has lead several national key projects of CHINA, including national 973 projects, 863 projects and NSFC Key projects. His research interests include high performance routers, internet routing, high performance computing, wireless networks and information security. He is a member of the ACM and IEEE.