Attribute Set Based Signature Secure in the Standard Model

Baohong Li[1], Yinliang Zhao[1] and Hongping Zhao[2]
[1] School of Electronics and Information Engineering, Xi'an Jiaotong University
Xi'an, 710049 - China
[e-mail: bhli@mail.xjtu.edu.cn]
[2] 518th Hosp. of PLA
Xi'an, 710049 - China
*Corresponding author: Baohong Li

Received August 19, 2014; revised January 29, 2014; accepted March 2, 2015; published April 30, 2015

Abstract

We introduce attribute set based signature (ASBS), a new cryptographic primitive which organizes user attributes into a recursive set based structure such that dynamic constraints can be imposed on how those attributes may be combined to satisfy a signing policy. Compared with attribute based signature (ABS), ASBS is more flexible and efficient in managing user attributes and specifying signing policies. We present a practical construction of ASBS and prove its security in the standard model under three subgroup decision related assumptions. Its efficiency is comparable to that of the most efficient ABS scheme.

Keywords: Attribute set based signatures, signing policy, subgroup decision assumptions, dual system encryption

This work is partially supported by the Natural Science Foundation of China under Grant No. 61173040.

http://dx.doi.org/10.3837/tiis.2015.04.014 **ISSN : 1976-7277**

1. Introduction

1.1 Motivation

Flexible and privacy-preserving authentication scheme is a fundamental concern in modern cryptography, and attribute based signature (ABS) [1] is proposed as a potential solution. However, as observed by Bobba, Khurana and Prabhakaran [2], attribute based systems are far from being flexible in managing user attributes and specifying signing policies, mainly due to the fact that they organizes user attributes into a single set, and a user can use all possible combinations of attributes issued in her attribute keys to satisfy a policy. For example, consider a graduate student who has enrolled in course 525 and she is also the TA for course 101. Here, both '*CourseID: 101*' and '*Role: TA*' are valid attributes but their combination is not allowed. To prevent such undesirable combination, these attributes have to be appended into two strings of '*Role:TA_CourseID:101*' and '*Role:Grad_CourseID:525*'. However, this approach leads to another undesirable consequence where a singleton attribute, say, '*Role:TA*' , can't be used alone to satisfy a policy, since attribute based systems only checks the equality of strings and can't extract a single attribute from a string.

To addresses these limitations, Bobba, Khurana and Prabhakaran present ciphertext policy attribute set based encryption (CP-ASBE) [2] as an enhancement to attribute-based encryption (ABE). By organizing user attributes into a recursive set based structure, CP-ASBE allows users to impose dynamic constraints on how those attributes may be combined to satisfy a policy, that is, policies can be specified to restrict decrypting users to use attributes from a single set or allow them to combine attributes from multiple sets. Moreover, CP-ASBE can efficiently support multiple value assignments and attribute revocation. Due to these powerful features, CP-ASBE provides an attractive primitive to realize a flexible and fine-grained access control in distributed systems.

Since ABS suffers from almost same problems as ABE, a natural question is whether there exists a practical signature analog of CP-ASBE, to realize a flexible and privacy-preserving authentication mechanism.

1.2 Contribution

In this paper, we introduce attribute set based signature (ASBS) as an enhancement to ABS. Since ASBS also organizes user attributes into a recursive set based structure, it shares same flexibility and other desirable features with CP-ASBE; that is, ASBS allows dynamic constraints to be imposed on how those attributes may be combined to satisfy a signing policy, and it also can efficiently support compound attributes, multiple value assignments and attribute revocation. We also construct a practical ASBS and prove its security in the standard model under three subgroup decision related assumptions. The efficiency of our construction is comparable to that of the most efficient ABS scheme.

Although our ASBS may be thought as the signature analog of CP-ASBE, it can't be straightforwardly converted from Bobba, Khurana and Prabhakaran's construction, since it can only be proved secure under both generic group model and random oracle model. In addition, perfect privacy is a specific requirement in our ASBS.

Roughly speaking, a signature in our construction is generated by re-randomizing a secret key, and it is verified by first distributing shares of a secret exponent across some verification components and then checking whether the secret can be recovered by the signature. To do so,

we need to extend the re-randomization technique from the ABS scheme due to Okamoto and Takashima [3], in order to accommodate our more complicated key structure. We also adapt the dual system encryption of Lewko and Waters to prove the security of our ASBS. Their technique is originally developed to prove the full security of IBE and ABE [4, 5].

The primary challenge in our construction is how to manage the recursive set based attribute structure. Specifically, our construction should allow a signing policy to specify whether and how a signer can combine attributes from multiple sets within her attribute structure. To tackle the issue, we introduce the *translating keys* and *translating verification components*. Given a signer's attribute structure, we randomize attribute keys in each attribute set with a unique exponent. Then a translating key is issued for each attribute set, used to translate the exponent into a constant such that attribute keys in this attribute set can be combined with other attribute keys to satisfy a signing policy. However, this combination can only be performed with the help of the corresponding translating verification component, which is generated by the verifier when the signing policy does specify to do so.

1.3 Related Work

Because ABS is the most related concept to ours ASBS, we briefly review it as follows.

Maji, Prabhakaran and Rosulek [1] presented the first ABS in 2008. This primitive allows a signer to convince a verifer that she holds a set of attributes satisfying the signing policy and has endorsed the message. Their scheme supports very expressive signing policies and is almost optimally efficient, but its security is only proven in generic group model, an artificial model which is not solid enough to guarantee the security.

Motivated by the limitation of [1], several ABS schemes [6-8, 15-17] were presented to be secure *in the standard model*, but they can only achieve *selective security*, a weaker notion of unforgeability than *adaptive security*. In this security model, an adversary is required to announce the target signing policy she intends to attack before seeing the public parameters. In addition, signing policies in most of these ABS schemes are limited to threshold predicate, which is less expressive than necessary.

Maji, Prabhakaran and Rosulek [9] presented the first ABS which is adaptively secure in the standard model, but it is much less efficient (in terms of signature size) since it employed the Groth-Sahai NIZK system as building blocks, in order to prove relations between the bits of elements in the group. By using the technology of dual pairing vector spaces, Okamoto and Takashima [3] also presented an ABS scheme which is adaptively secure in the standard model, and their scheme is more expressive, i.e., it allows non-monotone predicates to express signing policies. They further improved it to accommodate the setting of multi-authority [10]. However, the efficiencies of their two schemes are several times worse than that of Maji, Prabhakaran and Rosulek's ABS scheme [1].

As we mentioned earlier, all of these ABS schemes are not flexible enough in managing user attributes and specifying signing policies. So our goal is to present a construction of ASBS that satisfies at the same time the following properties: (1) it is proved adaptively secure in the standard model, (2) it admits general signing policies, and (3) it is efficient in terms of signature size.

The remainder of the paper is organized as follows. In Section 2, we review some definitions and complexity assumptions. Section 3 defines ASBS and formulizes its security. In Section 4, we present our construction of ASBS and prove its security. Finally, Section 5 concludes the whole paper.

2. Preliminaries

2.1 Bilinear Groups of Composite Order and Complexity Assumptions

Definition 1 (**Bilinear group of composite order**): A (symmetric) bilinear group of composite order is a tuple (N, G, G_T, $\hat{e}$) where $N = p_1 p_2 p_3$ for distinct primes p_1, p_2, p_3, G, G_T are cyclic groups of composite order N, and $\hat{e}: G \times G \to G_T$ is an efficiently computable function with the following properties:

(1) $\forall g, h \in G$, $\forall a, b \in \mathbf{Z}_N$, $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$

(2) The element $\hat{e}(g, g)$ is a generator of G_T.

The security of our construction relies on three subgroup decision related assumptions on bilinear groups of composite order. They have been used by Lewko and Waters to prove the security of their IBE [4] and ABE [5]. To be concise, in the sequel we use G_{p_i} or $G_{p_i p_j}$ to denote the subgroup of order p_i or $p_i p_j$ in G.

Assumption 1: Given a bilinear group of composite order (N, G, G_T, $\hat{e}$) and random $g \in G_{p_1}$, $X_3 \in G_{p_3}$, it is hard to distinguish a random element $T_1 \in G_{p_1 p_2}$ from a random element $T_2 \in G_{p_1}$

Assumption 2: Given a bilinear group of composite order (N, G, G_T, $\hat{e}$) and random $g, X_1 \in G_{p_1}$, $X_2, Y_2 \in G_{p_2}$, $X_3, Y_3 \in G_{p_3}$, it is hard to distinguish a random element $T_1 \in G$ from a random element $T_2 \in G_{p_1 p_3}$.

Assumption 3: Given a bilinear group of composite order (N, G, G_T, $\hat{e}$) and random $\alpha, s \in \mathbf{Z}_N$, $g \in G_{p_1}$, $X_2, Y_2, Z_2 \in G_{p_2}$, $X_3 \in G_{p_3}$, it is hard to distinguish the element $T_1 = \hat{e}(g, g)^{\alpha s}$ from a random element $T_2 \in G_T$.

2.2 Linear Secret Sharing Scheme and Attribute Structure

Definition 2 (**Monotone access structure** [11]): Let $\mathsf{P} = \{ P_1, P_2, \ldots, P_l \}$ be a set of parties (attributes in our setting). A collection $\Lambda \subseteq 2^{\mathsf{P}}$ is *monotone* if $\forall B, C$: if $B \in \Lambda$ and $B \subseteq C$ then $C \in \Lambda$. A monotone access structure is a collection Λ of non-empty subsets of P , and elements in Λ are called as authorized sets.

Definition 3 (**Linear secret sharing scheme (LSSS)** [11]): A secret sharing scheme over a set of parties $\mathsf{P} = \{ P_1, P_2, \ldots, P_l \}$ is called *linear* (over $\mathbf{Z}_N$) if,

(1) The shares of each party form a vector over $\mathbf{Z}_N$, and

(2) There exists a matrix A with l rows and n columns, and a function ρ which maps A_x , the x-th row of A, to the party $\rho(x)$ for $x = 1, \ldots, l$. Given a random vector $v = (s, v_2, \ldots, v_n) \in \mathbf{Z}_N^n$, $A \cdot v$ is the vector of l shares of the secret s, and $A_x \cdot v$ belongs to party $\rho(x)$.

Using standard techniques [12], any monotonic boolean formulas can be converted into an LSSS matrix A to represent its access structure Λ . For each authorized set $B \in \Lambda$, there exists a reconstruction vector $\omega = (\omega_1, \omega_2, \ldots, \omega_l) \in \mathbf{Z}_N^l$, where $\omega_x = 0$ if $\rho(x) \notin B$, such that $\omega \cdot A = (1, 0, \ldots, 0)$ and $\sum_{\rho(x) \in B} \omega_x A_x \cdot v = s$. On the contrary, no reconstruction vector exists if $B \notin \Lambda$.

Definition 4 (**Attribute structure**): An attribute structure in ASBS is a recursive set where each element of the set is either a set itself (i.e. an attribute structure) or an attribute.

The *depth* of an attribute structure is defined as the number of its recursive levels. Like the CP-ASBE [2], our construction mainly focuses on the attribute structure with depth 2, where elements at depth 1 can either be attributes or sets but elements at depth 2 can only be

attributes, belonging to some sets at depth 1. For each set at depth 1, an *index*, numbered from 1, is assigned to identify it. For the sake of consistency, attributes at depth 1 can be thought to form a set with index of 0. We also abuse the notion of index to an individual attribute, which is defined as the combination of the index of the set it belongs to and its sequence number in the set.

In our construction, a signing policy is expressed by a monotonic boolean formula and a signer's attributes are described by an attribute structure. If a signer's attribute structure satisfies the signing policy, the signing policy is realized by the corresponding LSSS matrix and the reconstruction vector is used to generate a signature.

3. Definitions

3.1 Attribute Set based Signature

Definition 5 (**Attribute Set based Signature**): An attribute set based signature ASBS = {Setup, KGen, Sign, Verify} is a tuple of polynomial time algorithms.

–Setup$(1^\lambda) \rightarrow (PK, MSK)$. On input security parameter 1^λ, the setup algorithm outputs the public parameters *PK* and a master secret key *MSK* for the attribute authority.

–**KGen**$(PK, MSK, AS) \rightarrow SK$. Given (*PK*, *MSK*) and an user's attribute structure *AS*, the key generation algorithm outputs a secret key *SK* for the user.

–**Sign**$(PK, SK, M, (A, \rho)) \rightarrow \sigma$. The signing algorithm takes in *PK*, a secret key *SK*, a message *M* and a LSSS access structure (A, ρ) over the universe of attributes. If the attribute structure in *SK* satisfies the LSSS access structure, it outputs a signature σ.

–Verify$(PK, M, (A, \rho), \sigma) \rightarrow 1/0$. Given *PK*, a message *M*, a LSSS access structure (A, ρ) and a signature σ, the verification algorithm outputs 1 if σ is a valid signature on *M* or 0 otherwise.

3.2 Security model for ASBS

Perfect privacy and unforgeability are two security properties that ASBS should satisfy. We give their formal definitions as follows.

Definition 6 (**Perfect Privacy**): An ASBS scheme is perfect privacy, if, for all $(PK, MSK) \leftarrow \mathsf{Setup}(1^\lambda)$, all messages *M*, all attribute structures $AS^{(1)}$ and $AS^{(2)}$, all secret keys $SK^{(1)} \leftarrow \mathsf{KGen}(PK, MSK, AS^{(1)})$ and $SK^{(2)} \leftarrow \mathsf{KGen}(PK, MSK, AS^{(2)})$, all access structures (A, ρ) such that both $AS^{(1)}$ and $AS^{(2)}$ satisfy (A, ρ), the distributions of $\sigma^{(1)} = \mathsf{Sign}(PK, SK^{(1)}, M, (A, \rho))$ and $\sigma^{(2)} = \mathsf{Sign}(PK, SK^{(2)}, M, (A, \rho))$ are equal.

Definition 7 (**Existential Unforgeability**): For an adversary $\mathcal{A}$, we define $\mathrm{GameAdv}_{\mathcal{A}}$ to be its success probability in the following game. An ASBS scheme is existentially unforgeable if $\mathrm{GameAdv}_{\mathcal{A}}$ is negligible for any polynomial-time adversary $\mathcal{A}$.

–Setup Run $(PK, MSK) \leftarrow \mathsf{Setup}(1^\lambda)$ and give *PK* to $\mathcal{A}$.

–Queries $\mathcal{A}$ can adaptively makes a polynomial number of *q* queries of the following two types:

(1) Key Queries $\mathcal{A}$ can request a secret key for any attribute structure *AS*.

(2) Signing Queries $\mathcal{A}$ can request a signature for any message, attribute structure *AS* and access structure (A, ρ), with the restriction that the attribute structure *AS* should satisfy the access structure (A, ρ).

–Forgery $\mathcal{A}$ outputs a signature σ^* on a message M^* and an access structure (A^*, ρ^*). $\mathcal{A}$ succeeds in the game if

(1) Verify$(PK, M^*, (A^*, \rho^*), \sigma^*) = 1$.

(2) $\mathcal{A}$ has never queried a secret key for an attribute structure AS^* satisfying (A^*, ρ^*).

(3) $\mathcal{A}$ has never queried a signature on message M^* and (A^*, ρ^*).

4. OUR ASBS

4.1 Construction

We construct our ASBS in a bilinear group of composite order $(N, G, G_T, \hat{e})$, where keys and signatures occur in the subgroups $G_{p_1 p_3}$, while the subgroup G_{p_2} is used as the semi-functional space to prove the security of our construction.

–Setup(1^λ) The setup algorithm performs the following steps:

(1) Generate a bilinear group $(N, G, G_T, \hat{e})$ of composite order $N = p_1 p_2 p_3$.

(2) Pick random value α, $a \in \mathbf{Z}_N$, $g \in G_{p_1}$, $X_3 \in G_{p_3}$, and choose $u_0, u_1, \ldots, u_\eta \in G_{p_1}$ as Water's hash function [13], where η is the length of a message. To be concise, we define $F(M) = u_0 \prod_{i=1}^{\eta} (u_i)^{\mu_i}$ for a message $M = (\mu_1, \mu_2, \ldots, \mu_\eta)$.

(3) Let the universe attribute structure be $AS = \{ AS_0, AS_1, \ldots, AS_m \}$. For each attribute set $AS_i = \{ at_{i,1}, \ldots, at_{i,n_i} \}$, $1 \le i \le m$, pick a unique random exponent $b_i \in \mathbf{Z}_N$. For the j-th attribute appearing in set AS_i, $0 \le i \le m$, $1 \le j \le n_i$, pick a unique random exponent $h_{ij} \in \mathbf{Z}_N$.

The algorithm outputs public parameters and the master secret key as:

$$PK = (N, G, G_T, \hat{e}, g, g^a, \hat{e}(g,g)^\alpha, u_0, u_1, \ldots, u_\eta, g^{b_i} \forall i, g^{a/b_i} \forall i, H_{ij} = g^{h_{ij}} \forall i \forall j)$$

$$MSK = (\alpha, b_i \forall i, X_3)$$

–KGen(PK, MSK, AS) The key generation algorithm chooses a unique random $t_i \in \mathbf{Z}_N$ for each set $AS_i \in AS$, $0 \le i \le m$. It also chooses random elements $R_0, R_i, R_i', R_{ij} \in G_{p3}$ and outputs the secret key as:

$$K = g^\alpha g^{a t_0} R_0,$$
$$K_i = g^{t_i} R_i, \quad K_{ij} = (H_{ij})^{t_i} R_{ij}, \quad 0 \le i \le m, \ 1 \le j \le n_i$$
$$L_i = g^{a(t_0 + t_i)/b_i} R_i', \quad 1 \le i \le m$$

–Sign$(PK, SK, M, (A, \rho))$ Let A be an $l \times n$ matrix and A_x be the x-th row of A. The function ρ maps each row number x to an attribute's index $\rho(x)$, while the function φ maps x to the index of the set the attribute $\rho(x)$ belongs to. The algorithm proceeds as follows:

(1) Choose random $t_0', t'_{\varphi(x)} \in \mathbf{Z}_N$ for $1 \le x \le l$, and blind the secret key SK as:

$$K' = K(g^a)^{t_0'} = g^\alpha g^{a(t_0 + t_0')} R_0',$$
$$K'_{\varphi(x)} = K_{\varphi(x)} g^{t'_{\varphi(x)}} = g^{t_{\varphi(x)} + t'_{\varphi(x)}} R_{\varphi(x)},$$
$$K'_{\rho(x)} = K_{\rho(x)} (H_{\rho(x)})^{t'_{\varphi(x)}} = (H_{\rho(x)})^{t_{\varphi(x)} + t'_{\varphi(x)}} R_{\rho(x)}$$
$$L'_{\varphi(x)} = L_{\varphi(x)} (g^{a/b_{\varphi(x)}})^{t_0' + t'_{\varphi(x)}} = g^{a(t_0 + t_0' + t_{\varphi(x)} + t'_{\varphi(x)})/b_{\varphi(x)}} R'_{\varphi(x)}$$

(2) Find a reconstruction vector $\omega = (\omega_1, \omega_2, \ldots, \omega_l) \in \mathbf{Z}_N^l$ and two random vectors $\beta_1 = (\beta_{11}, \beta_{12}, \ldots, \beta_{1l})$, $\beta_2 = (\beta_{21}, \beta_{22}, \ldots, \beta_{2l})$, such that $\beta_1 \cdot A = \beta_2 \cdot A = (0, 0, \ldots, 0)$. We discuss the

existence of β_1, β_2 in the proof of Theorem 1. Then compute:

$$\sigma_a = K' \prod_{x=1}^{l} ((K'_{\rho(x)})^{\omega_x} (H_{\rho(x)})^{\beta_{1x}}) F(M)^r, \ \sigma_b = g^r$$

$$\sigma_{1,x} = (K'_{\varphi(x)})^{\omega_x} g^{\beta_{1x}}, \ \sigma_{2,x} = (L'_{\varphi(x)})^{\omega_x} (g^{a/b_{\varphi(x)}})^{\beta_{2x}}, \ 1 \le x \le l$$

The algorithm outputs the signature as $\sigma = (\sigma_a, \sigma_b, (\sigma_{1,x}, \sigma_{2,x})_{x=1,\ldots,l})$.

–Verify$(PK, M, (A, \rho), \sigma)$ Given a signature σ on a message M, the algorithm first chooses a random vector $v = (s, v_2, \ldots, v_n) \in \mathbf{Z}_N^n$ and generates a group of verification components as follow:

$$C = g^s, \ C_x = g^{aA_x v} (H_{\rho(x)})^{-s}, \ E_x = g^{b_{\varphi(x)} A_x v}, \ 1 \le x \le l$$

Then the algorithm outputs 1 if the following verification equation holds

$$\hat{e}(\sigma_a, C) = \hat{e}(g,g)^{\alpha s} \, \hat{e}(F(M)^s, \sigma_b) \prod_{x=1}^{l} \hat{e}(E_x, \sigma_{2,x}) / \hat{e}(C_x, \sigma_{1,x}) \quad (1)$$

We name the component L_i in a secret key and E_x in the verification components as the *translating key* and the *translating verification component*, respectively. During verification, only *simultaneous* use of both a translating key and the corresponding translating verification component can translate the exponents t_i in K_i and K_{ij} into the constant t_0, and the verification succeeds only when all t_i are translated into t_0.

4.2 Efficiency

To show the efficiency and security of our ASBS, we compare it with existing ABS schemes which support general signing policies in **Table 1**, where l and n are the size of the underlying LSSS matrix, and λ is the security parameter (e.g., 128). The signature size and complexity are measured in terms of the number of group elements and the number of paring operations to verify a signature, respectively. In addition, since there are two constructions are presented in [9], the more efficient construction based on Waters signature is used to measure the signature size, and the underlying Groth-Sahai NIZK system is instantiated under DLIN Assumption.

Table 1. Comparison with ABS schemes supporting general signing policies

Schemes	Size	complexity	Model	Policy	Multi-authority
MPR08 [1]	$l + n + 2$	$n\,l + n + 3$	generic group	monotone	No
MPR11 [9]	$36l + 2n + 9\lambda + 12$	$36l$	standard	monotone	No
OT11a [3]	$7l + 11$	$7l + 15$	standard	non-monotone	No
OT11b [10]	$13l$	$13l$	standard	non-monotone	Yes
Our ASBS	$2l + 2$	$2l + 2$	standard	monotone	No

Although some ABS schemes, such as [6-8, 15-17], have short or even constant signature size, we need not compare with them in **Table 1**, since they only admit threshold signing policies, which can't provide enough flexibility than necessary. Although some of them, such as [17], can be extended to admit general signing policies, the signature size, as the authors of [17] denoted, will drastically increase.

Since the ABS scheme due to Maji, Prabhakaran and Rosulek [1] has the best efficiency among all ABS schemes, we consider it as a benchmark. Compared with this ABS scheme, our ASBS has almost same signature size but less complexity.

4.3 Perfect Privacy

Theorem 1. Our construction of ASBS can achieve perfect privacy.

Proof. Given a LSSS access structure (A, ρ), if rank$(A) < l$, there obviously exist polynomial numbers of vectors β such that $\beta \cdot A = (0, 0, \ldots, 0)$. If rank$(A) = l$, there exists only one $\beta = (0, 0, \ldots, 0)$ such that $\beta \cdot A = (0, 0, \ldots, 0)$, but the signing policy is limited to a (n, n)-threshold predicate, which means there is no attribute privacy at all. So we only need to consider the case of rank$(A) < l$.

To show that two signatures of $\sigma^{(1)} = \mathsf{Sign}(PK, SK^{(1)}, M, (A, \rho))$ and $\sigma^{(2)} = \mathsf{Sign}(PK, SK^{(2)}, M, (A, \rho))$ have equal distributions, it suffices to prove that, by choosing proper random t_i', r and β_1, β_2, same signatures can be generated from different keys $SK^{(1)}$ and $SK^{(2)}$. To do so, we choose $r^{(1)} = r^{(2)}$, $t_0^{(1)} + t_0'^{(1)} = t_0 = t_0^{(2)} + t_0'^{(2)}$ and $t_{\varphi(x)}^{(1)} + t_{\varphi(x)}'^{(1)} = t_{\varphi(x)} = t_{\varphi(x)}^{(2)} + t_{\varphi(x)}'^{(2)}$ for $1 \le x \le l$. Given $\beta_1^{(1)} = (\beta_{11}^{(1)}, \ldots, \beta_{1l}^{(1)})$ and $\beta_2^{(1)} = (\beta_{21}^{(1)}, \ldots, \beta_{2l}^{(1)})$, we also choose $\beta_1^{(2)}$ and $\beta_2^{(2)}$ by setting $\beta_{1x}^{(2)} = (\omega_x^{(1)} - \omega_x^{(2)}) t_{\varphi(x)} + \beta_{1x}^{(1)}$, $\beta_{2x}^{(2)} = (\omega_x^{(1)} - \omega_x^{(2)})(t_0 + t_{\varphi(x)}) + \beta_{2x}^{(1)}$. It is easy to check that $\beta_1^{(2)} \cdot A = \beta_2^{(2)} \cdot A = (0, 0, \ldots, 0)$ and $\sigma^{(1)} = \sigma^{(2)}$.

4.4 Existential Unforgeability

We adapt the dual system encryption technique [4, 5] to prove the unforgeability of our construction. Specifically, we define two types of semi-functional signatures and an additional semi-functional verification algorithm VerifySF. We also define a sequence of games where the first game is the real unforgeable game and in the last game the success probability of any polynomial-time adversary is negligible. Then, by standard hybrid arguments, we prove several lemmas to show that the adversary's success probabilities are indistinguishable among these games. The unforgeability of our construction follows from these lemmas.

–Semi-functional signatures of type 1 Given a normal signature $\sigma' = (\sigma_a', \sigma_b', (\sigma_{1,x}', \sigma_{2,x}')_{x=1,\ldots,l})$, a semi-functional signature of type 1 is generated by choosing a generator $g_2 \in G_{p_2}$, random exponents $d, e \in \mathbf{Z}_N$ and $f_x \in \mathbf{Z}_N$, and outputting $\sigma = (\sigma_a' g_2^d, \sigma_b', (\sigma_{1,x}' (g_2^e)^{\omega_x}, \sigma_{2,x}' (g_2^{f_x})^{\omega_x})_{x=1,\ldots,l})$.

–Semi-functional signatures of type 2 Given a normal signature $\sigma' = (\sigma_a', \sigma_b', (\sigma_{1,x}', \sigma_{2,x}')_{x=1,\ldots,l})$, a semi-functional signature of type 2 is generated as $\sigma = (\sigma_a' g_2^d, \sigma_b', (\sigma_{1,x}', \sigma_{2,x}')_{x=1,\ldots,l})$.

–VerifySF This algorithm is same as Verify except that it chooses a random exponent $c \in \mathbf{Z}_N$, two random vectors $u, w \in \mathbf{Z}_N^n$, random values $\gamma_x \in \mathbf{Z}_N$ for each x, and random values $z_{\rho(x)} \in \mathbf{Z}_N$ for each attribute $\rho(x)$, and generates semi-functional verification components as:

$$C = g^s g_2^c,\ C_x = g^{aA_x v} (H_{\rho(x)})^{-s} g_2^{A_x u - z_{\rho(x)}},\ E_x = g^{b_{\varphi(x)} A_x v} g_2^{\gamma_x A_x w},\ 1 \le x \le l$$

– Game$_{real}$: The real security game defined in Section 3.

– Game$_0$: The real security game with the exception that the algorithm VerifySF is used to verify a forged signature.

– Game$_{k,1}$, $1 \le k \le q$: Same as Game$_0$ except that the first $k-1$ signatures are semi-functional signatures of type 2 and the k-th key is a semi-functional signature of type 1.

– Game$_{k,2}$, $1 \le k \le q$: Same as Game$_{k,1}$ except that the k-th signature is also a semi-functional signature of type 2.

– Game$_{final}$: In this game, all signatures are semi-functional signatures of type 2, and the

forged signature is verified by VerifySF.

Lemma 1. Suppose there exists an adversary $\mathcal{A}$ such that $\mathrm{Game}_{real}\mathrm{Adv}_{\mathcal{A}} - \mathrm{Game}_0\mathrm{Adv}_{\mathcal{A}} = \varepsilon$, then we can build a simulator S that has advantage ε in breaking the Assumption 1.

Proof. The simulator S begins by taking in an instance (g, X_3, T) of Assumption 1, where $T = g^s g_2^c$ or g^s for unknown $s, c \in \mathbf{Z}_N$. It simulates Game_{real} or Game_0 as follows:

–Setup S generates *PK* and *MSK* by running the algorithm Setup.

–Queries Since S knows the *MSK*, it can run algorithms KGen and Sign to outputs any secret keys and signatures.

–Forgery Upon receiving a forged signature, S chooses a random vector $v' = (1, v_2', ..., v_n') \in \mathbf{Z}_N^n$, and generates verification components as follows:

$$C = T,\ C_x = T^{aA_x v'} T^{-s_{\rho(x)}},\ E_x = T^{b_{\phi(x)} A_x v'}$$

If $T = g^s$, we have $C = g^s$, $C_x = g^{aA_x sv'} H_{\rho(x)}^{-s}$, $E_x = g^{b_{\phi(x)} A_x sv'}$. They are properly distributed normal verification components with the implicit setting of $v = sv'$.

If $T = g^s g_2^c$, we have $C = g^s g_2^c$, $C_x = g^{aA_x sv'} H_{\rho(x)}^{-s} g_2^{A_x cav' - cs_{\rho(x)}}$, $E_x = g^{b_{\phi(x)} A_x sv'} g_2^{cb_{\phi(x)} A_x v'}$. They are well-formed semi-functional verification components if we further set $u = cav'$, $z_{\rho(x)} = cs_{\rho(x)}$, $cb_{\varphi(x)} = \gamma_x$ and $w = v'$. Although some values in G_{p1} parts of the verification components, such as a, v', $s_{\rho(x)}$, $b_{\varphi(x)}$, are reused in G_{p2} parts, it is still properly distributed, since these values modulo p_2 are uncorrelated from their values modulo p_1 due to the Chinese Remainder Theorem. Hence S can use the output of $\mathcal{A}$ to break the Assumption 1 with advantage ε.

Lemma 2. Suppose there exists an adversary $\mathcal{A}$ such that $\mathrm{Game}_{k-1,2}\mathrm{Adv}_{\mathcal{A}} - \mathrm{Game}_{k,1}\mathrm{Adv}_{\mathcal{A}} = \varepsilon$, then we can build a simulator S that has advantage ε in breaking the Assumption 2.

Proof. Given an instance (X_1X_2, X_3, Y_2Y_3, T) of Assumption 2, where $T = g^t g_2^e R$ or $g^t R$, for some unknown $t, e \in \mathbf{Z}_N$ and $R \in G_{p3}$, S simulates $\mathrm{Game}_{k-1,2}$ or $\mathrm{Game}_{k,1}$ as follows.

–Setup S generates *PK* and *MSK* by running the algorithm Setup.

–Queries Since S knows the *MSK*, it can generate any secret keys by running the algorithm KGen. S can also generate normal signtures for requests > k by running the algorithm Sign.

To generate the first $k-1$ semi-functional signatures, S first generates a normal signature $\sigma' = (\sigma_a', \sigma_b', (\sigma_{1,x}', \sigma_{2,x}')_{x=1,...,l})$. Then it choose random $d' \in \mathbf{Z}_N$ and outputs the signature as $\sigma = (\sigma_a'(Y_2Y_3)^{d'}, \sigma_b', (\sigma_{1,x}', \sigma_{2,x}')_{x=1,...,l})$. It is a properly distributed semi-functional signature of type 2 if we implicitly set $g_2^d = Y_2^{d'}$ and $R_0 = R_0' \prod_{x=1}^{l} (R_{\rho(x)}')^{\omega_x} Y_3^{d'}$.

To generate the k-th signature, S chooses random $\xi_{\varphi(x)} \in \mathbf{Z}_N$, $R_0', R_{\varphi(x)}', R_{\varphi(x)}'', R_{\rho(x)}' \in G_{p3}$ and $r \in \mathbf{Z}_N$. It also chooses a reconstruction vector ω and two random vectors β_1, β_2 such that $\beta_1 \cdot A = \beta_2 \cdot A = (0, 0, \ldots, 0)$, and outputs:

$$\sigma_a = (g^{\alpha} T^a R_0') \prod_{x=1}^{l} ((T g^{\xi_{\varphi(x)}})^{s_{\rho(x)}} R_{\rho(x)}')^{\omega_x} (H_{\rho(x)})^{\beta_{1x}} F(M)^r,\ \sigma_b = g^r$$

$$\sigma_{1,x} = (T g^{\xi_{\varphi(x)}} R_{\rho(x)}')^{\omega_x} g^{\beta_{1x}},\ \sigma_{2,x} = ((T^2 g^{\xi_{\varphi(x)}})^{a/b_{\varphi(x)}} R_{\varphi(x)}'')^{\omega_x} (g^{a/b_{\varphi(x)}})^{\beta_{2x}},\ 1 \le x \le l$$

If $T = g^t g_2^e R$, we have a properly distributed semi-functional signature of type 1 as follows, where we implicitly set $t_0 = t$, $t_{\varphi(x)} = t + \xi_{\varphi(x)}$, $f_x = 2ae / b_{\varphi(x)}$, $R_0 = R_0' R^a$, $R_{\rho(x)} = R^{s_{\rho(x)}} R_{\rho(x)}'$,

$g_2^d = (g_2^e)^a \prod_{x=1}^{l} (g_2^e)^{s_{\rho(x)}\omega_x}$, $R_{\varphi(x)} = RR'_{\rho(x)}$, $R'_{\varphi(x)} = R^2 R''_{\varphi(x)}$.

$$\sigma_a = g^{\alpha} g^{at} R_0' R^a \prod_{x=1}^{l} (((g^{t+\xi_i})^{s_{\rho(x)}} R^{s_{\rho(x)}} R'_{\rho(x)})^{\omega_x} (H_{\rho(x)})^{\beta_{1x}}) F(M)^r (g_2^e)^a \prod_{x=1}^{l} (g_2^e)^{s_{\rho(x)}\omega_x}$$

$$\sigma_{1,x} = (g^{t+\xi_{\varphi(x)}} RR'_{\rho(x)})^{\omega_x} g^{\beta_x} (g_2^e)^{\omega_x} , \ 1 \le x \le l$$

$$\sigma_{2,x} = ((g^{t+t+\xi_{\varphi(x)}})^{a/b_{\varphi(x)}} R^2 R''_{\varphi(x)})^{\omega_x} (g^{a/b_{\varphi(x)}})^{\beta_{2x}} ((g_2^e)^{2a/b_{\varphi(x)}})^{\omega_x} , \ 1 \le x \le l$$

If $T = g^t R$, we have a properly distributed normal signature with the implicit setting of $t_0 = t$, $t_{\varphi(x)} = t + \xi_{\varphi(x)}$.

It can be checked that the *k*-th signature, in both cases of $T = g^t g_2^e R$ or $T = g^t R$, can be successfully verified by the algorithm VerifySF, so neither $\mathcal{A}$ nor S can distinguish two games by verifying the *k*-th signature.

–Forgery To simulate the algorithm VerifySF, S implicitly sets $g^s = X_1$, $g_2^c = X_2$ and obtains $\hat{e}(g,g)^{\alpha s}$ by computing $\hat{e}(g, X_1 X_2)^{\alpha}$. It also chooses a random vector $u' = (a, u_2', ..., u_n')$ and generates verification components as follows:

$$C = X_1 X_2 , \ C_x = (X_1 X_2)^{A_x u'} (X_1 X_2)^{-s_{\rho(x)}} , \ E_x = (X_1 X_2)^{b_{\phi(x)} A_x a^{-1} u'} , \ 1 \le x \le l$$

We can check that $C = g^s g_2^c$, $C_x = g^{aA_x s a^{-1} u'} H_{\rho(x)}^{-s} g_2^{A_x cu' - cs_{\rho(x)}}$ and $E_x = g^{b_{\varphi(x)} A_x s a^{-1} u'} g_2^{cb_{\varphi} A_x a^{-1} u'}$, thus it is a properly distributed verification components with the implicit setting of $v = sa^{-1}u'$, $u = cu'$, $z_{\rho(x)} = cs_{\rho(x)}$, $\gamma_x = cb_{\varphi(x)}$ and $w = a^{-1}u'$.

To sum up, depend on whether $T \in G$ or $T \in G_{p1p3}$, S can properly simulate $\text{Game}_{k-1,2}$ or $\text{Game}_{k,1}$, thus can use the output of $\mathcal{A}$ to break the Assumption 2 with advantage ε.

Lemma 3. Suppose there exists an adversary $\mathcal{A}$ such that $\text{Game}_{k,1}\text{Adv}_{\mathcal{A}} - \text{Game}_{k,2}\text{Adv}_{\mathcal{A}} = \varepsilon$, then we can build a simulator S that has advantage ε in breaking the Assumption 2.

Proof. S simulates the games in almost the same way as it does in the lemma 2, except that it chooses a random exponent $h \in \mathbf{Z}_N$ and generates the *k*-th signature as:

$$\sigma_a = (g^{\alpha} T^a R_0') \prod_{x=1}^{l} ((T g^{\xi_{\varphi(x)}})^{s_{\rho(x)}} R'_{\rho(x)})^{\omega_x} (H_{\rho(x)})^{\beta_{1x}} F(M)^r (Y_2 Y_3)^h , \ \sigma_b = g^r$$

$$\sigma_{1,x} = (T g^{\xi_{\varphi(x)}} R'_{\rho(x)})^{\omega_x} g^{\beta_{1x}} , \sigma_{2,x} = ((T^2 g^{\xi_{\varphi(x)}})^{a/b_{\varphi(x)}} R''_{\varphi(x)})^{\omega_x} (g^{a/b_{\varphi(x)}})^{\beta_{2x}} , \ 1 \le x \le l$$

Depend on $T \in G$ or $T \in G_{p1p3}$, it is a properly distributed semi-functional signature of type 2 or normal signature, and in both cases it can't be verified by the algorithm VerifySF due to the extra term $(Y_2 Y_3)^h$. Thus S can also use the output of $\mathcal{A}$ to break the Assumption 2 with advantage ε.

Lemma 4. Suppose there exists an adversary $\mathcal{A}$ such that $\text{Game}_{q,2}\text{Adv}_{\mathcal{A}} - \text{Game}_{final}\text{Adv}_{\mathcal{A}} = \varepsilon$, then we can build a simulator S that has advantage ε in breaking the Assumption 3.

Proof. Given an instance ($g^{\alpha} X_2$, X_3, $g^s Y_2$, Z_2, T) of Assumption 3, where $T = \hat{e}(g,g)^{\alpha s}$ or $\hat{e}(g,g)^r$ for unknown $\alpha, s, r \in \mathbf{Z}_N$, S simulates $\text{Game}_{q,2}$ or Game_{final} as follows.

–Setup S chooses random exponents a, b_i , s_{ij} , $u_0, u_1, \ldots, u_{\eta} \in \mathbf{Z}_N$, and sets PK as:

$$PK = (N, G, G_T, \hat{e}, g, g^a, \hat{e}(g, g^{\alpha} X_2), u_0, u_1, \ldots, u_{\eta}, g^{b_i} \forall i , g^{a/b_i} \forall i , H_{ij} = g^{h_{ij}} \forall i \forall j)$$

–Queries S generates a secret key as follows:

$$K = g^{\alpha} X_2 g^{at_0} (Z_2)^{t_0} R_0$$

$$K_i = g^{t_i} R_i,\ K_{ij} = (H_{ij})^{t_i} R_{ij},\ 0 \le i \le m,\ 1 \le j \le n_i$$
$$L_i = g^{a(t_0+t_i)/b_i} R_i',\ 1 \le i \le m$$

Given such a secret key, S generates a signature by running the algorithm Sign. The resulting signature is a properly distributed semi-functional signature of type 2 with the implicit setting of $g_2^d = X_2(Z_2)^{t_0}$.

–Forgery To simulate the algorithm VerifySF, S chooses a random vector $u' = (a, u_2', ..., u_n')$, and outputs verification components as follows:

$$C = g^s Y_2,\ C_x = (g^s Y_2)^{A_x u'} (g^s Y_2)^{-s_{\rho(x)}},\ E_x = (g^s Y_2)^{b_{\varphi(x)} A_x a^{-1} u'}$$

It is a properly distributed semi-functional verification components, with the implicitly setting of $g_2^c = Y_2$, $v = sa^{-1}u'$, $u = cu'$, $z_{\rho(x)} = cs_{\rho(x)}$, $\gamma_x = cb_{\varphi(x)}$ and $w = a^{-1}u'$. Then S verifies the signature by the verification equation of $\hat{e}(\sigma_a, C) = T\,\hat{e}(F(M)^s, \sigma_b)\prod_{x=1}^{l}\hat{e}(E_x, \sigma_{2,x}) / \hat{e}(C_x, \sigma_{1,x})$.

If $T = \hat{e}(g,g)^{\alpha s}$, it is the original verification equation (1) used in VerifySF. Otherwise, due to the random T in the verification equation, S rejects the forged signature with overwhelming probability. Thus S can use the output of $\mathcal{A}$ to break the Assumption 3 with advantage ε.

Theorem 2 (Unforgeability). Our construction of ASBS is existential unforgeable under Assumptions 1, 2 and 3.

Proof. It follows from lemma 1 to 4, and the fact that the success probability of any polynomial-time adversary in Game $_{final}$ is negligible.

5. Conclusion

In this paper, we introduce attribute set based signature as an enhancement to attribute based signature. ASBS organizes user attributes into a recursive set based structure such that dynamic constraints can be imposed on how those attributes may be combined to satisfy a signing policy. Thus it is more flexible and efficient in managing user attributes and specifying signing policies. We present a practical construction of ASBS and prove its security in the standard model under three subgroup decision related assumptions. Its efficiency is comparable to that of the most efficient ABS scheme.

Our construction relies on a bilinear group of composite order. This requires large group orders to guarantee security. So an interesting direction for future research is to convert it to prime order groups, probably using the technique from [14].

References

[1] H. K. Maji, M. Prabhakaran and M. Rosulek, "Attribute-based signatures: achieving attribute-privacy and collusion-resistance," *IACR Cryptology ePrint Archive*, pp. 328, 2008. Article (CrossRef Link)

[2] R. Bobba, H. Khurana and M. Prabhakaran, "Attribute-sets: a practically motivated enhancement to attribute-based encryption," in *Proc. of 14th European Symposium on Research in Computer Security*, pp. 587-604, 2009. Article (CrossRef Link)

[3] T. Okamoto and K. Takashima, "Efficient attribute-based signatures for non-monotone predicates in the standard model," in *Proc. of the 14th Int. Conf. on Public Key Cryptography*, pp. 35-52, 2011. Article (CrossRef Link)

[4] A. Lewko and B. Waters, "New techniques for dual system encryption and fully secure HIBE with

short ciphertexts," in *Proc. of 7th Theory of Cryptography Conferenc*e, pp. 455-479, 2010. Article (CrossRef Link)

[5] A. Lewko, T. Okamoto, A. Sahai et al, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in *Proc. of Advances in Cryptology - EUROCRYPT 2010*, pp. 62-91, 2010. Article (CrossRef Link)

[6] J. Li, M. H. Au, W. Susilo et al, "Attribute-based signature and its application," in *Proc. of 5th ACM Symposium on Information, Computer and Communications Security*, pp. 60–69, 2010. Article (CrossRef Link)

[7] S. F. Shahandashti and R. Safavi-Naini, "Threshold attribute-based signatures and their application to anonymous credential systems," in *Proc. of Progress in Cryptology – AFRICACRYPT 2009*, pp. 198-216, 2009. Article (CrossRef Link)

[8] S. Guo and Y. Zeng, "Attribute-based signature scheme," in *Proc. of International Conference on Information Security and Assurance*c, pp. 509-511, 2008. Article (CrossRef Link)

[9] H. K. Maji, M. Prabhakaran and M. Rosulek, "Attribute-based signatures," in *Proc. of Cryptographers'Track at the RSA Conference*, pp. 376-392, 2011. Article (CrossRef Link)

[10] T. Okamoto and K. Takashima, "Decentralized attribute-based signatures," in *Proc. of* Public-Key Cryptography, pp. 125-142, 2013. Article (CrossRef Link)

[11] A. Beimel, "Secure schemes for secret sharing and key distribution," *PhD thesis, Isral institute of technology*, 1996.

[12] M. Karchmer and A. W. Karchmer, "On span programs," in *Proc. of the 8th IEEE Structure in Complexity Theory Conference*, pp. 102-111, 1993. Article (CrossRef Link)

[13] B. Waters, "Efficient identity-based encryption without random oracle," in *Proc. of Advances in Cryptology - EUROCRYPT 2005*, pp. 114-127, 2005. Article (CrossRef Link)

[14] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Proc. of Advances in Cryptology - EUROCRYPT 2012*, pp. 318-335, 2012. Article (CrossRef Link)

[15] J. Herranz, F. Laguillaumie, B. Libert and C. Ràfols, "Short attribute-based signature for threshold predicates," in *Proc. of Int. Conf. of the cryptographers' Track at the RSA*, pp. 51-67, 2012. Article (CrossRef Link)

[16] C. Chen, J. Chen, H. W. Lim et al, "Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures," in *Proc. of Int. Conf. of the cryptographers' Track at the RSA*, pp. 50-67, 2013. Article (CrossRef Link)

[17] A. Escala, J. Herranz and P. Morillo, "Revocable attribute-based signatures with adaptive security in the standard model," in *Proc. of Progress in Cryptology - AFRICACRYPT 2011*, pp. 224-241, 2011. Article (CrossRef Link)

Baohong Li, Lecturer in School of Electronics and Information Engineering, Xi'an Jiaotong University (XJTU), China. He received his M.S. degree and Ph.D. degree in computer science from XJTU in 2002 and 2006. His current research interests include Network Security and Cryptography.

Yinliang Zhao, Professor in School of Electronics and Information Engineering, Xi'an Jiaotong University (XJTU), China. He received his M.S. degree and Ph.D. degree in computer science from XJTU in 1988 and 1995. His research interests include Network Security, Parallel Computing thread-level parallelism and Compiler Optimization.

Hongping Zhao, She received her M.S. degree and Ph.D. degree from the Fourth Military Medical University in 2003 and 2006. She is currently a chief engineer in 518th Hosp. of PLA. Her research interests include network security and Management Information System.