Short and Efficient Expressive Attribute-Based Signature in the Standard Model

Aijun Ge[1], Cheng Chen[2], Chuangui Ma[1] and Zhenfeng Zhang[2]

[1]Department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou, Henan 450002, China
[2]State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing, 100190, China
Email: geaijun2011@gmail.com, chencheng@is.iscas.ac.cn

Abstract. Attribute-based signature allows the signer to announce his endorsement using a signing policy without revealing the identity, and only the signer whose attributes satisfy the signing policy can generate a valid signature. Attribute-based signature can provide flexible access control policy and has many application in real scenarios requiring both privacy and authentication. In this paper, we present a new construction of efficient attribute-based signature schemes based on Waters' ciphertext-policy attribute-based encryption schemes. Our scheme is existentially unforgeable in the standard model for the selective adversary and can achieve perfect privacy. Compared with other attribute-based signature schemes supporting the general access structure, our scheme has the shortest signature size and the best computation efficiency.

Keywords: attribute-based signature, ciphertext policy, existential unforgeability, general access structure

1 Introduction

In traditional public key cryptography, the communication model is one-to-one, that is, any encrypted message using a particular public key can be decrypted only with the corresponding secret key. For example, when one wants to distribute a message to a specific set of users, he has to encrypt it under each user's public key or identity, which is inefficient as the ciphertext size and computational cost of encryption/decryption algorithms increase linearly with the number of receivers. In other application scenarios, it is desirable to be able to encrypt without exact knowledge of the public key of intended receivers. In most cases, the qualified receivers share some common attributes, such as working location, gender or age range. Attribute-based cryptography was proposed as an efficient method to solve these problems.

Since the introduction of attribute-based encryption (ABE for short) by Sahai and Waters [21] in 2005, a lot of ABE schemes [21,7,5,9,16,2,24] have been proposed. Goyal *et al.* [7] further extended ABE and introduced two variants: key policy attribute-based encryption (KP-ABE, e.g., [7,2]) and ciphertext policy attribute-based encryption (CP-ABE, e.g., [5,9,16,24]). In a CP-ABE system, a user's private key is associated with a set of attributes and encrypted ciphertext will specify an access policy over attributes. A user can decrypt if and only if his attributes satisfy the ciphertext's policy. While in a KP-ABE system, the situation is reversed: the private key is associated with an access policy, and the ciphertext is associated with a set of attributes. A user can decrypt if and only if the attributes associated with the ciphertext satisfy the user's private key policy. Attribute-based cryptosystem has significant advantages over the traditional public key cryptosystems since it provides flexible policy, which is an important tool for secure and fine-grained data sharing and access control.

With the development of ABE, research in attribute-based signature (ABS for short) has been a very active area in recent years. The notion of ABS was introduced explicitly in the first version of [15] by Maji *et al.* Up to now, a number of ABS schemes have been proposed. According to the structure of access policy, these schemes can be divided into two categories: ABS for single threshold structure (such as [12,19,11,10]) and ABS for general access structure (such as [15,17,6,18]).

In an ABS scheme, users receive a secret key from a master entity depending on the attributes that they hold. Later, a user can use the private key and a signing policy (which must be satisfied by his attributes) to compute the signature on a message. The verifier is convinced that some user holding a set of attributes satisfying the signing policy has endorsed the message. In particular, the signature releases no other identity or attribute information about the actual signer. ABS has found many important applications, such as attribute-based messaging, attribute-based authentication, trust negotiation and leaking secrets (see [15] for detailed descriptions of applications).

Let us take a private access control mechanism for example: a server first chooses and publishes an access policy, and then users who want to access the restricted resource should first sign on a fresh challenge message chosen by the server. Note that only the user whose attributes satisfy the access policy can generate a valid signature on the challenge message and the access policy. For others whose attributes do not satisfy the policy, it is (computationally) difficult to produce a valid ABS even through collusion.

Our Contributions. Motivated by the recent work of [17,18], we give a new general construction of ABS scheme based on Waters CP-ABE schemes [24]. Our concrete scheme is provably secure against selective attacks under the assumption of computation q-Diffie-Hellman Exponent problem (qDHE), which is a modified assumption of the decisional q-Bilinear Diffie-Hellman Exponent problem (qBDHE). qBDHE assumption is introduced and shown to be hard in the generic group model by Boneh *et al.* in [3], and has been proved useful for constructing hierarchical identity-based encryption [3], broadcast encryption [4] and ABE [2,24] schemes. Compared with other ABS schemes supporting general access structures, this construction provides better efficiency in terms of the computational cost and communicational cost. What is more, the signature size in our construction is even shorter than Maji *et al.*'s [15] most efficient construction, the security for which is only proven in the generic group model. It is well

known that the generic group model is not a standard model in the security proof, while our scheme can be proved secure in the standard model (that is, without using the random oracle or the generic group heuristic). Note that this general construction can also work for the fully secure CP-ABE scheme of [13], and can turn their scheme into a fully secure ABS in composite order groups.
Organization. The rest of the paper is organized as follows. In the next section, we review some preliminaries, including the bilinear map, hardness assumption and the syntax definition of ABS. The security model of ABS scheme is given in Section 3. We present our concrete ABS scheme in Section 4. Further, we compare the efficiency and security of the proposed ABS scheme with others existing ABS schemes in Section 5. Finally, conclusions will be made in Section 6.

2 Preliminaries

2.1 Bilinear Maps

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be cyclic multiplicative groups of prime order p. Let g be a generator of $\mathbb{G}_1$ and h be a generator of $\mathbb{G}_2$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ has the following properties:

- **Bilinearity**: For all $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$ and all $a, b \in \mathbb{Z}_p^*$, we have $e(g^a, h^b) = e(g,h)^{ab}$.
- **Non-degeneracy**: $e(g,h) \neq 1$.
- **Computability**: There's an efficient algorithm to compute $e(u,v)$ for any $u \in \mathbb{G}_1$ and $v \in \mathbb{G}_2$.

2.2 Hardness Assumption

The security of our construction is based on the computational q-Diffie-Hellman Exponentiation assumption, which is modified from the decisional q-Bilinear Diffie-Hellman Exponentiation (qBDHE)[3].

Let $\mathbb{G}$ be a bilinear group with prime order p, the qBDHE problem in $\mathbb{G}$ is stated as follows: Given the following $2q+1$ elements $(g, h, g^a, g^{a^2}, ..., g^{a^q}, g^{a^{q+2}}, ..., g^{a^{2q}}) \in \mathbb{G}^{2q+1}$ as input, where a is chosen at random from $\mathbb{Z}_p$, the goal of the qBDHE problem is to output $e(g,h)^{a^{q+1}}$.

Definition 1. *(q-Diffie-Hellman Exponentiation, qDHE) We say the (t, ε) qDHE assumption holds in a group $\mathbb{G}$, if there is no probabilistic polynomial time adversary who is able to compute $g^{a^{q+1}}$ just given $(g, g^a, g^{a^2}, ..., g^{a^q}, g^{a^{q+2}}, ..., g^{a^{2q}})$ running in time at most t with probability at least ε, where $a \in \mathbb{Z}_p$ and $g \in \mathbb{G}$ are chosen independently and uniformly.*

2.3 Access Structure and Linear Secret Sharing Scheme

Definition 2. *(Access structure[1]) Let $\{P_1, P_2, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,...,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in$*

$\mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, ..., P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, ..., P_n\}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Definition 3. *(Linear secret sharing scheme [1], LSSS) A secret sharing scheme Π over a set of parties $\mathcal{P}$ is called linear if*

1. *The shares for each party form a vector over $\mathbb{Z}_p$;*
2. *There exists a matrix M with ℓ rows and k columns called the share generating matrix for Π. For all $i = 1, ..., \ell$, the ith row of M we let the function ρ defined the party labeling row i as $\rho(i)$. When we consider the column vector $v = (s, r_2, ..., r_k)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, ..., r_k \in \mathbb{Z}_p$ are randomly chosen, then Mv is the vector of ℓ shares of the secret s according to Π. The share $(Mv)_i$ belongs to party $\rho(i)$.*

Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be an authorized set, and let $I \subseteq \{1, 2, ..., \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}_{i \in I}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \lambda_i w_i = s$. It is shown in [1] that these constants w_i can be found in polynomial in the size of the share generating matrix M.

Using standard techniques [1], one can convert any monotonic boolean formulas into an LSSS representation. An access tree of ℓ nodes will result in an LSSS matrix of ℓ rows. It is shown in [14] that, one can convert any access tree into an LSSS matrix. The signing predicate we will use in our signature scheme is LSSS access structure.

2.4 Syntax of ABS Scheme

According to [15], an ABS scheme is made up of four algorithms: **Setup**, **Extract**, **Sign** and **Verify**. For a fixed security parameter λ, these algorithms work as follows:

- **Setup**(λ): The Setup algorithm takes input the security parameter λ, and it returns some public parameters *params* and the master secret key *msk*. The public parameters contain the universe of attributes $\mathbb{U}$.
- **Extract**($msk, params, S$) The Extract algorithm takes the master secret key *msk*, the public parameters *params* and a user's attribute set $S \subseteq \mathbb{U}$ as input, and the attribute authority computes the attribute private key SK_S as the algorithm's output.
- **Sign**($m, params, \Upsilon, SK_S$): The Sign algorithm takes a message m, the public parameters *params*, a signing predicate Υ and a user's attribute key SK_S with attribute set S satisfying predicate Υ (we say that an attribute set S satisfies the predicate Υ if $\Upsilon(S) = 1$) as input, and outputs a valid signature σ.

- **Verify**$(m, params, \Upsilon, \sigma)$: The Verify algorithm takes the public parameters $params$, the signing predicate Υ, the message m and its signature σ as input, and outputs a boolean value either "valid" or "invalid".

Correctness. For any correctly generated signature on the message m by a signer with attributes w satisfying the claim-predicate Υ, we have

$$\textbf{Verify}(params, \Upsilon, m, \textbf{Sign}(params, \Upsilon, SK_S, m)) = valid.$$

3 Security Models

An ABS scheme must satisfy two security properties: unforgeability and perfect privacy. The definition of unforgeability is based on the idea that an adversary should not be able to generate a valid signature if his attribute set S does not satisfy the predicate Υ. For the perfect privacy, the signature reveals nothing about the identity or attributes of the signer. Our security model combines the security model proposed by Maji *et al.* [15] and Li *et al.*[11]

3.1 Unforgeability

Our scheme is existentially unforgeable against selective predicate attacks, which is a weaker security model than the adaptive chosen predicate attack. The difference is that the adversary of the selective predicate model has to give the challenge predicate at the beginning of the security game. This model has also been used in other attribute-based scheme [7,2,11,12,19]. The formal definition is given in the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

- **Initial Phase**: The adversary $\mathcal{A}$ declares a challenge predicate Υ^* , which will be used in the forgery signature.
- **Setup Phase**: After receiving the challenge predicate Υ^*, the challenger $\mathcal{C}$ chooses a security parameter λ and runs Setup algorithm to generate the master secret key msk and public parameters $params$. $\mathcal{C}$ gives $params$ to the adversary $\mathcal{A}$, while keeps the msk secretly.
- **Queries Phase**: The adversary $\mathcal{A}$ can adaptively query Extraction Oracle and Signing Oracle for a polynomially bounded times, and $\mathcal{C}$ answers the queries with the master secret key msk:
  - Extraction Oracle: $\mathcal{A}$ can request a private key SK_S for any attribute set $S \subseteq \mathbb{U}$.
  - Signing Oracle: $\mathcal{A}$ can request a signature for any message m and predicate Υ.
- **Forgery Phase**: Finally, the adversary $\mathcal{A}$ outputs a signature σ^* on a message m^* with respect to the challenge predicate Υ^*. We say that the adversary wins the game if
  1. $\mathcal{A}$ has not made Extraction Oracle for any attribute set S^* that S^* satisfying Υ^*;
  2. (m^*, Υ^*) has not been queried to the Signing Oracle;

3. σ^* is a valid signature on the message m^* and predicate Υ^*.

The advantage $Adv_{ABS,A}^{sP-CMA-EUF}$ is defined as the probability that $\mathcal{A}$ wins above game.

Definition 4. *An adversary $\mathcal{A}$ $(t, q_K, q_S, \varepsilon)$ breaks an ABS scheme if $\mathcal{A}$ runs in time at most t, and makes at most q_K and q_S times Extraction Oracle queries and Signing Oracle queries, while the advantage $Adv_{ABS,A}^{sP-CMA-EUF}$ is at least ε. A signature scheme is $(t, q_K, q_S, \varepsilon)$ existentially unforgeable if there is no forger that can $(t, q_K, q_S, \varepsilon)$ break it.*

3.2 Perfect Privacy

Definition 5. *An ABS scheme satisfies perfect privacy if for any two attribute sets S_1 and S_2, a message m, a signature σ on predicate Υ with $\Upsilon(S_1) = \Upsilon(S_2) = 1$, any adversary $\mathcal{A}$ cannot identify which attribute set S_1 or S_2 is used to generate the signature σ better than random guessing.*

If the perfect privacy holds, then a signature does not leak which set of attributes of signing key was used to generate it. This holds even the adversary has unbounded computational power and has access to the signer's private keys, that is, the signature is simply independent of everything except the message and the predicate.

4 New Construction of ABS Scheme

In this section, we present an efficient ABS scheme with short signature size, which is proven selective secure under the qDHE assumption (Definition 1) in the standard model. We also show a fully security construction in composite order groups based on [13] in Appendix B.

- **Setup:** Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be cyclic multiplication groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ is an efficient bilinear map. Note that this construction can also work on asymmetric pairing groups, where $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and $\mathbb{G}_1 \neq \mathbb{G}_2$. The universal set of attributes $\mathbb{U}$ $(|\mathbb{U}| = U, U \in \mathbb{Z}_p)$ used in this system is public known. First, a random generator $g \in \mathbb{G}_1$, two exponents $\alpha, a \in \mathbb{Z}_p^*$ are chosen. Then, pick random $h_1, h_2, ..., h_U, u_0, u_1, u_2, ..., u_n \in \mathbb{G}_1$. We use $u_0, u_1, u_2, ..., u_n$ as the Waters' hash function [23], and n is the length of a message used in this system. The master key is g^α and the public parameters are $params = (g, g^a, e(g, g)^\alpha, h_1, h_2, ..., h_U, u_0, u_1, u_2, ..., u_n)$.
- **Extract:** The private key for a user with attributes $S \subseteq \mathbb{U}$ is generated as follows:
  1. The attribute authority randomly chooses $t \in \mathbb{Z}_p$, then computes $K = g^\alpha g^{at}$, $L = g^t$;
  2. For each attribute $x \in S$, the attribute authority computes $K_x = h_x^t$;
  3. Finally, the attribute authority outputs the private key: $SK_S = (K, L, \{K_x\}_{x \in S})$.

- **Sign:** Let the signing predicate Υ can be represented by an LSSS access structure (M, ρ), that is, M is an $\ell \times k$ matrix, and ρ is an injective function that associates rows of M to attributes. The signature for the message $m = (\mu_1, \mu_2, ..., \mu_n) \in \{0,1\}^n$ is constructed as follows:
  1. The signer should first blind his private key SK_S as follows: The signer chooses a random $t' \in \mathbb{Z}_p$, and sets $K' = Kg^{at'} = g^{\alpha} g^{a(t+t')}, L' = Lg^{t'} = g^{t+t'}, \forall x \in S : K'_x = K_x h_x^{t'} = h_x^{(t+t')}$. Then the signer's new private key $SK'_S = (K', L', \{K'_x\}_{x \in S})$.
  2. As the signer's attribute set S should satisfy the signing predicate Υ, that is $\Upsilon(S) = 1$, then the signer can find $\overrightarrow{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_\ell)$ that satisfies $\overrightarrow{\alpha} M = (1, 0, ..., 0)$. In additional, the signer find another random $\overrightarrow{\beta} = (\beta_1, \beta_2, ..., \beta_\ell)$ that satisfies $\overrightarrow{\beta} M = (0, 0, ..., 0)$. For the existence of $\overrightarrow{\beta}$, we have a discussion in Section 5.2.
  3. For each $i \in [1, \ell]$, the signer computes $s_i = (L')^{\alpha_i} g^{\beta_i}$, and sets $y = \prod_{i=1}^{\ell} ((K'_{\rho(i)})^{\alpha_i} (h_{\rho(i)})^{\beta_i})$. Then, the signer random chooses $r \in \mathbb{Z}_p$ and computes: $\sigma_1 = yK'(u_0 \prod_{j=1}^{n} (u_j)^{\mu_j})^r$, $\sigma_2 = g^r$.
     Note: The signer may not have the key $K'_{\rho(i)}$ for every attribute $\rho(i)$ in the computation of y. However, in this case, $\alpha_i = 0$, and so the value is not needed.
  4. Finally, the signer outputs the signature $\sigma = (s_1, ..., s_\ell, \sigma_1, \sigma_2)$.
- **Verify:** Given a signature $\sigma = (s_1, ...s_\ell, \sigma_1, \sigma_2)$ of a message $m = (\mu_1, \mu_2, ..., \mu_n) \in \{0,1\}^n$ for the predicate Υ (corresponding to $(M_{\ell \times k}, \rho)$), firstly, the verifier chooses randomly $v_1 = 1, v_2, ..., v_k$ from $\mathbb{Z}_p$, sets $\overrightarrow{v} = (1, v_2, ..., v_k)$ and computes $\lambda_i = \sum_{j=1}^{k} v_j M_{i,j}$, where M_i is the vector corresponding to the ith row of matrix M. The verifier accepts the signature if and only if the following equation holds, otherwise rejects it.

$$e(g,g)^{\alpha} e((u_0 \prod_{j=1}^{n} (u_j)^{\mu_j}), \sigma_2) \prod_{i=1}^{\ell} e(g^{a\lambda_i} h_{\rho(i)}, s_i) \stackrel{?}{=} e(g, \sigma_1).$$

5 Security Analysis

The correctness of this **Verify** algorithm follows from the correctness of the Waters' CP-ABE scheme [24], and is shown in Appendix A.

5.1 Existential Unforgeability

Theorem 1. *The new ABS scheme is existentially unforgeable under the selective predicate attack, assuming that the qDHE assumption holds in $\mathbb{G}_1$.*

Proof. Suppose there exists an adversary $\mathcal{A}$ with non-negligible advantage ε against our scheme, then we can construct a probability polynomial time algorithm that can solve the qDHE problem. We define the game between a challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ as follows:

Init: The challenger $\mathcal{C}$ is given qDHE challenge $(g, g^a, g^{a^2}, ..., g^{a^q}, g^{a^{q+2}}, ..., g^{a^{2q}})$ and asked to compute $g^{a^{q+1}}$. The game begins with $\mathcal{A}$ sends a challenge predicate $\Upsilon^*(M^*, \rho^*)$, where the size of the the challenge predicate matrix M^* is $\ell^* \times k^*$ with $k^* \leqslant q$.

Setup: $\mathcal{C}$ first defines the parameters $h_1, h_2, ..., h_U$ as follows. For each attribute x in this system, $\mathcal{C}$ chooses a random value $z_x \in \mathbb{Z}_p$. If attribute x appeared in the challenge predicate $\Upsilon^*(M^*, \rho^*)$, that is, there exists an i such that $\rho^*(i) = x$, then let $h_x = g^{z_x} g^{aM^*_{i,1}} g^{a^2 M^*_{i,2}} ... g^{a^{k^*} M^*_{i,k^*}}$. Otherwise $h_x = g^{z_x}$. Further more, $\mathcal{C}$ randomly chooses $\alpha', b_j \in \mathbb{Z}_p, a_j \in \{-1, 0, 1\}$ for each $0 \leqslant j \leqslant n$, and sets $e(g,g)^\alpha = e(g^a, g^{a^q}) \cdot e(g,g)^{\alpha'}$, $u_j = g_1^{a_j} g^{b_j} (0 \leqslant j \leqslant n)$, with $g_1 = g^{a^q}$. We note that the master secret key $g^\alpha = g^{a^{q+1}} \cdot g^{\alpha'}$ is unknown to $\mathcal{C}$. For convenience, let $a(m) = a_0 + \sum_{i=1}^n a_i m_i$, $b(m) = b_0 + \sum_{i=1}^n b_i m_i$.

Queries: The adversary $\mathcal{A}$ can adaptively query the following oracles for a polynomially bounded times, and $\mathcal{C}$ answers these queries in the following way:

- Extraction Oracles: Suppose the adversary $\mathcal{A}$ asks the private key of attributes set S with the restriction that S does not satisfy M^*. The challenger $\mathcal{C}$ first randomly chooses $t_1 \in \mathbb{Z}_p$, then it finds a vector $\overrightarrow{w} = (w_1, ..., w_{k^*}) \in \mathbb{Z}_p^{k^*}$ such that $w_1 = -1$ and for all i where $\rho^*(i) \in S$ we have $\overrightarrow{w} \cdot M^*_i = 0$. By implicitly defining $t = t_1 + w_1 a^q + w_2 a^{q-1} + ... + w_{k^*} a^{q-k^*+1}$, $\mathcal{C}$ outputs the private key $SK_S = (K, L, \{K_x\}_{x \in S})$ as follows:

$$K = g^\alpha g^{at} = g^{\alpha'} g^{at_1} \prod_{i=2,...k^*} (g^{a^{q+2-i}})^{w_i}$$
$$L = g^t = g^{t_1} \prod_{i=1,...k^*} (g^{a^{q+1-i}})^{w_i}$$

$K_x = L^{z_x} \prod_{j=1,...k^*} \Big(g^{a^j t_1} \prod_{\substack{k=1,...,k^* \\ k \neq j}} (g^{a^{q+1+j-k}})^{w_k}\Big)^{M^*_{i,j}}$ for $\rho^*(i) = x \in S$, and

$K_x = (h_x)^t = L^{z_x}$ for attribute x does not appear in the challenge predicate, that is, there is no i such that $\rho^*(i) = x$.
- Signing Oracles: Consider a query for a signature of $m = (\mu_1, \mu_2, ..., \mu_n) \in \{0,1\}^n$ on predicate $\Upsilon(M_{\ell \times k}, \rho)$ with attribute set S that satisfies $\Upsilon(S) = 1$. The challenger will construct a signature in the following way:
  1. If S does not satisfy the challenge predicate Υ^*, $\mathcal{C}$ can get the private key of S by querying the Extraction Oracles, and then generate the valid signature normally using these private keys
  2. If S does satisfy the challenge predicate Υ^*, that is $\Upsilon^*(S) = 1$. $\mathcal{C}$ first check $a(m) = a_0 + \sum_{i=1}^n a_i \mu_i$ equals 0 or not. If $a(m) = 0$, $\mathcal{C}$ outputs "failure" and stops this game. Otherwise, when $a(m) \neq 0$, $\mathcal{C}$ first compute $\overrightarrow{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_\ell)$ and $\overrightarrow{\beta} = (\beta_1, \beta_2, ..., \beta_\ell)$ with $\overrightarrow{\alpha} M = (1, 0, ..., 0)$, $\overrightarrow{\beta} M = (0, 0, ..., 0)$ as $\Upsilon(S) = 1$. $\mathcal{C}$ randomly chooses $r', t' \in \mathbb{Z}_p$ and

assigns $r = r' - (a/a(m))$, and then computes the signature:

$$s_i = g^{\alpha_i t' + \beta_i} (i = 1, 2..., \ell)$$
$$y = \prod_{i=1}^{\ell} (h_{\rho(i)})^{\alpha_i \cdot t' + \beta_i}$$
$$\sigma_1 = y g^{\alpha + a t'} (g_1^{a(m)} g^{b(m)})^r) = y(g^{\alpha'} (g^a)^{t'} (g^{a^q})^{r' a(m)} g^{b(m) \cdot (r' - a/a(m))})$$
$$\sigma_2 = g^{r' - (a/a(m))} = g^{r'} (g^a)^{-(1/a(m))}$$

Forgery: Finally, the adversary $\mathcal{A}$ outputs a forged signature $\sigma^* = (s_1^*, ...s_\ell^*, \sigma_1^*, \sigma_2^*)$ of a message $m^* = (\mu_1^*, \mu_2^*, ..., \mu_n^*) \in \{0, 1\}^n$ for the challenge predicate Υ^* (the corresponding monotone span program is $(M^*_{\ell^* \times k^*}, \rho^*)$). If $a(m^*) \neq 0$, $\mathcal{C}$ will abort. Otherwise, $\mathcal{C}$ can give the solution to the qDHE problem. $\mathcal{C}$ implicitly sets $\overrightarrow{v} = (s = -1, -a, -a^2..., -a^{k^*-1})$, and computes $\lambda_i = \overrightarrow{v} \cdot M_i^* = -\sum_{j=1}^{\ell^*} M^*_{i,j} a^{j-1}$. As

$$h_{\rho*(i)} = g^{z_{\rho*(i)}} g^{a M^*_{i,1}} g^{a^2 M^*_{i,2}} ... g^{a^{k^*} M^*_{i,k^*}} = g^{z_{\rho*(i)}} g^{-a\lambda_i},$$

$\mathcal{C}$ can compute

$$g^\alpha \sigma_2^{b(m^*)} (\textstyle\prod_{i=1}^{\ell^*} s_i^{z_{\rho*(i)}}) = \sigma_1$$

According to $g^\alpha = g^{a^{q+1}} \cdot g^{\alpha'}$, $\mathcal{C}$ outputs

$$g^{a^{q+1}} = \sigma_1 (\sigma_2)^{-b(m*)} g^{-\alpha'} \prod_{i=1}^{\ell^*} s_i^{-z_{\rho*}(i)}$$

as the solution to the submitted instance of the qDHE problem, which contradicts with qDHE assumption.

Probability: For the simulation to complete without aborting, we require the following conditions fulfilled

1. For each m_i to be queried in the Signing Oracle queries, we have that $a(m_i) \neq 0$;
2. For the outputting forgery message m^*, we have that $a(m^*) = 0$.

Using the technique of $(1, q_S, 0, P)$ programmable hash function in [20], we can get the probability of $\mathcal{C}$ not aborting as $P = \mathcal{O}(\frac{1}{q_S \sqrt{n}})$, where q_S is the maximum number of Signing Oracle queries the adversary $\mathcal{A}$ can make. Therefore, we can get the probability of solving qDHE problem as $\varepsilon' \geqslant \varepsilon \cdot P$, if the adversary $\mathcal{A}$ succeeds breaking our ABS scheme with probability ε. □

5.2 Perfect Privacy

Theorem 2. *The attribute-based signature scheme we proposed in Section 4 can achieve perfect privacy.*

Proof. First we note that the role of the vector $\overrightarrow{\beta}=(\beta_1,\beta_2,...,\beta_\ell)$ is to hide the real attributes the signer used to sign the message. For any predicate Υ (The corresponding LSSS access structure is $(M_{\ell\times k},\rho)$), if $rank(M)<\ell$, it is obvious that there exists polynomial numbers of $\overrightarrow{\beta}$ that satisfies $\overrightarrow{\beta}M=(0,0,...,0)$. For a simple example, M could be $(1,1,1,1)^T$ (a column vector, which is the transpose of $(1,1,1,1)$ for a policy of A_1 or A_2 or A_3 or A_4 (following the way of [14] in the construction of M). In that case, $\overrightarrow{\beta}=(\beta_1,\beta_2,\beta_3,\beta_4)$ could be $(1,-1,1,-1)$ or $(1,1,1,-3)$ or any other values that satisfies $\beta_1+\beta_2+\beta_3+\beta_4=0$. On the other hand, if the matrix M is full rank, then there is only one $\overrightarrow{\beta}=(0,0,...,0)$ that satisfies $\overrightarrow{\beta}M=(0,0,...,0)$. But in this case, as $\overrightarrow{\alpha}M=(1,0,...,0)$, $\overrightarrow{\alpha}=(1,1,...,1)$, which means the signing predicate are limited to conjunction or (n,n)-threshold predicate, and there is no attribute privacy at all. So here, we just consider the case of $rank(M)<\ell$.

The perfect privacy game begins with the challenger running **Setup** to get the public parameters *params* and the master key g^α. The challenger also gives the adversary *params* and g^α. After these interactions, the adversary outputs a challenge predicate Υ and two attributes S_1 and S_2 with $\Upsilon(S_1)=\Upsilon(S_2)=1$. Assume the challenger or adversary has generated the private keys as

$$SK_{S_1}=(g^\alpha g^{at_1},g^{t_1},\{(h_x)^{t_1}\}_{x\in S_1}),$$

$$SK_{S_2}=(g^\alpha g^{at_2},g^{t_2},\{(h_x)^{t_2}\}_{x\in S_2}).$$

Then the adversary outputs a message m and asks the challenger to generate a signature on the message m and predicate Υ with the private key SK_S either SK_{S_1} or SK_{S_2}. The challenger chooses a random bit $b\in\{1,2\}$, and outputs a signature $\sigma^*=(s_1^b,...,s_\ell^b,\sigma_1^b,\sigma_2^b)$ by running algorithm **Sign** with the private key SK_{S_b}

Since the challenger can generate a valid signature either by using SK_{S_1} or SK_{S_2}, and for our purpose, we just have to show that the distributions of signatures created by using SK_{S_1} or SK_{S_2} are identical. Here, we show that a signature created by using SK_{S_1} can also be generated by using SK_{S_2}:

Using the private key SK_{S_1}, the challenge signature $\sigma^*=(s_1^1,...,s_\ell^1,\sigma_1^1,\sigma_2^1)$ is in the form of

$$\{s_i^1=g^{\alpha_i^1t+\beta_i^1}\}_{i=1,2,...,\ell}$$

$$y^1=\prod_{i=1}^{\ell}(h_{\rho(i)})^{\alpha_i^1t+\beta_i^1}$$

$$\sigma_1^1=y^1g^\alpha g^{at}(u_0\prod\nolimits_{j=1}^{n}(u_j)^{\mu_j})^{r^1})$$

$$\sigma_2^1=g^{r^1},$$

where $t=t_1+t_1'$(t_1' and r^1 are randomly chosen by the challenger), $\overrightarrow{\alpha^1}=(\alpha_1^1,\alpha_2^1,...,\alpha_\ell^1)$ satisfies $\overrightarrow{\alpha^1}M=(1,0,...,0)$ according to attributes set S_1, and the vector $\overrightarrow{\beta^1} = (\beta_1^1, \beta_2^1, ..., \beta_\ell^1)$ satisfying $\overrightarrow{\beta^1} M = (0, 0, ..., 0)$ is also randomly chosen by the challenger.

For private key SK_{S_2}, there is another vector $\overrightarrow{\alpha^2} = (\alpha_1^2, \alpha_2^2, ..., \alpha_\ell^2)$ satisfies $\overrightarrow{\alpha^2} M = (1, 0, ..., 0)$. We have random t_2' and r^2 satisfy $t_2 + t_2' = t = t_1 + t_1'$, $r^2 = r^1$, respectively.

In addition, we set $\overrightarrow{\beta^2} = (\beta_1^2, \beta_2^2, ..., \beta_\ell^2)$ and for each $\beta_i^2 = (\alpha_i^1 - \alpha_i^2)t + \beta_i^1$. We can check that

$$\begin{aligned}
\overrightarrow{\beta^2} M &= \sum\nolimits_{i=1}^{\ell} \beta_i^2 M_i \\
&= \sum\nolimits_{i=1}^{\ell} (\alpha_i^1 - \alpha_i^2) t M_i + \sum\nolimits_{i=1}^{\ell} \beta_i^1 M_i \\
&= t(\overrightarrow{\alpha^1} M) - t(\overrightarrow{\alpha^2} M) + (\overrightarrow{\beta^1} M) \\
&= (t, 0, ..., 0) - (t, 0, ..., 0) + (0, 0, ..., 0) \\
&= (0, 0, ..., 0)
\end{aligned}$$

So, choosing proper random t_2', r^2 and $\overrightarrow{\beta^2}$, the challenger can generate the same valid signature $\sigma^* = (s_1^1, ..., s_\ell^1, \sigma_1^1, \sigma_2^1)$ from the private key SK_{S_2}. By using the similar proof, one can also get the following result: if a signature is generated by the private key SK_{S_2}, it can also be generated form private key SK_{S_1}. From the proof, we have shown that the ABS scheme satisfies perfect privacy. □

5.3 Efficiency

In this section, we compare our scheme with other existing ABS schemes that support the general access structure in the literature. Since the instantiations 1,2 in [15] and the schemes in [6] can be seen as the general constructions, and these schemes are much complicated and very inefficient as they will employ the Groth-Sahai non-interactive proof [8], which can be indicated by the comparison in [17]. We refer interested readers to [17] for the complexity of ABS scheme using the Groth-Sahai proof [8], so here, we do not compare their schemes in Table 1. Let ℓ and k represent the size of the underlying access structure matrix $M_{\ell \times k}$ for a signing predicate. The signature size of our scheme is $\ell + 2$ group elements, and the complexity is only $\ell + 3$ paring operations (Here we only consider the complex pairing operations for our convenience in the Table 1).

Note that the ABS scheme [10] with constant size signatures can be extended to admit some other more expressive kinds of monotone predicates (such as hierarchical threshold predicates [22] but not the fully expressive access structure), in which case the signature size is no longer constant. Our scheme is more efficient compared with [10] for the expressive access structure such as the CNF or DNF form.

Table 1. Comparison with other ABS schemes

Schemes	MPR11 [15]	OT11 [17]	OT12 [18]	Ours
Signature Size	$\ell + k + 2$	$7\ell + 11$	13ℓ	$\ell + 2$
Complexity	$k\ell + k + 3$	$7\ell + 15$	13ℓ	$\ell + 3$
Security	full	full	full	selective
Model	generic group	standard	random oracle	standard
Predicate	monotone	non-monotone	non-monotone	monotone
Multi-authority	No	Yes	Yes	No

6 Conclusion

In this paper, we give a new construction of ABS scheme based on Waters' CP-ABE framework [24]. Under the qDHE assumption, this scheme can be proved existentially unforgeable against selective predicate attack in the standard model. Compared with other ABS schemes that supports the general signing policies, our scheme can achieve the best efficiency in terms of the signature size and computation costs, at the expense of a weaker security.

References

1. Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, Isral institute of technology, Technion, Haifa, Israel, 1996.
2. Attrapadung N, Libert B, Panafieu E. Expressive key policy attribute-based encryption with constant size ciphertexts. In *Proc. the 14th Int. Conf. on Public Key Cryptography* (*PKC 2011*), Taormina, Italy, *LNCS 6571*, Springer-Verlag, 2011, pp.90-108.
3. Boneh D, Boyen X, Goh E. Hierarchical identity-based encryption with constant size ciphertext. In *Proc. Advances in Cryptology* (*EUROCRYPT 2005*), Aarhus, Denmark, *LNCS 3494*, Springer-Verlag, 2005, pp.440-456.
4. Boneh D, Gentry C, Waters B. Collusion-resistant broadcast encryption with short ciphertexts and private keys. In *Proc. Advances in Cryptology* (*CRYPTO 2005*), Santa Barbara, CA, USA, *LNCS 3621*, Springer-Verlag, 2005, pp.258-275.
5. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In *Proc. the IEEE Symposium on Security and Privacy*, Oakland, Washington, DC, USA, 2007, pp.321-334.
6. Escala A, Herranz J, Morillo Paz. Revocable attribute-based signatures with adaptive security in the standard model. In *Proc. Advances in Cryptology* (*AFRICACRYPT 2011*), Dakar, Senegal, *LNCS 6737*, Springer-Verlag, 2011, pp.224-241.
7. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of the 13th ACM Conference on Computer and Communications Security* (*CCS 2006*), Alexandria, VA, USA, 2006, pp.89-98.
8. Groth J, Sahai A. Efficient non-interactive proof systems for bilinear groups. In *Proc. Advances in Cryptology* (*EUROCRYPT 2008*), Istanbul, Turkey, *LNCS 4965*, Springer-Verlag, 2008, pp.415-432.

9. Herranz J, Laguillaumie F, Rafols C. Constant size ciphertexts in threshold attribute-based encryption. In *Proc. the 13th Int. Conf. on Public Key Cryptography* (*PKC 2010*), Paris, France, *LNCS 6056*, Springer-Verlag, 2010, pp.19-34.
10. Herranz, J., Laguillaumie, F., Libert B., Rafols, C.: Short attribute-based signature for threshold predicates. In *Proc. Int. Conf. of the cryptographers' Track at the RSA* (*CT-RSA 2012*), San Francisco, USA, *LNCS 7178*, Springer-Verlag, 2012, pp.51-67.
11. Li J, Au M, Susio W, Xie D, Ren R. Attribute-based signature and its applications. In *5th ACM Symposium on Information, Computer and Communications Security* (*ASIACCS 2010*), Beijing, China, 2010, pp.60-69.
12. Li J, Kim K. Hidden attribute-based signatures without anonymity revocation, *Information Sciences*, 2010, 180, pp.1681-1689.
13. Lewko A, Okamoto T, Sahai A, Takashima K, Waters B. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In *Proc. Advances in Cryptology* (*EUROCRYPT 2010*), Monaco and Nice, French, *LNCS 6110*, Springer-Verlag, 2010, pp.62-91.
14. Lewko A, Waters B. Decentralizing attribute-based encryption. In *Proc. Advances in Cryptology* (*EUROCRYPT 2011*), Tallinn, Estonia, *LNCS 6632*, Springer-Verlag, 2011, pp.568-588.
15. Maji H, Prabhakaran M, Rosulek M. Attribute-based signature: Achieving attribute privacy and collusion-resistance. In *Proc. Int. Conf. of the cryptographers' Track at the RSA* (*CT-RSA 2011*), San Francisco, USA, *LNCS 6558*, Springer-Verlag, 2011, pp.376-392. The first version available at http://eprint.iacr.org/2008/328.
16. Okamoto T, Takashima K. Fully secure functional encryption with general relations from the decisional linear assumption. In *Proc. Advances in Cryptology* (*CRYPTO 2010*), Santa Barbara, CA, USA, *LNCS 6223*, Springer-Verlag, 2010, pp.191-208.
17. Okamoto T, Takashima K. Efficient attribute-based signatures for non-monotone predicates in the standard model. In *Proc. of the 14th Int. Conf. on Public Key Cryptography* (*PKC 2011*), Taormina, Italy, *LNCS 6571*, Springer-Verlag, 2011, pp.35-52.
18. Okamoto T, Takashima K. Decentralized attribute-based signatures. Technical Report, Cryptology ePrint Archive, Report 2011/701, 2011, http://eprint.iacr.org/2011/701
19. Shahandashti S, Safavi-Naini R. Threshold attribute-based signature and their application to anonymous credential systems. In P*roc. Advances in Cryptology* (*AFRICACRYPT 2009*), Gammarth, Tunisia, *LNCS 5580*, Springer-Verlag, 2009, pp.198-216.
20. Schage S, Schwenk J. A CDH-based ring signature scheme with short signatures and public keys. In *Proc. of the 14th Int. Conf. on Financial Cryptography and Data Security* (*FC 2010*), *LNCS 6052*, Springer-Verlag, 2010, pp.129-142.
21. Sahai A, Waters B. Fuzzy identity-based encryption. In *Proc. Advances in Cryptology* (*EUROCRYPT 2005*), Aarhus, Denmark, *LNCS 3494*, Springer-Verlag, 2005, pp.457-473.
22. Tassa T. Hierarchical threshold secret sharing. Journal of Cryptology, 20(2), pp. 237-264, 2007.
23. Waters B. Efficient identity-based encryption without random oracle. In *Proc. Advances in Cryptology* (*EUROCRYPT 2005*), Aarhus, Denmark, *LNCS 3494*, Springer-Verlag, 2005, pp.114-127.

24. Waters B. Ciphertext policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proc. the 14th Int. Conf. on Public Key Cryptography* (*PKC 2011*), Taormina, Italy, *LNCS 6571*, Springer-Verlag, 2011, pp.53-70.

A Correctness

If $\sigma = (s_1, ..., s_\ell, \sigma_1, \sigma_2)$ is a valid signature of the message $m = (\mu_1, \mu_2, ..., \mu_n) \in \{0,1\}^n$ for the predicate Υ (corresponding to $(M_{\ell \times k}, \rho)$), then

$$\begin{aligned}\sigma_1 = &\ yK'(u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j})^r) \\ &= K' \prod_{i=1}^{\ell} ((K'_{\rho(i)})^{\alpha_i} (h_{\rho(i)})^{\beta_i})(u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j})^r \\ &= g^\alpha g^{at} \prod_{i=1}^{\ell} (h_{\rho(i)}^{\alpha_i t + \beta_i})(u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j})^r\end{aligned}$$

So,

$$\begin{aligned}e(g, \sigma_1) &= e(g, g^\alpha g^{at} \prod_{i=1}^{\ell} (h_{\rho(i)}^{\alpha_i t + \beta_i})(u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j})^r) \\ &= e(g,g)^\alpha e(u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j}, g^r) e(g, g^{at} \prod_{i=1}^{\ell} (h_{\rho(i)}^{\alpha_i t + \beta_i})) \\ &= e(g,g)^\alpha e(u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j}, \sigma_2) e(g, g^{at}) \prod_{i=1}^{\ell} e(g^{\alpha_i t + \beta_i}, h_{\rho(i)}) \\ &= e(g,g)^\alpha e(u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j}, \sigma_2) \prod_{i=1}^{\ell} e(s_i, g^{a\lambda_i} h_{\rho(i)})\end{aligned}$$

Note that $\lambda_i = \sum_{j=1}^{k} v_j M_{i,j},$, the last equality is derived by: $\sum_{i=1}^{\ell} \lambda_i(\alpha_i t + \beta_i) = t \sum_{i=1}^{\ell} \lambda_i(\alpha_i) + \sum_{i=1}^{\ell} \lambda_i \beta_i) = t \cdot 1 + 0 = t$.

B Fully Secure ABS Scheme

– **Setup:** The Setup algorithm first chooses a bilinear group $\mathbb{G}$ of order $N = p_1 p_2 p_3$ (three distinct primes). We let $\mathbb{G}_{p_i}$ denote the subgroup of order p_i in $\mathbb{G}$. Note that the universal set of attributes $\mathbb{U}$ ($|\mathbb{U}| = U, U \in \mathbb{Z}_p$) used in this system is public known. A random generator $g \in$ $mathbbG_{p_1}$ and two exponents $\alpha, a \in \mathbb{Z}_p^*$ are chosen. Then, pick random $h_1, h_2, ..., h_U, u_0, u_1, u_2, ..., u_n \in \mathbb{G}_1$. We use $u_0, u_1, u_2, ..., u_n$ as the Waters' hash function [23]. The master key is g^α and a generator X_3 of $\mathbb{G}_{p_3}$, while the public parameters are $params = (g, g^a, e(g,g)^\alpha, h_1, h_2, ..., h_U, u_0, u_1, u_2, ..., u_n)$.

- **Extract** The private key for a user with attributes $S \subseteq \mathbb{U}$ is generated as follows:
  1. The attribute authority randomly chooses $t \in \mathbb{Z}_p$, $R_0, R'_0 \in G_{p_3}$, then computes $K = g^\alpha g^{at} R_0$, $L = g^t R'_0$;
  2. For each attribute $x \in S$, the attribute authority randomly chooses $R_x \in G_{p_3}$, and computes $K_x = h_x^t R_x$;
  3. Finally, the attribute authority outputs the private key: $SK_S = (K, L, \{K_x\}_{x \in S})$.
- **Sign:** Let the signing predicate Υ can be represented by an LSSS access structure (M, ρ), that is, M is an $\ell \times k$ matrix, and ρ is an injective function that associates rows of M to attributes. The signature for the message $m = (\mu_1, \mu_2, ..., \mu_n) \in \{0,1\}^n$ is constructed as follows:
  1. The signer should first blind his private key SK_S as follows: The signer chooses a random $t' \in \mathbb{Z}_p$, and sets $K' = Kg^{at'} = g^\alpha g^{a(t+t')} R_0, L' = Lg^{t'} = g^{t+t'} R'_0, \forall x \in S : K'_x = K_x h_x^{t'} = h_x^{(t+t')} R_x$. Then the signer's new private key $SK'_S = (K', L', \{K'_x\}_{x \in S})$.
  2. As the signer's attribute set S should satisfy the signing predicate Υ, that is $\Upsilon(S) = 1$, then the signer can find $\overrightarrow{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_\ell)$ that satisfies $\overrightarrow{\alpha} M = (1, 0, ..., 0)$. In additional, the signer find another random $\overrightarrow{\beta} = (\beta_1, \beta_2, ..., \beta_\ell)$ that satisfies $\overrightarrow{\beta} M = (0, 0, ..., 0)$. For the existence of $\overrightarrow{\beta}$, we have a discussion in Section 5.2.
  3. For each $i \in [1, \ell]$, the signer computes $s_i = (L')^{\alpha_i} g^{\beta_i}$, and sets $y = \prod_{i=1}^{\ell} ((K'_{\rho(i)})^{\alpha_i} (h_{\rho(i)})^{\beta_i})$. Then, the signer random chooses $r \in \mathbb{Z}_p$ and computes:
     $\sigma_1 = \ yK'(u_0 \prod_{j=1}^{n} (u_j)^{\mu_j})^r)\ , \sigma_2 = g^r$.
  4. Finally, the signer outputs the signature $\sigma = (s_1, ..., s_\ell, \sigma_1, \sigma_2)$.
- **Verify:** Given a signature $\sigma = (s_1, ...s_\ell, \sigma_1, \sigma_2)$ of a message $m = (\mu_1, \mu_2, ..., \mu_n) \in \{0,1\}^n$ for the predicate Υ (corresponding to $(M_{\ell \times k}, \rho)$), firstly, the verifier chooses randomly $v_1 = 1, v_2, ..., v_k$ from $\mathbb{Z}_p$, sets $\overrightarrow{v} = (1, v_2, ..., v_k)$ and computes $\lambda_i = \sum_{j=1}^{k} v_j M_{i,j}$, where M_i is the vector corresponding to the ith row of matrix M. The verifier accepts the signature if and only if the following equation holds, otherwise reject it.

$$e(g,g)^\alpha e((u_0 \prod\nolimits_{j=1}^{n} (u_j)^{\mu_j}), \sigma_2) \prod_{i=1}^{\ell} e(g^{a\lambda_i} h_{\rho(i)}, s_i) \stackrel{?}{=} e(g, \sigma_1).$$