An online/offline distributed attribute-based authentication scheme

Xin Liu1,2*, Man Guo3, Xuzhou Li1,2 and Bin Zhang1,2

1 School of Information Engineering, Shandong Youth University of Political Science, Jinan, Shandong, 250013, China
2 Key Laboratory of Information Security and Intelligent Control in Universities of Shandong (Shandong Youth University of Political Science), Shandong Youth University of Political Science, Jinan, Shandong, 250013, China
3 Public Course Teaching Department, Shandong University of Arts & Crafts, Jinan, Shandong, 250300, China
*Corresponding author’s e-mail: lxonne@163.com

Abstract. Attribute-based authentication is an effective cryptography mechanism, which makes it possible for service providers to implement fine-grained access control on cloud resources. Although many attribute-based authentication schemes have been proposed, most of them only support single attribute authority and users have to perform a large amount of computation in the authentication phase. By extending the ciphertext policy attribute-based encryption scheme of Rouselakis et al., a distributed attribute-based authentication scheme was designed. The feature of the new scheme is to optimize the online computation efficiency of both service providers and users, i.e., by introducing the technique of online/offline attribute-based encryption, the online computational burden of semi-trusted servers is greatly reduced. On the other hand, by introducing outsourcing decryption, users’ computation in the authentication stage is independent of the size of underlying access structures. Compared with previous schemes, the new scheme satisfies several ideal properties, that is, introducing distributed authorities, supporting outsourcing computation, satisfying anonymity and unlikability, and so on.

1. Introduction

With the development of cloud computing technology, more and more data and electronic resources are stored in the “cloud”. Service Provider (SP) is responsible for the maintenance of cloud servers and ensures that only legitimate users can access online resources through the authentication mechanism. However, traditional public key based anonymous authentication cannot meet the demands of SP to enforce fine-grained access control. Attribute-based Authentication (ABA) is a promising cryptography mechanism, which allows users to perform anonymous authentication with SP by using their attribute keys, and make SP be sure that their own attributes can satisfy the pre-specified predicate (also called as access structure).

In [1], Li et al. proposed an Attribute-based Signature (ABS) scheme supporting threshold predicate, and constructed a multi-authority ABA scheme based on it. In [2], Shahandashti et al put forward the concept of threshold ABS (t-ABS), and constructed a secure threshold ABA scheme in the universal composability framework. Lian et al. [3] designed an adaptive ABA scheme based on ABS. Li
et al [4] proposed a multi-authority ABA scheme. However, when some attribute authority needs to be added or removed, the remaining attribute authorities need to re-execute the setup protocol. At the same time, users need to re-execute the key generation protocol with each attribute authority. In [5, 6], Yang et al. proposed two ABA schemes respectively. The first scheme was obtained by combining the technique of Ciphertext Policy Attribute-based Encryption (CP-ABE) scheme with group signature. Its characteristic is that in the authentication phase, SP defines a one-time attribute tree as the access structure. The second scheme was constructed on attribute-based group signature [7], which has the advantage that when the underlying attribute tree changes, users do not need to apply for attribute credentials again. In [8], Kaaniche et al. compared the security model of Anonymous Credential (AC) and ABS, and proposed a general method to construct AC scheme based on ABS. Recently, Liu et al. [9] put forward a k-times ABA scheme, which supports Linear Secret Sharing Scheme (LSSS) access policies, and can be deployed on trusted computing platforms. As a summary, most of previous ABA schemes have the following limitations: (1) In the authentication phase, user’s computation cost grows with the size of attribute-based access policy. (2) There is only one attribute authority, that is, a single point of failure cannot be overcome.

1.1. Related work
In the design of an ABA scheme, there is a knotty problem, that is, the calculation efficiency of the underlying ABS or ABE scheme is not satisfactory, which means that users’ computation cost depend on the size of predicates. In [10], Green et al. proposed an outsourcing decryption method for ABE schemes. With this technique, a user can outsource a ciphertext to a semi-trusted server, which is responsible for converting it into a short ElGamal type ciphertext, thus transforming the complex calculation problem of decrypting an ABE ciphertext into the simple calculation problem of decrypting an ElGamal type ciphertext. Later, Lai et al. [11] pointed out that the outsourcing decryption mechanism of Green et al. could not ensure a server to perform the ciphertext conversion honestly. Therefore, they put forward a revised security model for ABE and a specific scheme to meet the verifiability of outsourcing decryption. However, their scheme is still unsatisfactory. Specifically, the computation of users in the encryption and decryption phase is still large, and the ciphertext length of their scheme is twice that of a standard ABE scheme. Therefore, many researches focus on how to improve the efficiency of outsourcing decryption and how to verify the correctness of the results of outsourcing decryption (such as [12]). Similar to ABE schemes, ABS schemes also have the disadvantage that the computation of sign and verification phase is linearly dependent on the claimed predicate. Cui et al. [13] proposed the concept of server-aided ABS, which allows the main computing tasks of a signer / verifier to be outsourced to a server. In addition, bilinear pairing operation is also a heavy task for computation constrained devices. Therefore, it is also considered to use techniques such as pair delegation protocol to help users avoid pairing operations (for example, [14]) during the authentication phase.

1.2. Contribution
Inspired on the idea of Li et al. [15], we combined Shamir’s threshold secret sharing scheme with the CP-ABE of Rouselakis et al. [16], and constructed an efficient distributed ABA scheme. In order to reduce the computation burden of SP in the authentication stage, the generation process of ciphertext is divided into two stages by using the technique of Hohenberger et al. [17], i.e. generating intermediate ciphertexts in the offline stage and synthesizing the challenge ciphertext in the online stage. The new scheme allows users to outsource their decryption task of the challenge ciphertext to a server, and uses the technique of Lin et al. [12] to verify the returned decryption result. In addition, in the authentication stage, users need to prove to SP that they have legal member credentials. We use the technique in [18] to realize this proof, such that users can avoid performing bilinear pairing operations in the construction of the proof.
1.3. Organization
In Section 2, we introduce the technical preliminaries. In Section 3, we propose a syntax definition of
distributed ABA schemes. In Section 4, we provide the description of the new scheme. Next, we make
security analysis and performance analysis in Section 5 and 6 respectively. We conclude in Section 7.

2. Preliminaries

2.1. Bilinear pairing
Let \( \mathbb{G} = \langle g \rangle, \mathbb{G}_T \) denote the cyclic multiplicative group of order \( p \), which is a prime number. In
addition, define a bilinear mapping \( \hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T \). The mapping satisfies the following properties:

1) Non-degenerate, i.e., \( \hat{e}(g, g) \neq 1_{\mathbb{G}_T} \). (2) Bilinear, that is, for all elements \( a, b \in \mathbb{Z}_p \), the equation
\( \hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab} \) holds. We call \((\mathbb{G}, \mathbb{G}_T)\) a bilinear group pair.

2.2. Online / offline attribute-based encryption
In [17], Hohenberger et al. proposed the formal definition of an online/offline ABE scheme. Compared with standard ABE schemes, this scheme divides the execution process of encryption
algorithm Encrypt() into two stages. In the offline phase, a server executes the Offline.Encrypt() algorithm to generate an intermediate ciphertext \( ICT \). In the online phase, the server generates the final
ciphertext \( CT \) with \( ICT \) and access structure \( A \), by invoking the Online.Encrypt() algorithm. The
advantage of this scheme is that most of the computation tasks in the generation of the ciphertext are
scheduled to be completed in the offline phase, thus alleviating the computation burden of the server
in the online phase.

2.3. Shamir’s threshold secret sharing scheme
Shamir’s secret sharing scheme provides a method for \( n \) participants to generate a key \( K \), so that any
\( t \leq n \) participants (e.g., \( P_1, ..., P_t \)) can cooperate to recover \( K \), but any \( t-1 \) participants cannot be able to
do this [19]. The details of the scheme is as follows: for \( i = 1, ..., n \), a dealer \( \mathcal{D} \) selects \( x_i \in \mathbb{Z}_p \) and
assigns \( x_i \) as the exclusive public identity of \( P_i \). \( \mathcal{D} \) constructs a polynomial \( f(x) \) of degree \( t-1 \), so that
\( K = f(0) \). Then, \( \mathcal{D} \) gets the secret share \( s_i = f(x_i) \) of \( K \) for each \( P_i \). Finally, any \( t \) participants
(assumed to be \( P_1, ..., P_t \)) can use the following methods for secret reconstruction, i.e.,
\[ K = \sum_{i=1}^{t} \Delta_i s_i \mod p \], where \( \Delta_i \) is called the Lagrange coefficient.

2.4. Key derivation function
The key derivation function \( KDF() \) is used to help users generate their private keys [12] that satisfy
the pseudo randomness. In other words, assuming that the output length of the function \( KDF() \) is \( l \)
bits (i.e., \( KDF(DK, l) \)), for any probabilistic polynomial time adversary, the output of the function is
indistinguishable from a randomly selected string \( R \) in \( \{0,1\}^l \).

3. Syntax of distributed ABA schemes
A distributed attribute-based authentication scheme consists of the following algorithms/protocols:

- **Setup.** The algorithm is performed by a central authority \( \mathcal{CA} \). The algorithm takes as input a
security parameter \( \kappa \) and an attribute universe \( \mathcal{U} \), and returns the system public parameter \( \text{params} \) and \( \mathcal{CA} ’s \) private key \( sk_{\mathcal{CA}} \).

- **AASetup.** This is a protocol executed by an attribute authority \( \mathcal{AA} \) and \( \mathcal{CA} \). The executive
process of the protocol is divided into two stages. In phase I, \( \mathcal{AA} \) obtains the identity identifier \( \text{aid}_i \)}
and the credential \( cred_i \) provided by \( CA \). In phase II, \( AA_i \) and \( CA \) jointly generate the master key \( MK \). Finally, \( AA_i \) outputs their secret share \( sk_i \) of \( MK \).

**-URegistration.** This is a protocol executed by a user \( U \) and \( CA \). \( U \) takes as input \( params \), \( CA \) takes as input \( sk_{CA} \). Finally, \( U \) outputs his/her public/private key pair \( (PK_U, SK_U) \) and member credential \( cred_U = (\alpha, e) \).

**-AttKeyGen.** This is a protocol executed by \( U \) and \( AA_i \). \( U \) takes as input \( params, (PK_U, SK_U), cert_U, AA_i \) takes as input \( params, sk_i \). Finally, \( U \) outputs his/her private key share \( SK_{S_i} \) of attribute set \( S \) from \( AA_i \). After collecting \( t \) shares, \( U \) composes his/her private key \( SK_S \).

**-AttAuthentication.** This is a protocol executed by \( U \) and \( SP \). The executive process of the protocol is divided into two stages. In phase I, \( SP \) generates an intermediate ciphertext \( ICT \). In phase II, \( U \) interacts with \( SP \). \( U \) takes as input \( params, (PK_U, SK_U), cert_U, SK_S \), \( SP \) takes as input \( params, ICT, A \). Finally, if the protocol succeeds, \( SP \) accepts \( U \)'s authentication.

**-GenTK.** The algorithm takes as input \( SK_S \), and returns \( SSTK, RK \). Following the notations in [10], we call \( STK \) as the transmission key and \( RK \) as the retrieving key.

**-outTransform.** The algorithm takes as input \( STK, CT \), and returns a partially decrypted ciphertext \( PCT \).

**-outDecrypt.** The algorithm takes as input \( CT, PCT, RK \), and returns the plaintext \( nonce \).

4. The new scheme

4.1. High level description

In the Setup phase, \( CA \) generates the public system parameter \( params \), in which an element \( Y = \hat{e}(g, g)^\alpha \) is generated by \( n \) attribute authorities using Shamir’s secret sharing scheme. \( \alpha \) can be regarded as the master key of the Rouelakis-Waters CP-ABE Scheme [16] (RW scheme for short). User \( U \) first executes the URegistration protocol with \( CA \) to obtain his/her member credential \( cred_U \). In the AttKeyGen protocol, only by showing his/her member credential to attribute authorities can \( U \) obtain the qualification to apply for his/her attribute key share \( SK_{S_i} \). It should be noted that \( SK_S \) is not generated by an attribute authority. Instead, \( U \) needs to execute the AttKeyGen protocol separately with \( t \) different authorities to obtain the attribute key share \( SK_{S_j} \) provided by \( AA_i \). In other words, \( SK_S \) is obtained by combining \( t \) key shares with the Lagrange interpolation formula. The AttAuthentication protocol can be divided into two stages. In phase I, \( SP \) generates an intermediate ciphertext \( ICT \) offline in advance. In phase II, \( SP \) defines a LSSS (Linear Secret Sharing Schemes) access structure \( A \) and generates the RW scheme ciphertext \( CT \) of a random number \( nonce \) based on \( ICT \). The condition for \( U \) to pass the authentication is that he can decrypt the ciphertext \( CT \) to get \( nonce \) and use this element to generate a non-interactive zero-knowledge proof \( \Pi_U^{[1]} \), which proves that he owns a valid member credential. For \( U \), he/she can perform the decryption process in two ways. Obviously, he/she can decrypt \( CT \) directly with \( SK_S \). As an alternative, \( U \) performs the GenTK algorithm to generate a transmission key \( TK_S \). Then, \( U \) outsources \( CT \) and \( TK_S \) to \( CS \). When the latter returns the partially decrypted ciphertext \( PCT \), \( U \) invokes algorithm Decrypt\textsubscript{out} to recover the plaintext \( nonce \) from \( PCT \). In this process, \( U \) should use the technique of Lin et al. [12] to verify the correctness of \( PCT \).
4.2. Detailed description

The operation of the new scheme is as follows:

- **Setup.** Taking a security parameter $\kappa$ as input, $\mathcal{CA}$ performs the following operations:
  
  (1) Defines bilinear group $(\mathbb{G}, \mathbb{G}_t, \mu, \hat{\cdot})$, where $\hat{\cdot} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_t$.
  
  (2) Selects $\bar{g}_0, \bar{g}_1, \bar{h} \in_r \mathbb{G}$, selects $\gamma \in_r \mathbb{Z}_p$, and sets $\overline{v} = \bar{h}^\gamma$. Selects $g, g_1, g_2, h, u, v, w \in_r \mathbb{G}$, defines an attribute universe $\mathbb{U} = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$, and sets $SK_{\mathcal{CA}} = \gamma$.
  
  (3) Sets the parameters of Shamir’s threshold secret sharing scheme to $(t, n)$.
  
  (4) Defines a collision-resistant hash function $\mathcal{H}_c : \{0, 1\}^t \rightarrow \mathbb{Z}_p^*$. 

- **AASetup.** $\mathcal{AA}$ and $\mathcal{CA}$ do the following steps:
  
  (1) $\mathcal{AA}$ selects $i, e \in_r \mathbb{Z}_p$, and calculates $\text{cred}_i = (\bar{g}_0 \cdot PK_i)^{\mathcal{H}_c(e)}$. $\mathcal{CA}$ then returns $\text{aid}_i, (\text{cred}_i, e)$ to $\mathcal{AA}$. If the equality $\hat{\cdot}(\text{cred}_i, \overline{v}) = \hat{\cdot}(\bar{g}_0 \cdot PK_i, \overline{v})$ is established, $\mathcal{AA}$ keeps the member credential $(\text{cred}_i, e)$ and the public/private key pair $(PK_i, SK_i)$, where $SK_i = x_i$.

- **URegistration.** $\mathcal{U}$ and $\mathcal{CA}$ do the following steps:
  
  (1) $\mathcal{U}$ selects $sk_u \in \mathbb{Z}_p$ and calculates $PK_u = g^{sk_u}$. $\mathcal{U}$ generates a zero-knowledge proof $\Pi_u^{\mathcal{U}} = PK\{x\} : PK_u = g^{sk_u}$. $\mathcal{U}$ sends $PK_u, \Pi_u^{\mathcal{U}}$ and a registration request to $\mathcal{CA}$.
  
  (2) If $\Pi_u^{\mathcal{U}}$ is valid, $\mathcal{CA}$ selects $e \in_r \mathbb{Z}_p$ and calculates $A = (\bar{g}_0 \cdot PK_u)^{\mathcal{H}_c(e)}$. $\mathcal{CA}$ generates a zero-knowledge proof $\pi_u^{\mathcal{CA}} = PK\{\gamma\} = A = (\bar{g}_0 \cdot PK_u)^{\mathcal{H}_c(e)}$ and returns $A, e, \pi_u^{\mathcal{CA}}$ to $\mathcal{U}$.
  
  (3) If $\pi_u^{\mathcal{CA}}$ is valid, $\mathcal{U}$ keeps the public/private key pair $(PK_u, SK_u)$ and the member credential $\text{cred}_u = (A, e)$.

- **AttKeyGen.** In order to apply for the attribute key, $\mathcal{U}$ needs to carry out the current protocol with any $t$ attribute authorities respectively. For simplicity, suppose $\mathcal{U}$ and $\mathcal{AA}$ do the following steps:
  
  (1) $\mathcal{U}$ generates a zero-knowledge proof $\Pi_u^i = PK\{(A, e), SK_u\} : A = (\bar{g}_0 \cdot g_i^{sk_u})^{\mathcal{H}_c(e)}$ and sends the attribute set $S$, his/her public key $PK_u$ and $\Pi_u^i$ to $\mathcal{AA}$.
  
  (2) If $\Pi_u^i$ is valid, $\mathcal{AA}$ selects $r_{j_1}, r_{j_2}, \ldots, r_{j_{|S|}} \in \mathbb{Z}_p$ and sets $K_{o_j} = g^r_{i_1}w_{i_2}^{r_{j_2}}, K_{j_3} = g^r_{i_3}$. For $i = 1, \ldots, |S|$, $\mathcal{AA}$ sets $K_{j, i} = g^{r_{j_1}}h^{r_{j_2}}\gamma^{r_{j_3}}$. Finally, $\mathcal{AA}$ returns $SK_{S_j} = (K_{o_j}, K_{j_3}, K_{j, i})$ to $\mathcal{U}$.
  
  (3) $\mathcal{U}$ sets $K_0 = \prod_{i=1}^{|S|} (K_{o_j})^{\hat{y}_i}, K_i = \prod_{i=1}^{|S|} (K_{j, i})^{\hat{y}_i}$. For $i = 1, \ldots, |S|$, $\mathcal{U}$ sets $K_{j_2} = \prod_{i=1}^{|S|} (K_{j, i})^{\hat{y}_i}$.
  
  (4) $\mathcal{U}$ keeps $SK_S = (K_0, K_{j_2}, K_{j, i})$.

- **AttAuthentication**(phase I). SP does the following steps:
  
  (1) Selects $s \in \mathbb{Z}_p$ and sets $DK = Y^s = \hat{\cdot}(g, g)^s, C_0 = g^s$.
  
  (2) For $j = 1, \ldots, P$, selects $\lambda_j, \lambda_j, \gamma_j, t_j, e_j \in \mathbb{Z}_p$ and sets $C_{j, 1} = w^{t_j}\gamma_j, C_{j, 2} = (u^j h)^{\gamma_j}, C_{j, 3} = g^{t_j}$.
  
  (3) Sets an intermediate ciphertext $ICT = \text{key}, s, C_0, (\lambda_j, \lambda_j, \gamma_j, t_j, e_j, C_{j, 1}, C_{j, 2}, C_{j, 3})_{j=1,\ldots,P}$.

- **AttAuthentication**(phase II). $\mathcal{U}$ and SP do the following steps:
  
  (1) $\mathcal{U}$ sends an authentication request $\text{Req}$ to $\text{SP}$.
(2) $SP$ defines an access structure $A = (M, \rho)$, where $M$ represents a matrix with $l$ rows and $n$ columns and $\rho$ is a mapping function, s.t., $\rho : M \rightarrow A_{\rho(i)}$, where $l \leq P$. $SP$ selects $y_1, ..., y_n \in \mathbb{Z}_p$ and sets $\tilde{v} = (s, y_2, ..., y_n)^T, \tilde{\lambda} = M\tilde{v} = (\lambda_1, ..., \lambda_l)^T$.

(3) For $j = 1, ..., l$, $SP$ sets $C_{i,j} = \lambda_j - \lambda_{j-1}, C_{j,j} = t_j(x_j - \rho(j))$.

(4) $SP$ selects $\text{nonce}, r \in \{0,1\}^*$ and sets $C_1 = g_{1}^{\text{nonce}} \cdot g_{2}^{\rho(1)}$, $C_2 = (\text{nonce} \parallel r) \parallel KDF(DK, I)$. Then, $SP$ sends the challenge ciphertext $CT = (A, C_0, C_1, C_2, \{C_{j,1}, ..., C_{j,l}\}_{j\in[1,l]})$ to $U$.

(5) $U$ parses $CT$ into $CT = (A, C_0, C_1, C_2, \{C_{j,1}, ..., C_{j,l}\}_{j\in[1,l]})$. If $S \neq A$, $U$ terminates the protocol. Let $I$ represent a set $I = \{i : \rho(i) \in S\}$. $U$ calculates a set of constants $\{\omega_i\}_{i\in I} \in \mathbb{Z}_p^*$, such that $\sum_{i \in I} \omega_i M_i = (1,0,...,0)$. $U$ computes

$$DK = e(w^{\sum_{i \in I} \omega_i c_{i,0}}, K_1) \cdot \prod_{i \in I} e(C_{i,1}, K_1) \cdot e(C_{i,2}, u^{C_{i,5}}, K_{i,2}) \cdot e(C_{i,3}, K_{i,3})$$

and calculates $C_2 \oplus KDF(DK, I) = \text{nonce} \parallel r$.

(6) If the equation $g_1^{\text{nonce}} \cdot g_2^{\rho(c)} = C_1$ holds, $U$ generates a zero-knowledge proof $\Pi_{i}^U = \text{PK} \{A, e, SK_{i,j} : A = (g_0 \cdot g_{\text{nonce}})^{C_{1,0}}\}$ and provides $\Pi_{i}^U$ to $SP$.

The specific generation process of $\Pi_{i}^U$ is as follows:

(a) $U$ selects $c \in \mathbb{Z}_p$ and calculates $A' = A^c, \tilde{A} = g_0 \cdot PK_{i,j} A^{-c}$. $U$ selects $\rho_1, \rho_2, \rho_3 \in \mathbb{Z}_p$ and calculates $R = g_0^c g_{\text{nonce}} A^{\rho_3}$.

(b) $U$ sets $c = H_{i}(A' \parallel \tilde{A} \parallel R \parallel \text{nonce})$ and calculates $\xi_1 = \rho_1 + c \cdot r \mod p, \xi_2 = \rho_2 + c \cdot SK_{i,j} \cdot r \mod p, \xi_3 = \rho_3 + c \cdot e \mod p$.

(c) $U$ sets $\Pi_{i}^U = (A', \tilde{A}, c, \xi_1, \xi_2, \xi_3)$.

(7) If $\Pi_{i}^U$ is valid, $SP$ allows $U$ to access the desired cloud resources.

The verification process of $\Pi_{i}^U$ is as follows:

(a) $SP$ parses $\Pi_{i}^U$ into $\Pi_{i}^V = (A^c, \tilde{A}, c, \xi_1, \xi_2, \xi_3)$ and calculates $\tilde{R} = g_0^c g_{\text{nonce}} A^{\rho_3} \tilde{A}^{-c}$.

(b) $SP$ accepts $\Pi_{i}^U$ on the condition that the following relationships are satisfied simultaneously, i.e., (i) $c = H_{i}(A' \parallel \tilde{A} \parallel \tilde{R} \parallel \text{nonce}), (ii) \hat{e}(\tilde{A}, h) = \hat{e}(A', w), (iii) A' \neq 1_{\mathbb{G}}$.

- GenTK. Taking $SK_x$ as the input, the algorithm performs as follows:

(1) Parses $SK_x$ into $SK_x = (K_0, K_1, \{K_{i,2}, K_{i,3}\}_{i\in[l]})$.

(2) Selects $z \in \mathbb{Z}_p^*$ and calculates $K'_0 = K_{i,2}, K'_i = K_{i,3}$. For $i = 1, ..., |S|$, calculates $K'_{i,2} = K_{i,2}, K'_{i,3} = K_{i,3}$.

(3) Outputs $TK_x = (K'_0, K'_1, \{K'_{i,2}, K'_{i,3}\}_{i\in[l]}), RK_x = z$.

- Transform out. Taking $(PK, TK_x, CT)$ as the input, the algorithm performs as follows:

(1) If $S \neq A$, aborts. Else, parses $CT$ into $CT = (A, C_0, C_1, C_2, \{C_{j,1}, ..., C_{j,l}\}_{j\in[1,l]})$.

(2) Let $I$ represent a set $I = \{i : \rho(i) \in S\}$. Calculates a set of constants $\{\omega_i\}_{i\in I} \in \mathbb{Z}_p^*$, such that $\sum_{i \in I} \omega_i M_i = (1,0,...,0)$, and calculates

$$PCT = \frac{e(C_{0,0}, K'_0)}{e(w^{\sum_{i \in I} \omega_i c_{i,0}}, K'_1) \cdot \prod_{i \in I} e(C_{i,1}, K'_1) \cdot e(C_{i,2}, u^{C_{i,5}}, K'_{i,2}) \cdot e(C_{i,3}, K'_{i,3})}^{\omega_i}$$

(3) Outputs $PCT$. 

6
Taking \((CT, PCT, RK_s)\) as the input, the algorithm performs as follows:

1. Parses \(CT\) into \(CT = (A, C_0, C_1, C_2, \{C_{j,1}, \ldots, C_{j,l}\})\).
2. Calculates \(DK' = PCT^{DK}, nonce' \mid r' = C_2 \otimes KDF(DK', l)\).
3. Checks if \(g_1^{\hat{h}(nonce')}, g_2^{\hat{h}(r')} = C_1\). If so, outputs \(nonce'\), otherwise, outputs \(\perp\).

### 5. Security analysis

In this section, we provide the security analysis of the new scheme.

**Correctness.** We divide the proof of correctness into the proof of the following four facts.

**Fact 1:** In the AASetup protocol, the master key \(MK = \alpha\) jointly generated by \(CA\) and \(AA_1, \ldots, AA_n\) is exactly the master key of the RW scheme. This fact can be derived from the correctness of the underlying Shamir’s secret sharing scheme.

**Fact 2:** \(U\) obtains the set of attribute key shares \(\{SK_{x,s}\}_{s=1}^t\) by executing the AttKeyGen protocol with \(AA_1, \ldots, AA_n\). With this set, \(U\) can generate the attribute key \(SK\) of the RW scheme by itself. This fact can also be derived from the correctness of the underlying Shamir’s secret sharing scheme.

Specifically, assume that the attribute key of \(U\) in the RW scheme is \(g^{w'}, g^z, (u^kh)^{v^r}\). Now, suppose \(U\) has the set of attribute key shares \(\{SK_{x,s}\}_{s=1}^t\), where \(SK_{x,s} = (K_0, K_{ij}, \{K_{ij,1}, K_{ij,3}\})\). To generate \(SK\), \(U\) sets
\[
K_0 = \prod_{i=1}^l (K_{0,i})^h = \prod_{i=1}^l (g^{ah})^h = g^{\hat{h}(ah)},
K_i = \prod_{i=1}^l (K_{ij,i})^h = \prod_{i=1}^l (g^{z_i})^h = g^{zh},
K_s = \prod_{i=1}^l (K_{ij,i})^h = \prod_{i=1}^l (u^kh)^{v^r} = (u^kh)^{\hat{h}(v^r)}.
\]

**Fact 3:** In the AttAuthentication protocol, \(U\) can decrypt the challenge ciphertext \(CT\) directly, and recover the random number \(nonce\). The reason is that it’s easy to prove that the decryption equation
\[
CK = e(C_0, K_0)\prod_{i=1}^l e(C_{i,1}, K_i) e(C_{i,2}, u^{C_{i,3}}, K_{i,2}) e(C_{i,3}, K_{i,3})^\alpha
\]
is equivalent to the decryption equation of the RW scheme, i.e.,
\[
DK = e(C_0, K_0)\prod_{i=1}^l e(C_{i,1}, K_i) e(C_{i,2}, u^{C_{i,3}}, K_{i,2}) e(C_{i,3}, K_{i,3})^\alpha
\]

**Fact 4:** In the AttAuthentication protocol, \(U\) can also outsource the challenge ciphertext \(CT\) to recover the random number \(nonce\) by help of a server. The reason is that, it’s easy to prove
\[
PCT = e(C_0, K_0') \prod_{i=1}^l e(C_{i,1}, K_i') e(C_{i,2}, u^{C_{i,3}}, K_{i,2}') e(C_{i,3}, K_{i,3}')^\alpha = DK^{1/2}
\]

Therefore, \(U\) can firstly calculate \(DK = PCT\), and then recover the plaintext \(nonce\) by performing the exclusive-or operation on \(C_0\) and \(KDF(DK, l)\).

**Ciphertext security.** The CP-ABE scheme (OORW scheme for short) used in this scheme is obtained by modifying the RW scheme as follows: (1) The master key \(MK = \alpha\) is generated by \(AA_1, \ldots, AA_n\) cooperately. (2) The ciphertext generation process is divided into online stage and offline stage. However, the above changes will not affect the security of the underlying RW scheme. It can be
proved that if the RW scheme satisfies the selective CPA-security, the OORW scheme also satisfies this property. Specifically, we can define a selective-CPA security experiment (Experiment 1) executed by a reduction algorithm \( \mathcal{B} \) and an adversary \( \mathcal{A} \). At the same time, \( \mathcal{B} \) and a challenger \( \mathcal{C} \) carries out the selective-CPA security experiment of the RW scheme (Experiment 2). In Experiment 1, \( \mathcal{B} \) receives the public key of the OORW scheme provided by \( \mathcal{C} \), and sends it to \( \mathcal{A} \). Moreover, \( \mathcal{B} \) hides the details of \( AA \), generating secret key \( sk \) and the master key \( MK \) from \( A \). In the challenge phase of Experiment 1, when \( A \) submits the challenge messages \( m_0, m_1 \), \( B \) sends them to \( C \). Lastly, \( B \) receives \( CT'_{RW} \), which is the challenge ciphertext of the RW scheme returned by \( C \). Now, \( \mathcal{B} \) can simulate the challenge ciphertext \( CT \) of the OORW scheme by using \( CT'_{RW} \). Obviously, if \( A \) can break through the selective-CPA security of the OORW scheme, \( B \) can also break through the selective-CPA security of the RW scheme.

**Soundness.** As long as \( A \) cannot corrupt \( t \) attribute authorities, it cannot decrypt the challenge ciphertext \( CT \). So \( A \) cannot pass the authentication process with \( SP \).

**Completeness.** As long as more than \( t \) attribute authorities are not corrupted and remain online, the new system can perform normally.

**Anonymity & Unlinkability.** In the AttAuthentication protocol, \( SP \) can only obtain a transcript of the AttAuthentication protocol \((\mathcal{A}', \mathcal{A}', \pi'_U)\). In view of \( SP \), elements \( \mathcal{A}', \mathcal{A}' \) have the same distribution with the random elements in group \( G \). According to the simulatability of \( \pi'_U \), \( SP \) cannot extract the identity information of \( U \) from \( \pi'_U \), nor can he/she correlate the previous authentication behavior of \( U \).

### 6. Performance analysis

In Table 1, we compare the main properties of this scheme with several existing schemes. First of all, except for the new scheme and the scheme in [1, 4], the remaining schemes only consider the case of one central attribute authority. In the authentication phase, only the new scheme and the scheme in [9] allow users to outsource their computing tasks to cloud server. In the Authenticate phase, each scheme adopts different access policies. Compared with threshold access policy, LSSS policy and attribute tree based policy are more flexible. It should be noted that ABS and ABE are the core mechanisms to realize attribute authentication. In this paper, the new scheme was constructed by using the technique of online/offline ABE (OOABE), the scheme in [9] was constructed based on ABE, and other schemes are constructed by using ABS or Attribute-based Group Signature (ABGS).

| Scheme | Number of AA | Outsourcing Computation | Type of Access Policy | Core Mechanism |
|--------|--------------|--------------------------|-----------------------|---------------|
| [1]    | multiple     | N                        | threshold             | ABS           |
| [2]    | single       | N                        | threshold             | ABS           |
| [3]    | single       | N                        | LSSS                  | ABS           |
| [4]    | multiple     | N                        | threshold             | ABS           |
| [5]    | single       | N                        | attribute tree        | ABS           |
| [6]    | single       | N                        | attribute tree        | ABGS          |
| [8]    | single       | N                        | LSSS                  | ABS           |
| [9]    | single       | Y                        | LSSS                  | ABE           |
| Ours   | multiple     | Y                        | LSSS                  | OOABE         |

Next, we analyze the operation efficiency of \( SP \) and \( U \) in the AttAuthentication protocol of the new scheme. It is assumed that \( SP \) generates an intermediate ciphertext \( ICT \) in advance through the
offline phase of the AttAuthentication protocol. At the same time, \( \mathcal{U} \) invokes the GenTK algorithm in advance to generate \( TK_x, RK_x \). We use the symbols \( Exp(\mathbb{G}) \) and \( \mathbb{P} \) to represent one exponential operation in group \( \mathbb{G} \) and one bilinear pairing operation respectively. It is easy to conclude that the online operation of \( SP \) and \( \mathcal{U} \) is about \( 2Exp(\mathbb{G}) + 2\mathbb{P} \) and \( 3Exp(\mathbb{G}) \) respectively. Obviously, their online computing costs are independent of the size of the attribute-based access policy.

7. Conclusion
This paper proposes an online/offline distributed ABA scheme for cloud environment. The feature of the new scheme is that multiple distributed attribute authorities are responsible for maintaining the attribute universe, thus overcoming the drawback of a single point of failure. Moreover, the new scheme optimizes the online computing efficiency of users and service providers respectively. In future research, we will define a formal security model of distributed ABA schemes, provide more rigorous security analysis, design a lightweight ABA scheme suitable for IoT environment, etc.

Acknowledgments
This work was supported in part by the Project of Shandong Province Higher Educational Science and Technology Program under Grant J17KA081, in part by the Undergraduate Education Reform Project of Higher Education in Shandong Province under Grant M2018X245, in part by the Scientific Research Project of Shandong Youth University of Political Science under Grant XJPY2021, and in part by the Project of Educational Reform and Research in Shandong Youth University of Political Science under Grant JG201905.

References
[1] Li, J., Au, M. H., Susilo, W. (2010) Attribute-based signature and its applications. In: Proceedings of ASIA CCS 2010. Beijing. pp. 60-69.
[2] Shahandashti, S. F., Safavi-Naini, R. (2009) Threshold attribute-based signatures and their application to anonymous credential systems. In: Proceedings of AFRICACRYPT 2009. Gammarth. pp.198-216.
[3] Lian, Y., Huang, X., Mu, Y., et al. (2014) SA 3: Self-adaptive anonymous authentication for dynamic authentication policies. Future Generation Computer Systems, 30: 133-139.
[4] Li, J., Chen, X., Huang, X. (2015) New attribute-based authentication and its application in anonymous cloud access service. International Journal of Web and Grid Services, 11: 125-141.
[5] Yang, H., Oleshchuk, V. A. (2015) An efficient traceable attribute-based authentication scheme with one-time attribute trees. In: Proceedings of NordSec 2015. Stockholm. pp. 123-135.
[6] Yang, H., Oleshchuk, V. A. (2015) A dynamic attribute-based authentication scheme. In: Proceedings of C2SI 2015. Rabat. pp. 106-118.
[7] Emura, K., Miyaji, A., Omote, K. (2009) A dynamic attribute-based group signature scheme and its application in an anonymous survey for the collection of attribute statistics. Information and Media Technologies, 44:1060-1075.
[8] Kaaniche, N., Laurent, M. (2016) Attribute-based signatures for supporting anonymous certification. In: Proceedings of ESORICS 2016. Heraklion. pp. 279-300.
[9] Liu, X., Xu, Q., Zhang, B., et al (2018) k-times attribute-based authentication scheme using direct anonymous attestation. Journal on Communications, 2018, 39: 113-133.
[10] Green, M., Hohenberger, S., Waters, B. (2011) Outsourcing the decryption of abe ciphertexts. In: Proceedings of SEC 11, San Francisco. pp. 34-34.
[11] Lai, J., Deng, R. H., Guan, C., et al. (2013) Attribute-based encryption with verifiable outsourced decryption. IEEE Transactions on information forensics and security, 8: 1343-1354.
[12] Lin, S., Zhang, R., Ma, H., et al. (2015) Revisiting attribute-based encryption with verifiable outsourced decryption. IEEE Transactions on Information Forensics and Security, 10: 2119-2130.

[13] Cui, H., Deng, R. H., Liu, J. H., et al. (2018) Server-aided attribute-based signature with revocation for resource-constrained industrial-internet-of-things devices. IEEE Transactions on Industrial Informatics, 14: 3724-3732.

[14] Liu, X., Xu, Q., Zhang, B. (2016) Lightweight multi-coupon system for multi-merchant environments with DAA. Journal on Communications, 37: 30-45.

[15] Li, W., Xue, K., Xue, Y. (2016) TMACS: a robust and verifiable threshold multi-authority access control system in public cloud storage. IEEE Transactions on Parallel and Distributed Systems, 27:1484-1496.

[16] Rouselakis, Y., Waters, B. (2013) Practical constructions and new proof methods for large universe attribute-based encryption. In: Proceedings of ACM CCS 2013. Berlin. pp. 463-474.

[17] Hohenberger, S., Waters, B. (2014) Online/offline attribute-based encryption. In: Proceedings of PKC 2014. Buenos Aires, pp. 293-310.

[18] Camenisch, J., Drijvers, M., Hajny, J. (2016) Scalable revocation scheme for anonymous credentials based on n-times unlinkable proofs. In: Proceedings of ACM WPES 2016, Vienna. pp. 123-133.

[19] Stinson, D. R., Paterson, M. (2018) Cryptography theory and practice (4th edition). CRC Press, Boca Raton.