Anonymous decentralized attribute-based access control for cloud-assisted IoT

Hassan Nasiraee, Maede Ashouri-Talouki

PII: S0167-739X(19)32238-1
DOI: https://doi.org/10.1016/j.future.2020.04.011
Reference: FUTURE 5581

To appear in: *Future Generation Computer Systems*

Received date : 23 August 2019
Revised date : 5 April 2020
Accepted date : 6 April 2020

Anonymous Decentralized Attribute-based Access Control for Cloud-assisted IoT

Hassan Nasiraee , Maede Ashouri-Talouki*

Abstract

Attribute-Based Encryption (ABE) has emerged as powerful cryptographic tools to bring fine-grained access control with widespread applications such as Cloud-assisted IoT data sharing. Subsequently, decentralized ABE with untrusted attribute authorities is proposed to remove the online Trusted Authority (TA). In the decentralized architecture, a user as a data customer (e.g., IoT-device) submits his attributes to the untrusted authorities to get the private keys. In the architecture, user's privacy, against the untrusted authorities, is a significant challenge that must be ensured (e.g., *E-health Cloud* application). In this paper, we address the privacy issue in the decentralized ABE and propose a novel anonymous and decentralized attribute-based encryption in the standard model. It preserves the user's anonymity against the authorities in an efficient manner. In our solution, we use cryptographic accumulators to verify the user's attributes anonymously. Then, we include the accumulator in the ciphertext to ensure the ABE access control against unauthorized users.

Moreover, in some applications, access structures (encryption/decryption policy) include sensitive information and should be obfuscated from everyone minus the users whose secret key attributes meet the access structures. To ensure the hidden policy, we propose an efficient and decentralized policy *obfuscation technique* to preserve the privacy of the policy against the Public Cloud Server (PCS). It is exciting for a decentralized environment where the authori-

*Corresponding author.
H. Nasiraee and M. Ashouri-Talouki are with the IT Engineering Department of the University of Isfahan, Isfahan, Iran (e-mail: nasiraee@eng.ui.ac.ir, m.ashouri@eng.ui.ac.ir)

ties are untrusted and may collude with the PCS.

To be applicable for IoT resource-constrained devices, we outsource the expensive decryption computation over powerful Cloud servers. Then, we formally analyze the security properties of the proposed scheme and conduct experimental results to show its efficiency. Finally, we briefly explain how the features of the proposal meet the requirements of some real-world applications.

Keywords: Attribute-based encryption, decryption outsourcing, privacy, hidden policy, Cloud-assisted IoT.

1. Introduction

Cloud computing is a new computing paradigm that assists in many essential technologies. One of these is the Internet of Things (IoT), which connects resource-constrained devices. The widespread use of IoT-devices and the rise of cloud-assisted IoT information sharing applications require high necessities for data security, privacy protection, and access control. To ensure a secure fine-grained access control system along with preserving the privacy and anonymity issues, ABE is a powerful cryptographic tool for data sharing applications over untrusted cloud storage. However, ABE requires costly pairing operations, and the data customer user may be a resource-constrained device (such as an IoT device). Thus, crucially regarding the efficiency issue in any new scheme for such a decentralized IoT data sharing application is required.

In a typical Cloud-based data sharing scenario (Fig. 1), a Data Owner (DO) requests a public key from the attribute authority to encrypt the data. Then, the DO sends the ciphertext (CT) along with the Access Structure (AS), to the PCS to realize the fine-grained access control (Ciphertext Policy-ABE) [1]. Later, any data user (known as Data Customer DC), requests the private keys, associated with his attributes, from the authority to access the encrypted data. He decrypts the ciphertext if his set of attributes satisfy the access structure (known as the access policy) included in the CT.

However, in a real-world scenario, we may need to share data according to an access policy comprised of attributes or credentials distributed by various organizations which indicates a decentralized architecture. For example, we might want to share Electronic Health Records (EHRs) exclusively with a data customer who has the attribute of "Cardiologist" distributed by a medical organization and the attribute "Researcher" distributed by a clinical trial. On financial relevance, some companies such as Airbus and Siemens might both distribute attributes as part of a joint plan [2, 3, 4]. Then, many real-world applications require a decentralized and distributed approach. Moreover, a centralized architecture has the *key escrow problem*, due to its knowledge of all system secret parameters. Consequently, a decentralized ABE (DABE) with semi-honest authorities is more fascinated. In the following, we shortly review the main privacy-related challenges of such systems, along with a brief history of existing solutions. Then we entitle our contributions.

The User's Anonymity. In the decentralized architecture, there are N semi-honest and untrust authorities where each of them is responsible for a disjoint subset of the attribute universe. In this architecture, a user must present the same Global-IDentifier (GID) to each authority to obtain the private keys. Thus, it is straightforward for colluding authorities to pool their information and build a "complete profile" of all of the attributes corresponding to each GID, which leads to compromise of the user's privacy. This is undesirable, particularly if the user uses the ABE system in many different settings, and wishes to keep private the information about some of those settings. Note that in a centralized system, a single and online trusted authority knows all attributes and the GIDs, and then the user's anonymity is not an issue.

To ensure the user's anonymity, some previous works addressed the user's *identity-anonymity*, during the process of the private key requests from the authority. The first work to satisfy the user's anonymity is accomplished by Chase et al. [3] in 2009. They proposed an Anonymous DABE (ADABE) scheme without the trusted authority and addressed the user's GID-anonymity. The proposal was a breakthrough in the field, and many subsequent types of research have taken the idea. In their scheme, the user's privacy is provided by anonymizing the GID (*identity-anonymity*). But, as described in [5], the idea has a *privacy-leakage problem* against untrusted authorities. The *problem* is the *identity-leakage* due to the existence of many sensitive or individual attributes in real-world applications. In such applications, an authority can gradually build the user's profile and identify or guess the user's identity (GID) by gathering received attributes of the user. For more clarity, consider the given example of Fig. 1 and assume that we know that the Head of the Central Hospital is Mary Jackson. The attribute set of the two users are known as $User1$={Status=”Emergency”, Affiliation=”Central Hospital”, Dept.=”Repository”, SSN:”241-271-97” , Sex=”Male”} and $User2$={Position=”Head”, Affiliation=”Central Hospital”, Dept.=”Cardiologist”, Sex=”Female”}. Form this knowledge, it can be deduced that the attribute set User2 is Mary's attribute-set even if we are not aware of her GID [5]. Consequently, the *GID-anonymizing* idea (*identity-anonymity*) does not guarantee the user's anonymity against the authorities. Then, controlling the release of the individual and sensitive attributes is required.

To resolve the *identity-leakage problem*, and to answer the user's anonymity question, some works such as [5, 4, 6] have ensured *attribute-anonymity* instead of the *identity-anonymity*. Up to now, the main interesting approaches to provide the attribute-anonymity are Camenisch's membership-proof [7] (such as [5]) and Oblivious Transfer (OT) (such as [4, 6]). Han et al. [5] proposed an attractive anonymous and decentralized scheme in the standard model, which provides the user's *attribute-anonymity*. The system enjoys the Camenisch's membership-proof, and the authority (here, A_k) generates the private key based on *attribute-publics* (such as g^{x_i}) instead of *attribute-secrets* (such as x_i). But, in section 2, we show the approach needs efficiency and security enhancements. Moreover, in the following, we explore why the OT approach (alongside its high computation cost), is unable to guarantee the user's anonymity. In OT_1^n, the server is unknown about the *user's interested one* from the set of n-attributes. But, the server is confident about the user's possession of all n-attributes. In particular, in OT_1^n based Private Information Retrieval (PIR), the user must be

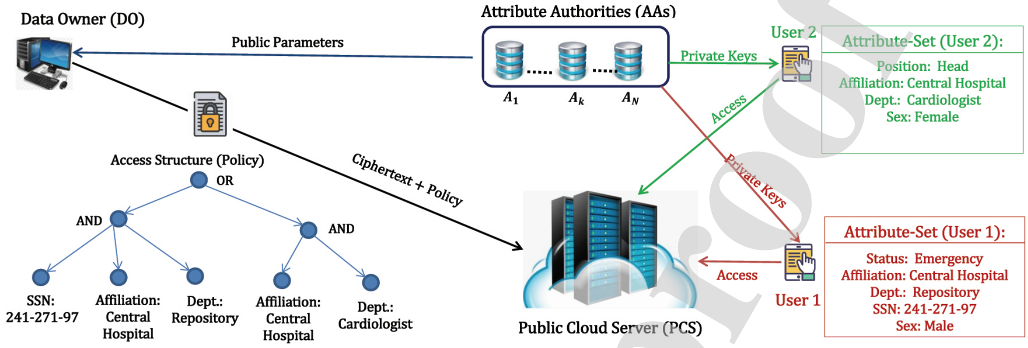

Figure 1: A System Model for an ABE-based Data Access Control

legitimate for all n-items, otherwise will be rejected. In general settings of the OT [8], the sender holds a set A, of n-items, and the transfer limitations are particularized by a set B of permitted subsets of A. The receiver may get any subset of the items in A that appeared in the set B. Consequently, we need a new approach (Section 3.2) to guarantee the user's anonymity in an efficient manner.

The Hidden Policy. In a usual ABE scheme, an encryption/decryption policy is comprised of attributes and is explicitly transferred along with a ciphertext (e.g., an encrypted EHR). Anyone who obtains the ciphertext knows the associated policy. Therefore, such ABE is inappropriate for the applications (e.g., E-health) where policies comprised of sensitive attributes. For more clarity, again consider the example of Fig. 1. A hospital encrypts EHRs using the ABE under a policy "(Affiliation="Central Hospital" AND Dept.=" Repository" AND SSN: "241-271-97") OR (Affiliation="Central Hospital" AND Dept.=" Cardiologist")". Then, it sends the ciphertext together with the policy to the cloud. The policy indicates that only a Cardiologist in Central Hospital or an emergency patient with a social security number (SSN) 241-271-97 in the hospital can access the EHRs. Then, if such an ABE scheme is used, an unauthorized user for the policy can infer that an emergency user with social security number 241-271-97 is experiencing a heart disease problem. This will break the user's privacy and reveals the attributes of the access policies in ABE

[9, 10]. In fact, the data owner encrypts his data to prevent leakage of sensitive information to the PCS because untrusted trading firms ordinarily accomplish the cloud. Consequently, preserving the privacy of the included policy into the ciphertext (known as *hidden policy*) is another privacy requirements that should be ensured. In the following, we shortly review the most interesting solutions for the *hidden policy problem* and entitle their drawbacks.

- *Predicate Encryption (PE).* As the first solution, one can provide hidden access structures from PE and Inner Product Encryption (IPE) [11]. In the encryption scheme, secret keys correspond to predicates, and ciphertexts are related to a set of attributes. So that, a secret key SK_f correspond to a predicate f can be used to decrypt a ciphertext related to an attribute set I if and only if $f(I) = 1$. A hidden-policy ABE system can be built from IPE. However, it happens in a super-polynomial blowup in size for arbitrary formulas [12].

- *Match-Finding schemes.* In [13, 9, 14, 15], some secure and innovative hidden policy schemes have proposed in a centralized architecture. The solution is *finding a minimum subset of the user's attribute-set, which can satisfy the policy (successful decryption occur).* In a case of a tree access structure (similar to an access matrix), the *k-elements* are leaf-nodes of the tree and are *tested* whether the tree can be satisfied (match is found) or not. Mathematically, the solution is to find an *ordered k-elements* from an *n-elements index set* I so that the order of the *k-elements* must be taken into account. Then, the solution can be viewed as a *k-permutation of n,* and then, the *test space (search-space)* is $P(n, k) = \frac{n!}{(n-k)!}$. Trivially, it is impractical apart from the computation cost (i.e., number of pairing operations) per each test. As a usual scenario, if the access tree is encrypted with 20 attributes and the user has 10 attributes (and the corresponding private keys), the size of the *search-space* (the computational complexity) is around $P(20, 10) \approx 10^{13}$.

- *Policy obfuscation.* The policy obfuscation solution [16] is an attractive solution for hidden policy purposed in recent researches (such as [17, 10, 18]). In this solution, the data owner obfuscates the policy along with generating the ciphertext, and the user de-obfuscates it before the decryption. This solution is more efficient than the two previous solutions. But, the schemes of [16, 17, 10, 18] use the expensive pairing operations at the end-users (n pairing operations for n attributes at the users for *de-obfuscation* purpose). Some other schemes ([6, 19, 20]) used the Bloom Filters and hash-function for obfuscation/de-obfuscation purposes. In the existing obfuscation schemes, the collusion between untrusted authorities and the PCS is not allowed. Because the collusion will reveal the policy for the PCS and the untrusted authorities.

In this paper, we introduce a security and efficiency enhanced *hidden-policy scheme* based on the *obfuscation idea*, an extensively applied idea in recent researches [17, 10, 18, 6, 19, 20].

Outsourcing Decryptions. Finally, the attribute-based encryption access control system is based on expensive asymmetric cryptography operations (e.g., Pairing Operations), which requires high computation cost. Then, any solution must take into care the efficiency of the end-users, which may be power-limited. To make our scheme efficient and applicable for such users (in a cloud-assisted IoT data sharing application), we securely outsource the cost-expensive operations of the decryption algorithm to the untrusted cloud servers.

Our Contributions. Now, after the above brief review of the existing solutions for the challenges, we entitle the main contributions of this paper as following:

- A novel anonymous and decentralized ABE (ADABE) scheme that is secure against the collusion of the untrusted authorities, is presented. Similar to [3, 4], our system is executed among the N authorities and then is secure against the collusion of at most *N-2* authorities.

- It guarantees the user's anonymity against the untrusted authorities and ensures both of the*identity-anonymity* and *attribute-anonymity*, by efficiently leveraging the cryptographic accumulators [21] (section 3.2).
- To ensure the hidden policy, the scheme uses a *decentralized obfuscation solution* which is resistant against collusion of at most *N-2* authorities and the PCS (section 3.5). Moreover, it is pairing-free and then is more efficient at the end-user side.
- To enhance the computational efficiency (at the end-users), a new verifiable computation outsourcing over the public cloud is presented.
- The scheme ensures the above features along with using an expressive access policy (i.e., a tree access structure with AND/OR and threshold gates).

Organization. The remainder of this article is organized as follows. In Section 2, some previous related works in the field are briefly reviewed. Section 3 gives the preliminaries required to understand the proposal. The security model of the scheme is presented in Section 4. Our construction is detailed in Section 5. Section 6 presents formal proofs, efficiency discussion, and experimental results. Section 7 gives the applications of the proposal in some real-world scenarios. Conclusions are provided in Section 8.

2. Related Works

The traditional access control systems (e.g., RBAC Model) provide flexible data access mechanisms for trusted storages. But, the ABE systems can provide the access control mechanisms for untrusted storages (e.g., Cloud Storage Servers). Subsequently, the decentralized extension of the ABE system enables it without an online trusted authority. In addition, an anonymous and decentralized extension ensures the data customer's anonymity, along with guaranteeing the hidden policy (ciphertext anonymity) in a decentralized environment. In the following, we review the previous works to resolve the challenges of this field.

2.1. Attribute-based Encryption systems

The ABE notion is introduced by Sahai and Waters [22]. In the ABE, both the ciphertext and the private key are labeled with a set of attributes. To decrypt the ciphertext, a user must find a match between his attributes and the included (plain) attributes into the ciphertext. There are two flavors of ABE schemes which are dual of each other: Ciphertext-Policy ABE (CP-ABE) [1] and Key-Policy ABE (KP-ABE) [23]. At first, an access policy is included in the ciphertext, while the private keys are connected to the attributes. At the second one, the ciphertext is bind to the attributes, while the access policy is included in the private keys.

2.2. Anonymity in Decentralized ABE systems

In 2009, Chase et al. [3] presented the first Anonymous DABE (ADABE) scheme. In this scheme, the main idea of building a distributed architecture is to gather partial secret parameters generated by different untrusted authorities. However, it uses a restricted access structure (limited expressiveness) and has the demonstrated *identity-leakage problem.* Lewko and Waters [2], in 2011, proposed an innovative and secure DABE in random oracle model with some novel ideas without considering user's privacy.

As providing the user's *attribute-anonymity* with the OT approach, Jung et al. [5] in 2015 proposed an interesting ADABE without any trusted entity. But, as discussed, the approach has a privacy problem. As well as [3], the main idea in the scheme is generating the user's private keys by gathering partial private parameters. Moreover, the system has two security issues. (the paper notations) The first is about collusion between a user and an authority that has in possession the requested attributes but response as she does not possess (response only g^{d_k} without $H(att(i))^{r_i}$, where $1 \leq k \leq N$). The collusion leads to the generation of $g^{\sum d_k}$, and then, by using $D = g^{\sum v_k + \sum d_k}$ to recover $g^{\sum v_k}$, the whole network will be compromised. The second is about collusion among users for *key combination attacks.* Because the $\sum d_k$ in the function $DecNode(x)$ is the same for all users. It is due to outsourcing $\hat{C} = D^{h^{-1}}$ by the data owner in the encryption time, as part of the ciphertext. To make different the parameter $\sum d_k$ for various users, it has to be renewed for each new user. It means the data owner has to be online to continuously update the ciphertext $\hat{C}$ (per each new user arrivals), which is impractical.

As providing the user's *attribute-anonymity* with the next approach, Han et al. [5] proposed an ADABE to preserve the user's anonymity against the authorities in a distributed environment. But, the scheme has two security issues along with inefficiency at the users. The first is about the *key combination attack*. In the system, the private keys are anonymously tied to the user's GID, and the GID cannot be extracted from the private keys. But, the keys from different users (with different GIDs) can be included in its decryption algorithm and satisfying the equations. As the second, each authority generates Camenisch' Tag [7] and corresponding public key with *attribute-secret* $a_{i,j}$ (the paper notations), which is used in the membership-proof process. But, the point is that since the authority herself is the generator of the Tag and the public key, then she knows the corresponding pairs of the *attribute-secrets* (e.g., $a_{i,j}$) and the Tags. Consequently, at the end of the process, the authority can identify the corresponding *attribute-secret* of a given Tag.

Qin et al. [24], proposed efficient outsourcing decryption to enhance efficiency at users but in a single authority environment and disregarded the user's privacy. Zhang et al. [25], in 2017, proposed a secure hidden policy ABE in a single authority environment with a new matching phase for efficiency enhancement in the encryption algorithm. Li at al. [26] proposed a light-weight verifiable outsourced decryption algorithm in single trusted authority architecture and disregarded the privacy problems. In the same year, Wang et al. [27] proposed an *identity-anonymity* preserving scheme using a trusted authority to issue anonymous credentials. The scheme uses a *frequent change of Pseudo-IDentity (PID)* (as a traditional trick to enhance the anonymity in *identity-anonymity* methods). In 2018, Li et al. [28] proposed an *identity-anonymity* supporting scheme. In the scheme, a pre-determined authority knows the user's GID to ensure *malicious user tracing* purposes. In the same year, Dai et al.

[29] proposed an efficiency enhanced lattice-based ABE. But it is still expensive for resource-constrained devices and, of course, disregarded the anonymity problems.

2.3. Hidden-Policy ABE systems

In 2008, Nishide et al. [30] proposed a hidden policy ABE scheme with a special kind of limited access structure. The idea is improved in later works, such as in [25]. As hidden policy schemes by the obfuscation idea, Yang et al. [19] introduced a centralized trusted authority based hidden policy scheme based on [31], which uses the Bloom Filter to *obfuscate the access structure.* Han et al. [6] proposed improvement over Yang et al. [19] to preserve privacy against attribute authority with the OT primitive. In 2019, Hao et al. [20] proposed improvement over Yang et al. [19] proposal to fix its vulnerability against the *dictionary attack* by fuzzing the used Bloom filter. Very recently, in [17, 10, 18], the Lewko-Waters scheme [2] is extended with an *obfuscation technique* to ensure the hidden policy in DABE with untrusted (semi-honest and curious) authorities. But, in the extensions, the collusion of the untrusted authorities and PCS is not allowed. Moreover, as well as the *obfuscation* in [16], the computation cost at the users is $O(n)$ expensive asymmetric paring operations, where n the size of the users' attribute set. Also, the schemes disregarded the user's anonymity against the untrusted authorities.

Finally, Table 1 compares the previous works in different aspects.

3. Preliminaries

3.1. Bilinear Maps

Assume G_1 is a cyclic multiplicative group of a large prime order q with generator g. The bilinear pairing map e is defined as $e : G_1 \times G_1 \to G_T$ with the following properties, where G_T is the codomain of e:

- Bilinearity: $e(g^a, g^b) = e(g,g)^{ab}$ for all a,$b \in Z_q$.
- Non-degeneracy: There exists $g \in G_1$ so that $e(g,g) \neq 1_{G_T}$.

Table 1: Comparison of different schemes where DEC stands for Decentralized

Schemes	Identity Anonymity	Attribute Anonymity	Hidden Policy	Security Model	Standard Model	Access Structure	Collusion Resistant	Decentralized	Decryption Outsourcing
Chase et al. 2009 [3]	✓	×	×	CCA	✓	AND gates	✓	✓	×
Lewko et al. 2011 [2]	×	×	×	CCA	ROM	LSSS	✓	✓	×
Hur et al. 2013 [16]	×	×	✓	CPA	ROM	LSSS	–	×	×
Qin et al. 2015 [24]	×	×	×	RCPA	ROM	LSSS	–	×	✓
Han et al. 2015 [5]	✓	✓	×	CCA	✓	LSSS	✓	✓	×
Jung et al. 2015 [4]	×	✓(OT)	×	CCA	ROM	Tree	×	✓	×
Yang et al. 2017 [19]	×	×	✓	CPA	✓	LSSS	–	×	×
Li et al. 2017 [26]	×	×	×	RCPA	✓	AND gates	–	×	✓
Zhang et al. 2018 [9]	×	×	✓	CCA	✓	LSSS	–	×	×
Zhong et al. 2018 [18]	×	×	✓	CPA	ROM	LSSS	✓	✓	×
Belguith et al. 2018 [10]	×	×	✓	RCPA	ROM	LSSS	✓	✓	✓
Cui et al. 2018 [14]	×	×	✓	CPA	ROM	LSSS	–	×	×
Qi Han et al. 2018 [6]	×	✓(OT)	✓	CPA	✓	LSSS	–	×	×
Fan et al. 2019 [17]	×	×	✓	CPA	ROM	LSSS	–	✓	✓
Hao et al. 2019 [20]	×	×	✓	CPA	✓	LSSS	–	×	×
Xiong et al. 2019 [15]	×	×	✓	CPA	✓	LSSS	–	×	✓
Proposed	✓	✓	✓(DEC)	RCPA	✓	Tree	✓	✓	✓

- Symmetry: $e(a, b) = e(b, a)$ for all $a, b \in G_1$.

where 1_{G_1} and 1_{G_T} are the identity elements of G_1 and G_T, respectively.

3.2. Accumulators and Attributes

A cryptographic accumulator protocol [21, 32] is used to accumulate a set $X = \{x_1, ..., x_n\}$ into a succinct value acc_X. To certify the membership of element x_i, a so-called witness wit_{x_i} will be efficiently computed. Moreover, for any non-accumulated value y in X, it is computationally infeasible to find a witness (collision freeness). With a cryptographic accumulator (acc), the user can perform membership-proof of his attributes to the authority, without disclosing the attribute x_i.

Here we briefly present the bilinear map accumulator, firstly introduced by Nguyen et al. [21] where g is the group generator, and e is the bilinear map. Let the authority A_k is responsible for the set S_k of n_k attributes ($S_k = \{x_i\}_{1 \leq i \leq n_k}$) and the random secret $sk = v_k$ is in Z_p^*. With the above settings, the accumulator value of the set S_k, acc_k, as an element in G_1 is defined as below:

$$acc_k = g^{\prod_{i=1}^{n_k}(v_k+x_i)}$$

The witness of an attribute $x_i \in acc_k$ is,

$$W_{i,S_k} = g^{\prod_{j=1,j\neq i}^{n_k}(v_k+x_j)}$$

The *Public Param (PP)* of the acc_k is the set $\{g^{v_k}, \{g^{x_i}, W_{k,i}\}_{1\leq i\leq n_k}\}$ where $n_k = |S_k|$ is the cardinality of the attribute-set which the authority is responsible for. Moreover, the private key of the acc_k is $sk = v_k$, which is known only by the accumulator builder. The authority A_k (Verifier), certifies the membership of attribute x_i to the accumulator acc_k by

$$e(g^{x_i}.g^{v_k}, W_i) = e(acc(S_k), g)$$

which is mathematically equivalent to:

$$W_{i,acc(S_k)}^{(v_k+x_i)} = acc(S_k)$$

without learning secret v_k.

The accumulator is collision-free under the Strong Diffie-Hellman (SDH) assumption [33], which is introduced by Boneh and Boyen. In our construction, the attribute element x_i of the accumulator acc_k is unknown for the authorities. Only the authorized users and the *accumulator builder* know it.

3.3. Zero-Knowledge Proof

To prove the possession of attribute i to the authority A_k, a user u applies the Zero-Knowledge Proof (ZKP) g^{x_i} ($PoK = \{(x_i) : g^{x_i}\}$), where x_i is the attribute and can be written as the public g^{x_i}. Similar to [27], we assume the attribute secret x_i can be written as a public form (in our case, such as g^{x_i}) without revealing the secret. To do this, we apply the efficient and Non-Interactive Schnorr' ZKP (NIZKP) [34] which is as follow:

The user (as prover) selects $w \in_R Z_q^*$ and computes $a = g^w$, $c = H(a, g^{x_i})$ and $r = (cx_i + w) \bmod q$ where p,q and g are system parameters.

The authority A_k (as verifier), calculates $c' = H(a, g^{x_i})$, and then checks a is equal to $g^r.g^{-x_i.c'}(\text{mod } p)$.

The user u, along with the PID, anonymously and efficiently proves the possession of attribute i to the A_k (as verifier).

3.4. User's Individual Public Token (IPT)

In an anonymous credential system [35], a participant can get a credential of an issue, which involves the participant's alias and attributes. By doing that, the participant, by engaging interactive protocols [36, 7], can prove a third party that he gets a credential, including the distributed alias and attributes, without publishing any more knowledge. Similarly, we use a publicly available *user's individual token*, including his PID, GID, and attributes (such as $token = f_t(PID|GID|\{x_i\})$ where f_t is a secure function such an accumulator (SDH secure). In the case of the accumulator, membership verification is anonymous and efficient at the user side. As we see later, it is to prevent collusion among users for key combination attack purposes. Moreover, for user u with the attribute set size n_u, the *offline TA* generate the token $token_{PID} = f_t(PID|GID|\{x_i\}_{1\leq i\leq n_u})$ as the user's token IPT.

In spite of some previous schemes (such as [27] to ensure *identity-anonymity*) that requires *frequent change of PID* (As a traditional trick to enhance the anonymity), we are free of this obligation due to provide the both of *identity-anonymity* and *attribute-anonymity.*

3.5. Decentralized Policy Obfuscation Idea

In this paper, we propose a pairing-free decentralized obfuscation method, so that the data owner (in encryption time) replaces the attribute x_i with an *obfuscated-attribute* s_i (to generate the obfuscated access tree T_{Obf}). It is calculated with a *secret* Obf_i ($s_i = H(Obf_i)$), that is generated after the cooperation of the N *untrusted authorities.* Then, it is resistant against collusion of at most $N - 2$ authorities. In decryption, to produce the *obfuscated* s_i, an authorized user combines *N-secrets*, taken from the N-authorities (here, ${Obf_i}^k$ from authority A_k), during the key generation algorithm. Before starting the decryption algorithm, the user calculates the obfuscated s_i by multiplying the *N-secrets*, as $s_i = H(Obf_i = \prod_{k=1}^{N} {Obf_i}^k)$. More details about Obf_i and ${Obf_i}^k$ are given in sections 5.2 and 5.3.

3.6. Complexity Assumptions

Let g be a public element of prime order p in a cyclic multiplicative group, G be generated by g and $a, b, c, z, \zeta \in_R Z_q^*$. Now, we shortly review the four well-known computational complexity assumptions.

Assumption 1 : Discrete Logarithm problem (DL). Assume $h \in G$ is publicly known, the ADVantage (ADV) of any probabilistic polynomial-time (PPT) *adversary* to compute x such that $h = g^x$ is negligible.

Assumption 2 : Decision Diffie-Hellman problem (DDH). Assume $g^a, g^b, h \in G$ are publicly known, the ADV of any PPT *adversary* to determine whether or not $h = g^{ab}$ is negligible.

Assumption 3: Decision Bilinear Diffie-Hellman problem (DBDH). Let g^a, g^b, g^c and $e(g,g)^z$ are known, the ADV of any PPT *adversary* to determine whether $e(g,g)^{abc} = e(g,g)^z$ is negligible.

Assumption 4: q-SDH [33]. Assume the set $\{g^{{v_k}^i} : 0 \leq i \leq q\}$ is publicly known, the ADV of any PPT *adversary* to find a pair $(t, g^{\frac{1}{v+t}}) \in Z_p^* \times G$ is negligible.

3.7. Access Structure

We use a tree access structure in which the leaf nodes are attributes, and non-leaf nodes are AND, OR and threshold gates. For a given tree T and a node x (corresponding to attribute i), if we set c_x as the number of the node's children, then the corresponding threshold value t_x satisfies $0 < t_x \leq c_x$. If at least t_x children nodes have been assigned true values, and then the node x is assigned a true value. Then, for an OR-gate, we have $t_x = 1$, and for AND-gate, we have $t_x = c_x$. As *satisfying rules* for an attribute set S, if $T(S) = 1$

or $x(S) = 1$, then the tree T or the node x is satisfied by the user's attributes. T is recursively calculated as follows. If x is a non-leaf and at least t_x child nodes return 1, then we have $x(S) = 1$. If x is a leaf node and $att(x) \in S$ then $S(x) = 1$. For root node R, $T(S) = 1$ only if $R(S) = 1$.

We use $L_{i,S}$ for $i \in Z_p$, and a set S of elements in Z_p as the Lagrange coefficient: $L_{i,S}(x) := \prod_{j\in S, j\neq i} \frac{x-j}{i-j}$. It is used for polynomial interpolation in the decryption algorithm. Moreover, we need an irreversible hash function $H : \{0,1\}^* \rightarrow G_T$.

4. Security Model

In the following, we define four algorithms; *Setup*, *KeyGen*, *Encrypt* and *BlindDecrypt* for our security model.

- $Setup(1^\lambda) \rightarrow (MSK_k, PP_k)$. The *setup* algorithm in the *offline trusted installation* phase, takes a security parameter as input. It outputs secret and public parameters of an authority (here A_k).

- $Encrypt(M, T, params) \rightarrow CT$: The algorithm takes the message M, the access tree and the public parameters. The DO runs it, and output the ciphertext CT (Included obfuscated access tree T_{Obf}).

- $KeyGen(MSK_k, token_{PID}, \{{W_{k,i}}^{b_k}\}, \{{W_{k,i}}^{h}\}) \rightarrow (\{K_{1,i}, K_{2,i}, {Obf_i}^{k}\})$. The key generation algorithm, which is run by the authority A_k, takes authority A_k master key, the user's token (IPT), and two sets of publics. It outputs a set of the private keys and *attribute-obfuscator* parameter (${Obf_i}^{k}$).

- $BlindDecrypt(CT, \{K^z_{1,i}, K^z_{2,i}, s_i\}_{1\le i\le n_{u,T}}, E^z_1, E^z_2,) \rightarrow (A^1_z, A^2_z)$: The algorithm takes the ciphertext, blinded private keys, obfuscated-attributes and two blinded publics (E^z_1, E^z_2). It is run by the untrusted PCS and outputs the two *blinded results*, which is used to extract the message M (plaintext) with simple arithmetic operations at the user.

Now we formalize our security model by the following two games. The first is to ensure the ABE access control, and the second is for the user's anonymity against the authorities and the PCS.

Game 1. As well as the previous works where considered outsourcing decryption (such as [10, 37, 24, 38]) we adopt Replayable CPA (RCPA) introduced by Canetti et al. [39]. It means the ciphertext can be modified to a new ciphertext without changing the message in a meaningful way. The adversary is an unauthorized user, and only the authorized users who have enough attributes can successfully decrypt the ciphertext.

Init. The adversary commits to an access tree T_0.

Setup. The adversary has in possession system parameters and a set of attributes, which, of course, are not enough to satisfy the access tree. She has in control at most $N-2$ authorities, which can be malicious. But still, at least two authorities (which are responsible for some attributes in the tree T_0) are in control of the challenger.

Phase 1. The adversary (as an unauthorized user) is allowed to query for private keys from the *DBDH simulator*.

Challenge. The adversary chooses two messages m_0 and m_1. Then, she sends them to the challenger. The challenger, flips an unbiased coin $\varphi \in_R \{0,1\}$ and encrypts m_φ. Then returns m_φ to the adversary.

Phase 2. Phase 1 is repeated, adaptively.

Guess. The adversary guesses φ as $\varphi' \in_R \{0,1\}$. She will succeed if the probability $\Pr(\varphi' = \varphi) - \frac{1}{2}$ is non-negligible.

Definition 1. If all the PPT adversaries have at most a negligible ADV in the security game, then the proposed scheme can securely ensure access control against unauthorized access and ensures confidentiality.

Game 2. This game model the user's anonymity against the authorities (the *identity-anonymity* and *attributes-anonymity*).

Setup. The adversary who has in control at most $N-2$ (compromised) authorities and the PCS. The adversary has all the secrets of the authorities

Table 2: Main Notations in our Construction

GID, PID	The user's Global-ID and Pseudo-ID
$acc_k, \{W_{k,i}\}$	The accumulator/wittness-set for authority A_k
v_k	The secret of accumulator acc_k
x_i, g^{x_i}	The secret and public for attribute i
a_k, c_k	The master keys of the authority A_k
n_k, n_T, n_u	The number of attributes in authority A_k, tree T and user u, respectively ($n_k = \|S_k\|, n_T = \|S_T\|, n_u = \|S_u\|$)
$n_{k,u}, n_{k,T}$	$n_{k,u} = \|S_k \cap S_u\|, n = n_{T,u} = \|S_T \cap S_u\|$.
$token_{PID}, b_{k,u}$	The user's token and the corresponding secret in the authority A_k
h	The DO's secret for HP purpose
$C, \{C_{1,x}, C_{2,x}\}$	The ciphertext CT
E_1, E_2	The publics for decryption
q_x, s	The polynomial at node x and the root's secret
T_{Obf}	The obfuscated tree (policy)
${Obf^k}_i$	The *partial obfuscator* of authority A_k for attribute x_i
Obf_i	The *obfuscator* for attribute x_i
s_i	The *obfuscated value* of attribute x_i
H	A secure one-way hash function
$K_{1,i}, K_{2,i}$	The private keys of attribute i
$r_{k,u}$	The random of authority A_k for the user's private key
z	The blinding param for blinded- decryption
${A^1}_z, {A^2}_z$	The blinded results (outputs) of the blinded-decryption

and also all the system public parameters (*params*).

Phase 1. The adversary is allowed to query for private keys from the simulator and the non-corrupted authorities.

Challenge. The adversary chooses two challenge messages m_0 and m_1 and two challenge access tree T_0 and T_1. Then submits them to the challenger. The challenger flips an unbiased coin $\varphi \in_R \{0, 1\}$ and then return back the encrypted message m_φ and the obfuscated T_φ.

Phase 2. Phase 1 is repeated, adaptively.

Guess. The adversary guess φ as $\varphi' \in_R \{0, 1\}$ and success if $\Pr(\varphi' = \varphi) - \frac{1}{2}$ is non-negligible.

Definition 2. If all PPT adversaries have at most a negligible ADV in the game, then the user's anonymity is guaranteed.

5. The Construction

In our proposal, as well as Chase et al. [3] and Jung et al. [4], the private keys are generated by the cooperation of the untrusted (semi-honest) N authorities. In the following, we detail the four algorithms of our proposal. For more convenience, the main notations used throughout our construction are summarized in table 2.

5.1. Setup

The *offline TA* in the *trusted installation phase* generates the system secret and public parameters. A set of n_k attributes ($S_k = \{x_i\}_{1 \le i \le n_k}$) will be securely accumulated into an accumulator acc_k corresponding to the authority A_k. The secret and public values of the accumulator acc_k are as follows.

$$SK_{acc_k} = \{v_k, \{x_i\}_{1 \le i \le n_k}\}$$

$$PP_{acc_k} = \{acc_k, g^{v_k}, \{(g^{x_i}, W_{k,i})\}_{1 \le i \le n_k}\}$$

The secret values of the accumulator are kept secret by the *offline TA*. Each authority A_k runs the *Setup* algorithm to generate its master keys and public parameters as the following:

$$MSK_k = \{a_k, c_k\} \text{ where } a_k, c_k \in_R Z_q^*$$

$$PP = \{PP_{acc_k}, \{W_{k,i}{}^{c_k}, g^{(x_i+v_k)a_k{}^2}\}_{1 \le i \le n_k}, acc_k{}^{a_k}\}$$

The public parameters will be (publicly) published or send over an *authenticated public bulletin board (or channel)*, available for every participant (user, authorities, DO, and PCS), as well as [40, 41].

5.2. Encryption

For our decentralized hidden policy scheme (Section 3.5), the DO selects a random secret $h \in_R Z_q^*$ and publicly publishes $\{\{W_{k,i}{}^{h}\}_{1 \le i \le n_{k,T}}\}_{1 \le k \le N}$ (or sends over the *bulletin board*), the n_T is the number of attributes in the tree and

$n_{k,T} = |S_k \cap S_T|$. The authority A_k uses the public parameters to generate the $Obf_i{}^k$ in the $KeyGen$ algorithm for the authorized users (Section 5.3). Here, the DO uses the secret h and public-values $\{W_{k,i}{}^{c_k}\}$ to create the *obfuscated-attribute* $s_i = H(Obf_i)$, where $\{Obf_i = (\prod_{j=1}^{N} W_{k,i}{}^{c_j})^h\}_{1 \leq i \leq n_{k,T}}$ and H is a secure hash function (or an accumulator [21]). To generate the obfuscated tree T_{Obf}, the DO replaces the attribute x_i (in the tree T) with s_i.

To generate the ciphertext, for each node x of the tree T, the algorithm selects a polynomial q_x of degree d_x where d_x is one less than the threshold value t_x ($d_x = t_x - 1$). The algorithm starts from the root node R and chooses a random secret $s \in_R Z_q^*$ and sets $q_R(0) = s$. Then it randomly chooses other coefficients of q_R to define it completely. For any other node x, it sets $q_x(0) = q_{parent(x)}(index(x))$ and randomly selects other coefficients to define q_x as well. Finally, the ciphertext is generated as follows and will be sent to the PCS.

$$CT =< T_{Obf}, C, \{C_{1,x}, C_{2,x}\}_{1 \leq i \leq n_T} > \text{ where,}$$

$$C = M.e(g,g)^s,\ C_{1,x} = (acc_k^{a_k} \times g)^{q_x(0)},\ C_{2,x} = g^{((x_i+v_k).a_k{}^2)q_x(0)}$$

5.3. Key Generation

The following processes are done between user u and the authorities to generate the user's attribute private keys.

- Similar to [16, 17, 10, 18], the user u anonymously establishes an anonymous channel with the authorities by the *one-way anonymous key issuing scheme* [42].
- After receiving the user's request, all of the authorities (here A_k) calculates secret $b_{k,u}$ as $b_{k,u} = H(token_{PID}|a_k)$, generates and publicly publishes $\{\{W_{k,i}{}^{b_k}, g^{b_k}\}_{1 \leq i \leq n_k}\}$ or send to the *bulletin board*, where H is a secure one-way function.
- The user with the PID proves possession of the attribute x_i to the authority A_k by the schnorr's NIZKP, as $PoK = \{(x_i) : g^{x_i}\}$. Also, the authority A_k verifies the accumulation of the attribute x_i into the acc_k with the

corresponding accumulator membership equality test $e(g^{x_i}.g^{v_k}, W_{k,i}) = e(acc_k, g)$ without knowing the attribute x_i.

- The authority A_k generates the private keys $(K_{1,i}, K_{2,i})$ for the user u as follows:

$$K_{1,i} = ((\prod_{j=1,j\neq k}^{N} {W_{j,i}}^{b_{j,u}})({W_{k,i}}^{b_{k,u}}))^{1/a_k} = W_{k,i}^{(\sum_{k=1}^{N} b_{k,u})/a_k}$$

$$\text{and similarly, } K_{2,i} = W_{k,i}^{(1+\sum_{k=1}^{N} b_{k,u})/a_k}$$

  where $n_{k,u}$ is the number of user's attributes which the authority A_k is responsible for $(n_{k,u} = |S_k \cap S_u|)$.

- Now, all of the N authorities generate the *attribute-obfuscators*. To compute their secret part of the *obfuscator*, the authority A_k calculates ${Obf_i}^k = ({W_{k,i}}^h)^{c_k}$.

- Finally, the authority sends the private keys and the *obfuscators* to the user through the anonymous channel.

As noted before, due to preserving both of *identity-anonymity* and *attribute-anonymity*, the *frequent change of PID as a trick to increase user's anonymity* ([27]) is not required in our case. Then, the given *token* to the user u (included the user's PID) at *user's registration phase* is enough during the user life-time in the system.

5.4. Decryption

At first, the user calculates the *obfuscated-attribute* $Obf_i = \prod_{j=1}^{N} {Obf_i}^j = \prod_{j=1}^{N} {W_{k,i}}^{hc_j}$ and $s_i = H(Obf_i)$ to generate the leaf-nodes of the tree T_{Obf} for blind-decryption purpose over the PCS. Outsourcing the expensive ABE decryption over the PCS servers and performing a blind-decryption (partially decryption) on the servers, decreases the computational load of the users and made the proposal applicable for resource-constrained devices and users.

Blinding and unblinding can simply be achieved in our proposal. To this purpose, the user generates a secret *Blinding-Parameter (BParam) z*. Then, computes and sent the blinded parameters ($\{(K_{1,i})^z, (K_{2,i})^z\}_{1\leq i\leq n_{u,T}}$, ${E_1}^z$, ${E_2}^z = (g.E_1)^z$) along with the leaf-node set of tree T_{Obf} as $\{s_i\}_{1\leq i\leq n_{u,T}}$ to the PCS, where $E_1 = \prod g^{b_k}$.

The blind-decryption algorithm over the PCS will be blindly invoked at the root of the tree T_{Obf} and then is recursively executed at each node. Thus, the algorithm calculates the leaf-nodes (such as node x) as follows.

$$BlindDecNode_A(s_i, CT, {K_{1,i}}^z, {E_1}^z) = \frac{e(C_{1,x}, {E_1}^z)}{e(C_{2,x}, {K_{1,i}}^z)}$$

$$= \frac{e((acc_k^{a_k} \times g)^{q_x(0)}, g^{z\left(\sum_{k=1}^{N} b_k\right)})}{e(g^{(x_i+v_k)a_k{}^2 q_x(0)}, W_{k,i}^{z\left(\sum_{k=1}^{N} b_k / a_k\right)})}$$

$$= \frac{e(acc_k^{a_k q_x(0)}, g^{z\left(\sum_{k=1}^{N} b_k\right)}).e(g^{q_x(0)}, g^{z\left(\sum_{k=1}^{N} b_k\right)})}{e(g^{q_x(0)}, acc_k{}^{a_k.z.\left(\sum_{k=1}^{N} b_k\right)})}$$

$$= e(g,g)^{z.\left(\sum_{k=1}^{N} b_k\right).q_x(0)}$$

Similarly, $BlindDecNode_B(s_i, CT, {K_{2,i}}^z, {E_2}^z) = e(g,g)^{z.\left(1+\sum_{k=1}^{N} b_k\right).q_x(0)}$

Now, the algorithm recursively calculates the interpolation at the non-leaf (internal) nodes. In other words, the above-calculated results at leaf-node x are used to calculate the secret value at internal node y ($q_y(0)$) by the interpolation. The calculation at internal nodes is as follows.

$$F_y = \prod_{z\in S_y} {F_z}^{L_{i,S'_y(0)}}, where \quad \begin{matrix} i = index(z) \\ S'_y = \{index(z) : z \in S_y\} \end{matrix}$$

$$= \prod_{z\in S_y} (e(g,g)^{z.(\sum_{k=1}^{N} b_k).q_y(0)})^{L_{i,S'_y(0)}}$$

$$= \prod_{z\in S_y} (e(g,g)^{z.(\sum_{k=1}^{N} b_k).q_{parent(z)}(index(z))})^{L_{i,S'_y 0)}}$$

$$(by\, construction)$$

$$= \prod_{z\in S_y} (e(g,g)^{z.(\sum_{k=1}^{N} b_k).q_y(i).L_{i,S'_y(0)}})$$

$$= e(g,g)^{z.(\sum_{k=1}^{N} b_k).q_y(0)} (using\, polynomial\, interpolation)$$

After traversing the tree and calculating the secret value at the root node R of the tree ($s = q_R(0)$), we have:

$$A^1{}_z = e(g,g)^{z.\left(\sum_{k=1}^{N} b_k\right).q_R(0)} = e(g,g)^{z.\left(\sum_{k=1}^{N} b_k\right).s}$$

Similarly, $A^2{}_z = e(g,g)^{z.\left(1+\sum_{k=1}^{N} b_k\right).s}$

Then, the $A^1{}_z$ and $A^2{}_z$ as the *blind-decryption results* will be sent to the user. Finally, at the end-user, the message M will be extracted as following with simple arithmetic operations.

$$M = \frac{C\times(A^1{}_z)^{1/z}}{(A^2{}_z)^{1/z}}$$

Since the PCS is not trusted, the *blind-decryption results* must be verified. To verify the result, the data owner sets a parameter V as $V = M|H(M)$. Then, includes it into the ciphertext C (as $C = V.e(g,g)^s$). The H is a secure hash function (a usual hash function, an accumulator or any unforgeable tag).

Since the user's *BParam (z)* is unknown for the PCS, she can not cheats the user with some faked values (such as $M', H(M'), V' = M'|H(M'), A^1{}_z{}', A^2{}_z{}'$). It means, unblinding the faked results $A^1{}_z{}'(\neq A^1{}_z)$ and $A^2{}_z{}'(\neq A^2{}_z)$ by the user (with the $BParam = z$) gives a random values ($A^{1'}(\neq A^1)$ and $A^{2'}(\neq A^2)$) and finally leads to a random V' (instead of V) where the probability of $\Pr(V' = V)$ is negligible for any PPT Adv. Consequently, the authorized user with enough set of attributes and the corresponding private keys can satisfy the access-tree (extract $A^1{}_z$ and $A^2{}_z$ (or A^1 and A^2)). So that can successfully decrypts the ciphertext and then extracts the message M.

The user' *individual* ($\sum b_{k,u}$) (originated from the user's IPT token) is included into the private keys ($K_{1,i}$ and $K_{2,i}$), and into the decryption-results at each leaf-nodes ($e(g,g)^{z.(\sum b_k).q_R(0)}$). It prevents the collusion among the users. It means combination of private keys (from different users) results in random (during the Lagrange interpolation calculations). It is similar to the *tied private keys with the GID* in the Lewko-Waters scheme [2], and of course, in an anonymous manner and in the standard model.

6. Analysis

6.1. Security and Privacy

In the following, we show that our proposal is secure and ensures the data access for the authorized users and prevent access for unauthorized users (Game 1 and Theorem 6.1). Also, we show it provides user's privacy against the authorities and the PCS (Game 2 and Theorem 6.2). It is formally done in the security model defined by the security games.

Theorem 6.1: *If there is a PPT adversary that can solve the security game 1 in a non-negligible advantage, then she can solve the DBDH problem in a non-negligible ADV, as well.*

Let G_1 represent a multiplicative cyclic group of prime order q, and its generator is g. Our bilinear map function e is defined as $e : G_1 \times G_1 \rightarrow G_T$. The challenger flips a random coin ψ and sets record y as the following.

$$y = (g, acc, D, B, L, R)$$
$$= \begin{cases} (g, acc_k^{a_k}, \{g^{(x_i+v_k).{a_k}^2}\}, e(g,g)^s, e(g,g)^{s.\sum_{k=1}^{N} b_k}, e(g,g)^{s.(1+\sum_{k=1}^{N} b_k)}) & if\ \psi = 0 \\ (g, acc_k^{t_1}, \{g^{(x_i+v_k).{t_1}^2}\}, e(g,g)^{t_2}, e(g,g)^{t_3}, e(g,g)^{t_4}) & if\ \psi = 1 \end{cases}$$

Where a_k, b_k and s are valid secrets, but $t_i \in_R Z_q^*$ (is random) and then $(acc_k^{t_1}, e(g,g)^{t_2}, e(g,g)^{t_3}, e(g,g)^{t_4})$ are randoms, too. The simulator *sim*, which plays as a DBDH challenger in the game, receives the record y from the challenger.

Init. The adversary commits to an access tree T_0.

Setup. The adversary knows the system parameters and a set of attributes, which of course, are not enough to satisfy the tree T_0. Some of the attributes are in control of at least two non-compromised authorities.

Phase 1. The adversary is allowed to query the private keys from the *sim*.

Challenge. The adversary chooses two messages m_0 and m_1 and sends the two to the challenger. The challenger flips a fair coin $\varphi \in_R \{0, 1\}$ and encrypts $V_\varphi = m_\varphi | H(m_\varphi)$ with the record y and the access tree and then returns it back to the adversary.

$$CT^* = \left\langle \{{C^*}_{1,x}, {C^*}_{2,x}\}_{1\le x\le n_{T_{Obf}}}, C^*, {A_z^1}^*, {A_z^2}^* \right\rangle = \begin{cases} \left\langle \{C_{1,x}, C_{2,x}\}_{1\le x\le n_{T_{Obf}}}, C, A_z^1, A_z^2 \right\rangle, if\ \psi = 0 \\ \left\langle \{{C'}_{1,x}, {C'}_{2,x}\}_{1\le x\le n_{T_{Obf}}}, C', {A_z^1}', {A_z^2}' \right\rangle, if\ \psi = 1 \end{cases}$$

Where,

$$\text{(Valid CT } (\psi = 0))\ CT = \langle\ \{C_{1,i} = (acc_k^{a_k} \times g)^{q_x(0)}, C_{2,x} = g^{(x+v_k){a_k}^2 q_x(0)}\}_{1\le i\le n_{T_{Obf}}}, C = V_\varphi.e(g,g)^s, {A^1}_z = e(g,g)^{z.s.\sum_{k=1}^{N} b_k}, {A^2}_z = e(g,g)^{z.s.(1+\sum_{k=1}^{N} b_k)}\ \rangle$$

$$\text{(Invalid CT } (\psi = 1))\ CT^{'} = \langle\ \{{C'}_{1,x} = (acc_k^{t_1} \times g)^{t_5}, {C'}_{2,x} = g^{(x+v_k){t_1}^2 t_6}\}_{1\le i\le n_{T_{Obf}}}, C^{'} = V_\varphi.e(g,g)^{t_2}, {A^1}_z{}' = e(g,g)^{t_3}, {A^2}_z{}' = e(g,g)^{t_4}\ \rangle$$

Phase 2. Phase 1 is repeated, adaptively.

Guess. The PPT *adversary* guesses φ as $\varphi' \in_R \{0,1\}$. She will be a success if the probability $\Pr(\varphi' = \varphi) - \frac{1}{2}$ is a non-negligible.

The successful adversary must distinguish between CT and CT' from CT^* with non-negligible advantage.

The adversary reveals his guess φ' of φ. The *sim* checks if $\varphi' = \varphi$ then sets and releases $\psi' = 0$; otherwise, $\psi' = 1$. The former means a valid (DBDH) three-elements record as $(C, {A^1}_z, {A^2}_z)$, and the later means a random record $(C', {A^1}_z{}', {A^2}_z{}')$.

The ADV of the adversary Adv in this security game for each of the three elements will be calculated as the following (Let assume the adversary advantage in the DBDH problem as ϵ):

- Firstly, when $\psi = 1$ (invalid CT), the Adv can not learn any information about φ, and we have $\Pr[\varphi \neq \varphi' | \psi = 1] = \Pr[\varphi = \varphi' | \psi = 1] = 1/2$. Moreover, the *sim* sets and releases $\psi' = 1$ if $\varphi \neq \varphi'$. Consequently, we have $\Pr[\psi' = \psi | \psi = 1] = \Pr[\varphi \neq \varphi' | \psi = 1] = 1/2$.
- Second, when $\psi = 0$, the Adv is given a valid ciphertext, but she has not enough attributes (and corresponding private keys) to satisfy the

tree. Then the adversary has to distinguish between the two tree-elements records, as $(C, {A^1}_z, {A^2}_z)$ and $(C', {A^1}_z{}', {A^2}_z{}')$. Since her ADV in DBDH problem is defined as ε, then we have $\Pr[\varphi = \varphi' | \psi = 0] = 1/2 + \varepsilon$ to distinguish each element. Moreover, sim sets and releases $\psi' = 0$ if $\varphi = \varphi'$. Consequently, we have $\Pr[\psi' = \psi|\psi = 0] = \Pr[\varphi = \varphi'|\psi = 0] = 1/2 + \varepsilon$.

The ADV of the adversary to distinguish the records is :
$\Pr[\psi' = \psi|\psi = 0] \Pr[\psi = 0] + \Pr[\psi' = \psi|\psi = 1] \Pr[\psi = 1] - 1/2 =$
$(1/2)(1/2 + \varepsilon) + (1/2)(1/2) - 1/2 = \varepsilon/2$

One of the conclusions obtained from the above proof is that, if ADV of the adversary in the DBDH is ε, then the overall ADV of the adversary in this game will be $\frac{1}{3}({}^{\varepsilon}\!/_2 + {}^{\varepsilon}\!/_2 + {}^{\varepsilon}\!/_2) = {}^{\varepsilon}\!/_2$, which is a negligible advantage in polynomial time. Hence, our scheme securely ensures *ABE access control.*

Theorem 6.2 *If there is a PPT adversary Adv that can solve the security game 2 in non-negligible ADV, then she can solve the DBDH problem in non-negligible ADV, as well.*

Sketch of Proof. Regarding the *user's anonymity against the authorities* (even in collusion with the all $N - 2$ authorities), the user u proves the possession of attribute x_i with the schnorr's NIZKP ($PoK = \{(x_i) : g^{x_i}\}$), without revealing the attribute. Also, the authority A_k performes the membership-proof $(e(g^{x_i}.g^{v_k}, W_{k,i}) = e(acc_k, g))$ for the attributes; again the user does not reveal the attribute x_i.

Moreover, we use a token which (similar to an anonymous credential) includes the user's PID, GID and attributes (such as $token = f_t(PID|GID|\{x_i\})$) where the token is used to generate secret $b_{k,u}$. Since, f_t and f_k are two secure hash functions (such as an accumulator [36, 21]), their outputs ($token_{PID}$ and $b_{k,u}$) do not leak any information about their inputs with non-negligible advantage. Moreover, the public params, which includes the attributes $(\{x_i\}_{1 \le i \le n_u})$ such as the acc_k, $W_{k,i}$ and g^{x_i} are secure (the two first are SDH secure and the third is DL hard problem). In game 2, the adversary submits two messages m_0 and m_1, and two access tree T_0 and T_1, then the challenger returns the CT^*.

Similar to the calculations for game 1, the ADV of the adversary in the security game 2, to distinguish the DBDH record CT $(C, {A^1}_z, {A^2}_z)$ from the random record CT' $(C', {{A^1}_z}', {{A^2}_z}')$, is as follows:

$\Pr[\psi' = \psi|\psi = 0] \Pr[\psi = 0] + \Pr[\psi' = \psi|\psi = 1] \Pr[\psi = 1] - 1/2 =$

$(1/2)(1/2 + \varepsilon) + (1/2)(1/2) - 1/2 = \varepsilon/2$

Hidden Policy (HP). Our proposal ensures policy privacy preservation, as privacy of the user and DO against the untrusted PCS, even in case of collusion of at most $(N - 2)$ authorities. In spite of the previous obfuscation techniques (such as [16, 17, 10, 18]), ours is executed among the N authorities. Then, the authorities (Which are allowed to collude together and to be compromised by the PCS), can not de-obfuscate the obfuscated access tree (T_{Obf}) or deduct any information about the attributes included into the ciphertext. In ours, the *obfuscated-attribute* x_i (Obf_i) is built at the user side by the received values from the authorities ($Obf_i = \prod_{k=1}^{N} {Obf_i}^k = \prod_{j=1}^{N} {W_{k,i}}^{hc_j}$). The parameter h is only known for the data owner. Thus, the authority to calculate ${Obf_i}^k$ or Obf_i has to find h (a.k.a c_k) as the exponent of ${W_{k,i}}^h$ (a.k.a ${W_{k,i}}^{c_k}$), which is a hard problem.

To be more specific, let $X = \prod_{j=1,j=\neq i}^{N} (x_j + v_k)$, then we have ${W_{k,i}}^h = (g^X)^h = (g_1)^h$ and ${W_{k,i}}^{c_k} = (g^X)^{c_k} = (g_1)^{c_k}$. Distinguishing $(g_1)^{c_k h}$ from $(g_1)^r$ $(r \in_R Z_q^*)$ with given $(g_1)^{c_k}$ and $(g_1)^{c_k}$ (a.k.a ${W_{k,i}}^{hc_k}$ from ${W_{k,i}}^r$ with given ${W_{k,i}}^h$ and ${W_{k,i}}^{c_k}$) is a DDH hard problem.

In compact, if the adversary submits the two messages and the two access trees, then the challenger returns the CT^* (encryption of message m_φ with obfuscated tree $T_{\varphi Obf}$). Similar to the above brief calculation in game 2, the ADV of the adversary to distinguish the valid record CT $(C, {A^1}_z, {A^2}_z)$ from the random CT' $(C', {{A^1}_z}', {{A^2}_z}')$ is $\varepsilon/2$.

6.2. Efficiency

Here, we analyze the computation cost and communication overhead of our proposal and give a comparison in table 3, where the relevant notations are summarized in Table 4. It is followed by a prototype focused on efficiency and performance at the end-users.

Computation Cost. The meaningful computation at the end-users is the pairing operations in the decryption algorithm. That is directly related to the number of attributes (n), which is outsourced over the PCS. Moreover, our *obfuscation/de-obfuscation* method is pairing-free and uses only multiplication. Then, the overall cost at the user side is: Two exponentiation in G_T (E_T) for unblinding of *blind-decryption results*, one hash value calculation (H) for the *result verification* and $O(n)$ multiplications for *de-obfuscation* which is very small.

The computation costs at the authority A_k is: One pairing (the accumulator verification), $3n+1$ exponentiations in G_T (E_T) for private keys and obfuscators, $2n$ multiplication in G_T (M) for private keys, and the light-weight two-round NIZKP cost.

The significant computation of the DO is: $2(n+1)$ exponentiations in G_T for the encryption algorithm, n_k multiplication in G_T (M) and two exponentiations ($2E_T$) for the *obfuscator*.

As summarized in table 3, our scheme has the lowest computation cost at the end-user device (as our motivation). The PCS blindly does the user's computations through the outsourcing procedure. It increases the computation cost at the PCS that is assumed as a powerful computation resource.

Communication overhead. Usually, the significant communication overhead at the end-user side is due to the transmission of the ciphertext and the blinded parameters to the PCS (for the blind-decryption algorithm). Consequently, the communication cost of such schemes [3, 2, 16, 4, 14, 19, 20, 9] (which do not use the decryption outsourcing), is small. Table 3 summarizes the user's communication overheads of our scheme in comparison with the previous works. For clarity, the overhead of the *one-way anonymous channel* and the *NIZKP* (shared among the schemes) are omitted. Note that, the less communication overhead of some previous works (table 3) in comparison to others

Table 3: Efficiency Comparison of Different Schemes.

Schemes	**Computations (PCS)**	**Computations (end-user)**	**Communication (end-user)**
Chase et al. 2009 [3]	-	$(2n+1)E_T+2(nN+3)P$	$(n_u+2N)\|G_T\|$
Lewko et al. 2011 [2]	-	$n.E_T+2n.P$	$n_u\|G_T\|$
Hur et al. 2013 [16]	-	$n.P$ (for HP) $+ 2(n+1)P+E_1+H$	$(2n_u+1)\|G_T\|$
Qin et al. 2015 [24]	$(2n+1)P+n.E_1$	$3H+M$	$(2n_T+1)\|G_1\|+\|G_T\|$
Han et al. 2015 [5]	-	$(4n_u+2n)P+(39+3n_{k,u}+8n_u+n)E_1+3n.E_T$	$(12+2n_u)\|G_1\|+(n_u+n_k+9)\|G_T\|$
Jung et al. 2015 [4]	-	$n_u.E_T$ (for OT) $+2(n+1)P+E_1+H$	${n_u}^2\|G_T\|$ (for OT)
Yang et al. 2017 [19]	-	$n.E_1+(2n+1)P+2dn_u.H$	$n_u\|G_T\|$
Li et al. 2017 [26]	$(2+n)P+2n.E_1$	$4P+E_T$	$(2n_T+3)\|G_1\|+\|G_T\|$
Zhang et al. 2018 [9]	-	$(4+n_u)P+3n_u.E_1+n_u.E_T$	$(n_u+1)\|G_T\|$
Zhong et al. 2018 [18]	-	$n_u.P$ (for HP) $+n.E_T+2n.P$	$(n_u+3)\|G_T\|$
Belguith et al. 2018 [10]	$2n.P$	$n_u.P$ (for HP) $+E_T+3H$	$(n_u+1)\|G_T\|+(6n_u+1)\|G_1\|$
Cui et al. 2018 [14]	-	$(6n+1)P+n.E_T$	$(n_u+2)\|G_T\|$
Qi Han et al. 2018 [6]	-	$n_u.E_T$(for OT) $+(2n_u+1)E_T+(3n_u+1)P+2dn_u.H$	${n_u}^2\|G_T\|$ (for OT)
Fan et al. 2019 [17]	$Nn.E_T+N(2n+1)P$	$n_u.P$ (for HP) $+E_T+H$	$(3n_T+1)\|G_1\|+(n_T+2)\|G_T\|$
Hao et al. 2019 [20]	-	$n.n_u.E_1+(l+n_u)P+2dn_u.H$	$n_u\|G_T\|$
Xiong et al. 2019 [15]	$(7n+2)P+n.E_T$	$3P+3E_T+2n_u.H$	$(2+5n)\|G_1\|$
Proposed	$4n.P$	$2E_T+H+n.M$	$(2n+2)\|G_T\|$

(including ours) is due to disregarding one or more aspects of the anonymity and the privacy issues (table 1) or ignoring the decryption outsourcing. Finally, our 720 proposal is free of a secure multi-party communication (MPC) protocol, which results in a low computation cost comparing to the previous works ($O(N^2)$) [3, 4].

Table 4: Notations for Efficiency Analysis

HP	Hidden Policy (de-obfuscation)
d	number of hash functions
P	Time of Pairing
E_1, E_T	Exponentiation times in G_1 and G_T
$l \times t$	Dimensions of access matrix

Experimental results. We experimentally measured decryption and de-725 obfuscation (for Hidden Policy purpose) time as computation cost at the end-user. We used cpabe-toolkit source code [43] to implement our proposal and conduct a comparison. The evaluation is done on Linux Kali 32-bit with Intel

7700HQ, Core i7, 16GB RAM, and pbc-0.5.14 library. We used a 160-bit elliptic curve g group and non-singular elliptic curve type A, $y^2 = x^3 + x$, over a 512-bit finite filed, as well as the cpabe-toolkit [43].

We implemented four more relevant to ours (Jung et al. 2015 [4], Han et al. 2015 [5], Belguith et al. 2018 [10], Fan et al. 2019[17]) and our scheme to measure the processing time (Fig.2 (a)) and the ciphertext communication overhead (Fig.2 (b)) at the end-user.

In our experiment, most expensive cost operations are pairing (6-ms) in G_T, exponentiations (6.2-ms) in G_1, and exponentiations (0.67-ms) in G_T. Moreover, for reading random from linux kernel (/dev/urandom), we measured 15-ms and 1.5-ms for G_1 and G_T, respectively. As light-weight arithmetic operations, we measured around 0.02-ms for both multiplication and division.

Fig. 2 (a) shows the *processing time* of the user regarding the number of attributes. We used the optimization methods detailed in Bethencourt et al. paper [1] to minimize the number of pairings and exponentiations in the access tree traverse. To avoid the domination of symmetric decryption (AES) over the ABE algorithm, we retained the small File Size as 50-KB. In the figure, the three more efficient schemes ([10, 17] and ours), have decryption outsourcing over the PCS. As it is clear from the figure, our proposed scheme is more efficient at the users, who may be resource-constrained while preserving privacy. Because, alongside the decryption outsourcing, our decentralized hidden policy solution reduces the $O(n)$-pairings (at previous relevant solutions [16, 10, 18]) to $O(n)$-multiplications.

Fig. 2 (b) shows the *communication overhead* of the user regarding the number of attributes. To have an exact comparison, only the cost of the ABE operations is calculated, and the AES encryption of the outsourced file is disregarded. In the figure, the communication overhead of the optimized version of our scheme is marked as *'Proposed-OP'*. The optimization is done through minimizing the transmitted bits (the whole ciphertext plus the blinded private-keys) to the PCS by the users. So, we assumed only downloading the obfuscated policy (disregarding the ciphertext). Then, after de-obfuscation, we only transmit

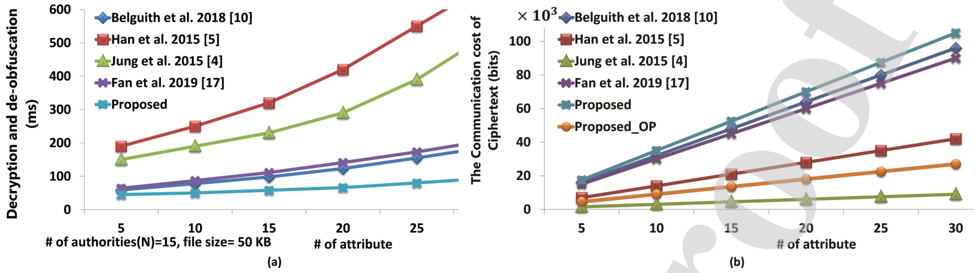

Figure 2 Experimental results. (a) Computation time (Decryption and De-Obfuscation) at the user. (b) The user's Communication Overhead.

the required parameters ($< \{(K_{1,i})^z, (K_{2,i})^z\}, {E_1}^z, {E_2}^z >$ and the tree T_{Obf}) to the PCS for the blind-decryption algorithm. Thus, the decryption outsourcing leads to significant computational efficiency (Fig. 2(a)) at the cost of a small increment in the communication cost (Fig. 2(b)) in our proposal. The schemes of [4] and [5] disregarded the decryption outsourcing (blind-decryption) which results in less communication-overhead, but a high computation cost of the users. Also, the interactive set-membership proof to ensure the attribute-anonymity leads to less efficiency of [5] than [4].

Finally, as the figures confirm, with the help of cloud resources, our scheme at the users, which may have the power limitations, has enough efficiency.

7. Applications

The ABE scheme has many interesting applications in the real world, such as Search over Encrypted Data, Social Networks, Forensic Analysis, Fog/Edge Computing, Targeted Broadcast Encryption [23], Selective Data Sharing [44], Group Key Management [44] and Content Addressable File System [45]. However, to enjoy the benefits, the applications must support the user's privacy and anonymity requirements. Our scheme ensures data customer's anonymity, along with guaranteeing the hidden policy privacy (ciphertext anonymity) in a decentralized environment. In the following, due to space limitations, we briefly show how our scheme properties meet the requirements of the first three applications.

Anonymous Search over Encrypted Data. As noted in [23], the basic ABE can be used to search over encrypted data. Our scheme can hide the set of attributes under which the data is encrypted, and then, can be used as an anonymous keywords search where the user's attributes are replaced with the keywords. When we use our construction for keyword search application, an index set I is encryption (obfuscation) of some keywords $\{w_1, w_2, ..., w_n\}$ that are extracted from arbitrary documents. In this case, the obfuscated tree (T_{Obf}) is the same as the set I. It is comprised of obfuscated keywords instead of the user's attributes. A user sends the trapdoor of keyword w (as $\mathcal{T}_w = H(Obf_w)$) to the PCS. The PCS run a function $Search(T_{Obf}, \mathcal{T}_w)$ to find a match. When a match is found, it will be notified to the user and the relevant document will be extracted. Similar to the usual keyword search, in this case, the PCS can not learn w from T_{Obf} or $\mathcal{T}_w$. However, the user knows that $\mathcal{T}_w$ is corresponding to the keyword w. To use the construction for both data access control and keyword search simultaneously, we need two trees, where the first one (tree T) includes the attributes and the second tree (such as tree $\bar{T}$) contains the keywords of the documents. Also, the users are anonymous (*identity-anonymity* and *attribute-anonymity*) while request the authority for keyword searches. To cover negative attributes and non-monotonic access structures, similar to [46], we can cover it at the cost of doubling the ciphertext size. Moreover, in our scheme, the anonymous search is partially and blindly done by the cloud computing power, which means minimizing the overhead of the user. We leave the specific extension for the application as future works.

Privacy Enhanced Social Networks. In social networks (Such as Facebook, Meetup, Bumble), people can find others with similar interests by publicly revealing personal information (e.g., name, sex, weight, height, address, interests, etc.). In an online dating service, a user can specify a policy for people who wish to access her profile. To be more specific, suppose Sara is only interested in people who are 'Male', 'Tall' and 'Wealthy' (e.g., Bob). Then, she enforces the policy P_{Sara}=('Male' AND 'Tall' AND 'Wealthy'), which can be provided by a basic ABE scheme [45]. However, assume Sara is not interested in revealing her policy for everyone. In this case, by hiding the policy (obfuscating the included attributes), our scheme guarantees this privacy requirement. Also, by using the applications with different settings to find people, Bob remains anonymous (The user's anonymity against various authorities).

Anonymous Forensic Analysis. The application of ABE systems for the 'Audit Log' analysis (as an essential part of the forensic analysis), is a well-studied topic [23]. But, as an extended application, let assume a joint plan of some corporations that jointly deliver their user's activity-logs to a cyber-security company for forensic analysis. Since the logs may disclosure essential knowledge for the enemies, the set of attributes (leaf-nodes of the tree policy) under which the data is encrypted should be hidden (the hidden policy property). Also, the analyst in the company has to request the keys from the corporations (also, different and decentralized authorities) in an anonymous manner to access the data (the user's anonymity). Our scheme, with the help of cloud powers, efficiently ensures the security and privacy requirements of the application.

8. Conclusions

In this paper, we present an anonymous and decentralized attribute-based access control with untrusted authorities in the standard security model where the users may collude with the untrusted authorities. To be applicable for cloud-assisted IoT networks with resource-constrained devices, the decryption cost outsourced over the cloud servers.

In the proposed scheme, the privacy of the user is entirely preserved. In particular, both of the *identity-anonymity* and the *attribute-anonymity* against the untrusted attribute authorities during private key generation are guaranteed. Moreover, *hidden policy*, as the privacy of an expressive access-structure against the PCS, is ensured through a novel distributed access structure obfuscating. So that, the PCS, in collusion with the untrusted authorities (at most $N-2$), can not learn the policy included in the ciphertext.

Finally, our security proofs and efficiency analysis are followed by briefly presenting the applications of our scheme in some fascinating real-world scenarios. The specific extensions for the applications are left for future works.

Acknowledgments

We thank anonymous reviewers for useful and constructive comments.

References

[1] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: 2007 IEEE symposium on security and privacy (SP'07), IEEE, 2007, pp. 321–334.

[2] A. Lewko, B. Waters, Decentralizing attribute-based encryption, in: Annual international conference on the theory and applications of cryptographic techniques, Springer, 2011, pp. 568–588.

[3] M. Chase, S. S. Chow, Improving privacy and security in multi-authority attribute-based encryption, in: Proceedings of the 16th ACM conference on Computer and communications security, ACM, 2009, pp. 121–130.

[4] T. Jung, X.-Y. Li, Z. Wan, M. Wan, Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption, IEEE transactions on information forensics and security 10 (1) (2015) 190–199.

[5] J. Han, W. Susilo, Y. Mu, J. Zhou, M. H. A. Au, Improving privacy and security in decentralized ciphertext-policy attribute-based encryption, IEEE transactions on information forensics and security 10 (3) (2015) 665–678.

[6] Q. Han, Y. Zhang, H. Li, Efficient and robust attribute-based encryption supporting access policy hiding in internet of things, Future Generation Computer Systems 83 (2018) 269–277.

[7] J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, in: Interna-

tional Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2001, pp. 93–118.

[8] Y. Ishai, E. Kushilevitz, Private simultaneous messages protocols with applications, in: Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems, IEEE, 1997, pp. 174–183.

[9] Y. Zhang, D. Zheng, R. H. Deng, Security and privacy in smart health: Efficient policy-hiding attribute-based access control, IEEE Internet of Things Journal 5 (3) (2018) 2130–2145.

[10] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, R. Attia, Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot, Computer Networks 133 (2018) 141–156.

[11] J. Katz, A. Sahai, B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in: annual international conference on the theory and applications of cryptographic techniques, Springer, 2008, pp. 146–162.

[12] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters, Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2010, pp. 62–91.

[13] J. Lai, R. H. Deng, Y. Li, Expressive cp-abe with partially hidden access structures, ASIACCS'12: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security 18 (2012) 146–162.

[14] H. Cui, R. H. Deng, J. Lai, X. Yi, S. Nepal, An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited, Computer Networks 133 (2018) 157–165.

[15] H. Xiong, Y. Zhao, L. Peng, H. Zhang, K.-H. Yeh, Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing, Future Generation Computer Systems 97 (2019) 453–461.

[16] J. Hur, Attribute-based secure data sharing with hidden policies in smart grid, IEEE Transactions on Parallel and Distributed Systems 24 (11) (2013) 2171–2180.

[17] K. Fan, H. Xu, L. Gao, H. Li, Y. Yang, Efficient and privacy preserving access control scheme for fog-enabled iot, Future Generation Computer Systems 99 (2019) 134–142.

[18] H. Zhong, W. Zhu, Y. Xu, J. Cui, Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage, Soft Computing 22 (1) (2018) 243–251.

[19] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, X. Shen, An efficient and fine-grained big data access control scheme with privacy-preserving policy, IEEE Internet of Things Journal 4 (2) (2017) 563–571.

[20] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, X. S. Shen, Fine-grained data access control with attribute-hiding policy for cloud-based iot, Computer Networks 153 (2019) 1–10.

[21] L. Nguyen, Accumulators from bilinear pairings and applications, in: Cryptographers' Track at the RSA Conference, Springer, 2005, pp. 275–292.

[22] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2005, pp. 457–473.

[23] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM conference on Computer and communications security, Acm, 2006, pp. 89–98.

[24] B. Qin, R. H. Deng, S. Liu, S. Ma, Attribute-based encryption with efficient verifiable outsourced decryption, IEEE Transactions on Information Forensics and Security 10 (7) (2015) 1384–1393.

[25] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, I. You, Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing, Information Sciences 379 (2017) 42–61.

[26] J. Li, Y. Wang, Y. Zhang, J. Han, Full verifiability for outsourced decryption in attribute based encryption, IEEE transactions on services computing (2017) 1–1.

[27] H. Wang, D. He, J. Han, Vod-adac: Anonymous distributed fine-grained access control protocol with verifiable outsourced decryption in public cloud, IEEE transactions on services computing 5 (4) (2017) 298–308.

[28] J. Li, X. Chen, S. S. Chow, Q. Huang, D. S. Wong, Z. Liu, Multi-authority fine-grained access control with accountability and its application in cloud, Journal of Network and Computer Applications 112 (2018) 89–96.

[29] W. Dai, Y. Doröz, Y. Polyakov, K. Rohloff, H. Sajjadpour, E. Savaş, B. Sunar, Implementation and evaluation of a lattice-based key-policy abe scheme, IEEE Transactions on Information Forensics and Security 13 (5) (2018) 1169–1184.

[30] T. Nishide, K. Yoneyama, K. Ohta, Attribute-based encryption with partially hidden encryptor-specified access structures, in: International conference on applied cryptography and network security, Springer, 2008, pp. 111–129.

[31] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in: International Workshop on Public Key Cryptography, Springer, 2011, pp. 53–70.

[32] I. Damgård, N. Triandopoulos, Supporting non-membership proofs with bilinear-map accumulators, IACR Cryptology ePrint Archive 2008 (2008) 538.

[33] D. Boneh, X. Boyen, Short signatures without random oracles, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2004, pp. 56–73.

[34] C.-P. Schnorr, Efficient signature generation by smart cards, Journal of cryptology 4 (3) (1991) 161–174.

[35] G. Persiano, I. Visconti, An efficient and usable multi-show non-transferable anonymous credential system, in: International Conference on Financial Cryptography, Springer, 2004, pp. 196–211.

[36] J. Camenisch, A. Lysyanskaya, Dynamic accumulators and application to efficient revocation of anonymous credentials, in: Annual International Cryptology Conference, Springer, 2002, pp. 61–76.

[37] J. Li, X. Huang, J. Li, X. Chen, Y. Xiang, Securely outsourcing attribute-based encryption with checkability, IEEE Transactions on Parallel and Distributed Systems 25 (8) (2013) 2201–2210.

[38] M. Green, S. Hohenberger, B. Waters, et al., Outsourcing the decryption of abe ciphertexts, in: USENIX Security Symposium, Vol. 2011, 2011.

[39] R. Canetti, H. Krawczyk, J. B. Nielsen, Relaxing chosen-ciphertext security, in: Annual International Cryptology Conference, Springer, 2003, pp. 565–582.

[40] F. Hao, P. Zieliński, The power of anonymous veto in public discussion, in: Transactions on Computational Science IV, Springer, 2009, pp. 41–52.

[41] D. Chaum, The dining cryptographers problem: Unconditional sender and recipient untraceability, Journal of cryptology 1 (1) (1988) 65–75.

[42] A. Kate, G. Zaverucha, I. Goldberg, Pairing-based onion routing, in: N. Borisov, P. Golle (Eds.), Privacy Enhancing Technologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 95–112.

[43] Ciphertext-policy attribute-based encryption toolkit. [online]. available: http://acsc.cs.utexas.edu/cpabe/ , accessed 2020.

[44] L. Cheung, C. Newport, Provably secure ciphertext policy abe, in: Proceedings of the 14th ACM conference on Computer and communications security, ACM, 2007, pp. 456–465.

[45] M. Pirretti, P. Traynor, P. McDaniel, B. Waters, Secure attribute-based systems, Journal of Computer Security 18 (5) (2010) 799–837.

[46] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: Proceedings of the 14th ACM conference on Computer and communications security, ACM, 2007, pp. 195–203.

Highlights:

- We propose a decentralized anonymous ABE access control for cloud-assisted IoT data sharing application.
- We present a user's privacy-preserving access control against untrusted attribute authorities. So, the untrusted authority can not guess or identify the user's identity during his lifetime.
- We propose a decentralized and pairing-free approach to ensure hidden policy as an improvement over the previous policy obfuscation approaches. So that, the untrusted cloud servers can not learn the policy even in collusion with the untrusted attribute authority.
- We introduce an efficient and verifiable method to outsource expensive computation cost over the untrusted cloud servers to make it efficient at the end-users.
- The formal proofs and experimental results confirm the security, privacy, and efficiency of the proposed scheme.
- We show how the properties of the proposal meet the requirements of some interesting nowadays applications.

***Hassan Nasiraee** is currently pursuing the Ph.D. degree in information security with the University of Isfahan, Isfahan, Iran. He had been selected as top students in Mathematical Olympiad at national wide. Research interests include applied mathematics, applied cryptography, Functional Encryption and their practical implementation.*

***Maede Ashouri-Talouki** is an assistant Professor of IT Engineering department of the University of Isfahan (Iran). She received her B.S. degree and M.S. degree in Computer Engineering from the University of Isfahan (Iran) in 2004 and 2007, respectively. In 2012, she received her Ph.D. degree at University of Isfahan in computer engineering. In 2013, she joined the University of Isfahan (Iran). Her research interests include mobile networks security, user privacy and anonymity, cryptographic protocols, distributed cryptography protocols and network security.*

MANUSCRIPT TITLE:

Anonymous Decentralized Attribute-based Access Control for Cloud-assisted IoT

AUTHORSHIP STATEMENT:

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. We certify that this paper consists of original, unpublished work which is not under consideration for publication elsewhere. Moreover, all authors have checked the manuscript and have agreed to the submission.

AUTHORSHIP CONTRIBUTIONS:

Conceptualization:	Hassan Nasiraee, Maede Ashouri-Talouki
Methodology:	Hassan Nasiraee, Maede Ashouri-Talouki
Software:	Hassan Nasiraee, Maede Ashouri-Talouki
Validation:	Hassan Nasiraee, Maede Ashouri-Talouki
Formal Analysis:	Hassan Nasiraee, Maede Ashouri-Talouki
Investigation:	Hassan Nasiraee, Maede Ashouri-Talouki
Resources:	Hassan Nasiraee, Maede Ashouri-Talouki
Data Curation:	Hassan Nasiraee, Maede Ashouri-Talouki
Writing - Original Draft:	Hassan Nasiraee
Writing - Review & Editing:	Hassan Nasiraee, Maede Ashouri-Talouki
Supervision:	Maede Ashouri-Talouki
Project Administration:	Maede Ashouri-Talouki

With the best of our knowledge, we have not any conflict of interest. Thanks