

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

6-2021

Efficient attribute-based encryption with repeated attributes optimization

Fawad KHAN

Hui LI

Yinghui ZHANG

Singapore Management University, yinghuizhang@smu.edu.sg

Haider ABBAS

Tahreem YAQOOB

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Information Security Commons](#)


Citation

KHAN, Fawad; LI, Hui; ZHANG, Yinghui; ABBAS, Haider; and YAQOOB, Tahreem. Efficient attribute-based encryption with repeated attributes optimization. (2021). *International Journal of Information Security*. 20, (3), 431-444.

Available at: https://ink.library.smu.edu.sg/sis_research/7117

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Efficient attribute-based encryption with repeated attributes optimization

Fawad Khan^{1,2}  · Hui Li¹ · Yinghui Zhang^{3,4} · Haider Abbas² · Tahreem Yaqoob²

Abstract

Internet of Things (IoT) is an integration of various technologies to provide technological enhancements. To enforce access control on low power operated battery constrained devices is a challenging issue in IoT scenarios. Attribute-based encryption (ABE) has emerged as an access control mechanism to allow users to encrypt and decrypt data based on an attributes policy. However, to accommodate the expressiveness of policy for practical application scenarios, attributes may be repeated in a policy. For certain policies, the attributes repetition cannot be avoided even after applying the boolean optimization techniques to attain an equivalent smaller length boolean formula. For such policies, the evaluated secret shares are also multiple for repeated attributes; hence, the ciphertext computed for those irreducible policies is long and computational effort is more. To address this issue, a new CP-ABE scheme is proposed which employs our Repeated Attributes Optimization algorithm by which the Linear Secret Sharing Scheme matrix sent along with ciphertext will contain the access structure of policy including attributes appearing multiple times, but the ciphertext will only be evaluated for unique non-repeated attributes. Security and performance analysis show that the proposed construction fulfils its goals of achieving desired security with low communication overhead and computational cost for resource-constrained devices.

Keywords Optimization · Cloud computing · Internet of things

1 Introduction

Internet of Things (IoT) is a framework where ubiquitous sensors (or devices) are connected to physical world via the Internet [1,2]. The three core components of IoT consist of devices, internet and connectivity [3]. IoT creates exciting new application scenarios where data collection intersects data analysis including making real time decisions based on the collected data. However, the resource-constrained smart

devices have problems in handling the enormous amount of data generated in the era of Big Data.

On the other hand, cloud computing [4–6] has emerged to provide on-demand, scalable access to resources as a service. Therefore, to fully utilize the benefits provided by cloud computing, most IoT applications outsource their data to cloud infrastructure. However, due to privacy and security concerns of data in the cloud, the data are encrypted prior to outsourcing it and access control is enforced to retain data ownership.

Attribute-based encryption (ABE) [7] has evolved as a potential candidate to provide data access in cloud computing systems. To assure data security, the data owner constructs a ciphertext over an access control policy for a set of user attributes in Ciphertext Policy-ABE (CP-ABE). User has access to data if the attributes in user key satisfy the policy in ciphertext. Boolean formulas or access tree structures are mostly preferred by ordinary users to represent policies because of their simplicity, whereas the highly expressive and provably secure CP-ABE schemes [8–12] require LSSS matrices for enforcing policies in ciphertexts. For CP-ABE using LSSS, policy is not sent in plaintext; instead an LSSS matrix (A, ρ) is evaluated for the policy and sent along

✉ Fawad Khan
fawadkhan@mcs.edu.pk

¹ State Key Laboratory of Integrated Service Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, Shaanxi, People's Republic of China
² National University of Sciences and Technology, Islamabad 44000, Pakistan
³ National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, People's Republic of China
⁴ School of Information Systems, Singapore Management University, Singapore, Singapore

with the ciphertext. Lewko and Waters detailed an algorithm [9], for converting any monotone boolean access formula to corresponding LSSS matrix. For all highly expressive and proven secure constructions [8–12], ciphertext size linearizes correspondingly to the associated rows of LSSS matrix. Moreover, nature of LSSS matrix relies completely on the underlying policy for which it is constructed, i.e., if any attribute is repeated in policy its image will also appear in LSSS matrix. Attributes repetition in LSSS matrix directly affects the encryption operation of data owner, i.e., he has to compute the ciphertext for repeated attributes multiple times which leads to large ciphertext size and computational cost.

Disjunctive normal form (DNF) policies can be arbitrarily expressed as boolean formulas containing attributes appearing once, or even multiple times. Consider a scenario where an enterprise and a firm have a collective database which is updated by both the staff of enterprise and firm. The data need to be shared between the staff of enterprise, firm and the BoG with the policy: $(\text{Staff} \wedge (\text{Enterprise} \vee \text{Firm})) \vee (\text{BoG} \wedge \text{Enterprise} \wedge \text{Firm})$. There is a repetition of two attributes if we express the policy in any minimal representation.

Another scenario in which a patient suffering from a cardiac and nasal polyp disease admitted at hospital with his reports being shared with Cardiologist, Otolaryngologist and a nurse on duty in-charge of patient with the policy specified by: $(\text{Hospital} \wedge \text{Doctor} \wedge (\text{Cardiologist} \vee \text{Otolaryngologist})) \vee (\text{Nurse} \wedge \text{Cardiology} \iff \text{Cardiologist} \wedge \text{Nasal polyp} \iff \text{Otolaryngologist})$. As any specific disease is treated by its specialist doctor, both $(\text{Cardiology} \iff \text{Cardiologist})$ refer to the same attribute. In this scenario, the attributes Cardiologist and Otolaryngologist are repeated twice in policy. For both scenarios (of irreducible policies) discussed above, due to repetition of attributes in policy the corresponding LSSS matrix will contain a replica of repeated attributes, and so the generated attribute shares are multiple leading to enhanced computation cost and ciphertext size. Generally, an irreducible policy is that for which the attributes repetition cannot be avoided even after applying the optimization algorithms to obtain an equivalent smaller length boolean formula.

In real life, people working in different organizations have affiliations pertaining to those organizations, and their specific secret keys can be generated by attribute authorities relating to those organizations. A particular instance can be a doctor, nurse belonging to a hospital attribute authority and an enterprise employees belonging to its respective attribute authority. In that case, the authorities belonging to hospital and enterprise should be working independently in a decentralized fashion to generate the public parameters for attributes and secret keys for users attributes. Moreover, the parameters should be set in a manner that any particular data owner can be able to define a policy over attributes belonging to multiple distant attribute authorities.

Ensuring privacy of an individual or entity is an important parameter in the design of an interconnected system. This privacy breach can be alarming in the context of IoT, where resource-constrained devices are used to regularly monitor data and send it to the specified entity. In the context of a hospital, several IoT-enabled sensors are connected to the patient to monitor his disease symptoms, and the data are shared only with the specified doctors and nurses for treatment purposes. The patient’s privacy can be ensured by specifying an access control policy over IoT-enabled sensors data, where only legitimate doctors and nurses on duty in-charge of the patient can have access to his data.

Based on a particular application or scenario, attributes may be repeated in policy. Due to repetition of attributes, the ciphertext is computed for repeated attributes multiple times elongating the ciphertext size and its evaluation cost. Hence, there is a need to explore this issue and devise an algorithm which lessens the computational cost and size of ciphertext even when attributes appear multiple times in a policy.

Priorly, several constructions addressed the issues of attributes repetition in different contexts. The author’s [9] proposed that the repeated attributes should be replaced by k new replicas, where each replication of attribute is considered as a new attribute. This formulation can be applied in security proof as well because the security proof restricted that attributes should not be repeated in policy or access matrix. Moreover, an algorithm is proposed for constructing an LSSS matrix to reduce the ciphertext size for threshold-based policies [13]. Lewko and Water [14] employed dual system encryption technique for proving their multi-use attributes construction secure in the standard model. Takashima [15] proposed a CP-ABE with non-redundant key components for multi-use of attributes. All of the schemes either improved performance or elaborated the security versus performance tradeoff [8]; however, none of the construction has focused to address the issue of repeated attributes in the context of an irreducible policy. We propose a repeated attributes optimization (RAO) algorithm to address the issue for an irreducible policy. The intuition behind RAO for optimizing the evaluated attributes shares is to re-randomize some shares values by exploiting the secret reconstruction property $\sum c_x \lambda_x \in W_i = s$. The satisfaction of any attribute set W_i of policy by the user leads to the reconstruction of the same secret s and formally access to same data; hence, this fact is utilized for optimization.

1.1 Research contributions

In this paper, the authors’ motivation is to address attributes repetition in expressive LSSS-based irreducible policies which leads to enhanced computational cost in encryption and ciphertext size. Table 1 shows a comparison of schemes

Table 1 Comparison of attributes optimization for an irreducible policy (Ir-Pol) containing repeated attributes

Scheme	Access structure	Policy type	Ir Pol	Proposed remedy	Reduced cost
[9]	LSSS-based	Expressive	No	Replicated with a different name	No
[12]	LSSS-based	Expressive	No	Replicated with a different name	No
[13]	Threshold-based	Less expressive	Yes	Algorithm for small size threshold matrix	Yes
[21]	LSSS-based	Expressive	No	Limited to multi-message CP-ABE	No
This work	LSSS-based	Expressive	Yes	Optimization of repeated attributes shares	Yes

based on addressing the issue of repeated attributes and their proposed remedies.

The main contributions of our work are summarized as:

- An efficient decentralized multi-authority CP-ABE with repeated attributes optimization is proposed. The scheme employs our proposed repeated attributes optimization (RAO) algorithm in encryption algorithm that takes as input the LSSS matrix and its corresponding attribute’s shares for irreducible policy and returns an optimized single share value of each repeated attribute. Hence, RAO algorithm significantly reduces the encryption cost at owner for an irreducible policy and the communication cost by reducing the ciphertext size.
- The proposed scheme is proven to be secure against Chosen Plaintext Attack (CPA) in the Generic Gro-up Model. Performance evaluation depicts the effectiveness of the proposed scheme in comparison to existing standard constructions.
- Comprehensive performance analysis depicts the effectiveness of the proposed scheme. The scheme can be a possible solution for resource-constrained IoT devices by reducing the computation and communication cost.

1.2 Related work

The first CP-ABE construction was put forward by Bethencourt et al. [7], supporting monotonic access structure. Key Policy - ABE (KP-ABE) for the first time was proposed by Goyal et al. [16]. Kapadia et al. [17] proposed a hidden policy CP-ABE; however, it suffered from collusion. Nishade [18], then proposed a hidden policy CP-ABE based on less expressive AND-based access structures. Mellisa and Chase [19] proposed the first multi-authority KP-ABE scheme where multiple authorities were responsible to manage, distribute attributes and to assign attribute keys to users. First, CP-ABE multi-authority scheme was proposed by Muller et al [20]. The scheme required a central coordination authority to manage other authorities unless the first decentralized CP-ABE [9], was proposed by Lewko and Water.

CP-ABE schemes currently support And-based [18], Threshold-based [13] and the most expressive LSSS-based

access structures [9,21]. Lewko and Water’s proposed an algorithm [9] to provide ordinary users with capability for converting any monotone boolean access formula to corresponding LSSS matrix using AND-OR gate tree. A boolean access formula can contain repeated attributes with their image appearing in corresponding LSSS matrix leading to large communication and computation cost.

Prior approach to address the repeated attributes issue were [9,12] which stated that repeated attribute X should be replaced by k new “attributes” $X : 1, \dots, X : k$, and these were controlled by the same authority that controlled X . For policy $(A \wedge (B \vee C) \vee (B \wedge C \wedge D))$ containing attributes “ B, C ” twice; the attributes “ B, C ” are replaced by “ $B1, B2$ ” and “ $C1, C2$ ” respectively. The policy is rewritten with distinct attribute names as $(A \wedge (B1 \vee C1) \vee (B2 \wedge C2 \wedge D))$. Considering only attributes “ $B1, B2$ ” in this case, the public parameters for $B1$ and $B2$ are different and generated separately for each of them by the authority; hence, the evaluated ciphertext corresponding to them is also different. Moreover, in the LSSS access matrix, the attributes $B1$ and $B2$ are explicitly mentioned and for any user satisfying the policy should have the required decryption key explicitly for $B1$ or $B2$ from the authority instead of just B from the authority. Drawback of this approach is the large size of public parameters, more ciphertext evaluation cost and its size.

To cater the above issue, the authors in [13] proposed an algorithm by extending the work in [9] to support threshold policy by incorporating threshold gate (t, n) access tree, thereby obtaining smaller size LSSS matrices with no attributes repetition. The limitation of this approach is that it benefited only in reducing the ciphertext size and computational operations for threshold gate policies only. Khan et al. [21], have proposed a basic algorithm for removing repeated attributes shares from LSSS matrix of multi-message CP-ABE. However, the applicability of algorithm is limited to policies only for particular scenarios employing multi-message CP-ABE. Moreover, it cannot be scaled to cater irreducible policies.

Another issue focused by researchers is the security versus efficiency tradeoff by repetition of attributes. Water [8] presented schemes to be secure under different security

assumptions at the expense of an increase in public parameters and secret key sizes by lifting the restriction of attributes appearing only once in policy. Further, Lewko and Water [14] eliminated the efficiency loss by incorporating the concept of dual system encryption and proved the scheme to be secure in the standard model by relying on q-based complexity assumption. Recently, Takashima [15] proposed adaptively secure KP-ABE and CP-ABE constructions under a static assumption. The KP-ABE and CP-ABE schemes correspondingly achieved the non-redundant ciphertext and key components for multi-use attributes employing the CNF formula for policy. Besides the introduction of a new sparse matrix; the Dual Pairing Vector Spaces (DVPS) along with Inner Product Encryption (IPE) are employed. In comparison to existing constructions [8,14], the public and secret key sizes also depend on the number of columns of the access matrix besides the number of involved attributes.

The rest of the paper is organized as follows. Section 2 presents cryptography background, while Sect. 3 describes the definitions of system and security model. The proposed construction is presented in Sect. 4. In Sect. 5, the security and performance analysis of the proposed scheme are detailed, while Sect. 6 concludes the paper.

2 Preliminary background

The overview of bilinear map, LSSS and access structures is presented in this section.

2.1 Bilinear map

Let two multiplicative cyclic groups be G and G_T of prime order p where g is a generator of group G . There exists a bilinear map $e : G \times G \rightarrow G_T$ between the groups with the following properties.

- 1) Bilinearity: $e(g^a, g^b) = e(g, g)^{ab} \forall a, b \in \mathbb{Z}_p, g \in G$
- 2) Non-degeneracy: $e(g, g) \neq 1$
- 3) Computable: There must be an algorithm to efficiently compute $e(g, g) \forall g \in G$

2.2 Access structures

Definition 1 Access Structure [22]. For a set of parties $\mathcal{P}'_1, \dots, \mathcal{P}'_n$, collection $\mathcal{L}' \subseteq 2^{\{\mathcal{P}'_1, \dots, \mathcal{P}'_n\}}$ is monotone if $\forall \mathcal{M}', \mathcal{N}' : \text{if } \mathcal{M}' \in \mathcal{L}' \text{ and } \mathcal{M}' \subseteq \mathcal{N}'$, then $\mathcal{N}' \in \mathcal{L}'$. Monotone access structure is a collection of non-empty subsets of $\mathcal{P}'_1, \dots, \mathcal{P}'_n$. Sets in \mathcal{L}' are authorized sets.

Attributes are equivalent to parties with consideration of monotone access structures only, in this work.

2.3 Linear Secret Sharing Scheme

The proposed scheme utilizes Linear Secret Sharing Scheme (LSSS) [22].

Definition 2 A secret sharing scheme Π is linear for a set of parties \mathcal{P}' if a vector over \mathbb{Z}_p is formed by combining shares from all \mathcal{P}' . Moreover, a share generating matrix A with m rows and n columns exists for Π where x^{th} row in matrix A maps to party $\mathcal{P}'(x)$. A sharing vector $\mathbf{v} = \{s, v_2, \dots, v_n\} \in \mathbb{Z}_p^{\mathcal{R}}$ exists, such that $s \in \mathbb{Z}_p^{\mathcal{R}}$ is the shared secret. The product $A \cdot \mathbf{v}$ forms a vector of m shares for s according to Π . For each party $\mathcal{P}'(x)$, its share is evaluated by $\lambda_x = (A \cdot \vec{v})_x$.

Here, \mathcal{X}' indicates an attributes set and $\mathcal{Y}' \subseteq \{1, 2, \dots, m\}$ as $\mathcal{Y}' = \{x | \mathcal{P}'(x) \in \mathcal{X}'\}$. A vector $(1, 0, \dots, 0)$ exists in the span of A_x indexed by \mathcal{Y}' . To linearly reconstruct, the constants of the form $\{c_x \in \mathbb{Z}_p\}_{x \in \mathcal{Y}'}$ exists so that, if λ_x are valid secret shares of s accordingly for Π , then “ s ” can be reconstructed by $\sum_{x \in \mathcal{Y}'} c_x \lambda_x = s$.

3 System and security model

3.1 System model

The system model of a repeated attributes optimization-based CP-ABE scheme is shown in Fig. 1. It consists of Cloud Service Provider (CSP), Attribute Authorities (AA), owner and user.

- **CSP**: It is an entity for providing computational and storage services to users. It is a semi-trusted entity which follows the protocol but curious about learning the encrypted data placed over it.
- **AA**: Attribute authorities ensure the access control mechanism by providing decryption keys to users based on their attributes. Each attribute authority generates the Public Key (PK) and Secret Key (SK) parameters for every attribute in its domain. The public key parameters are utilized by data owners to encrypt data under an attributes policy while secret key parameters by AA to generate user decryption keys based on their identities GID and attributes possessed by them. All AA work in a decentralized fashion without coordination in between them.
- **Owner**: Data owners are resource-constrained devices who encrypt their data and define an access control policy to outsource data to CSP.
- **User**: An entity who wish to retrieve and access data based on the access privileged granted to him based on the attributes he possess. Users may collude with each other to have access to data that they are not entitled to have individually.

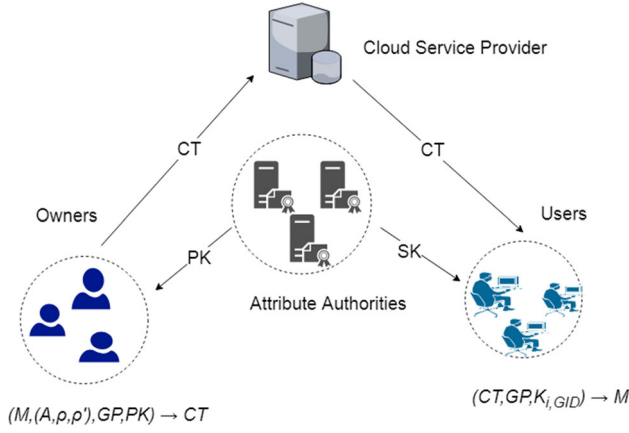


Fig. 1 System model

Our proposed scheme consists of the following algorithms.

- **Global Setup** $(\lambda) \rightarrow GP$: It takes as input a security parameter λ to output the global parameters GP .
- **Authority Setup** $(GP) \rightarrow SK, PK$: Each AA runs this algorithm by taking as input GP to return the secret, public key parameters SK, PK .
- **Encrypt** $(M, (A, \rho, \rho'), GP, PK) \rightarrow CT$: Data owner runs this algorithm by taking message M, GP, PK and access structure (A, ρ, ρ') for the policy as input to output ciphertext CT . A, ρ is the exact representation of an irreducible policy expressed using LSSS matrix A with ρ mapped to the rows of A , while ρ' contains the unique non-repeated attribute names appearing in ρ or in irreducible policy. Moreover, the length of ρ' is than ρ , and it is mapped to CT .
- **KeyGen** $(GID, GP, l, SK) \rightarrow K_{l, GID}$: AA runs this algorithm. It takes the user identity GID, GP, SK as input to generate the output key $K_{l, GID}$ corresponding to user attribute l .
- **Decrypt** $(CT, GP, \{K_{l, GID}\}) \rightarrow M$: User gets access to data by employing this algorithm. Taking key, CT and GP as input, it returns M if the key satisfy access structure in CT .

3.2 Security model

We consider the following indistinguishable game against the Chosen Plaintext Attacks (CPA) between an adversary \mathcal{A} and challenger \mathcal{C} . In security game, \mathcal{A} corrupts authorities statically but can make key queries adaptively.

- **Setup**: The global setup algorithm is run. Out of the total authorities S in the system, adversary \mathcal{A} specifies a subset of corrupt authorities $S' \subset S$. Challenger \mathcal{C} then obtains

the public and private keys of good (non-corrupt) $S - S'$ authorities by running authority setup and gives the public keys to \mathcal{A} .

- **Phase 1**: Adversary \mathcal{A} is given access to Key Generation Oracle (KGO). \mathcal{A} issues queries for good authorities keys (i, GID) corresponding to attribute, i and identity GID . Challenger replies to adversary with the keys $K_{i, GID}$.
- **Challenge**: \mathcal{A} then specifies two messages M_0, M_1 and a challenge access structure (A, ρ, ρ') under the restriction that union of queries made in Phase 1 and the keys possessed by \mathcal{A} for corrupt authorities should not include a span of $(1, 0, \dots, 0)$ in challenged access structure. Also, \mathcal{A} shares the public keys of corrupt authorities attributes appearing in challenged access structure. Challenger \mathcal{C} chooses randomly $\beta \in \{0, 1\}$ and sends an encrypted M_β under (A, ρ, ρ') to \mathcal{A} .
- **Phase 2**: \mathcal{A} makes further key queries (i, GID) similar to Phase 1, unless the constraint of challenged (A, ρ, ρ') is not violated.
- **Guess**: Adversary \mathcal{A} outputs a guess β' for β . Advantage of \mathcal{A} in the game is $Pr[\beta = \beta'] - \frac{1}{2}$.

Definition 3 The repeated attributes optimization-based multi-authority ciphertext policy-attribute-based encryption scheme is secure (against static corruption of authorities), if all polynomial bounded adversaries have at most a negligible advantage against challenger in above security game.

4 Proposed efficient CP-ABE with repeated attributes optimization

In this section, we detail our proposed decentralized multi-authority CP-ABE with repeated attributes optimization (CP-ABE-RAO) scheme.

4.1 Main idea

A DNF policy [23] can be generically represented as

$$P = \bigvee_{j=1}^N \left(\bigwedge_{X \in W_j} X \right)$$

where N attribute sets W_1, W_2, \dots, W_N denote attributes that occur in the j -th conjunction of P . An irreducible policy $IR - Pol$ with arbitrary attributes (A, B, C, D, E) for elaboration purpose can be written generically as: $IR - Pol = (A \wedge B) \wedge (C \vee D) \vee (C \wedge D \wedge E)$. For this particular policy, there are three attribute sets, namely $W_1 = \{A, B, C\}$, $W_2 = \{A, B, D\}$, $W_3 = \{C, D, E\}$. The attributes (C, D) are repeated twice in policy because they appear in more than one attribute set.

In traditional CP-ABE schemes, the data access is provisioned subject to the satisfaction of any attribute set W_i in policy. Hence, different user's having variant sets of attributes and satisfying the different W_i in policy will have access to same data, because a single secret s is shared for all the different W_i in policy.

The intuition behind RAO for optimizing the evaluated attributes shares is to re-randomize some shares values by exploiting the secret reconstruction property $\sum c_x \lambda_x \in W_i = s$. As the satisfaction of any attribute set W_i of policy by the user leads to the reconstruction of the same secret s and formally access to same data; hence, this fact is utilized for optimization. The repeated attributes share values belonging to different attribute sets of policy are optimized by fixing some shares values to a constant, while changing the others with secret s fixed for all W_i , such that after optimization they combine to reconstruct the secret s again. A necessary condition for optimization is that after performing it, unauthorized attribute sets should not be able to reconstruct the secret, i.e., perfect secret sharing condition should be maintained. Hence, the optimization can be performed to help reduce computation and communication cost for an irreducible policy in which both repeated and non-repeated attributes appear in a particular attribute set. For the particular example of policy $IR - Pol$, the attribute set W_3 contains attributes C, D, E among which the attributes C, D are repeated attributes which appear in W_1, W_2 as well, while E is non-repeated attribute which appears only in W_3 . Hence, for policies of this type the optimization can be performed for attribute E by fixing shares for attributes C and D . In fact, to the best of authors' knowledge, it is an open problem to perform attributes optimization when no optimization variables or non-repeated attributes are present with repeated attributes in an irreducible policy.

4.2 CP-ABE-RAO scheme

The proposed scheme consists of the following algorithms.

- **Global Setup**(λ) $\rightarrow GP$: In global setup, a bilinear group G of prime order p' is chosen. Global parameters are set to p', g and H ; where g is a generator of group G and H is a hash function that maps global identities GID to elements in G .
- **Authority Setup**(GP) $\rightarrow SK, PK$: Each authority selects for itself a random value $r \in Z_p$. For each attribute l that belongs to authority in attribute universe, it chooses a random value $\beta_l \in Z_p$. It keeps values $\{r, \beta_l \forall l\}$ as secret key, SK and publishes $\{g^r, e(g, g)^{\beta_l \forall l}\}$ as public key, PK .
- **Encrypt**($M, (A, \rho, \rho'), GP, PK$) $\rightarrow CT$: For encryption, firstly the access policy is converted into LSSS matrix A . The algorithm takes as input a message M ,

global parameters, an access matrix A of size $m \times n$ with ρ containing a map of its rows to attributes, and PK 's from relevant authorities. Also it takes as input ρ' which indicates the list of non-repeated distinct attributes appearing in ρ . Then, it chooses a random encryption exponent $s \in Z_p$, and $v \in Z_p^n$, where v is a column vector of length n and contains s as its first entry. Then, it computes $\lambda_x = A_x \cdot v$, where A_x is x^{th} row of A . Moreover, it choose a random vector $w \in Z_p^n$ of length n with secret $s' = 0$ as its first entry. Compute $w_x = A_x \cdot w$. Algorithm 1 summarizes the steps carried out for optimization.

The repeated attributes optimization (RAO) algorithm takes as input $A, \rho, \rho', s, \lambda_x, s', w_x$ and the number of attribute sets W_i in policy and proceeds as follows.

Lines 1-3 The coefficients c_x corresponding to attributes $\rho(x)$ belonging to all attribute sets W_i in policy are computed by the relation:

$$\sum_x c_x A_x = (1, 0, \dots, 0)$$

Lines 4-12 Counter variable $count_{\rho'(t)}$ is set to zero for all distinct non-repeated attribute names appearing in ρ' . Starting with the first attribute set and traversing through all of the them, the occurrences of all attributes appearing in attribute sets $\rho(x) \in W_i$ are counted by incrementing $count_{\rho'(t)}$ variables. This is used to record the repetition of attributes appearing multiple times in various different W_i .

Lines 13-21 Variable Add_i is initialized to 0 for every attribute sets W_i of policy and is incremented with $count_{\rho'(t)}$ for $\rho(x) \in W_i$ and $\rho(x) == \rho'(t)$ by satisfying the following relation:

$$Add_i = \left(\sum_{\rho(x) \in W_i} Count_{\rho'(t)} \text{ if } \rho(x) == \rho'(t) \right)$$

Notations: We represent attribute shares λ_x falling into two categories, namely: (1) fixed or optimized share $\lambda_{x-optimized}$, i.e., whose value have been fixed (set to a constant value). (2) Other share $\lambda_{x-other}$, i.e., whose optimized value is yet to be determined. Once the value is determined the status of attribute share is changed from $\lambda_{x-other}$ to $\lambda_{x-optimized}$. Moreover, we denote array K is a 3 dimensional array which keeps a record of optimized variable name, its $\lambda_{x-optimized}$ share value, and $w_{x-optimized}$ optimized share value.

Lines 22-32 To perform less optimization steps, the algorithm figures out that attribute set W_{lmax}

Algorithm 1 Repeated attributes optimization (RAO)

Require: A \triangleright LSSS access matrix for policy
Require: ρ \triangleright Mapping of attribute names to rows of A
Require: ρ' \triangleright Non-repeated attribute names of ρ
Require: W_i \triangleright Attribute sets in policy for (A, ρ)
Require: $s \in Z_p$ \triangleright Secret shared
Require: $s' = 0$ \triangleright Secret shared
Require: λ_x \triangleright Evaluated attribute shares for s wrt ρ
Require: w_x \triangleright Evaluated attribute shares for s' wrt ρ
Ensure: λ_t \triangleright Optimized attribute shares for s wrt ρ'
Ensure: w_t \triangleright Optimized attribute shares for s' wrt ρ'

- 1: **for all** $\rho(x) \in W_i$ **do**
- 2: compute c_x
- 3: **end for**
- 4: **for all** $\rho'(t)$ **do**
- 5: $count_{\rho'(t)} \leftarrow 0$
- 6: **end for**
- 7: \triangleright Count repeated attributes occurrences
- 8: **for all** $\rho(x) \in W_i$ **do**
- 9: **if** $\rho(x) == \rho'(t)$ **then**
- 10: $++ count_{\rho'(t)}$
- 11: **end if**
- 12: **end for**
- 13: \triangleright Evaluate W_i with largest # of repeated attributes
- 14: **for all** W_i **do**
- 15: $Add_i \leftarrow 0$
- 16: **end for**
- 17: **for all** $\rho(x) \in W_i$ **do**
- 18: **if** $\rho(x) == \rho'(t)$ **then**
- 19: $Add_i \leftarrow Add_i + count_{\rho'(t)}$
- 20: **end if**
- 21: **end for**
- 22: \triangleright Fix share values of a W_i with largest Add_i
- 23: **return** $lmax \leftarrow \text{index}(\text{maximum}(Add_i))$
- 24: $K = \{\}$ \triangleright Initialize a 3D array K of length ρ'
- 25: **for** $\rho(x) \in W_{lmax}$ **do**
- 26: $\lambda_{x-optimized} \leftarrow \lambda_x$
- 27: $w_{x-optimized} \leftarrow w_x$
- 28: $K_{j,1} \leftarrow \rho(x)$
- 29: $K_{j,2} \leftarrow \lambda_{x-optimized}$
- 30: $K_{j,3} \leftarrow w_{x-optimized}$
- 31: $K++$ \triangleright Increment K index by 1
- 32: **end for**
- 33: \triangleright Perform optimization (re-randomization of shares) for other W_i (excluding W_{lmax})
- 34: **for all** $\rho(x) \in W_{i,i \neq lmax}$ $\rho(x) \neq K$ **do**
- 35: $\lambda_{x-other} = (1/c_{x-other})(s - \sum_{x \in W_i, K} c_x \lambda_x)$
- 36: $\lambda_{x-optimized} \leftarrow \lambda_{x-other}$
- 37: $w_{x-other} = (1/c_{x-other})(s - \sum_{x \in W_i, K} c_x w_x)$
- 38: $w_{x-optimized} \leftarrow w_{x-other}$
- 39: $K_{j,1} \leftarrow \rho(x)$
- 40: $K_{j,2} \leftarrow \lambda_{x-optimized}$
- 41: $K_{j,3} \leftarrow w_{x-optimized}$
- 42: $K++$ \triangleright Increment K index by 1
- 43: **end for**
- 44: **if** $K_{*,1} == \rho'(t)$ **then**
- 45: $\lambda_t = K_{*,2}$
- 46: $w_t = K_{*,3}$
- 47: **end if**

which has the largest Add_i value or the greater repetition count. After determining W_{lmax} with largest Add_i , then fix its original attributes shares λ_x, w_x values to optimized values $\lambda_{x-optimized}, w_{x-optimized}$ and also appended to array K . These optimized-shares values $\lambda_{x-optimized}, w_{x-optimized}$ will replace the original λ_x in all other attribute sets W_i where these repeated attributes existed.

Lines 33-43 The optimization of all other-shares attributes $\lambda_{x-other}, w_{x-other}$ in other different W_i (not including W_{lmax}) is performed by employing the following relations:

$$\lambda_{x-other} = (1/c_{x-other})(s - \sum_{x \in W_i, K} c_x \lambda_x)$$
$$w_{x-other} = (1/c_{x-other})(s - \sum_{x \in W_i, K} c_x w_x)$$

Each new $\lambda_{x-optimized}, w_{x-optimized}$ is appended to array K . The optimization process completes when all attribute shares are optimized.

Lines 44-47 All the optimized values corresponding to attribute names in $\rho'(t)$ are assigned to λ_t and w_t .

Thereafter, it computes the ciphertext CT (for optimized new shares λ_t, w_t) as:

$$CT = \{C_0 = M \cdot e(g, g)^s, C_{1,t} = e(g, g)^{\lambda_t} \cdot e(g, g)^{\beta_{\rho'(t)} w_t}, C_{2,t} = g^{r w_t} \text{ for } t = \{1, 2, \dots, n'\}\}$$

This ciphertext CT is sent along with (A, ρ, ρ') to the cloud server, where A, ρ refers to LSSS matrix indicating actual policy with repeated attributes, and ρ' refers to optimized non-repeated attributes used for CT evaluation. In existing CP-ABE schemes [8,9], A, ρ and CT are mapped to each other; in-contrast as we have removed the repeated attributes occurrences in CT while still enforcing an irreducible policy, so here A is mapped to ρ , and ρ' is mapped to CT .

– **KeyGen** (GID, GP, l, SK) $\rightarrow K_{l,GID}$: To create a key for user GID corresponding to an attribute l of authority, it computes:

$$K_{l,GID} = g^{\beta_l/r} \cdot H(GID)^{1/r}$$

– **Decrypt** ($CT, GP, \{K_{l,GID}\}$) $\rightarrow M$:

To decrypt, the user will first determine which of his attributes satisfy the policy, and the index of ciphertexts components corresponding to those attributes. Algorithm 2 details the procedure.

The RAO-Check algorithm takes as input \mathbf{A} , ρ , ρ' , attribute sets W_i in policy, and decryption user attribute set S_{att} and proceeds as follows:

Algorithm 2 RAO Check

Require: \mathbf{A} \triangleright LSSS access matrix for policy
Require: ρ \triangleright Mapping of attribute names to rows of \mathbf{A}
Require: ρ' \triangleright Unique (non-repeated) attribute names of ρ
Require: W_i \triangleright Attribute sets in policy for (\mathbf{A}, ρ)
Require: S_{att} \triangleright User attribute set
Ensure: t \triangleright Location of Ciphertext CT components for user attributes in $\rho'(t)$ satisfying policy
1: \triangleright Determine user attributes satisfying policy
2: $S'_{att} \leftarrow Null$
3: **for all** W_i **do**
4: **if** $W_i \subseteq S_{att}$ **then**
5: $S'_{att} \leftarrow W_i$
6: **end if**
7: **end for**
8: **if** $S'_{att} \leftarrow Null$ **then**
9: “Abort” the program: Policy is not satisfied
10: **end if**
11: \triangleright Compute coefficient c_x values satisfying policy
12: **for all** $\rho(x) \in S'_{att}$ **do**
13: compute c_x
14: **end for**
15: **return** c_x
16: \triangleright Return location and attribute names
17: **for all** $\rho(x) \in S'_{att}$ **do**
18: **if** $\rho(x) == \rho'(t)$ **then**
19: **return** $t, \rho'(t)$
20: **end if**
21: **end for**

Lines 1-10 If any attribute set W_i of policy is subset of user attribute set S_{att} ($W_i \subseteq S_{att}$), then user attributes qualifying the policy are those attributes of that particular W_i ; otherwise, the policy is not satisfied, and user is not privileged for data access.

Lines 11-15 For user attributes S'_{att} satisfying the policy, compute and return the coefficients c_x from the relation $\sum_x c_x A_x = (1, 0, \dots, 0)$.

Lines 16-21 For each of his attribute in ρ satisfying policy, it will first check for condition where $\rho(x) == \rho'(t)$; then corresponding value of t in $\rho'(t)$ will give location of each attribute in ciphertext CT .

Then, decrypting user will combine his attribute keys $K_{\rho'(t), GID}$ with CT to decrypt as:

$$\begin{aligned} & \prod_t \left(\frac{C_{1,t}}{e(K_{\rho'(t), GID}, C_{2,t})} \right)^{c_x} \\ &= \prod_t \left(\frac{e(g, g)^{\lambda_t}}{e(H(GID), g)^{w_t}} \right)^{c_x} = e(g, g)^s \end{aligned}$$

After correctly finding $e(g, g)^s$ user will divide this by value of C_0 to obtain M .

4.3 Correctness

The proposed scheme is correct. Decrypting user needs to retrieve M embedded in ciphertext CT . If the user satisfies the policy, such that if any attribute set W_i of policy is the subset of user attribute set S_{att} , then user determines coefficients c_x values for the attributes in that particular W_i of policy from \mathbf{A} , ρ . After that, user determines the correct ciphertext CT components corresponding to attributes satisfying policy by checking out for condition where $\rho(x) == \rho'(t)$. Then, he combines his attribute keys with the ciphertext components to correctly decrypt as follows:

$$\begin{aligned} & \prod_t \left(\frac{C_{1,t}}{e(K_{\rho'(t), GID}, C_{2,t})} \right)^{c_x} \\ &= \prod_t \left(\frac{e(g, g)^{\lambda_t} \cdot e(g, g)^{\beta_{\rho'(t)} w_t}}{e(g^{\beta_{\rho'(t)}/r} H(GID)^{1/r}, g^{r w_t})} \right)^{c_x} \\ &= \prod_t \left(\frac{e(g, g)^{\lambda_t} \cdot e(g, g)^{\beta_{\rho'(t)} w_t}}{e(g, g)^{\beta_{\rho'(t)} w_t} e(H(GID), g)^{w_t}} \right)^{c_x} \\ &= \prod_t \left(\frac{e(g, g)^{\lambda_t}}{e(H(GID), g)^{w_t}} \right)^{c_x} = e(g, g)^s \end{aligned}$$

Finally, he retrieves M by $C_0 / e(g, g)^s$.

5 Analysis of the proposed scheme

This section overviews the security and performance analysis of the proposed scheme.

5.1 Security analysis

Theorem 1 *We show that our repeated attributes optimization-based decentralized multi-authority CP-ABE is secure for Chosen Plaintext Attacks under generic bilinear group model employed formerly in [7,9,24] by modeling H as random oracle.*

Proof Security model of scheme affirms that given black-box access for group operations and hash function H , adversary \mathcal{A} cannot succeed. We elaborate the model of generic bilinear group as in [24] by letting ψ_0, ψ_1 as two random encodings from additive group Z_p . Both ψ_0, ψ_1 form an injective map

from Z_p to $\{0, 1\}^m$, where $m > 3\log(p)$. The groups are formally represented as: $G_0 = \{\psi_0(x) : x \in Z_p\}$ and $G_1 = \{\psi_1(x) : x \in Z_p\}$ and we assume having access to oracles for determining the group operations in both of them. Moreover, we also have the oracle for computing the non-degenerate bilinear map $e : G_0 \times G_0 \rightarrow G_1$ and consider G_0 as bilinear group.

The attacker needs to identify the difference between $C_0 = M_0 e(g, g)^s$ and $C_0 = M_1 e(g, g)^s$ in security game. Consider an alteration in the game [7], in which the attacker has to distinguish between $C_0 = e(g, g)^s$ and $C_0 = e(g, g)^q$, for a random $q \in Z_p$. The simplified notations we utilize are: g represent $\psi_0(1)$, g^x represent $\psi_0(x)$, $e(g, g)$ represent $\psi_1(1)$ and $e(g, g)^y$ represent $\psi_1(y)$.

We now simulate the modified security game where C_0 is set to $e(g, g)^q$. Furthermore, S, U denote the authorities and attributes set, respectively. Global setup is run by simulator \mathcal{S} and g is given to the attacker \mathcal{A} . Thereafter, \mathcal{A} specifies and discloses set of corrupt authorities $S' \subset S$ to simulator. \mathcal{S} chooses at random $r \in Z_p$ for each un-corrupted authority, and $\beta_l \in Z_p$ for attributes $l \in U$ belonging to good (un-corrupted) authorities, and evaluates $g^r, e(g, g)^{\beta_l}$ by querying group oracles and gives these to attacker. Attacker \mathcal{A} asks $H(GID)$ for the first time. \mathcal{S} then replies to it by choosing a random value $h_{GID} \in Z_p$ and querying group oracles for computing $g^{h_{GID}}$. Moreover, \mathcal{S} also preserves a copy of $g^{h_{GID}}$, so that the future requested GID value will be furnished with same answer. \mathcal{A} requests key $K_{l,GID}$ corresponding to an identity GID and an attribute l , which \mathcal{S} determines by the help of group oracles and send back $g^{\beta_l/r} \cdot H(GID)^{1/r}$ to \mathcal{A} .

Thereafter, attacker \mathcal{A} specifies an access structure (A, ρ, ρ') for the challenge ciphertext under the restriction that combination of queries made in Phase 1 by \mathcal{A} , and the keys possessed by \mathcal{A} for corrupt authorities should not include a span of $(1, 0, \dots, 0)$ in challenged access structure. Moreover, \mathcal{A} shares the public keys of corrupt authorities attributes appearing in challenged access structure. Then, the simulator \mathcal{S} will choose s as the encryption exponent, and a vector $\mathbf{v} = (s, y_2, \dots, y_n) \in Z_p^n$, where y_2, \dots, y_n are used for sharing the encryption exponent s . For $x = 1$ to m , it computes $\lambda_x = \mathbf{A}_x \cdot \mathbf{v}$, where \mathbf{A}_x is the x^{th} row of \mathbf{A} . It chooses another vector $\mathbf{w} \in Z_p^n$ with 0 as its first entry, and evaluates $w_x = \mathbf{A}_x \cdot \mathbf{w}$. Then, these attribute shares values λ_x, w_x are passed to RAO algorithm which returns back optimized shares λ_t, ω_t for non-repeated attributes corresponding to ρ' . The simulator \mathcal{S} then computes the ciphertext CT with the help of group oracles as:

$$CT = \left\{ C_0 = M \cdot e(g, g)^q, C_{1,t} = e(g, g)^{\lambda_t} \cdot e(g, g)^{\beta_{\rho'(t)} w_t}, \right. \\ \left. C_{2,t} = g^{r w_t} \text{ for } t = \{1, 2, \dots, n'\} \right\}$$

Table 2 Possible query terms

β_i	r
h_{GID}	$\beta_i/r + h_{GID}/r$
$\lambda_t + \beta_{\rho'(t)} w_t$	$r w_t$
$\beta_i w_t + h_{GID} w_t$	h_{GID}/r
$h_{GID} \beta_i/r + h_{GID} h_{GID}'/r$	$\beta_i/r_j + h_{GID}/r_i r_j$
$h_{GID} h_{GID}'$	$(r w_t)(r' w_t')$
$(\beta_i/r_i + h_{GID}/r_i)(\beta_j/r_j + h_{GID}'/r_j)$	$r_i r_j$

Simulator \mathcal{S} sends the challenge ciphertext to attacker \mathcal{A} .

In simulation, we reason that with all but negligible probability, an \mathcal{A} consideration regarding if $C_0 = e(g, g)^s$ instead of $C_0 = e(g, g)^q$ is indistinguishable. This depicts that \mathcal{A} cannot gain non-negligible advantage in a modified version of security game; hence he cannot be able to attain non-negligible advantage in the real security game.

We restrict an \mathcal{A} querying for input values as the ones granted to it during simulation process, or in reply of former queries being made by him to oracles. The aforementioned event happens with greater probability. As ψ_0, ψ_1 are random injective mappings from Z_p into a set with more than p^3 elements; guessing an element's appearance in images of ψ_0, ψ_1 happens with a negligible probability. Under preceding situation, \mathcal{A} queries as a multi-variate polynomial in variables $q, \beta_i, r, \gamma_t, w_t, h_{GID}$, for i, r indicating un-corrupted authorities, t ranges over challenged access structure rows, and GID ranges over allowed user identities. We denote γ_t, w_t for the linear combination of variables $(s, v_2, \dots, v_n, w_2, \dots, w_n)$. Further, we state \mathcal{A} receives variant answers for each unlike queries in pair's to dissimilar polynomials, and there exists a nonzero difference for randomly assigning values to these variables correspondingly for different query polynomials. The above-mentioned event happens with greater probability to be realized by both union bound and Schwartz-Zippel lemma because polynomials hold at most degree 4.

As the appearance of q is only as $e(g, g)^q$, queries \mathcal{A} can look out relating to q will have the form $cq + \text{other terms}$, for a constant c . The view of \mathcal{A} can change only by making two dissimilar polynomial queries j and j' into G_1 , but if it replaces $q = s$; result will be the similar polynomial with the implication that $j - j' = cs - cq$ for a constant c . Hence, we conclude that a query of the form cs can be made by \mathcal{A} .

Now we will show a contradiction that query of the form cs cannot be put forward by \mathcal{A} . In Table 2, all possible queries that the \mathcal{A} can make are listed. By inspection, we conclude that queries put forward by \mathcal{A} are linear combinations of 1, q and other terms as seen in Table 2.

We remind that for corrupted authorities attacker \mathcal{A} knows the values of β_i and r ; hence, these values also appear as seen in Table 2. Recall that s can be retrieved from λ_t . For

ordering query cs ; \mathcal{A} needs to select constants ζ_t so that $\sum_t \lambda_t = cs$ by asking for query $\sum_t (\lambda_t + \beta_{\rho'(t)} w_t)$ in Table 2 to form $\sum_t \zeta_t (\lambda_t + \beta_{\rho'(t)} w_t)$. Attacker can construct polynomials of the form $-\zeta_t \beta_{\rho'(t)} w_t$ for corrupted authorities attributes to cancel out this term for the above polynomial. For attributes relating to uncorrupted authorities attacker needs to query $(\beta_{\rho'(t)} w_t + w_t h_{GID})$ this, leaving behind an extra term of $-\zeta_t w_t h_{GID}$. We state that attacker can access the term $(\beta_{\rho'(t)} w_t + w_t h_{GID})$ by requesting key for a specific attribute l and identity GID .

The assembling of terms correspondingly for each GID will remove $\zeta_t h_{GID} w_t$, under the condition of $(1, 0, \dots, 0)$ span existence in the rows A_x of A for corrupted authorities, or un-corrupted ones for whom \mathcal{A} obtained keys for $(\rho(x), GID)$. \mathcal{A} has not followed the rules of security game under this condition, by acquiring keys for GID giving himself the ability to decrypt the challenge ciphertext. \square

Hence, we demonstrated that attacker \mathcal{A} cannot build a query of the shape cs for a constant c . Under aforementioned conditions that hold with all but with negligible probability, we express that attacker's \mathcal{A} viewpoint that whether q is random or $q = s$ is identical. This proves that the attacker cannot attain a non-negligible advantage in the security game.

5.1.1 CCA extension

One limitation of the proposed scheme is that it is proven secure for Chosen Plaintext Attacks under generic bilinear group model. The scheme can be proven secure for chosen ciphertext attacks by incorporating a signature scheme in the security proof as illustrated in [25]. The attacker will be given access to a decryption oracle. The challenged ciphertext generated will also be signed, thereby limiting the attacker to fiddle with the ciphertext. We refer the readers for more insight as described in [25,26].

5.2 Performance evaluation

In the evaluation, firstly the theoretical computational costs with existing constructions are presented. Then, the practical simulations results are elaborated. Finally, the significance of RAO algorithm to provide help in the context of resource-constrained IoT devices is described.

We present the comparison of the proposed construction with [8–10,21,27,28] in Table 3 based on the parameters of the involved attribute authorities, security and provision for repeated attributes optimization (RAO). The notations used for security in Table 3 are SM for Standard Model, GGM for Generic Group Model, ROM for Random Oracle Model, CPA for Chosen Plaintext Attack, SS for Selective Secure, and FS for Fully Secure. Water's scheme [8] is CPA-based

Table 3 Parameters comparison

Scheme	Attribute authority	Security	RAO
[8]	Single	CPA SS SM	No
[9]	Multiple	CPA FS GGM	No
[10]	Multiple	CPA FS ROM	No
[21]	Multiple	CPA FS GGM	No
[27]	Single	CPA SS SM	No
[28]	Single	CPA SS SM	No
This work	Multiple	CPA FS GGM	Yes

CPA Chosen Plaintext Attack, SS Selective Secure, FS Fully Secure, SM Standard Model, GGM Generic Group Model, ROM Random Oracle Model

selectively secure in SM, while [9] and this work are adaptively secure in the GGM.

To exhibit performance, the cost of computational operations in key generation, encryption, decryption and ciphertext size of the proposed construction is presented and compared to [8–10,21,27,28] in Table 4.

The notations used in Table 4 are: m and l for number of user attributes and attributes authorities, respectively; n for number of attributes in access structure, z for users attributes satisfying policy, E for the exponential operation, P for the pairing operation, T for total attributes in universe, W for attribute sets in policy, $|G|$ for operations in source group, i.e., g , and $|G_T|$ for target group, i.e., $e(g, g)$ involving pairing. The proposed scheme has similar performance in encryption operation and ciphertext size in comparison to [8] but far better than [9] and [10]. The limitation of Water's scheme [8] is being based on a single authority; when compromised leads to entire system failure.

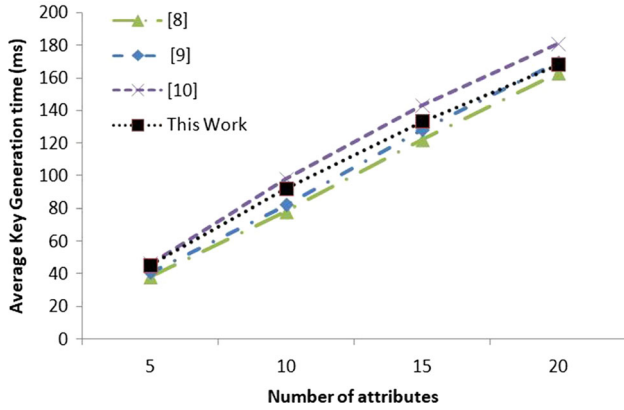
Moreover, [8] cannot be scaled to IoT context where several decentralized authorities are responsible to generate public parameters and ensure access control under their own domain with the designated attributes. The proposed CP-ABE with RAO scheme is the most efficient for decryption operation in comparison to other schemes, making it a reasonable choice for IoT devices with limited resources. The key generation time does not affect the timing of IoT devices because keys are generated once by the decentralized trusted attribute authorities and do not need to change frequently. Hence, it has no significance as such to decelerate the performance of IoT devices used to encrypt or decrypt data.

To practically demonstrate results, the proposed scheme is implemented in Charm [29,30]; a cryptographic tool designed to define and evaluate constructions specifically based on bilinear pairing. Moreover, it is scripted in "python" and utilize Pairing-Based Cryptography (PBC) library [31]. The simulation is executed on a Hyper-V VM running (with 3 GB allocated Ram) on a Dell Inspiron laptop i5-3337U

Table 4 Computational costs comparison

Scheme	KeyGen	Encryption	Decryption	User keysize	PP size	Ciphertext size
[8]	$(m + 2)E$	$(3n + 2)E$	$(2z + 1)P + zE$	$(m + 2) G $	$T(2 + G) + G_T $	$(2n + 1) G + G_T $
[9]	$(2m)E$	$(5n + 1)E$	$z(2P + E)$	$m G $	$T(G + G_T)$	$2n G + (n + 1) G_T $
[10]	$(l + 2m)E$	$(6n + 1)E + P$	$(3z)P + zE$	$m G + l G $	$T(G + G_T)$	$3nG + (n + 1) G_T $
[21]	$(2l + m)E$	$(3n + W)E$	$z(P + E)$	$m G $	$l G + T G_T $	$n G + (n + W) G_T $
[27]	$(4m + 3)E$	$(5n + 2)E$	$(4z + 1)P + 3zE$	$5 G + G_T + T Z_p $	$(2m + 2) G $	$(3n + 1) G + G_T $
[28]	$(3m + 7)E$	$(5n + 2)E$	$(3z + 1)P + zE$	$5 G + G_T + T Z_p $	$(2m + 3) G $	$(3n + 1) G + G_T $
This work	$(2l + m)E$	$(3n + 1)E$	$z(P + E)$	$m G $	$l G + T G_T $	$n G + (n + 1) G_T $

m number of user attributes, l : number of attributes authorities, n number of attributes in access structure, z users attributes satisfying policy, E Exponential operation, P : Pairing operation, T total attributes in universe, W attribute sets in policy, G Source group, i.e., g , G_T target group, i.e., $e(g,g)$ involving pairing

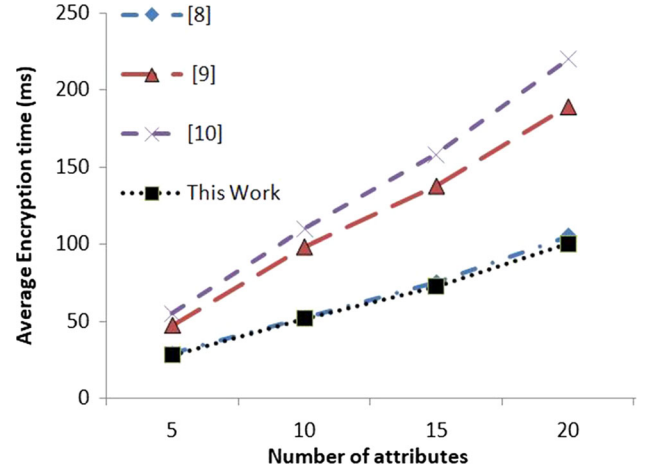
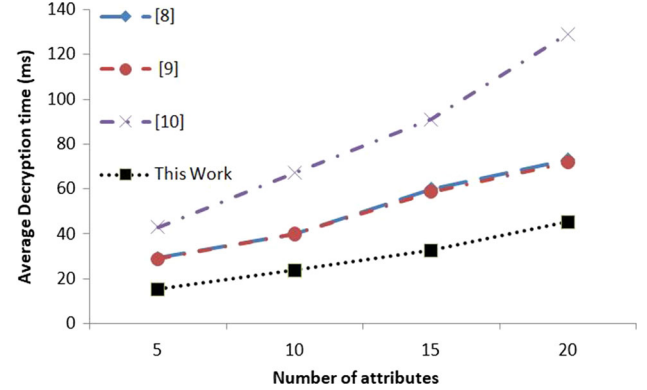
**Fig. 2** Average key generation time (ms)

CPU@ 1.80GHz with 8 GB Ram. The underlying OS was Ubuntu 14.04 with python library 3.4.3 and Charm-Crypto version 0.43. In Figs. 2, 3, 4, the average time in milliseconds (ms) of key generation, encryption and decryption operations versus the number of attributes is presented without taking repeated attributes optimization into consideration. The key generation time for all the schemes is almost similar as seen from Fig. 2. The encryption and decryption cost for the scheme is quite less as seen from Figs. 3 and 4 which further affirms the theoretical comparison presented in Table 4.

We now show the effect of attributes repetition on the size and computation cost of ciphertext. To elaborate, the applicability of the proposed scheme and specifically the RAO algorithm, we demonstrate that for a particular IoT-enabled hospital in which the patient suffering from a cardiac and nasal polyp disease is connected with sensors, and his confidential reports are being shared with doctors and nurse as: $IR - Pol = (Hospital \wedge Doctor \wedge (Cardiologist \vee Otolaryngologist)) \vee (Nurse \wedge Cardiologist \wedge Otolaryngologist)$.

There are overall seven (7) attributes in this policy.

If any sensor connected to patient periodically updates and sends the encrypted patient's report to the doctors and nurse, respectively, it needs to compute the ciphertext for

**Fig. 3** Average encryption time (ms)**Fig. 4** Average decryption time (ms)

all seven attributes. This periodic encryption process for a resource-constrained sensor device is costly in terms of computation and communication. The proposed CP-ABE with RAO encrypts the data only for non-repeated distinct attributes by performing optimization. After optimization through RAO in the encryption algorithm the number of attribute shares are reduced to five (5), namely Hospital,

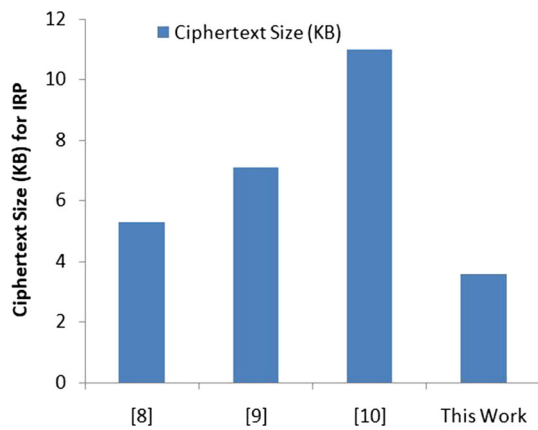


Fig. 5 Ciphertext size (KB) for IR-Pol policy

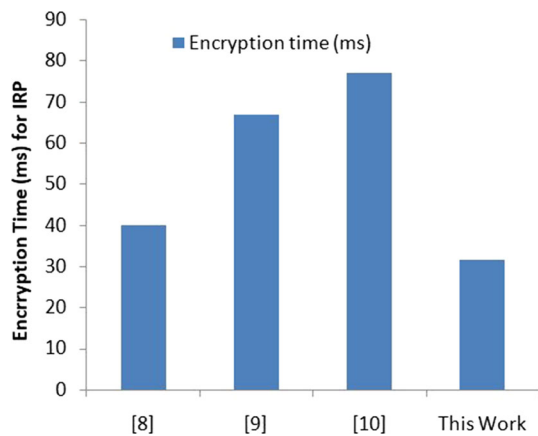


Fig. 6 Average encryption time (ms) for IR-Pol policy

Doctor, Cardiologist, Otolaryngologist and Nurse. So, the ciphertext will only be evaluated for these five attributes.

Hence, the optimization process eliminates the number of the computationally expensive exponential Exp operations in encryption operation for repeated attributes. This in effect reduces the ciphertext size as well in contrast to the other existing approaches [8–10]. Although the proposed scheme is already cost-efficient, the ciphertext size and its computation cost are reduced further by optimization as seen from Figs. 5 and 6 for irreducible policy $IR-Pol$. The ciphertext size and computation cost in the proposed scheme are linear with the number of attributes n in the access structure. However, for an irreducible policy, the ciphertext size and computation cost are reduced linearly with the number of repeated attributes. Suppose for a policy containing 7 attributes in total, after optimization by RAO algorithm is reduced to 5. In that case, the ciphertext size and computation cost will be correspondingly for 5 attributes instead of 7, leading to a reduction in computation and communication cost as depicted in Figs. 7 and 8.

To practically elaborate the difference and impact of RAO, Figs. 7, 8 demonstrate the effect of computational cost and

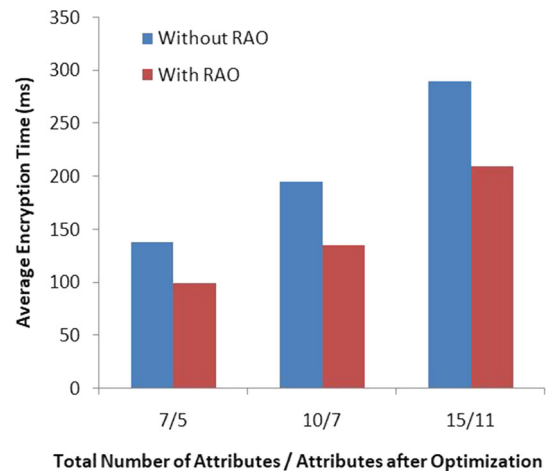


Fig. 7 Average encryption time (ms) with and without RAO

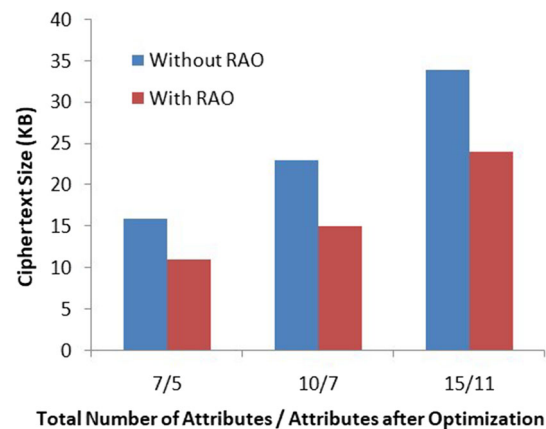


Fig. 8 Ciphertext size (KB) with and without RAO

ciphertext size reduction by with and without applying the RAO algorithm for the proposed CP-ABE scheme. For arbitrary irreducible policies with 7, 10 and 15 attributes, the number of attributes after optimization is reduced to 5, 7 and 11. This leads to reduced computational cost as seen in Fig. 7 where the resource-constrained data owner or (possibly an IoT sensor) have to perform for computing the ciphertext. Also, from Fig. 8, there is a reduction in ciphertext size leading to less communication overhead. RAO performs the optimization of attributes by making sure the usability of underlying policy for specifying access control pertaining to a particular scenario is adhered. Generally, as seen in Figs. 7 and 8 the cost decreases with the number of repeated attributes with in a particular irreducible policy. The reduced encryption time and ciphertext size ensures that the computation and communication costs are less for resource-constrained IoT devices, thereby making the proposed scheme a possible candidate for IoT.

6 Conclusion and future work

6.1 Conclusion

In this paper, an efficient CP-ABE scheme with repeated attributes optimization is proposed. The construction employs our proposed “RAO” algorithm for removal of repeated redundant attribute shares in encryption operation. This helps to reduce the ciphertext computational cost and its size for optimizable irreducible policies. The proposed scheme is proven secure for Chosen Plaintext Attacks (CPA) in the generic group model. Finally, the performance analysis including theoretical and simulation results exhibits its effectiveness for resource-constrained devices.

6.2 Future work

One limitation of the proposed scheme is that it is proven secure for Chosen Plaintext Attacks under generic bilinear group model. The scheme can be proven to be secure against the chosen ciphertext attacks by employing the methodology mentioned in 5.1.1. Another issue worth addressing is the optimization of an irreducible policy when no optimization variables are present with repeated attributes in an irreducible policy.

Funding This study was funded by National Natural Science Foundation of China under Grant U1401251, China 111 Project (No. B16037) and Shaanxi Key Basic Research Project 2016ZDJC-04.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of things: a survey. *Comput. Netw.* **54**(15), 2787–2805 (2010)
2. Li, S., Da Xu, L., Zhao, S.: 5G Internet of things: a survey. *J. Ind. Inf. Integr.* **10**, 1–9 (2018). <https://doi.org/10.1016/j.jii.2018.01.005>
3. Da Xu, L., He, W., Li, S.: Internet of things in industries: a survey. *IEEE Trans. Ind. Inf.* **10**(4), 2233–2243 (2014). <https://doi.org/10.1109/TII.2014.2300753>
4. Mell, P., Grance, T.: The NIST definition of cloud computing, 20–23 (2011)
5. Zheng, X., Martin, P., Brohman, K., Da Xu, L.: CLOUDQUAL: a quality model for cloud services. *IEEE Trans. Ind. Inf.* **10**(2), 1527–1536 (2014). <https://doi.org/10.1109/TII.2014.2306329>
6. Zheng, X., Martin, P., Brohman, K., Da Xu, L.: Cloud service negotiation in internet of things environment: a mixed approach. *IEEE Trans. Ind. Inf.* **10**(2), 1506–1515 (2014). <https://doi.org/10.1109/TII.2014.2305641>
7. Bethencourt, J., Sahai, A., & Waters, B.: Ciphertext-policy attribute-based encryption. In: *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pp. 321–334 (2007)
8. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: *International Workshop on Public Key Cryptography* (pp. 53–70). Springer, Berlin (2011)
9. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 568–588). Springer, Berlin (2011)
10. Rouselakis, Y., Waters, B.: Efficient statically-secure large-universe multi-authority attribute-based encryption. In: *International Conference on Financial Cryptography and Data Security* (pp. 315–332). Springer, Berlin (2015)
11. Rouselakis, Y., Waters, B.: New constructions and proof methods for large universe attribute-based encryption. *IACR Cryptology EPrint Archive* **583** (2012)
12. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 62–91). Springer, Berlin (2010)
13. Liu, Z., Cao, Z., Wong, D.S.: Efficient generation of linear secret sharing scheme matrices from threshold access trees (vol. 2010). *IACR Cryptology ePrint Archive* (2010)
14. Lewko, A., Waters, B.: New proof methods for attribute-based encryption: achieving full security through selective techniques. In: *Advances in Cryptology-CRYPTO 2012* (pp. 180–198). Springer, Berlin (2012)
15. Takashima, K.: New proof techniques for DLIN-based adaptively secure attribute-based encryption. In: *Australasian Conference on Information Security and Privacy* (pp. 85–105). Springer, Cham (2017)
16. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: *Proceedings of the 13th ACM Conference on Computer and Communications Security* (pp. 89–98). ACM (2006)
17. Kapadia, A., Tsang, P.P., Smith, S.W.: Attribute-based publishing with hidden credentials and hidden policies. In: *NDSS* (vol. 7, pp. 179–192) (2007)
18. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden cryptor-specified access structures. In: *International Conference on Applied Cryptography and Network Security* (pp. 111–129). Springer, Berlin (2008)
19. Chase, M.: Multi-authority attribute based encryption. In: *Theory of Cryptography Conference* (pp. 515–534). Springer, Berlin (2007)
20. Muller, S., Katzenbeisser, S., Eckert, C.: Distributed attribute-based encryption. In: *International Conference on Information Security and Cryptology* (pp. 20–36). Springer, Berlin (2008)
21. Khan, F., Li, H., Zhang, L.: Owner specified excessive access control for attribute based encryption. *IEEE Access* **4**, 8967–8976 (2016)
22. Beimel, A.: Secure schemes for secret sharing and key distribution, Ph.D. thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996). [Online]. Available: <https://www.cs.bgu.ac.il/beimel/Papers/thesis.pdf>
23. Muller, S., Katzenbeisser, S., Eckert, C.: On multi-authority ciphertext-policy attribute-based encryption. *Bull. Korean Math. Soc.* **46**(4), 803–819 (2009)
24. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: *Annual International*

- Conference on the Theory and Applications of Cryptographic Techniques (pp. 440–456). Springer, Berlin (2005)
25. Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N. (2011). Generic constructions for chosen-ciphertext secure attribute based encryption. In: International Workshop on Public Key Cryptography (pp. 71–89). Springer, Berlin
 26. Nandi, M., Pandit, T.: Generic conversions from CPA to CCA secure functional encryption. IACR Cryptol ePrint Archive **2015**, 457 (2015)
 27. Han, Q., Zhang, Y., Li, H.: Efficient and robust attribute-based encryption supporting access policy hiding in Internet of Things. Future Gener. Comput. Syst. **83**, 269–277 (2018)
 28. Cui, H., Deng, R. H., Qin, B., Weng, J.: Key regeneration-free ciphertext-policy attribute-based encryption and its application. Inf. Sci. pp. 217–229 (2020). <https://doi.org/10.1016/j.ins.2019.12.025>
 29. Akinyele, J.A., Garman, C., Miers, I., Pagano, M.W., Rushanan, M., Green, M., Rubin, A.D.: Charm: a framework for rapidly prototyping cryptosystems. J. Cryptogr. Eng. **3**(2), 111–128 (2013)
 30. Charm, <http://www.charm-crypto.io/>. Accessed 20 April 2019
 31. Pairing Based Cryptography Library (PBC), <https://crypto.stanford.edu/pbc/>. Accessed 20 April 2019

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Fawad Khan received the B.S. and M.S. degree in Electrical Engineering from UET Peshawar and CECOS University in 2010 and 2014 respectively. He received his Ph.D. degree from the School of Cyber Engineering, Xidian University in 2018. Currently he works at the National University of Science and Technology, Pakistan. His research interests include cryptography and information security.



Hui Li received his B.Sc. degree from Fudan University in 1990, M.A.Sc. and Ph.D. degrees from Xidian University, Xi'an, China, in 1993 and 1998, respectively. He was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Waterloo, Ontario, Canada, in 2009. Since June 2005, he has been a professor in the school of Telecommunications Engineering, Xidian University. His research interests are in the areas of cryptography, security of cloud computing, wireless network security, information theory, and network coding. He is the co-author of two books. He served as TPC co-chair of ISPEC 2009 and IAS 2009, general co-chair of e-forensic 2010, ProvSec 2011, and ISC2011.

computing, wireless network security, information theory, and network coding. He is the co-author of two books. He served as TPC co-chair of ISPEC 2009 and IAS 2009, general co-chair of e-forensic 2010, ProvSec 2011, and ISC2011.



Yinghui Zhang received his Ph.D degree in Cryptography from Xidian University, China, in 2013. He is a professor at School of Cyberspace Security, National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts and Telecommunications. He has published research articles in ACM ASIACCS, ACM Computing Surveys, IEEE TDSC, IEEE TSC, IEEE TCC, Computer Networks, Computers and Security, etc. He served for the program committee of several conferences and the editorial members of several international journals in information security. His research interests include public key cryptography, cloud security and wireless network security.



Haider Abbas (SM'15) received the M.S. degree in engineering and management of information systems and the Ph.D. degree in information security from the KTH-Royal Institute of Technology, Stockholm, Sweden, in 2006 and 2010, respectively. His professional career consists of activities ranging from research and development and industry consultations (government and private), through multi-national research projects, research fellowships, doctoral studies advisory services, international journal editorships, conferences/workshops chair, invited/keynote speaker, technical program committee member, and reviewer for several international journals and conferences. He is currently a Cyber Security Professional, an Academician, a Researcher, and an Industry Consultant who took professional trainings and certifications from the Massachusetts Institute of Technology, USA; Stockholm University, Sweden; the Stockholm School of Entrepreneurship, Sweden; IBM, USA; and the ECCouncil. He is also an Adjunct Faculty and Doctoral Studies Advisor at the Florida Institute of Technology, USA. In recognition of his services to the international research community and excellence in professional standing, he has been awarded one of the youngest Fellows of the Institution of Engineering and Technology, U.K.; a fellow of the British Computer Society, U.K.; and a fellow of the Institute of Science and Technology, U.K.



Tahreem Yaqoob has received her B.S. degree in Computer Science with emphasis in Security of cloud network from Fatima Jinnah Women University, Pakistan. She did M.S. in Information Security from National University of Sciences and Technology, Islamabad, Pakistan in 2018. Her research interests include security issues in healthcare environment and medical devices.