Computer Systems Science & Engineering
DOI:10.32604/csse.2022.017597
Article

Tech Science Press

Repeated Attribute Optimization for Big Data Encryption

Abdalla Alameen*

Department of Computer Science, College of Arts and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia
*Corresponding Author: Abdalla Alameen. Email: a.alameen@psau.edu.sa
Received: 04 February 2021; Accepted: 09 April 2021

Abstract: Big data denotes the variety, velocity, and massive volume of data. Existing databases are unsuitable to store big data owing to its high volume. Cloud computing is an optimal solution to process and store big data. However, the significant issue lies in handling access control and privacy, wherein the data should be encrypted and unauthorized user access must be restricted through efficient access control. Attribute-based encryption (ABE) permits users to encrypt and decrypt data. However, for the policy to work in practical scenarios, the attributes must be repeated. In the case of specific policies, it is not possible to avoid attribute repetition even after the application of Boolean optimization approaches to obtain a Boolean formula. For these policies, there exists a variety of evaluated secret shares for the repeated attributes. Therefore, the calculation of cipher text for these irreducible policies seems to be lengthy and computationally intensive. To address this problem, an improved meta-heuristic-based repeated attributes optimization on cipher-text policy-ABE (CP-ABE) is developed in this study. Here, the improved meta-heuristic concept is developed in the encryption phase, which returns the optimized single share value of each repeated attribute after considering all the attribute shares. The optimization process not only minimizes the encryption cost but also the communication cost. Herein, the improved sun flower optimization (SFO), called the newly updated SFO (NU-SFO) is used to perform the repeated attribute optimization in CP-ABE. Finally, the performance evaluation confirms the reliability and robustness of the developed scheme through comparisons with traditional constructions.

Keywords: Big data; repeated attribute optimization; cipher text policy; encryption

1 Introduction

Owing to the enhanced application of digitization and internet technology, data have become an important factor in organizational growth. Consequently, a novel paradigm known as big data, which refers to data that is huge in size, has emerged. Data can be semi-structured, unstructured, or structured. The veracity denotes the data, which is produced in a rapid format. Therefore, data must be gathered and processed in a rapid manner [1]. Recently, cloud computing technology has evolved as a quick development [2] and is considered as a significant area in computer science. It offers storage and computing services that help clients handle expanded data sharing. Data are saved to remote servers using

encryption. Therefore, they are not visible to malicious servers or unauthorized users. Moreover, digital content is safeguarded through access control. Although cloud computing offers a wide variety of benefits, businesses refrain from choosing this solution owing to privacy and security concerns. The data denotes storing it outside the server, which is untrustworthy. Cloud service providers (CSPs) disclose data to unauthorized users or access private data for business reasons. Therefore, data must be encrypted to ensure privacy. However, even after encryption, data remain visible to all the users. Consequently, users must be permitted to access only those data they are authorized to access [3].

Big data storage should consider user access control and privacy [4]. These issues are initially addressed by attribute-based encryption (ABE) [5]. Here, user identity is described using some attributes, which offer fine-grained access control and data privacy [6]. However, using these directly for access control and big data privacy is a complex task because it involves huge decryption and encryption computation costs arising from the necessity of exorbitant pairing operations and the size of the cipher-text, respectively. This overhead is minimized by outsourcing heavy decryption and encryption computations [7]. The CP-ABE restricts unauthorized users from using the data saved on remote servers. The CP-ABE schemes [8–11] were developed with distinct access structures such as the linear secret sharing scheme (LSSS) structure, AND gate with negative and positive attributes, AND gate, tree structure, and threshold [12–15]. However, they do not offer big data access control in a direct manner, owing to high computation cost. Recently, big data access control schemes [16,17] comprising LSSS were developed. Therefore, it is necessary to model a CP-ABE scheme such that minimum computation overhead is incurred [18].

The main contributions of this study are listed below.

• An enhanced CP-ABE was developed for addressing the computation efforts and calculations of cipher text for the irreducible policies.

• A new optimization algorithm, NU-SFO, was developed, which optimizes the random encryption exponent and the random vector, thereby minimizing the communication and encryption costs.

• The enhanced meta-heuristic concept was employed in the encryption phase by returning the optimized single share value for every repeated attribute after considering the entire attribute shares.

• The performance of the proposed NU-SFO-CP-ABE was validated against traditional algorithms in terms of encryption time analysis, decryption time analysis, and key generation analysis.

This paper is organized as follows. Section I provides an introduction to the CP-ABE for big data security. The works related to CP-ABE in literature are presented in Section II. Section III explains the system and security models of the recommended encryption process. Section IV describes the meta-heuristic-based CP-ABE in big data. The results of the experiments and allied discussions are provided in Section V. Section VI concludes the paper.

2 Literature Review

Premkamal et al. [19] have developed a novel CP-ABE for the access control and privacy of big data. Their scheme minimizes the computation overhead of decryption and encryption by outsourcing large computations. Furthermore, it checks the correctness of data in the outsourced computations. Additionally, it restricts data access for user groups, which is required for commercial applications. In terms of security analysis, this scheme proves to be secure against proxy, collusion, and chosen plain-text attacks. Furthermore, a performance analysis demonstrated the effectiveness of this scheme. Chen et al. [20] have developed an efficient extended file hierarchy files CP-ABE scheme (EFH-CP-ABE). This scheme is suitable for big companies and institutions comprising various hierarchical sectors because it saves computation cost and storage space. Moreover, this solution achieved flexible and secure access control for cloud storage users. The final step exhibited superior experimental outcomes. Qin et al. [21] addressed

an effective cipher text policy attribute-oriented encryption scheme through the reduction of pairing operations. Security analysis stated that the model was safe from user collusion and chosen-plaintext attacks. This scheme proved to be effective compared to traditional schemes. Guo et al. [22] developed a new framework to handle the access to EHRs. The fine-grained access control related to the EHRs was achieved by leveraging the CP-ABE approach for encrypting the tables released by hospitals, and it was stored in the database using a unique identity of the patient as the primary key. This framework enables distinct users with distinct privileges to search distinct database fields. The control of the field was emphasized inside the database. This scheme was evaluated using the datasets from University of California, Irvine. Li et al. [23] labeled an effective CP-ABE scheme. It minimized the computation cost of the PCSP and the communication and storage costs for the client. Moreover, the developed technique was secured under the bilinear Diffie–Hellman exponent (BDHE). Additionally, it was effective with respect to file and policy updates. Rasori et al. [24] proposed ABE-Cities, wherein the data were sensed from the city location and saved to the cloud in an encrypted format. Users were provided keys to decrypt the sensed data from the authorized zones or paths of the city. The sensors undergo lightweight symmetric-key encryption; therefore, constrained sensor devices such as battery powered motes were used. An expiration date was planned for every key, and the provided key was retracted in an unplanned manner. The existence of IoT gateways was leveraged by an advanced scheme to minimize the computation load. In 2018, Han et al. introduced a novel CP-ABE scheme to protect the attribute values of users against the attribute authority (AA)-oriented on 1-out-of-n oblivious transfer approach. Additionally, an attribute bloom filter was employed to protect the attribute format of the access policy. This scheme produced better security goals; however, there was no improvement in the computation overhead. Challagidad et al. [25] developed an effective multi-authority access control scheme. It comprises hierarchy access structure (HAS) and role hierarchy algorithm (RHA) to protect user data and provide fine-grained access to stored data. The HAS described an access structure for multi-authority and fine-grained access control. The results were effective considering the storage and time consumption for decryption and encryption. The benefits of this scheme are evident when the file count on the cloud storage server increases.

Major security concerns regarding big data are privacy and access control. One of the most adopted privacy algorithms related to big data is the ABE-based algorithm. Literature has suggested various advancements in the ABE-related algorithms; however, most of them face computation overhead while dealing with large data sources. Recently, CP-ABE has become a major research focus for the effective handling of big data in the cloud environment.

3 System and Security Models of the Proposed Encryption Process

The architectural model of a repeated attributes optimization (RAO)-oriented CP-ABE comprises the user, owner, attribute authorities, and CSP. CSP represents an entity that offers storage and computation services. It acts as a semi-trusted entity. AA employs the access control mechanism by providing decryption keys to the users on the basis of user attributes. Each AA produces the secret key and public key parameters. The secret key parameters are employed by the AA to generate user decryption keys on the basis of the attributes of users and the identities owned by them. The entire AA performs in a decentralized format without coordinating among them. Owner describes the resource-conditioned devices that encrypt their data for outsourcing them to the CSP. User is an entity who accesses and retrieves data on the basis of access privilege. Users conspire with one other to access the data that are not entitled to contain in an individual manner. Global Setup$(\lambda) \rightarrow GPC$: this takes as input, the security parameter, λ, and outputs the global parameters, GPC. Authority Setup$(GPC) \rightarrow SKC, PKC$: this algorithm is run by each AA by considering input GPC, and outputs the public and secret key parameters, $PKC$and SKC, respectively. Encrypt$(MC, (AC, \rho, \rho'), GPC, PKC) \rightarrow CTC$: the data owner

executes this algorithm by considering message MC, GPC, PKC and access structure (AC, ρ, ρ') that takes the policy as input, and returns cipher text CTC. An accurate representation of an irreducible policy is defined by AC, ρ, and it is described using the LSSS matrix, AC, wherein ρ is mapped to the rows of AC, and ρ' comprises the unique non-repeated attribute names that appear in ρ or the irreducible policy. Further, the length of ρ' is than ρ, and therefore, it is mapped to CTC. **KeyGen**$(GIDC, GPC, lc, SKC) \rightarrow KC_{lc,GIDC}$: this algorithm is implemented by the AA. It considers the user identity, $GIDC, GPC, SKC$, as input to generate the output key, $KC_{lc,GIDC}$, that is related to the user attribute 1. **Decrypt** $\left(CTC, GPC, \left\{KC_{lc,GIDC}\right\}\right) \rightarrow MC$: this algorithm allows users to access data. Here, GPC and CTC are considered as the input, and MC is returned as the output if the key fulfils the access structure in CTC.

The proposed scheme comprises the following algorithms.

Global Setup$(\lambda) \rightarrow GPC$: Here, a bilinear group, GC, of prime order pc' is selected. The global parameters are fixed to pc', gc and HC, wherein gc represents a generator of group GC, and HC denotes a hash function. Authority Setup$(GPC) \rightarrow SKC.PKC$: Every authority selects for itself a random value, $rc \in ZC_{pc}$. A random value $\beta_{lc} \in ZC_{pc}$ is selected. Values $\{rc, \beta_{lc} \forall lc\}$ are maintained as secret key, SKC, and $\left\{gc^{rc}, ec(gc, gc)^{\beta_{lc}} \forall lc\right\}$ is published as public key PKC.

Encrypt$(MC, (AC, \rho, \rho'), GPC, PKC) \rightarrow CTC$: In the case of encryption, the access policy is initially transformed into LSSS matrix AC. Here, the input is a message, MC, PKC's from relevant attributes, an access matrix AC of size $mc \times nc$ having ρ that comprises a map of its rows to attributes, and global parameters. Additionally, input ρ' that denotes the list of non-repeated distinct attributes that appear in ρ is also considered. Next, it selects a random encryption exponent, $sc \in ZC_{pc}$, and $vc \in ZC_{pc}^{nc}$, wherein vc represents a column vector of length nc and comprises sc as its initial entry. It then calculates $\lambda_{xc} = AC_{xc} \cdot vc$ in which AC_{xc} represents the xc^{th} row of AC. Further, it selects a random vector, $wc \in ZC_{pc}^{nc}$, of length nc, and having secret $sc' = 0$ as its initial entry. Then $\omega_{xc} = AC_{xc} \cdot wc$ is calculated. The steps performed during optimization are summarized in Algorithm 1. The RAO algorithm considers $AC, \rho, \rho', sc, \lambda_{xc}, sc', \omega_{xc}$ and the attribute set count WC_{ic} in policy as input and executes as follows.

Step-1: Coefficients cc_{xc} related to attributes $\rho(xc)$ that belong to the entire attribute sets, WC_{ic}, present in the policy are calculated using Eq. (1).

$$\sum_{xc} cc_{xc} AC_{xc} = (1, 0, \cdots, 0) \tag{1}$$

Step-2: For the entire different non-repeated attribute names that appear in ρ', the counter variable $count_{\rho'(tc)}$ is initialized to zero. The execution begins with the initial attribute set and traverses through all the attributes. The occurrences of the entire attributes that appear in the attribute sets, $\rho(xc) \in WC_{ic}$, are numbered by increasing the $count_{\rho'(tc)}$ variables. This records the attribute repetition that appears multiple times in several distinct WC_{ic}.

Step-3: For each attribute set, WC_{ic}, of policy, initialize variable Add_{ic} to 0, and subsequently increment it with $count_{\rho'(tc)}$ for $\rho(xc) \in WC_{ic}$ and $\rho(xc) == \rho'(tc)$ by fulfilling Eq. (2).

$$Add_{ic} = \left(\sum\nolimits_{\rho(xc) \in WC_{ic}} Count_{\rho'(tc)} \; if \; \rho(xc) == \rho'(tc)\right) \tag{2}$$

Notations: Attribute shares, λ_{xc}, is divided into two categories: (i) optimized or fixed share $\lambda_{xc-optimized}$, which is assigned a constant value, and (ii) Other share, $\lambda_{xc-other}$, in which the optimized value is yet to be described. When the value is described, the attribute share status is changed from $\lambda_{xc-other}$ to $\lambda_{xc-optimized}$. Furthermore, array KC is defined as a 3-dimensional array that maintains a record of optimized variable name, its $\omega_{xc-optimized}$ optimized share value, and the $\lambda_{xc-optimized}$ share value.

Step-4: To undergo few optimization steps, the algorithm points that attribute group $WC_{lc\ \max}$ contains the largest Add_{ic} value or the higher repetition count. Once $WC_{lc\ \max}$ having highest Add_{ic} is described, the original attribute shares values, λ_{xc}, ω_{xc}, are fixed to the optimized values, $\lambda_{xc-optimized}$, $\omega_{xc-optimized}$, and then appended to array KC. The original λ_{xc} is replaced by the optimized-shares values in the entire attribute groups, WC_{ic}, in which these repeated attributes are present.

Step-5: The optimization of the remaining shares attributes, $\lambda_{xc-other}$, $\omega_{xc-other}$, in remaining distinct WC_{ic}(excluding $WC_{lc\ \max}$) is done using Eqs. (3) and (4).

$$\lambda_{xc-other} = (1/cc_{xc-other})\left(sc - \sum_{xc \in WC_{ic}, KC} cc_{xc}\lambda_{xc}\right) \tag{3}$$

$$\omega_{xc-other} = (1/cc_{xc-other})\left(sc - \sum_{xc \in WC_{ic}, KC} cc_{xc}\omega_{xc}\right) \tag{4}$$

Every new $\lambda_{xc-optimized}$, $\omega_{xc-optimized}$ is appended to array KC. The optimization is said to be complete when all the attribute shares are optimized.

Step-6: All the optimized values related to the attribute names in $\rho'(tc)$ are allocated to λ_{tc} and ω_{tc}. Therefore, CT is calculated for the optimized novel shares λ_{tc} and ω_{tc} as follows.

$$CT = \left\{ \begin{array}{l} CC_0 = MC \cdot ec(gc, gc)^{sc}, CC_{1,tc} = ec(gc, gc)^{\lambda_{tc}} \\ \cdot ec(gc, gc)^{\beta_{\rho'(tc)}\omega_{tc}}, CC_{2,tc} = gc^{rc\omega_{tc}} \text{ for } tc = \{1, 2, \cdots, nc'\} \end{array} \right\} \tag{5}$$

Next, cipher text CT is transmitted to the cloud server along with (AC, ρ, ρ'), wherein AC, ρ defines the LSSS matrix that shows the actual policy with repeated attributes, and ρ' denotes the optimized non-repeated attributes that are utilized for the CT evaluation. Here, AC is mapped to ρ, and ρ' is mapped to CT.

KeyGen $(GIDC, GPC, lc, SKC) \rightarrow KC_{lc,GIDC}$: Generation of a key for user $GIDC$ that are related to an attribute lc of authority, is performed as follows.

$$KC_{lc,GIDC} = gc^{\beta_{lc}/rc} \cdot HC(GIDC)^{1/rc} \tag{6}$$

Decrypt $\left(CT, GPC, \left\{KC_{lc,GIDC}\right\}\right) \rightarrow MC$: For decryption, the user initially describes the attributes that fulfil the policy and the index of the cipher text components related to these attributes. The procedure is outlined in Algorithm 2. The RAO-check algorithm considers the input, AC, ρ, ρ', decryption user attribute set, SC_{att}, and attribute groups, WC_{ic}, in policy and proceeds as follows.

Step-1: If any attribute group ,WC_{ic}, of policy is a subset of user attribute set $SC_{att}(WC_{ic} \subseteq SC_{att})$, then the user attributes that satisfy the policy represent the attributes of that specific WC_{ic}, otherwise, the policy is unfulfilled, and the user is not eligible for data access.

Step-2: For user attributes SC'_{att} that fulfill the policy, calculate and return the coefficients, cc_{xc}, using Eq. (7).

$$\sum\nolimits_{xc} cc_{xc} AC_{xc} = (1, 0, \cdots, 0). \tag{7}$$

Step-3: For every attribute in ρ that fulfills the policy, the algorithm initially checks the condition in which $\rho(xc) == \rho'(tc)$. Then, the related value of tc in $\rho'(tc)$ provides the location of each attribute in cipher text CT. Subsequently, the decrypting user combines the attribute keys, $KC_{\rho'(tc),GIDC}$, having CT for decryption as shown in Eq. (8).

$$\prod_{tc}\left(\frac{CC_{1,tc}}{ec\left(KC_{\rho'(tc),GIDC},CC_{2,tc}\right)}\right)^{cc_{xc}} = \prod_{tc}\left(\frac{ec(gc,gc)^{\lambda_{tc}}}{ec(HC(GIDC),gc)^{\omega_{tc}}}\right)^{cc_{xc}} = ec(gc,gc)^{sc} \tag{8}$$

When the correct $ec(gc,gc)^{sc}$ is found, the user should divide it by CC_0 to obtain MC.

The proposed NU-SFO-based encryption for big data is used to optimize the random encryption component and the random vector to minimize the communication and encryption costs. The SFO [26] approach represents a population-oriented algorithm. It uses pollination and root velocity to provide robustness. It is assumed that every sunflower generates one pollen gamete and reproduces in an individual manner. The next significant nature-oriented optimization is the inverse square law radiation. It states that, "the intensity of the radiation is inversely proportional to the square of the distance." Otherwise, more the distance between the sun and the plant, less is the heat received. These steps are followed here to achieve the global optimum. The quantity of heat, QS, received by each plant is calculated using Eq. (9).

$$QS_{is} = \frac{PS}{4\pi rs_{is}^2} \tag{9}$$

Here, the distance between the current best and the plant is is defined by rs_{is}, and the power of the source is defined by PS. The direction of the path from the sunflower is given by Eq. (10).

$$\vec{ss}_{is} = \frac{XS^* - XS_{is}}{\|XS^* - XS_{is}\|}, \quad is = 1,2,\cdots,ns_{ps} \tag{10}$$

The sunflowers in direction ss are computed using Eq. (11).

$$ds_{is} = \lambda \times PS_{is}(\|XS_{is} + XS_{is-1}\|) \times \|XS_{is} + XS_{is-1}\| \tag{11}$$

In Eq. (11), the probability of pollination is defined by $PS_{is}(\|XS_{is} + XS_{is-1}\|)$, and a constant value for describing an "inertial" plant displacement is defined by λ. The maximum step is computed using Eq. (12).

$$ds_{\max} = \frac{\|XS_{\max} - XS_{\min}\|}{2 \times NS_{pop}} \tag{12}$$

Here, the plant count of total population is defined by NS_{pop}, and the upper and lower bound values are defined by $XS_{\max}$ and $XS_{\min}$, respectively. The new plantation is computed using Eq. (13).

$$\vec{XS}_{is+1} = \vec{XS}_{is} + ds_{is} \times \vec{ss}_{is} \tag{13}$$

The SFO suffers from some shortcomings such as its inability to (i) work with multiple suns, (ii) move in a randomly controlled manner, and (iii) perform random steps in a particular direction. Therefore, the algorithm was improved on the basis of fitness, and it is called NU-SFO. Here, the term, $count$, is initialized to zero. If $count \geq 5$, the solution is updated using Eq. (14) as follows.

$$XS^* = XS + CS_1(best - XS) + CS_2(XS - worst) \tag{14}$$

Here, the terms CS_1 and CS_2 represent the random numbers between (–1, 1). If $count < 5$, SFO is updated. The steps in pseudo code of the proposed NU-SFO are outlined in Algorithm 3, and the flowchart of the suggested NU-SFO is illustrated in Fig. 1.

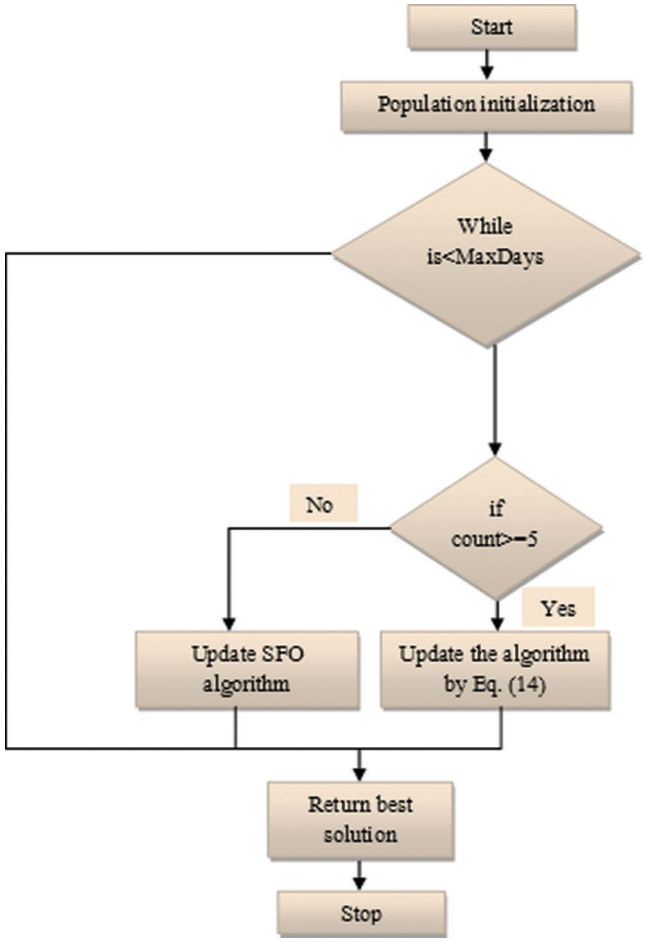

Figure 1: Flowchart of the proposed NU-SFO

4 Experimental Setup

The proposed NU-SFO-CP-ABE for enhanced CP-ABE in big data was implemented in MATLAB 2019a. The random encryption component and the random vector were optimized by the same proposed NU-SFO. The population size was considered to be 10, and the maximum iterations were considered to be 100. To prove the superiority of the proposed NU-SFO-CP-ABE, it was compared with several existing optimization algorithms such as GWO-CP-ABE [26], WOA-CP-ABE [27], BOA-CP-ABE [28], and SFO-CP-ABE [29].

4.1 Convergence Analysis

The convergence analysis of the recommended NU-SFO-CP-ABE against several heuristic optimization algorithms is presented in Fig. 2. It is evident from the figure that the cost function is greater with the proposed NU-SFO-CP-ABE. At the 6^{th} iteration, the cost function of NU-SFO-CP-ABE is 60%, 33.33%, 81.82%, and 73.91% higher than that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. For the 10^{th} iteration, the cost function of NU-SFO-CP-ABE is 44.44%, 30%, 54.76%, and 58.54% superior to that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Hence, the convergence analysis of NU-SFO-CP-ABE is superior to the traditional approaches.

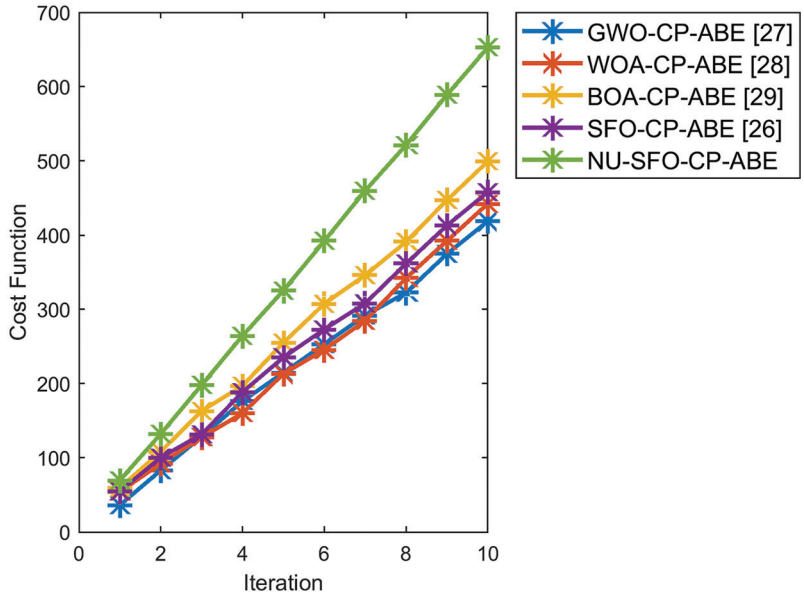

Figure 2: Convergence analysis of the proposed NU-SFO-CP-ABE in comparison with other heuristics for big data security

4.2 Encryption and Decryption Time Analysis

The encryption time analysis of the proposed NU-SFO-CP-ABE in comparison with different heuristic optimization algorithms is presented in Fig. 3. The proposed algorithm has the least encryption cost among all the existing methods, thereby proving its superiority. At the 20th attribute, the encryption time of NU-SFO-CP-ABE is 18.12%, 5.83%, 17.52%, and 9.6% superior to that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. For the 15th attribute, the encryption time of NU-SFO-CP-ABE is 14.71%, 8.66%, 10.08%, and 13.43% superior to that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Similarly, at the 10th attribute, the encryption time of NU-SFO-CP-ABE is 4.10%, 13.97%, 14.60%, and 10% superior to that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Overall, the encryption time analysis provides good results with the proposed NU-SFO-CP-ABE compared with all the traditional methods.

The decryption time analysis for big data security with the proposed NU-SFO-CP-ABE compared to the state-of-the-art algorithms is presented in Fig. 4. The proposed NU-SFO-CP-ABE achieves better results than the existing methods, achieving less decryption time with all number of attributes. At the 20th attribute, the decryption time of NU-SFO-CP-ABE is 15.38%, 16.67%, 12%, and 20.29% higher than that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. For the 15th attribute, the decryption time of NU-SFO-CP-ABE is 8.46%, 9.16%, 3.25%, and 13.14% superior to that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Furthermore, for the 10th attribute, the decryption time of NU-SFO-CP-ABE is 6.50%, 4.17%, 7.26%, and 17.27% higher than that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Therefore, the decryption time analysis results of the proposed NU-SFO-CP-ABE are better than those of the existing algorithms.

The key generation analysis for the proposed NU-SFO-CP-ABE compared to the traditional algorithms is presented in Fig. 5. In the case of 20 attributes, the results of the proposed NU-SFO-CP-ABE are better than those of the traditional algorithms. For the 20th attribute, the key generation of NU-SFO-CP-ABE is 16.03%, 17.29%, 15.38%, and 14.73% better than that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. For the 15th attribute, the key generation of NU-SFO-CP-ABE is 15.79%, 15.15%,

17.04%, and 18.84% better than that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Similarly, for the 10^{th} attribute, the key generation of NU-SFO-CP-ABE is 9.16%, 11.85%, 7.75%, and 12.5% superior to that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Therefore, the proposed NU-SFO-CP-ABE provides better results in the case of key generation analysis than the state-of-the-art algorithms.

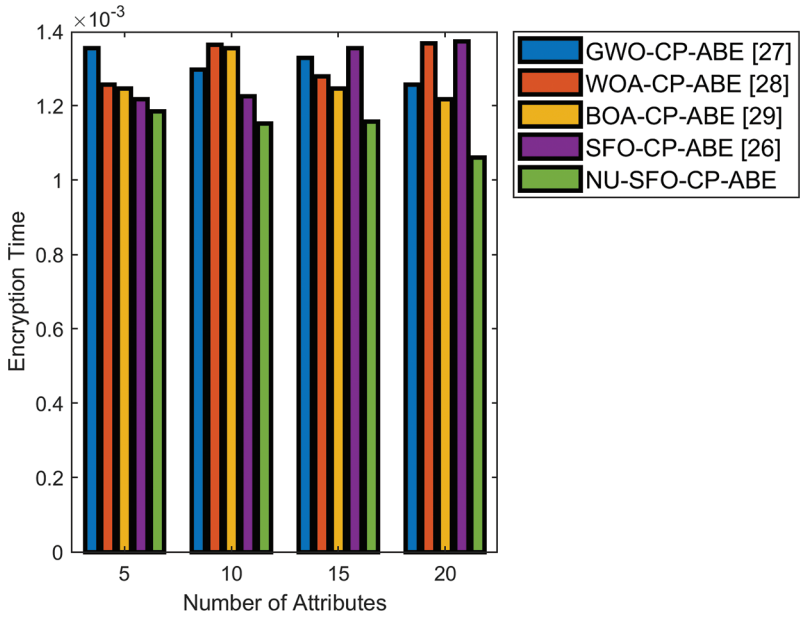

Figure 3: Encryption time analysis of the proposed NU-SFO-CP-ABE compared with other heuristics for big data security

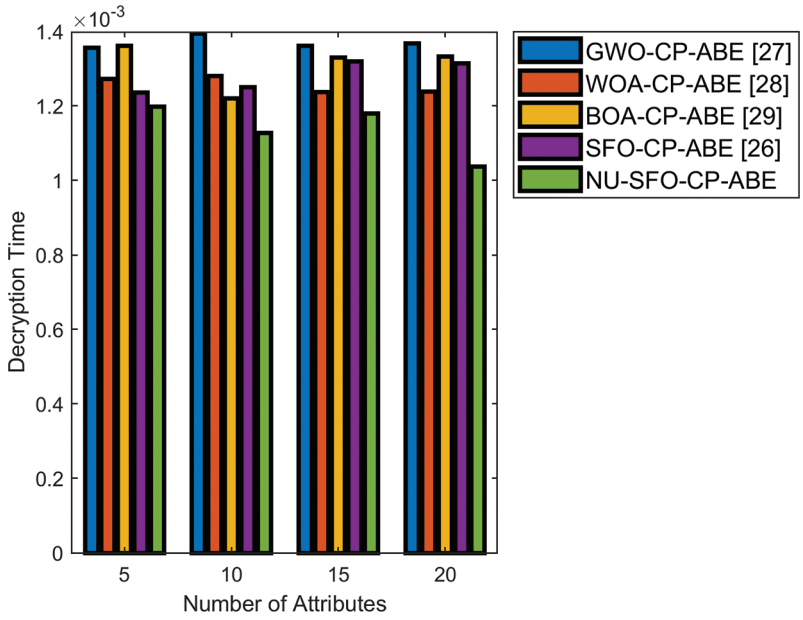

Figure 4: Decryption time analysis of proposed NU-SFO-CP-ABE compared with other heuristics for big data security

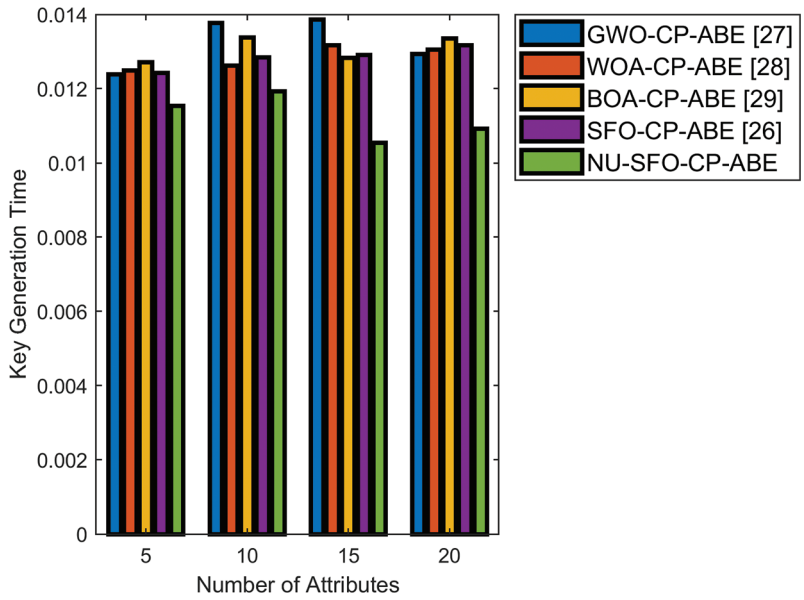

Figure 5: Key generation time analysis of the proposed NU-SFO-CP-ABE compared with other heuristics for big data security

5 Conclusion

In this study, an enhanced meta-heuristic-based RAO was developed on CP-ABE. The enhanced meta-heuristic concept was implemented in the encryption phase, which returned the optimized single share value after considering the entire attribute shares. The optimization concept minimized the encryption and communication costs by minimizing the CT size. The novel NU-SFO performed the RAO on CP-ABE by optimizing the random encryption exponent and the random vector. The performance evaluation demonstrated the robustness and reliability of the developed scheme compared to traditional algorithms. From the analysis, at the 20^{th} attribute, the encryption time of NU-SFO-CP-ABE was 18.12%, 5.83%, 17.52%, and 9.6% superior to that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Similarly, at the 20^{th} attribute, the decryption time of NU-SFO-CP-ABE was 15.38%, 16.67%, 12%, and 20.29% higher than that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Moreover, for the 20^{th} attribute, the key generation of NU-SFO-CP-ABE was 16.03%, 17.29%, 15.38%, and 14.73% better than that of BOA-CP-ABE, WOA-CP-ABE, and GWO-CP-ABE, respectively. Hence, the analysis of the proposed NU-SFO-CP-ABE is better with better performance than all the existing algorithms.

Acknowledgement: We thank Deanship of Research, Prince Sattam Bin Abdul-Aziz University, KSA, for providing an opportunity to conduct research.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Tao, Q. Qi, A. Liu and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, no. 1, pp. 157–169, 2018.

[2] M. Abdel-Basset, M. Mohamed and V. Chang, "NMCDA: A framework for evaluating cloud-computing services," *Future Generation Computer Systems*, vol. 86, no. 2, pp. 12–29, 2018.

[3] M. Kolhar, M. M. Abu-Alhaj and S. M. Abd El-atty, "Cloud data auditing techniques with a focus on privacy and security," *IEEE Security and Privacy*, vol. 15, no. 1, pp. 42–51, 2017.

[4] T. Khalid, M. A. Abbasi, M. Zuraiz and M. Aslam, "A survey on privacy and access control schemes in fog computing," *International Journal of Communication Systems*, vol. 34, no. 2, pp. 41–81, 2021.

[5] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh *et al.,* "Attribute based encryption with privacy protection and accountability for CloudIoT," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 1, 2020.

[6] J. Li, X. Chen, S. S. Chow, Q. Huang, D. S. Wong *et al.,* "Multi-authority fine-grained access control with accountability and its application in cloud," *Journal of Network and Computer Applications*, vol. 112, no. 15, pp. 89–96, 2018.

[7] R. R. Al-Dahhan, Q. Shi, G. M. Lee, M. G. and K. Kifayat, "Survey on revocation in ciphertext-policy attribute-based encryption," *Sensors*, vol. 19, no. 7, pp. 3–22, 2019.

[8] L. Li, T. Gu, L. Chang, Z. Xu, Y. Liu *et al.,* "A Ciphertext-policy attribute-based encryption based on an ordered binary decision diagram," *IEEE Access*, vol. 5, pp. 1137–1145, 2017.

[9] X. Liang, Z. Cao, H. Lin and J. Shao, "Attribute based proxy reencryption with delegating capabilities," in *Proc. Int. Sym. on Information Computer, and Communications Security*, Sidney, Australia, pp. 276–286, 2009.

[10] S. Luo, J. Hu, Z. Chen, M. Soriano, S. Qing *et al.,* "Ciphertext policy attribute-based proxy re-encryption," in *Lecturer Notes in Computer Science*, 1st. ed., vol. 3494. Berlin, Germany: Springer, pp. 401–415, 2010.

[11] X. Xu, J. Zhou, X. Wang and Y. Zhang, "Multi-authority proxy reencryption based on CP-ABE for cloud storage systems," *Journal of Systems Engineering and Electronics*, vol. 27, no. 1, pp. 211–223, 2016.

[12] X. Xie, H. Ma, J. Li and X. Chen, "An efficient ciphertext-policy attribute-based access control towards revocation in cloud computing," *Journal of Universal Computer Science*, vol. 19, no. 16, pp. 2349–2367, 2013.

[13] V. Goyal, A. Jain, O. Pandey and A. Sahai, "Bounded ciphertext policy attribute based encryption," *Automata Languages and Programming*, vol. 1, no. 1, pp. 579–591, 2008.

[14] X. Liang, Z. Cao, H. Lin and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," *Information Computer, and Communications Security*, vol. 2, no. 1, pp. 343–352, 2009.

[15] K. Yang, X. Jia and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, 2015.

[16] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su *et al.,* "An efficient and fine-grained big data access control scheme with privacy-preserving policy," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 563–571, 2017.

[17] P. P. Kumar, P. S. Kumar and P. J. Alphonse, "An efficient ciphertext policy-attribute based encryption for big data access control in cloud computing," in *Proc. Ninth Int. Conf. on Advanced Computing (ICoAC)*, Chennai, pp. 114–120, 2017.

[18] C. Guo, R. Zhuang, Y. Jie, Y. Ren, T. Wu *et al.,* "Fine-grained database field search using attribute-based encryption for e-healthcare clouds," *Journal of Medical Systems*, vol. 40, no. 11, pp. 2–35, 2016.

[19] P. K. Premkamal, S. K. Pasupuleti and J. P. Alphonse, "A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 7, pp. 2693–2707, 2019.

[20] J. Li, N. Chen and Y. Zhang, "Extended file hierarchy access control scheme with attribute based encryption in cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 1, 2019.

[21] X. Qin, Y. Huang and X. Li, "An ECC-based access control scheme with lightweight decryption and conditional authentication for data sharing in vehicular networks," *Soft Computing*, vol. 24, no. 24, pp. 18881, 2020.

[22] R. Guo, H. Shi, Q. Zhao and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, no. 1, pp. 11676–11686, 2018.

[23] J. Li, W. Yao, J. Han, Y. Zhang, J. Shen *et al.,* "User collision avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 1, no. 2, pp. 1767–1777, 2020.

[24] M. Rasori, P. Perazzo and G. Dini, "A lightweight and scalable attribute-based encryption system for smart cities," *Computer Communications*, vol. 149, no. 1, pp. 78–89, 2020.

[25] P. S. Challagidad and M. N. Birje, "Efficient multi-authority access control using attribute-based encryption in cloud storage," *Procedia Computer Science*, vol. 167, no. 1, pp. 840–849, 2020.

[26] M. Seyedali, M. Seyed and A. Lewis, "Grey Wolf optimizer," *Advances in Engineering Software*, vol. 69, no. 1, pp. 46–61, 2014.

[27] M. Seyedali and L. Andrew, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, no. 1, pp. 51–67, 2016.

[28] A. Sankalap and S. Satvir, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Computing*, vol. 1, no. 23, pp. 715–734, 2018.

[29] G. F. Gomes, S. Sebastiao and C. Carlos, "A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates," *Engineering with Computers*, vol. 35, no. 1, pp. 619–626, 2019.