Computer Systems Science & Engineering
DOI: 10.32604/csse.2022.022424
Article

Tech Science Press

Improved Secure Identification-Based Multilevel Structure of Data Sharing in Cloud Environments

Saraswathi Shunmuganathan[1,*], Sridharan Kannan[2], T. V. Madhusudhana Rao[3], K. Ambika[4] and T. Jayasankar[5]

[1]Department of Computer Science and Engineering, SSN College of Engineering, Chennai, Tamil Nadu, 603110, India
[2]Department of Computer Science and Engineering, J. K. K. Munirajah College of Technology, Erode, Tamil Nadu, 638506, India
[3]Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam, 530049, India
[4]Department of Computer Science and Engineering, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli, Tamil Nadu, 620024, India
[5]Department of Electronics and Communication Engineering, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli, 620024, India
*Corresponding Author: Saraswathi Shunmuganathan. Email: sarasuthan@yahoo.co.in
Received: 07 August 2021; Accepted: 17 November 2021

Abstract: The Cloud Computing Environment (CCE) developed for using the dynamic cloud is the ability of software and services likely to grow with any business. It has transformed the methodology for storing the enterprise data, accessing the data, and Data Sharing (DS). Big data frame a constant way of uploading and sharing the cloud data in a hierarchical architecture with different kinds of separate privileges to access the data. With the requirement of vast volumes of storage area in the CCEs, capturing a secured data access framework is an important issue. This paper proposes an Improved Secure Identification-based Multilevel Structure of Data Sharing (ISIMSDS) to hold the DS of big data in CCEs. The complex file partitioning technique is proposed to verify the access privilege context for sharing data in complex CCEs. An access control Encryption Method (EM) is used to improve the encryption. The Complexity is measured to increase the authentication standard. The active attack is protected using this ISIMSDS methodology. Our proposed ISIMSDS method assists in diminishing the Complexity whenever the user's population is increasing rapidly. The security analysis proves that the proposed ISIMSDS methodology is more secure against the chosen-PlainText (PT) attack and provides more efficient computation and storage space than the related methods. The performance of the proposed ISIMSDS methodology provides more efficiency in communication costs such as encryption, decryption, and retrieval of the data.

Keywords: Data sharing; cloud environments; big data; chosen-plaintext attack; security

1 Introduction

The allocation of resources to share the data with the secure method increases the convenience and storing of the most important data to increase the financial improvement in multilevel companies [1]. The efficient utilization of the resources with managing the data is measured using the parameters. The demand-based cloud accessibility and sharing of data minimize the cost of data management and increase the capacity of the CCEs [2]. The owners of the data have concerned about the security for sharing the valuable data in the CCE. The privilege to generate the access rights depends on a group of data. The sharing of data with the users was analyzed according to the level of privileges. The granting privilege is essential to enhance the Complexity of the techniques for data is shared in CCEs [3].

The access management methodology is related to the verification models than the identity matching technique. Every structure is assigned a specific role that understands the user privileges [4]. Moreover, the constraints of the model are very complex based on user feedback. The user privileges can be modified for the new prototype and role for the individual users. The rule is framed to manage the resource utilization for identifying the user roles. The cloud data can be accessed by many applications [5]. Third-Party Re-Encryption (TPRE) permits the transformation of a CipherText (CT) that is calculated under Alice's public key (PBK) into one that can be unlocked by Bob's Secret Key (SK). Not including Bob's SK, the encrypted mail is forwarded to Alice's co-worker Bob [6]. In this case, Alice, the agent authority, could assign a TPRE to convert her received mail into a design that Bob, the agent, can decrypt using his SK. Alice could offer her SK to the third party, but this entails an impracticable stage of trust in the third party [7]. TPRE enables a semi-trusted CS to transform a CT of encrypted data under the PBK of an agent into CT under the PBK of an agent without disclosing the primary encrypted messages of agent authority/agent to the third party [8]. The guarantee of the user's data security is getting more delayed acceptance of CCE [9]. To share sensitive data stored in cloud storage from data holder (say, Alice) to another approved user (say, Bob). Encouraged by the primitive of TPRE, Alice uses their own PBK, which can encrypt the sensitive data before uploading the shared data to the semi-trusted Cloud Server (CS) [10]. After receiving the DS request from Bob, Alice produces a TPRE key using her private key (PRK) and Bob's PBK and this TPRE key to the semi-trusted CS [11]. In the TPRE key, the CS can transform the encrypted CT with the PBK of Alice into encryption under the PBK of Bob [12]. By utilizing the TPRE primitive, the CT scan received by Bob can only be decrypted by him while the CS is incapable of learning the PT/PRKs of Alice/Bob. In conclusion, decrypting and downloading the received data with Bob's PRK have occurred [13].

Some issues occur while processing the CCE

a) Non-availability of standard interfaces and complex data recovery systems.
b) Limited access to distribute and remote data storage, computing resources and infrastructure.
c) Difficult to control and locate the data due to segregated evidence.
d) Cloud security and forensics services can be outsourced as per the requirements.
e) Not possible to destroy/modify the evidence as these may be stored in multiple locations.
f) Economically, it is feasible and favourable for the organization to apply the cloud security and privacy model.
g) Inter-relationship between the entities in the cloud makes it easy to investigate for the investigation team.

In this paper, an ISIMSDS is proposed to hold the DS of big data in CCEs. It constructs to minimize the issues of DS within the CCEs with composite hierarchical models. The prime contribution of the proposed method focuses on the access control of the group DS, security of the data, and user revocation. The research contribution of the method is as follows:

a) The complex file partitioning method is proposed to access the privilege-related framework that improves cloud DS in complex structures.
b) Access control-based EM is implemented to increase the security of the user data.
c) The Complexity of this proposed methodology is analyzed to provide the best authentication standard.
d) The most substantial EM is generated to prevent active attacks in the CCE.
e) The proposed method is organized to implement security and prevent the chosen-PT attacks.
f) The performance analysis for the proposed method is compared with the related methodologies to accomplish better performance.

2 Related Works

In the proxy signature's common construction, the signature of Bob is observed as a dual one that contains a signature from Alice and the proxy. The non-conversion between suitable signatures of Alice into Bob and the signing of Alice's SK is furnished to her by Bob or else by a reliable source [14]. A general structure for protecting mediated Certificate-less Public Key Encryption (CPKE) offers instant revocation. An outstanding investigation of CPKE provides ranks and methods, the different concepts of protection for a CPKE method are against an outside attacker, and the passive key generation center in the survey on EMs used to secure cloud storage methods [15,16]. For multi-signatures, proxy re-signatures can be used as a substitute, except for collective signatures, by allowing transformations between signatures on similar messages alone [17,18]. The attributes share the data that could be utilized by the new user in the CCE [19]. The value can assign the attribute that the PRK of every user implements the low level of threshold needs of the attributes that may implement the incorporated text among the CT. The flexibility is maintained for granting access to the user [20]. The attribute-related EMs have emerged to maintain the adaptability of sharing cloud data. The access-based policies are maintained for access granting methodology to key-based attribute EM with CT and policy reports. The PRK is incorporated with policy control [21]. The authorization is given to the active users; they can decrypt the CT and gather the PRK from the activation key generator. The key generator incorporates access-based policy management to maintain the flexibility in granting the privileges [22]. The DS methodology is used to implement the security in cloud-based collaborative proofing [23] as described in SRMSM [24], Privacy-preserving key-based security mechanism is implemented in PRMSM [25], and an effective healthcare-based security method is used in Moks [26]. Threshold Multi-keyword Search (TMS) is used for securing the data during sharing in the CCE. TMS methodology enhances access control and prevents chosen keyword attacks [27,28].

3 Proposed Work

3.1 Architecture

The proposed architecture is combined with storing and sharing the data in the CCEs. The cloud affords the user to utilize the storage services with efficient sharing services. Moreover, the cloud has the trusted environment for our proposed ISIMSDS method. The Group Manager (GM) has organized the parameter for generating the architecture for data encryption, group member authorization, identifying the group member, and eliminating the false members. The GM is the trusted member of the CCE. The Group members are formed as a group of users based on the dynamic communication model. The members can share the data within the active CCEs. Whenever the user stores the data in the CS, confidentiality should be maintained with the key agreement methodology related to the proposed architecture.

Consequently, a general SK is created to provide the encryption of the data stored in the CS to measure the confidentiality of the data for outsourcing purposes. The attackers for the CS could not be able to gather the data without knowing the SK. Additionally, the group signature is maintained to allow the user within the group to share the data in the CCE independently. The proposed architecture is demonstrated in Fig. 1. The users with the same group can share the data, and the user revocation guarantees that the SK is identified to encrypt and decrypt the data. The GM is capable of identifying the real group member. The GM executes the uploading of data and the access of DS. The attacker from the outside group can try to find the common SK to decrypt the shared data from the CCE. The member for revocation can also join with the attacker to find the common SK from the list of revocation files in the CS. A third-party member can create distinct related keys and broadcast the message during the key generation procedure. In our proposed method, integrity is maintained by verifying the data from the users.

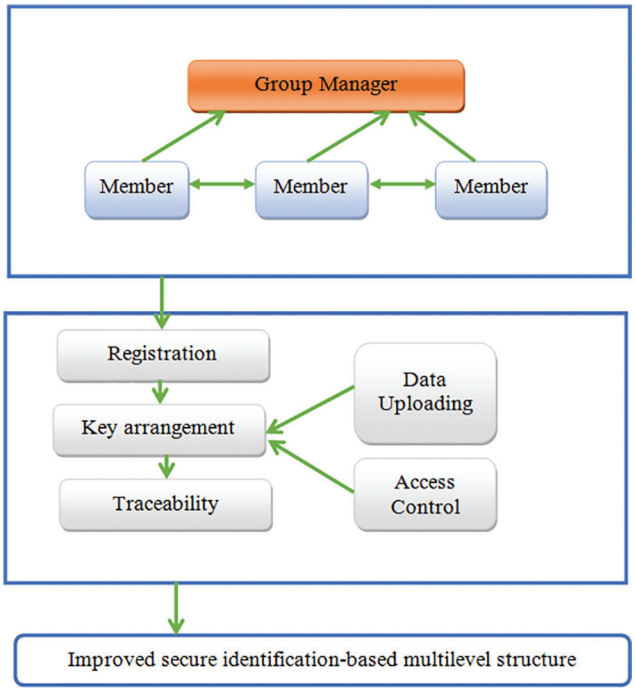

Figure 1: Architecture of the proposed method

3.2 Access Control Based Encryption Methodology

The proposed DS methodology based on the access control is implemented to accomplish the security compared to the related DS methods in the CCE. Accordingly, in the proposed methodology, the data owner is not connected online; the owner can dynamically share data with the user revocation, and incorporation is satisfied. The cloud storage method has multi-level attributes with different parameters like certificate authority, active users, servers, and data owners. The access control contains the Parameter Authority to register the initial configuration setup by assigning $global_{id}$ to every active user and $private_{id}$ for every parameter authority. During the registration process for implementing the access control, every $PA_x \in SK_A$ generates the SKs using the algorithm that determines the access policy for the more critical data in the CCE. For every data δ, the owners of the data initially identify the structure for accessing the data using the computation of $A = (\delta, \rho)$, to encrypt the data with this structure and delivered the encrypted information $Cipher_{text}$ to the intermediate CS. After the process of delivering the encrypted data, the active user $user_k$ may process the SK information SK_A and the relevant

$GlobalPublic_{key}$ to cloud for generating the Decryption Method (DM) measured using the intermediate CSs, and the user may complete the DM. The certificate authority, parameter authority, and intermediate CSs could not process the decryption without knowing the SK value. The attribute revocation is the primary process for monitoring every update in the encryption, and the entire process is showed in Fig. 2.

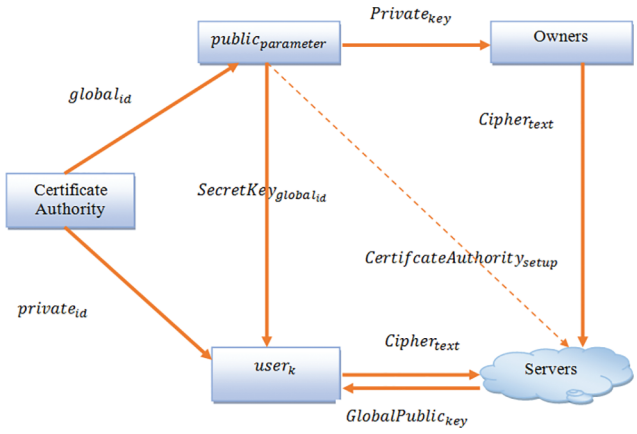

Figure 2: Access control based encryption process

3.2.1 Algorithm: Access Control Based Encryption

Step 1. Certificate authority defines the method with $CertifcateAuthority_{setup}$ function using Eq. (1)

Step 2. $$CertifcateAuthority_{setup}(1^m) \rightarrow \left(Master_{key},\ public_{parameter},\ (sig_{CA},\ ver_{CA})\right) \tag{1}$$

Step 3. The users forward the identity data to the certificate authority for user registration, which is proved in Eq. (2)

Step 4. $$User_{reg}\left(Public_{parameter}, Sig_{CA}, Info_m\right) \rightarrow \left(Global_{id},\ GlobalPublic_{key}, GlobalSecret_{key},\ cert(Global_{id})\right) \tag{2}$$

Step 5. After entering the user registration, every public parameter delivers the identity data to a certificate authority for $global_{id}$

Step 6. Every $Parameter_{authority_{setup}}$ includes the SK within the parameter authority, and it is computed using Eq. (3)

$$Parameter_{authority_{setup}}(PA_x, global_{id}) \rightarrow \left(SecretKey_{global_{id}},\ PublicKey_{global_{id}}\right) \tag{3}$$

3.2.2 Complexity Analysis of the Algorithm

The Complexity of the algorithm is analyzed with the possibility of attacks in the CCE. The authentication table information is used to compute all the possible values for each component in the list of passwords. The possibility for computing the encrypted text is also analyzed from the PT password text in the server. The permutation for converting the PT into encrypted text is computed using Eq. (4)

$$Per_{Au} = \alpha! \tag{4}$$

The amount of converted password text is calculated using Eq. (5)

$$Total_{Enc} = \left\{ \binom{\alpha - 1}{1} \binom{\alpha - 2}{1} \right\} \prod_{k=2}^{\alpha-1} \binom{k}{2} \alpha! \tag{5}$$

$$\prod_{k=2}^{\alpha-1} \binom{k}{2} = \frac{(\alpha - 1)!\,(\alpha - 2)!}{2^{\alpha-2}} \tag{6}$$

By simplifying Eqs. (5) and (6), we obtain Eq. (7)

$$Total_{Enc} = \frac{[(\alpha - 2)\alpha!]^3}{2^{\alpha-2}\alpha^2} \tag{7}$$

The hashed password is analysed using Eq. (8)

$$Password_{hash} = Hash\left(Passord_{plain}\right) \tag{8}$$

To avoid the active attack in the authentication table, the modified password is generated using Eq. (9)

$$Password_{hash}(modified) = Hash(Password_{plain}||Password_{modified}) \tag{9}$$

The access framework illustrates the policies for accessing the group of individual access for the SK. The framework determines the group of entities that an entity could generate to permit access to the SKs. The group of policies can be defined as $\{\mathrm{Po}_1, \mathrm{P}o_2, \ldots, \mathrm{P}o_n\}$. A group of members may rebuild the SK as a combination. The combination is defined using Eqs. (10) and (11)

$$\aleph \subseteq 2^{\{\mathrm{Po}_1, \mathrm{P}o_2, \ldots, \mathrm{P}o_n\}} \tag{10}$$

$$\beth \subseteq \mathbb{C} \rightarrow \mathbb{C} \in \aleph \tag{11}$$

The entire access framework is a monotone combination $\aleph$ of sub groups $\{Po_1, Po_2, \ldots, Po_n\}$ and it can be defined using Eq. (12)

$$\aleph \subseteq 2^{\{Po_1, Po_2, \ldots, Po_n\}} \backslash \varphi \tag{12}$$

The groups in $\aleph$ are called the approved groups, and the groups that are not in $\aleph$ is called the non-approved groups. In this paper, we utilize the parties to generate the elements. An access framework $\aleph$ can combine an approved and a non-approved group of elements. A Communication Tree (CT) β_i with level ω_i characterizes the framework that ensures the user for sharing data that can decrypt the CT is $cipher_i$ with the level of the tree construction. A β_i has several nodes with the leaf to frame the threshold values. The leaf node demonstrates the possibility for generating the components of the DS by the users. The nodes in the tree are assigned a threshold value using Eq. (13).

$$\gamma_l^i \in \beta_i \tag{13}$$

The AND gate is utilized to form a threshold value using Eq. (14).

$$key_{\gamma_l^i} = Children_{\gamma_l^i} \tag{14}$$

The OR gate is utilized for assigning the value for leaf node with threshold value using Eq. (15)

$$key_{\gamma_l^i} = 1 \tag{15}$$

The root node of every CT determines the SK value. The user can evaluate the group of attributes to generate the SK using Eq. (16)

$$key_{sec_l^i} \in \beta_i \tag{16}$$

Fig. 3 demonstrates the privileged multi-level access framework to minimize the replicated attributes for the CT with N amount of levels to provide the security in DS of CCEs. The threshold values are framed to represent the privacy preference methodology.

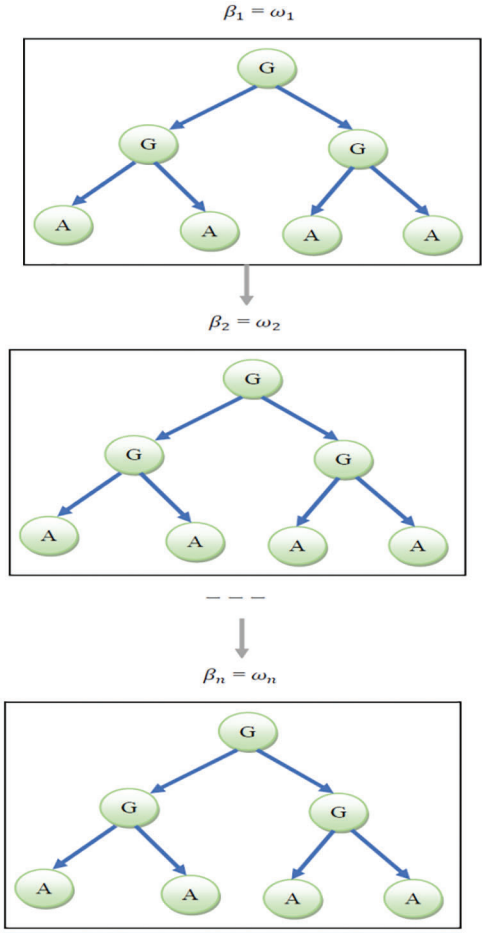

Figure 3: Privileged multi-level access framework

3.2.3 Algorithm 2: Data Security Specific Algorithm With the Aid of Multi-Level Access Framework

Step 1. **Start** algorithm

Step 2. **For Each** 'n' table in the target *database TD* $(1, \ldots, N)$

Step 3. Fetch K_1 for the whole table (n) and each attribute j in the table, the standard one is PRK

Step 4. Fetch K_2, and for the whole attribute j for each item (R_i, C_j) in the table (n), j PRK is the standard one.

Step 5. Fetch K_3, and for a tuple (1), i PBK is common

Step 6. Find the item length (R_I, C_j)

Step 7. $Jen = strlen((R_i, C_j))$

Step 8. **If** Jen NOT even **Then**

Step 9. Append '0' to (R_I, C_j)

Step 10. To each pair of characters/digit in (R_I, C_j)

Step 11. Convert the character into an integer

Step 12. Append to $(R_I, C_j)'$

Step 13. **End If**

Step 14. **End For**

Step 15. Store $(R_I, C_j)'$ in place of (R_I, C_j)

Step 16. **End For**

Step 17. **End**

On the side of the cloud service provider, full homomorphism encryption can be used. It was an adaptive method for authenticator and server. The remotely stored file is verified using the File 'f' that is preserved as a multi-valued vector, and for all blocks of a file, tag δ is created. The client sends the random challenge vector v, and the server returns the proof authentication. The tag is determined using Eq. (17).

$$\delta = \sum_i v_i, f_i \tag{17}$$

For the auditing process, homomorphism tags are used. In block-less verification, the server verifies the data and metadata that were aimed at each block. For every file, unique tags are created, numbered, and stored in the counters. By adding the combination of tags, the server can prove that the data is not modified. For n number of users, 'n' signatures are verified. The process is counted using Eq. (18).

$$(p_k, s_k) \leftarrow (Gen) \tag{18}$$

The stated information s_t is utilized for encoding the file using Eq. (19)

$$(f, s_t) \leftarrow Encode(f) \tag{19}$$

The parameter value μ is recognized to prove the facts of the PBK with cryptographic functions in a file f, and it is assessed in Eq. (20)

$$\mu = prove(p_k, f, c) \tag{20}$$

The *Vrfy* (*verify*) the methodology can generate the binary value of true/false as 1/0 and is identified in Eq. (21).

$$a = vrfy(p_k, c, \mu) \tag{21}$$

3.3 Data Sharing Technique

The real-time problem is identified while sharing the data to the user from the owner that the assurance of the security of DS among the group members who are joining dynamically and exit the group without sharing

data. A methodology that could sustain the users with dynamic alteration must assure that the latest joined users could access the stored data, and also, the user revocation can hold the data from the CCEs. The confidentiality of the data needs that the data is imperceptible to the CS and the non-registered users. The communication and the computational Complexity for sharing the SK among the group members will be dynamic. While sharing personal data, the method ensures that it cannot be shared with the attackers. The CS can afford the originality of the ownership of the data. The details of the owner's location should be stored on the server. Fault tolerance is another issue to organize the proposed method to identify the malicious members in the group. The DS based on privilege within the hierarchical data users is demonstrated in Fig. 4. The Data owner can hold the data file and selectively share several data users in the privacy privileges. The data user can share the data file with a rank-based model. The key generator may generate the PRK to access the data users in a group of attributes. The CS is the component for generating the encrypted components of the shared data.

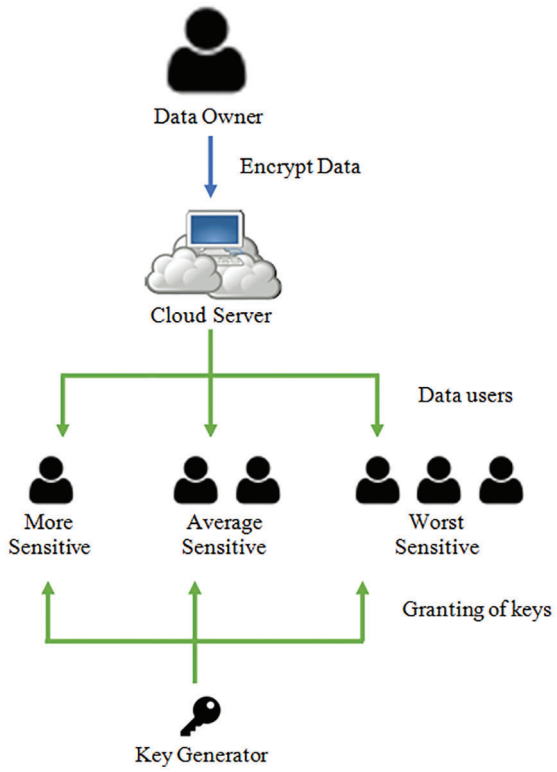

Figure 4: DS based on privilege

3.4 Partitioning of Data Files with Encryption

The file has been partitioned into a group of sectors like $File = \{File_1, File_2, \ldots, File_k\}$. Every file is called the latest file, which is joined with a sensitivity value to allocate the data users for accessing the privileges.

The procedure of partitioning the file is estimated according to the formation of the File. The single file is partitioned using records for several ways, as shown in Fig. 5. If the file contains only one record, then every file characterizes more than one attribute of records contained with the individual record. Whenever the file has several records, accessibility and flexibility are maintained to partition the data file. The agent is aware of the existence of the proxy. It is feasible for an agent to differentiate original encryption calculated under his

PBK using the Encrypt algorithm from a re-encryption CT on the similar message produced by the third party as the production of the *ReEnKey* algorithm. The input and the equivalent output of the *ReEnKey* algorithm in the transparent method cannot be linked. To overcome the security as mentioned earlier issues, we are implementing the TPRE. The definition of a unidirectional method consists of the following polynomial-time algorithms:

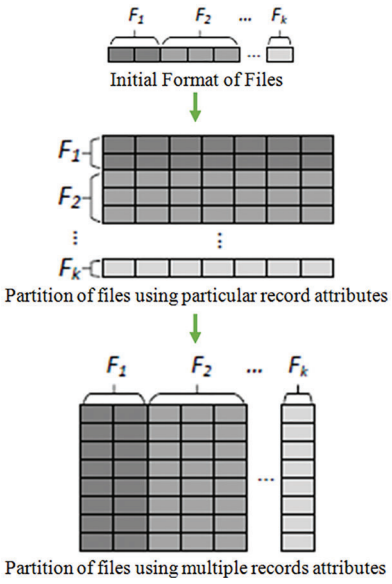

Figure 5: Partitioning of files

Key Generation

The input of security parameter *kEK*, is run by user '**B**' in this algorithm to produce its PBK/PRK pair $(pkb,\ skb)$.

ReEnKey

The input of key pair $(pkb,\ skb)$ for user '**V**' and a key pair $(pkb,\ skb)$ for client '**Q**' (skv is optional), the re-encryption key generation algorithm ReEnKey is achieved by user '**B**' to output a ReEnKey $rkb - v$. Since the ReEnKey $rkb - v$ allows transforming user '**B**'s' signature into user '**V**'s' signature, thus user '**Q**' acts as the agent authority, and user '**B**' acts as the agent.

Signature

The input of message *mEM* and its own PRK as *skb*, user '**B**' execute this algorithm to compute a corresponding signature *ob*.

ReSignature

In the input of a signature '**ob**' from user '**B**' and a re-signature key $rkb - v$, this algorithm is achived by the third party to produce a re-signed signature $ob - v$, if $Verify(pkb,\ m,\ ob)$ 1 hold; otherwise, this algorithm returns an error symbol indicating that '**ob**' is invalid.

Verify

In the input of a PBK of user b, the message *mEM* and an equivalent signature *ob*, a verifier achieves this algorithm to ensure the validity of the signature. If $Verify\ (pkb,\ m,\ ob)$ holds, it returns '1'; otherwise, it returns '0'. Tab. 1 illustrates the summary of notations used in this work.

Table 1: Summary of notations

Notation	Meaning
Per_{Au}	Permutation of authentication
α	A component in the password list
k	Key-Value
$Total_{Enc}$	Amount of converted password text
$Password_{hash}$	Hashed password
$Passord_{plain}$	Plain password
$Password_{modified}$	Modified password
Po	Policy
$\aleph$	Monotone combination
$\beth$	Secondary combination
$\mathbb{C}$	Access framework
φ	Null
β_i	Communication tree
ω_i	Level
$key_{\gamma_l^i}$	Threshold value
$key_{sec_l^i}$	SK
$Children_{\gamma_l^i}$	Value for leaf node
$key_{sec_l^i}$	SK value
δ	Tag value
v_i	Random challenge vector
f_i	File
p_k	PBK
s_k	PRK
s_t	State information
c	Challenge
a	The verification value
μ	Parameter value
$Encode$	Encoding value
kEK	Security parameter
b	User
$(pkb,\ skb)$	PBK/PRK pair
q	Client
$rkb - v$	ReEnKey
skb	Own PRK
$ob - v$	Re-signed signature

(Continued)

Table 1 (continued)

Notation	Meaning
$global_{id}$	The global identifier for every user
$private_{id}$	A private identifier for every parameter authority
PA_x	Parameter authority
SK_A	SK
δ	Data
A	Data owner
ρ	Constant value
$Cipher_{text}$	CT
$GlobalPublic_{key}$	Global PBK
$CertifcateAuthority_{setup}$	Certificate Authority setup
$Master_{key}$	Master key
$public_{parameter}$	Public parameter
sig_{CA}	Signature for certificate authority
ver_{CA}	Verification for certificate authority
$GlobalSecret_{key}$	Global SK
$user_{reg}$	User registration
$cert(global_{id})$	Certificate for global ID
$Parameter_{authority_{setup}}$	Parameter authority setup

4 Performance Evaluation

The implementation of the collaborative prototype is discussed in this section. The simulation experiment is validated using the various parameters. The proposed method ISIMSDS is compared with the related methods as SRMSM [24], PRMSM [25], and MOKS [26]. Fig. 6 demonstrates the Mean value of the parameters for producing a better cloud computing application. The result suggests that the mean value for the Security has the highest value according to the other parameters in the CCE. Fig. 7 demonstrates the efficiency comparison between the proposed method and the existing method that states remote-based administration and data management. The proposed method has shown better efficiency compared to the conventional methodology of the same level of configuration.

Fig. 8 demonstrates the computation time for the cloud key generation and identifies the proof for the integrity. The integrity of the collaborative CCE is analysed for various levels of iterations. For the first 10000 iterations, the time taken is up to 20000000 ms. Fig. 9 and 10 demonstrate that the time consumption for the EM and DM. The results proved that the proposed method has less time consumption for EM and DM.

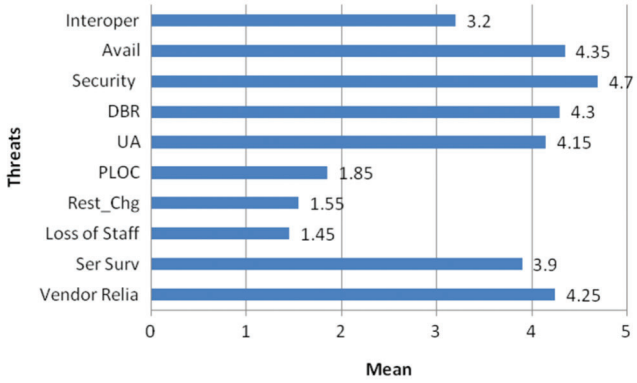

Figure 6: Threats *vs.* mean

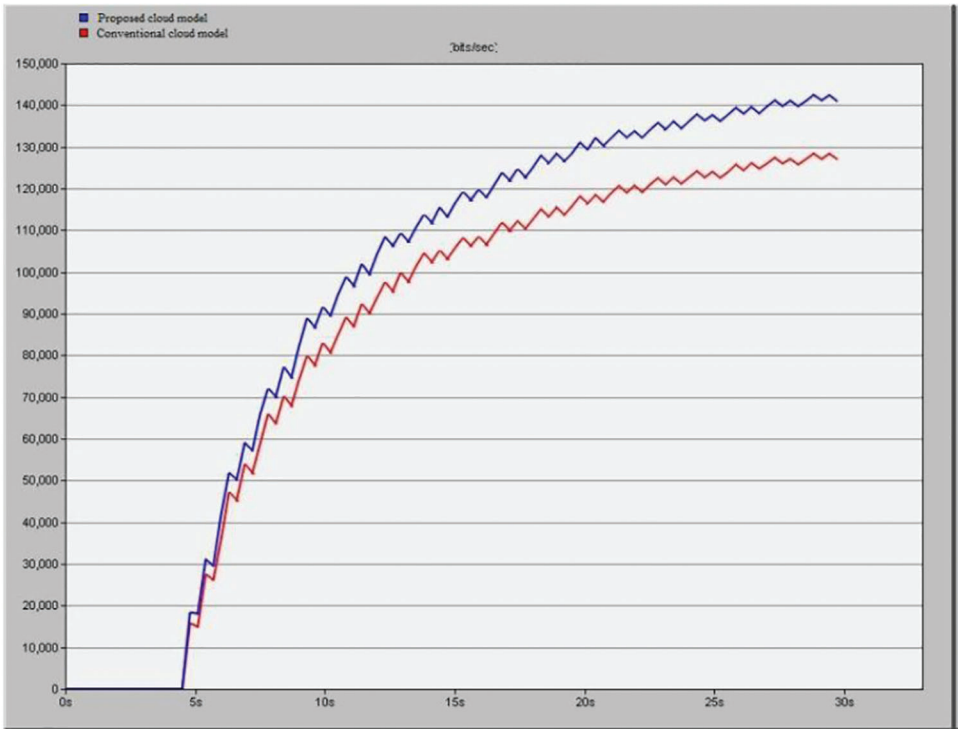

Figure 7: Efficiency comparison

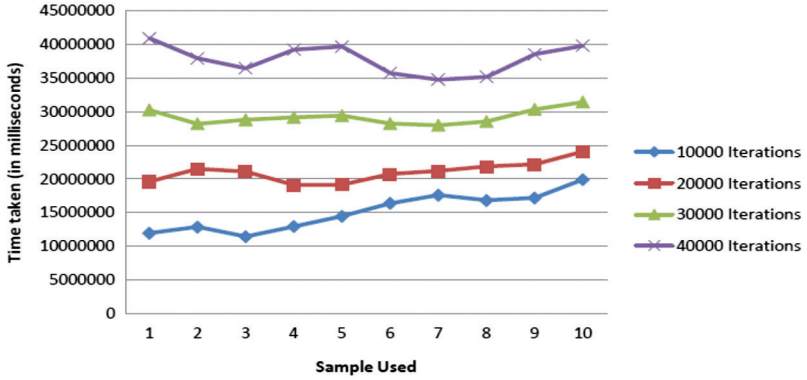

Figure 8: Computation time for iterations

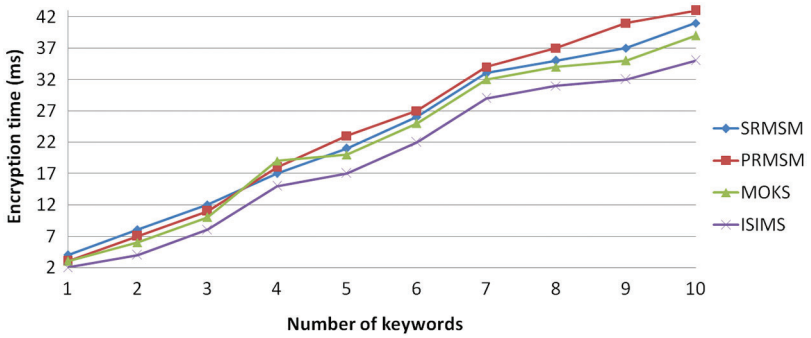

Figure 9: Time required for the encryption process

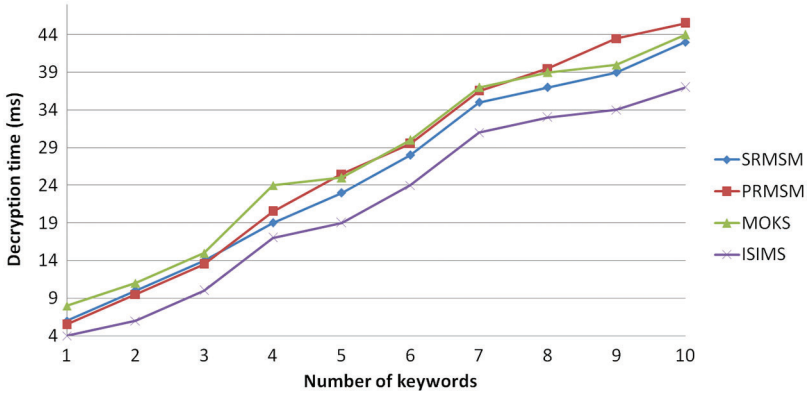

Figure 10: Time required for the DM

Fig. 11 illustrates the key generation time for computing the EM and DM. The security is enhanced by producing the key. Fig. 12 demonstrates the communication complexity for complete bits transmission in the proposed work. The result proved that the proposed method has less amount of communication complexity compared to the related methods. Fig. 13 demonstrates the retrieval time for constructing the CT to share a huge CCE data volume. Our proposed methodology is performed well for retrieval time compared to the related methods. Whenever the sharing of documents is very large, the correctness is checked for CT retrieval methodology.

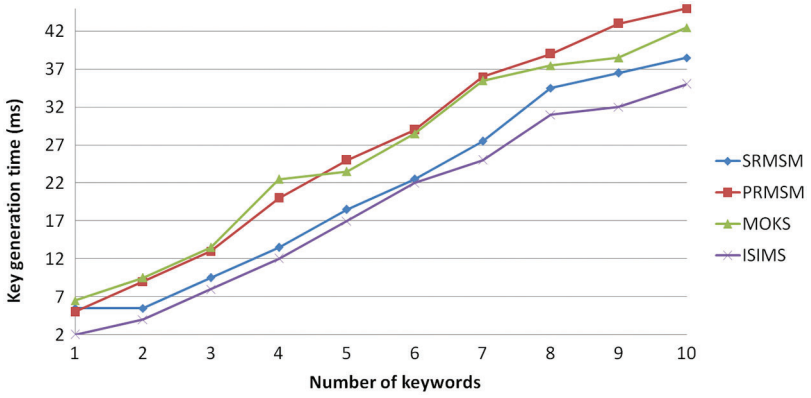

Figure 11: Key generation time

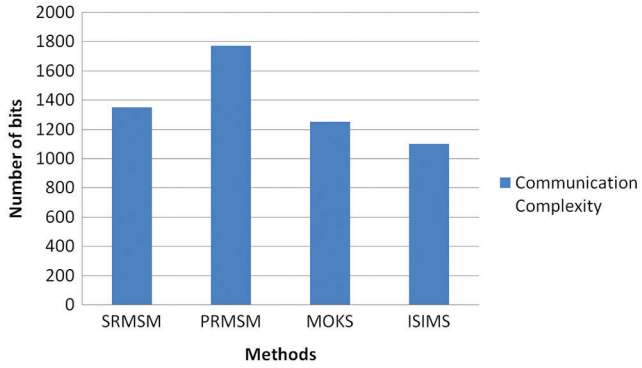

Figure 12: Communication complexity

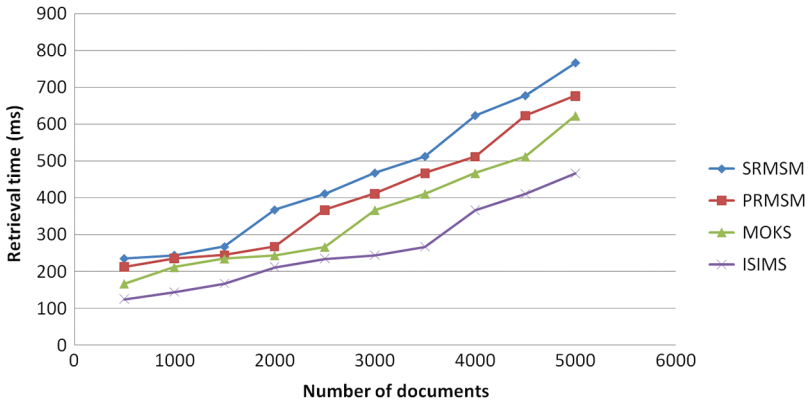

Figure 13: Retrieval time

The Complexity for computing the security parameters for the proposed method with the related methods is experimentally validated. The cost for computing the performance metrics of the proposed work is analyzed using various parameters like computing the mean value, efficiency comparison, computation time for iterations, encryption time, decryption, key generation and retrieval time, and communication complexity.

5 Conclusion

Several features are implemented by the CCEs that have a huge amount of multilevel structure to save the data and share among the users. This paper applies the security mechanism that the owner of the data shares in the CCE. The implementation is actively done with various parameters to provide security using the various key techniques. To overcome the performance issues, an ISIMSDS of DS is proposed to hold the DS of big data in CCEs. The proposed methodology partitions the data file into several segments related to the user's privileges and the sensitivity of the data. The outcome of the proposed ISIMSDS methodology discussed various parameters such as encryption, decryption, key generation, and computation time. These performances are better than the existing works. The proposed ISIMSDS methodology proved to prevent the chosen PT attack in various conditions and reduce the Complexity. The proposed work focuses on chosen-PT attacks because of provides the standard keys to the group. But threat issues occur due to internal attackers of the group members. The future work may focus on the internal attackers.

Funding Statement: The authors 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] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen *et al.,* "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1265–1277, 2016.

[2] G. Ateniese and S. Hohenberger, "Proxy re-signatures: New definitions, algorithms and applications," in *Proc. of the 12th Computer and Communications Security*, USA, pp. 310–319, 2005.

[3] F. Guo, Y. Mu, W. Susilo, D. S. Wong and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, 2014.

[4] K. Ganeshkumar and D. Arivazhagan, "Generating a digital signature based on new cryptographic scheme for user authentication and security," *Indian Journal of Science and Technology*, vol. 7, no. 12, pp. 1–5, 2014.

[5] R. Arun Prakash, T. Jayasankar and K. VinothKumar, "Biometric encoding and biometric authentication (beba) protocol for secure cloud in m-commerce environment," *Applied Mathematics and Information Sciences*, vol. 12, no. 1, pp. 255–263, 2018.

[6] P. Li, J. Li, Z. Huang, C. Z. Gao and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, vol. 21, no. 1, pp. 277–286, 2017.

[7] H. Liu, X. Yao, T. Yang and H. Ning, "Cooperative privacy preservation for wearable devices in hybrid computing-based smart health," *IEEE Internet Things Journal*, vol. 6, no. 2, pp. 1352–1362, 2019.

[8] H. Xiong, Z. Chen and F. Li, "Efficient privacy-preserving authentication protocol for vehicular communications with trustworthy," *Secure Communication Network*, vol. 5, no. 4, pp. 1441–1451, 2012.

[9] J. Yu, K. Ren, C. Wang and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Transaction Information Forensics Security*, vol. 10, no. 6, pp. 1167–1179, 2015.

[10] X. Chen, J. Li, X. Huang, J. Ma and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 546–556, 2015.

[11] B. Libert and D. Vergnaud, "Unidirectional chosen-cipher-text cipher-text secure proxy re-encryption," in *Int. Workshop on Public Key Cryptography*, Berlin, Heidelberg, Springer, pp. 360–379, 2008.

[12] G. Ateniese, K. Fu, M. Green and S. Honhenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 1, pp. 1–30, 2006.

[13] S. Lee, H. Park and J. Kim, "A secure and mutual-profitable DRM interoperability scheme," in *proc. IEEE Symposium on Computers and Communications*, Riccione, Italy, pp. 75–80, 2010.

[14] P. Vijayakumar, P. Pandiaraja, B. Balamurugan and M. Karuppiah, "A novel performance-enhancing task scheduling algorithm for cloud-based E-health environment," *International Journal of E-Health and Medical Communications*, vol. 10, no. 2, pp. 102–117, 2010.

[15] M. Mazini, B. Shirazi and I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 4, pp. 541–553, 2018.

[16] R. Kirubakaramoorthi, D. Arivazhagan and D. Helen, "Survey on encryption techniques used to secure cloud storage system," *Indian Journal of Science and Technology*, vol. 8, no. 36, pp. 1–7, 2015.

[17] A. M. Anusha Bamini and S. Enoch, "Optimized scheduling and resource allocation using evolutionary algorithms in cloud environment," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 5, pp. 125–133, 2017.

[18] K. Ganesh Kumar and S. Sudhakar, "Improved network traffic by attacking denial of service to protect resource using z-test based 4-tier geomark traceback (Z4TGT)," *Wireless Personal Communications*, vol. 114, no. 4, pp. 3541–3575, 2020.

[19] G. Jia, G. Han, H. Xie and J. Du, "Hybrid-LRU caching for optimizing data storage and retrieval in edge computing-based wearable sensors," *IEEE Internet Things Journal*, vol. 6, no. 2, pp. 1342–1351, 2019.

[20] R. Pitchai, S. Babu, P. Supraja and S. Anjanayya, "Prediction of availability and integrity of cloud data using soft computing technique," *Soft Computing*, vol. 23, no. 18, pp. 8555–8562, 2019.

[21] T. Yang, H. Xiong, J. Hu, Y. Wang and W. Xin, "A traceable privacy-preserving authentication protocol for VANETs based on proxy re-signature," in *Proc. of the 2011 IEEE 8th Int. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD'll)*, Shanghai, Chim, vol. 4, pp. 2217–2221, 2011.

[22] J. Li, Y. Zhang, X. Chen and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computer Security*, vol. 72, no. 4, pp. 1–12, 2018.

[23] S. Sudhakar and S. Chenthur Pandian, "Hybrid cluster-based geographical routing protocol to mitigate malicious nodes in mobile ad hoc network," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 21, no. 4, pp. 224–236, 2016.

[24] J. Shen, D. Liu, J. Shen, Q. Liu and X. Sun, "A secure cloud-assisted urban data sharing framework for ubiquitous-cities," *Pervasive Mobile Computing*, vol. 41, no. 6, pp. 219–230, 2017.

[25] W. Zhang, Y. Lin, S. Xiao, Q. Liu and T. Zhou, "Secure distributed keyword search in multiple clouds," in *IEEE 22nd Int. Sym. of Quality of Service (IWQoS)*, Hongkong, pp. 370–379, 2014.

[26] W. Zhang, Y. Lin and S. Xiao, "Privacy-preserving ranked multi-keyword search for multiple data owners in cloud computing," *IEEE Transactions on Computers*, vol. 6, no. 1, pp. 1566–1577, 2016.

[27] P. Vijayakumar, P. Pandiaraja, M. Karuppiah and L. J. Deborah, "An efficient secure communication for healthcare system using wearable devices," *International Journal of Electrical and Computer Engineering Systems*, vol. 63, no. 1, pp. 232–245, 2017.

[28] Y. Miao, R. Deng, K. K. R. Choo, X. Liu and H. Li, "Threshold multi-keyword search for cloud-based group data sharing," *IEEE Transactions on Cloud Computing*, vol. 99, pp. 1–18, 2020.