Wireless Personal Communications (2021) 116:1683–1702
https://doi.org/10.1007/s11277-020-07756-7

ECC-CRT: An Elliptical Curve Cryptographic Encryption and Chinese Remainder Theorem based Deduplication in Cloud

B. Rasina Begum[1] · **P. Chitra**[2]

Published online: 7 September 2020

Abstract
Cloud computing provides the data storage facility for the maintenance, management and remote backup of the data. The storage cost and data retrieving time has increased. To encrypt, the Elliptical Curve Cryptographic and to generate key the Chinese Remainder Theorem (ECC-CRT) based deduplication scheme is proposed. Data deduplication is the process of eliminating the repeated data in cloud storage. Cloud service provider receives the encrypted data and checks for duplication. Deduplication is performed by using cosine similarity checking. The advantage of the method is, it avoids malicious upload and downloads in storage space. The performance is compared with existing methods.

Keywords Chinese Remainder Theorem (CRT) · Elliptical Curve Cryptography (ECC) · Key generation · Deduplication and Cosine similarity

1 Introduction

Cloud computing is an emerging technology because of their ability to offer the cost efficient and on demand use of huge storage. Cloud is the terminology, which is commonly utilized to refer a bunch of components. Cloud computation is the practice of delivering computational services like servers, storages, data bases, networking, and applications. The groups that provide those services are termed as Cloud Service Provider (CSP). Cloud computing is a model for enabling convenient, on demand network access to shared pool of configurable computing resources. The service models of cloud computing was categorized into three such as Infrastructure as a Service (IaaS), [1] Platform as a Service (PaaS), and Software as a Service (SaaS). The cloud offers many benefits such as, fast deployment, pay for use, lower cast, scalability, rapid provisioning, rapid elasticity, ubiquitous network access, greater resiliency, hypervisor protection against network attacks, low cost

✉ B. Rasina Begum
rasinabegumphd@yahoo.com

[1] Department of Computer Science and Engineering, Mohamed Sathak Engineering College, Keelakarai, Tamil Nadu 623806, India

[2] Department of Computer Science and Engineering, Thiyagarajar College of Engineering, Madurai, Tamil Nadu 625 015, India

disaster recovery and data storage solutions, on demand security controls, real time detection of system tempering and rapid re-constitution of services. In cloud computing, it categorized into four, such as private cloud, public cloud, hybrid cloud and community cloud. Data deduplication is the process of reducing the storage space. It identifies the redundant data by using hash values and compares the data chunks and generates the logical values to other copies instead of storing other authentic copies of redundant data. Deduplication reduces the data volume so the disk space and network bandwidth are reduced, so it reduces the cost and energy consumption for running storage system. Data duplication is considered as a way to reduce or eliminate the redundant files, bytes or data blocks and ensures the unique data copy stored via duplicate data detection. It is used to control the data functionality and reduces the demands for capacity during the data protection. Single Instance Storage (SIS) merges the duplicate files into shared storage space. The duplicate files are replaced with file links. These links are operated like an original files. It saves the disk space occupied by duplicate files [2] and maximizes the use of current storage resources. Deduplication includes three methods, such as,

File level deduplication: it identify the files with different names and same content. These types of files are eliminated and stored with link or numerical values.

Block level deduplication [3] groups the data stream into fixed length or variable length data blocks and then matched with their stored data blocks to detect or identify the same data blocks. Byte level deduplication [4] compares the data stream with stored data stream byte by byte. It uses a cache disk to back up the data temporarily, before byte level data deduplication. So this process may be limited to set of backup data streams rather than entire backup.

In deduplication process, the different methods are used to evaluate the efficient method. The existing methods are RSA (RSA stands for Rivest Shamir Adleman considered as first public key cryptosystems used for data transmission in secure way), Elgamal, convergent encryption (CE), leakage resilient (LR) deduplication, Randomized Convergent Encryption (RCE). The cloud services have the ability to provide data backup. The users store their confidential data, which could be individual or business oriented. The user can retrieve their information from the cloud, after recovering from their calamity conditions. They can smoothly continue their regular process without any further delay. The existing techniques utilized the Elliptic Curve Cryptography (ECC) for encrypting the data. The scheme is formulated on the basis of algebraic structure of elliptic curve. In RSA scheme [5], the encrypting time is higher and it is not applicable to single user environment. Convergent Encryption (CE) is utilized to convert encrypted key from hash of plain text. In ByCE technique, the same encryption key is applied for two identical plain text to obtain same ciphertext. So the CSP is able to perform deduplication on ciphertext. CE offers data privacy in deduplication process. The secure CE is for efficient encryption and it considering the key management and block level deduplication.

Leakage resilient (LR) deduplication scheme: it is applied to solve the data integrity problems. LR enables the use of randomly selected key to encrypt the data, then the data encrypted key is encrypted under a KEK-(stands for Key Encryption Key which states as a key encryptes other key used for storage or transmission) which is derived from the data and distributed among the data holders after the Proof-of ownership (PoW) process. Data integrity is checked by using the data encryption key with same KEK.

The CRT generates the key to encrypt the data by using ECC and these data are stored in server at that time, the cosine similarity is used to identify the similar data. If the data are similar, then the similar data are required to generate the hash value by using hash function. If the data are not similar it will be stored in cloud. Then the data are stored in

cloud. So the data storage space is reduced in cloud. In this work, the novelty is implemented in key generation, encryption and decryption process for improving the security. It includes some limitations; secure proof of ownership (PoW) scheme in the standard model is an open problem and lack of dynamic ownership management between the data holders.

The remaining section of the paper are organized as follows: Section 2 presents a brief review of the existing research works related to the cloud security, and deduplication of cloud storage. Section 3 shows the detailed description of proposed work. Section 4 illustrates the performance analysis and comparative analysis of proposed work. Section 5 involves the short discussion of conclusion and future work.

2 Related Works

This section presents some of the existing works based on key distribution, attributed based encryption algorithm for cloud security. Ryan [6] surveyed the security issues in cloud computing. The safety and problems related to privacy in cloud systems kept on growing. Kshetri [7] framed a strategy to resolve the confidentiality and safety problems in cloud computing. The issues were categorized into technology based issues, human intrusion issues and corrupted data storing in cloud. Advanced research has been performed to explore the past history of a cloud organization before deploying them in cloud. Compatibility, complexness, observability and trialability had to be analyzed. Previous techniques were focused on security issues in data storage. Wei et al. [8] aimed to solve the computation security. A novel auditing algorithm termed Sec-Cloud was framed to bridge storage security and computational security. Different requirements of various users were controlled by batch verification method. The development process was to implement the algorithm in linear computation and data mining. Real cloud platforms such as EC2 and open stack were also planned for implementation. Sujithra et al. [9] gave a cryptographic tactic to preserve the information in cloud and also maintained performance level. Previously the cryptographic technique was implemented to individual system. Throughput required to be enhanced. Liu et al. [10] designed a novel security scheme for cloud system. In this scheme, the data sender encrypts the data with the identity parameter of the receiver and transmitted it to the cloud. The user was provided with a private key and a security device to access cloud. The user executed the same encryption algorithm to decrypt the data in cloud. If the security device was lost, the CSP would execute an algorithm that changed the decryption code so that the unauthorized use of the device was prevented. The improvement of the scheme was to enhance the efficiency of the algorithm. Cloud computing paved the path to data. The insurance process depended on the cloud system for their smooth operation. The exchanged data faced security issues. The Patient Health Record (PHR) management was the significant consideration of the scheme proposed by Li et al. [11], in which a patient-centric model was designed. Attribute Based Encryption (ABE) was executed to encrypt the PHR. It reduced the complexity in managing the key. The PHR owner decided to permit access rights to other agents over their data. The scheme required enhancement in managing trust of external users who request the access of PHR. Rapid improvement in cloud computation led to data privacy in addition to duplication of user identity. Jung et al. [12] proposed an anonymity control algorithm to completely prevent the user identity leakage and an efficient revocation of user to reduce network traffic in cloud. The accumulation of encrypted data in cloud increased rapidly. This slowed down the query process in cloud computation. Li et al. [13] developed a Key word Search

Function (KSF) and reduced the computational cost at both the owner and other trusted authorities. Future process was intended to frame the KSF for standard model which would be CCA secured. The Proxy Re-Encryption (PRE) was executed by the user to cancel the permission rights of any other user from accessing the private cloud data. Numerous PRE schemes were formulated. Liang et al. [14] made an enhancement to the PRE scheme, termed as Cypher text Policy Attribute Based PRE (CPB-PRE). It enhanced the CCA secure process of storing, forwarding and sharing data in cloud servers. The cloud provided a central data storage facility which was globally accessed by the users. Internet services costs was higher for the users to access their data in cloud. To reduce the cost, Tysowski and Hasan [15] offered, an improved encryption technique on attribute basis. The cloud service provider re-encrypted the data to eliminate revocation cost to users and in addition it provided data security. This scheme reduced computational complexity and network traffic. Security to data in cloud computation was accomplished by control of data access. Cipher text Policy Attribute Based Encryption (CP-ABE) technique gave the data owners to directly control their data accessibility in cloud. Yang and Jia [16] proposed, a revocable multi authority CP-ABE scheme to revoke attributes in addition to providing security to the data. The scheme was secure in random oracle model. Development works were to implement the model in complex remote storage system and social networks. The cloud network faced numerous safety and confidentiality problems. Cloud data accessing by individual users was surpassed by data sharing among organizations, both internal and external. Fabian et al. [17] developed, an attribute based data encryption technique, to share data among different cloud servers. The model contained individual cloud providers grouped under one cloud environment. Inter—organizational medical big data of patients was shared securely among these clouds. Next level of improvement was to manage the private and public keys of the cloud users to enhance security while sharing data. The online data sharing practice grew with the invention of cloud computing. Optimum security to online data was the key to successful cloud computing. Patranabis et al. [18] combined Chosen Plain text Attacks (CPA) and Chosen Cipher Attacks (CCA) to construct, Key Aggregate Cryptosystem (KAC) to solve the security issues in online sharing of license of any product and other online data sharing criticalities. The scheme utilized lower overhead cipher texts and key aggregates along with asymmetrical bilinear paring. The succeeding phase of the work was to enhance the scheme's scalability.

Jia et al. [19] imposed the Chinese Remainder Theorem in their study for generating secret reconstruction and share generation validity. This kind of strategy used for computational security. For recovery lesser difficulty and size of sharing also lesser which have been exhibited in this proposed study. The secret recovery computational complexity termed as O(t) related to the previous O(tlog2 t). Prasetyo et al. [20] exhibited the Chinese Remainder Theorem CRT and Boolean X-OR process to generate n number of shared images from the n number of secret images. By this proposed approach the security increased by shared color imaged merged into 2-D matrix representation. Yan et al. [21] designed a Chinese Remainder Theorem based on 2-in-1 image secret sharing with visual previewing capability, lossless recovery and grayscale stacking recovery decoding options. The CRT theorem encoded the grayscale secret image. Further it has been decoded through linear congruence set equations.

Wang et al. [22] The data reduction techniques for the storage and savings of I/O through duplicate removal content has considered as compression and De-duplication techniques and however for index management they also incur significant memory. To overcome this problem Austercache- a new flash caching method used for indexing the memory efficient when preserving the deduplication and compression benefits. This proposed

method focused on the Austercache management and various core methods proposed for cache replacement and data organization and implement the lightweight in memory index structures by eliminates the much indexing metadata. This proposed flash caching design supported the compression and deduplication and maintains the read hit ratio and write reduction ration and high I/O throughput obtained at the end. Yan et al. [23] The privacy of data holders preserved and the in cloud the data have stored in encrypted form. Yet the new challenges introduced by the encrypted data for the cloud data deduplication and it becomes important for cloud processing and big data storage. On encrypted data conventional deduplication schemes have not employed. Security weakness suffered by encrypted data deduplication's traditional solutions. The revocation and data access control cannot be flexibly supported. In this study, deduplicate encrypted data scheme proposed for cloud based storage on proxy re-encryption and ownership challenge. Cloud data deduplication with access control integrated. Based on computer simulations and extensive analysis the performance evaluated. For big data deduplication in storage of cloud the efficiency resulted. In this study, [24] for bipolar and bipolar neutrosophic set of interval-valued the weighted cosine similarity measures and cosine similarity measure proposed. These similarity measures ratios have been examined and based on proposed measures the two multi attribute decision making methods focused. The feasibility of proposed study made and found the bipolar disorder similarity using cosine similarity technique. [25] Among picture fuzzy set based cosine function the eight similarity measures presented by positive membership degree, neutral membership degree, refusal membership degree and negative membership degree have been considering in this study. For strategic decision making these similarity measures of weighted cosine function applied. For strategic decision making issues these similarity measures efficiency have been demonstrated.

3 Propose Work

In this section, the detailed description of proposed CRT-ECC based cloud architecture is explained for secure data sharing and reducing the storage space by using deduplication process. Figure 1 shows the overall flow of proposed work. Initially, the user generates the key using Chinese remainder theorem and the data are encrypted by using elliptical curve cryptography. Then the encrypted data are stored in the server. When the data are stored in the server then, the cosine similarity is used to check the similarity of the data. If the data are similar then the hash function is generated for the particular data otherwise the data are stored in the cloud. This method is used for reducing the storage space in cloud.

3.1 Key Generation Using CRT

The user selects the input data and service provider send the control information to Chinese remainder theorem for generating the secret key. The novelty is presented in this section. The CRT generates the key using control information for each user. The advantage of Chinese remainder theorem reduces the complexity of key generation in cryptography system. Let $m_1, m_2, \ldots, m_3$ are the collection of pairwise relatively prime integers and consider a_1, a_2,…a_m are the arbitrary integers. Based on this, the CRT illustrates the congruence system as follows,

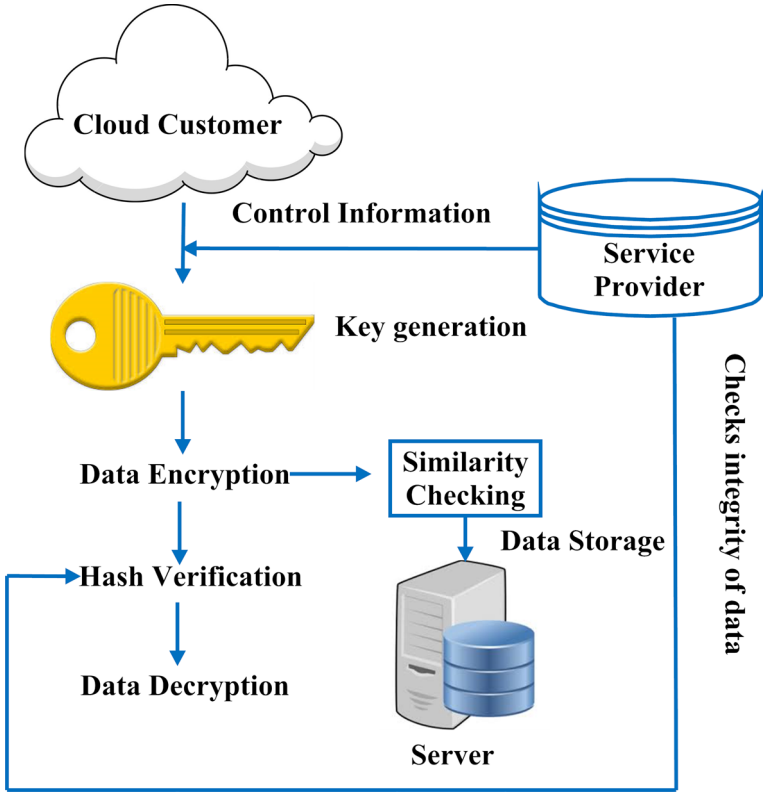


Fig. 1 Overall flow diagram of proposed work

$$\begin{aligned} Y &\equiv a_1 \left(modm_1 \right) \\ Y &\equiv a_m \left(modm_n \right) \end{aligned} \tag{1}$$

Here, the service provider provides the secret information based on their factorized value. Then the control information offers different information for each user. Finally, the CSP checks their factorized value to secret key.

Algorithm I: Chinese Remainder Theorem Based Key Generation

CRTK ← *Chinese remainder theorem key*

Given $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)$

If the secret is valid, then the CRT-based key generation is expressed as,

$$N = Y_1 * Y_2 * \ldots * Y_n$$

$$CRTK \equiv X_1 * N_1 * N_1^{-1} + \ldots\ldots\ldots + X_n * N_n * N_n^{-1} (mod\ N)$$

Where, $N_i = \frac{N}{y_i}$, $N_i^{-1} = N_i \bmod y_i$

3.2 ECC Encryption

Elliptical curve cryptography is used to encrypt the data then the encrypted data are stored in the cloud. The algebraic construction of an ellipse is basis behind formation of the technique. The equation for an ellipse given by

$$y^2 = x^3 + ax + b \tag{3}$$

Suppose if a user needs to exchange data with another, then they must know a secret key. The private key being d which is an integer, selected on a random basis within the interval [1, n-1]. The public key Q is also employed which is given by

$$Q = dG \tag{4}$$

d-Private key

G-base point

The sender's key be $(d_A Q_A)$ and the key of receiver be $(d_B Q_B)$.It is required that each users must possess the public key of the other user in order to exchange the data.

The sender of the data computes,

$$k = d_A Q_B \tag{5}$$

The receiver compute,

$$k = d_B Q_A \tag{6}$$

The above values are calculated by both sender and receiver are equal since,

$$d_A Q_B = d_A d_B G = d_B d_A G = d_B Q_A \tag{7}$$

This curve includes numerous value, characters and finite number of elements. The elliptical curve is derived by arithmetic (addition, subtraction, multiplication and division) operations for both encryption and decryption. ECC provides high security to the data with minimum cost.

Algorithm II: Modified ECC Encryption

Input: *CRTK, Raw Data*

Output: *encrypted data*

Sd←select data

Ed ← encrypted data

Pu ← public key

k = constant

Generate the CRT

\\ Set CRTK as secret key

Pr=CRTK

Initialize P as point on curve

Initialize K as random number generated between (1 to n-1)

Generate the public key

*Pu = Pr * p*

Encrypted data by below steps,

while *Sd.data*

$$Ed = Sd.data + (k * Pu) \tag{8}$$

end while

Transfer Ed from consumer to service provider

The encrypted data (Ed) is fully protected for secure cloud sharing. Authenticated users of the network can access *Ed*. The private key is generated by using Chinese remainder theorem. k represents the constant value and Pu represents public key. The encrypted data calculates by using Eq. (8). Then the encrypted data is transfered from consumer to service provider.

3.3 Deduplication

In deduplication process, the similarity checking is the main process. It is used to match the stored encrypted data. If the data are same, then it will generate the hash value. Otherwise the encrypted data are stored in cloud. If the cosine similarity value is less than 0.9, then the data are stored in cloud. If the cosine similarity value is greater than 0.9, then it is duplicate data so these data are not stored in cloud. But it generates the hash value. So the storage space of cloud is reduced. This is represented in Fig. 2. The cosine similarity of two non-zero vectors are derived by using Euclidean dot products,

The dot product of two vectors $A_k$and B_k is represented as,

$$A_k.B_k = A_k B_k \cos\theta$$

Here, θ is the measure of angle between $A_k and B_k$

The dot product is defined as the sum of product of each vector as,

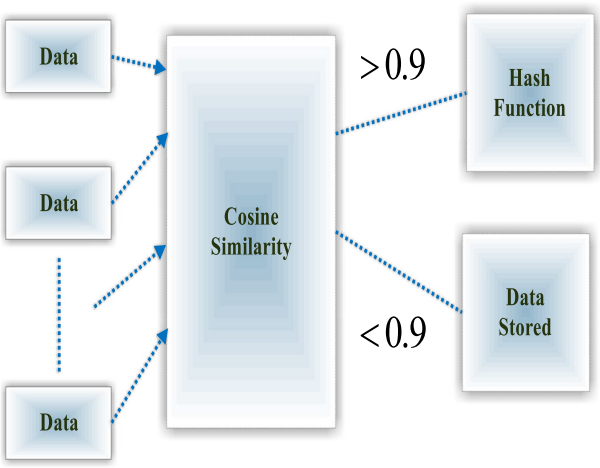


Fig. 2 Similarity matching

Algorithm III: Similarity Checking

Cs ← *cosine similarity*

Exd←*existing data*

For *i=0*: *Sd*

do

for *j=0*: *Exd*

do

Cosine similarity is calculated by using equation (10)

end for

end for

if $Cs_{ij} > 0.9$**then**

Duplicate data

else

New data

end if

$$A_k.B_k = A_{k1}.B_{k1} + A_{k2}.B_{k2} + A_{k3}.B_{k3} + \ldots \quad (9)$$

$$Cs_{ij} = \frac{\sum_{k=1}^{n} A_k B_k}{\sqrt{\sum_{k=1}^{n} A_k^2}\sqrt{\sum_{k=1}^{n} B_k^2}} \tag{10}$$

$A_k B_k$ – term frequency vector document

The above algorithm III represents the detail explanation of similarity checking. It is used to check the similarity of data and similarity data are termed as duplicated data. If the similarity matching is less than 0.9 these data are consider as new data.

3.4 Hash Key Verification

The hash key verification, the user stores the encrypted data, if data are not similar to any data then data is stored in server

After generating hash of data. If data is similar to already stored data then the access policy is updated for each data in server. Hash is transferred from service provider to customer. Then service provider gets request from customer and gets data from server. Service provider checks the integrity of data by compare hash. Figure 3 shows the flow diagram of hash function. Initially, the encrypted data generate the hash value and the values are compared. If the hash value of the data is checked whether it is existed in the cloud if so it's sets the pointer to existing data location or else it saves a new data in the cloud.

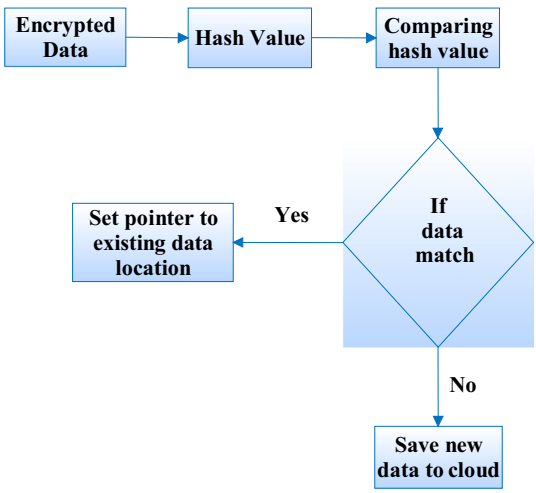


Fig. 3 Hash function

On upload (Deduplication)

Mh1 ← *message hash*

Mh2 ← *generate Ed.hash*

Transfer hash from service provider to consumer

Fr ← *file request*

Request transferred from consumer to service provider

Mh1 ← *message hash*

Mh2 ← *generate Ed.hash*

if *Mh2==Mh1* **then** // *Service provider side verification*

Hash verified

else

Hash not verified

end if

//On download (Integrity)

Transfer data from service provider to consumer

Mh1 ← *message hash*

Mh2 ← *generate Ed.hash*

if *Mh1==Mh2* **then** // *Consumer side verification*

Hash verified

else

Hash not verified

end if

3.5 ECC Decryption

Here the encrypted data are decrypted by using master keys. If the secret keys are matched with encrypted key, then the user decrypt the data otherwise, the user does not retrieve the original data. An algorithm IV represents the detail description of the ECC decryption. CT1 denotes the cipher text 1 and *Dd* denotes the decrypted data. The cipher text 1 is calculated by using Eq. (11) and decrypted data is calculated by using Eq. (12)

Algorithm IV: ECC Decryption

Input: *encrypted data*

Output: *decrypted data*

User send the request to service provider

Service provider send the control information to user

User generates the CRTK

Pr=CRTK

Decrypted data is obtained by below steps,

$CT1 = k * p$ (11)

while *Ed.data*

$$Dd = Ed.data - (CRTK * CT1) \quad (12)$$

Here,

Dd ←*decrypted data*

Pr ← *CRTK*

k = constant

P as point on curve

end while

Fig. 4 Encryption Time

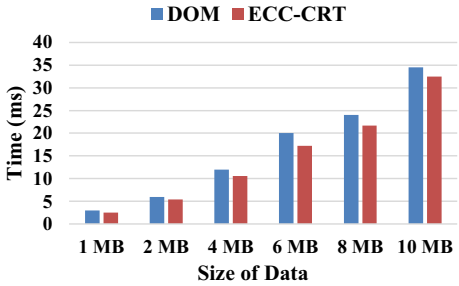
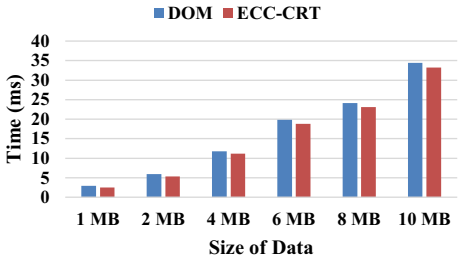


Fig. 5 Decryption



4 Performance Analysis

The performance results are compared with Data Deduplication with Dynamic Ownership Management (DOM), Convergent Encryption (CE), Leakage-Resilient (LR), Rivest Shamir Adleman (RSA), and Randomized Convergent Encryption (RCE). Here the proposed method provides the better result than existing techniques. An encryption time, decryption time, computational time, and memory consumptions are compared with existing methods.

4.1 Encryption Time

Encryption is the method of transforming any message form into unidentified scripts, for the purpose of protecting the data from unauthorized handling. The time consumed to encrypt any data is designated as encryption time. Figure 4 shows the encryption time with different data size for both existing [26] and proposed method. The X axis represents the size of the data and y axis represents the time in millisecond (ms).If the data size is 1 MB, then the existing method DOM takes 3 ms and the proposed method ECC-CRT takes only 2.5 ms. It clearly explains that the proposed method takes less time for encryption than existing technique. If data size increases, the encryption time also increases.

4.2 Decryption Time

It is the time taken by any technique to decipher or convert the encrypted data into their original format and it is termed as reverse process of encryption. The data decrypting does not require the same public key which is used encode the data. But instead, it

generates their specific private key from the attributes received from the user. Figure 5 shows the decryption with different data size for both existing [26] and proposed technique. If the data size is 1 MB, the existing method DOM takes, 2.9 ms but the proposed method takes 2.5 ms. If the data size is 10 MB, then the existing method DOM takes 34.4 ms and the proposed method takes 33.2 ms. Here, the decryption process takes less time than existing method. It is the one of the advantage of proposed technique. Decryption time increases when the data size are increased.

4.3 Computation Time Analysis

Here, the computational time represents both data uploading and downloading time. Figure 6a shows the computation time of data uploading. In x axis represents the data size in megabytes (MB) and y axis represents time in milliseconds (ms). Computation time is termed as time taken for their process (data uploading or data downloading).

If the size of data is 1 MB then the existing methods CE, LR, RCE, DOM and the proposed method ECC-CRT take 3 ms, 7 ms, 3 ms, 3 ms and 2 ms. If the data size is 10 MB then, the existing methods CE, LR,RCE, [27] DOM take 35 ms, 69 ms,35 ms,35 ms and the proposed method ECC-CRT takes 32 ms. The proposed method ECC-CRT takes minimum time then existing methods. If the data size increases the computation time also increases.

Figure 6b represents the computation time of data downloading. Here, the x axis represents the data size in megabyte (MB) and the y axis represents the time in milliseconds (ms). The data size is 10 MB then the existing methods CE, LR, RCE, DOM and the proposed method ECC-CRT takes 34.5 ms, 34.5 ms, 34.5 ms, 34.8 ms and 34.2 ms. The proposed method ECC-CRT takes minimum time for data downloading process then existing methods.

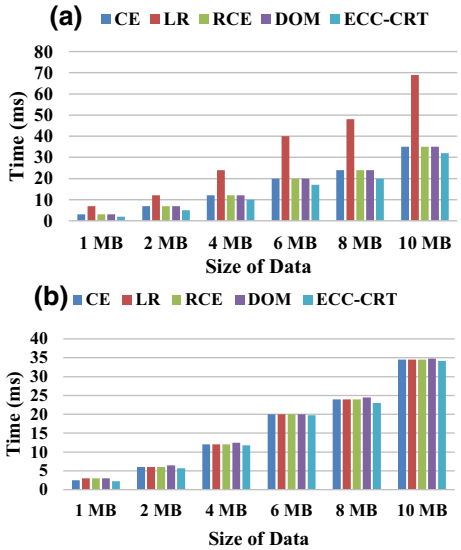


Fig. 6 **a** Uploading time, **b** Down loading time

Fig. 7 Memory consumption

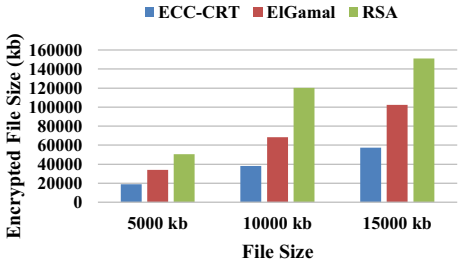
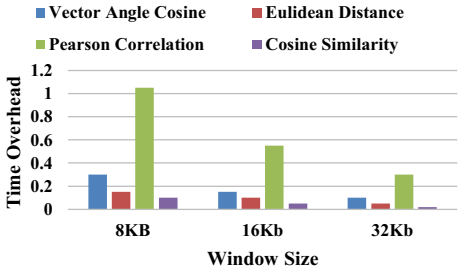


Fig. 8 Similarity comparison



4.4 Memory Consumption

Memory consumption represents the data which are occupied memory space in cloud. Figure 7 shows the memory computation of both existing ElGamal, RSA and proposed ECC-CRT. The encrypted data takes minimum storage space than existing methods of ElGamal and RSA.

4.5 Similarity Comparison

The similarity is compared with varied time period and different window size. The proposed cosine similarity is compared with existing vector angle cosine, Euclidean [28] distance, and Pearson correlation. In *X* axis represents the window size in kilobyte (Kb) and *Y* axis represents the time. The window size increases the time also increases. The proposed Cosine similarity takes less time than existing methods (Fig. 8).

4.6 Storage Space

Figure 9 shows the storage space for different existing methods of no deduplication, perfect deduplication, extreme [29] binning and proposed ECC-CRT. In this graph, the x axis represents the files and y axis represents the storage space in megabyte (MB). The file size is 25, then the existing methods of no deduplication, perfect deduplication extreme binning and proposed ECC-CRT take 950 MB, 280 Mb, 290 MB and 250 MB storage space. The proposed ECC-CRT method takes minimum space for storage than existing methods.

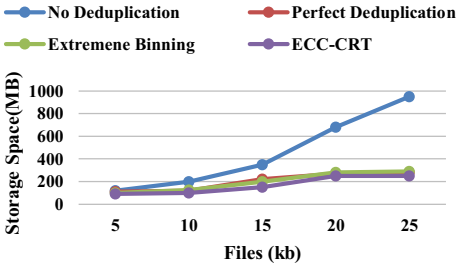


Fig. 9 Storage space

5 Conclusion

This paper presented the secure data sharing and increased security of cloud data. The main contribution of the paper has considered as to provide the high security to cloud data. For achieving this high security the efficient technique like Chinese Remainder theorem applied in ECC cryptography for key generation and also the in case of de-duplication the similarity identified by cosine similarity measure. The user selected the input data and CRT has been used to generate the key and it termed as master key. Then the master key has been used to encrypt the data by using elliptical curve cryptography. Next, the cloud service provider receives the encrypted data and check the deduplication by using cosine similarity. If the data have not showing similarity then these data have stored in cloud. If the data shows similarity, then the server generated the hash and the access policy has been updated. Then the hash has transferred from service provider to customer. In decryption, the customer sends the request to the service provider. Then the service provider checks the integrity of data by comparing with hash and then customer decrypt the data. In the above experiments, the performance of the existing and proposed techniques have been analyzed and evaluated. From the result, the proposed technique provides the better result when compare to other DOM, ElGamal, RSA, vector angle cosine, Euclidean distance, and Pearson correlation techniques. For future enhancement, the various other efficient algorithm can be used which focusing on the time consumption while generating the keys and also rapid computation to achieve superior performance in case of high security required.

References

1. Arora, S., & Beri, R. (2017). Adoption and Use of Cloud by Small and Medium Businesses (SMBS). *Advances in Computational Sciences and Technology, 10,* 529–536.
2. Kawtikwar, N. P., & Joshi, M. (2017). Data Deduplication in Cloud Environment using File-Level and Block-Level Techniques. *Imperial Journal of Interdisciplinary Research, 3*(5), 1294–1298.
3. Li, J., Li, Y. K., Chen, X., Lee, P. P., & Lou, W. (2015). A hybrid cloud approach for secure authorized deduplication. *IEEE Transactions on Parallel and Distributed Systems, 26,* 1206–1216.
4. Xu, L., Pavlo, A., Sengupta, S., & Ganger, G. R. (2017). Online Deduplication for Databases. In *Proceedings of the 2017 ACM International Conference on Management of Data*, (pp. 1355–1368).
5. Yarom Y., & Falkner, K. (2014). FLUSH + RELOAD: A high resolution, low noise, L3 Cache side-channel attack. In *USENIX Security Symposium*, (pp. 719–732).
6. Ryan, M. D. (2013). Cloud computing security: The scientific challenge, and a survey of solutions. *Journal of Systems and Software, 86,* 2263–2268.
7. Kshetri, N. (2013). Privacy and security issues in cloud computing: The role of institutions and institutional evolution. *Telecommunications Policy, 37,* 372–386.

8. Wei, L., Zhu, H., Cao, Z., Dong, X., Jia, W., Chen, Y., et al. (2014). Security and privacy for storage and computation in cloud computing. *Information Sciences, 258,* 371–386.
9. Sujithra, M., Padmavathi, G., & Narayanan, S. (2015). Mobile device data security: a cryptographic approach by outsourcing mobile data to cloud. *Procedia Computer Science, 47,* 480–485.
10. Liu, J. K., Liang, K., Susilo, W., Liu, J., & Xiang, Y. (2016). Two-factor data security protection mechanism for cloud storage system. *IEEE Transactions on Computers, 65,* 1992–2004.
11. Li, M., Yu, S., Zheng, Y., Ren, K., & Lou, W. (2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems, 24,* 131–143.
12. Jung, T., Li, X.-Y., Wan, Z., & Wan, M. (2015). Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *IEEE Transactions on Information Forensics and Security, 10,* 190–199.
13. Li, J., Lin, X., Zhang, Y., & Han, J. (2016). KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage. *IEEE Transactions on Services Computing*, *10*, 715–725.
14. Liang, K., Au, M. H., Liu, J. K., Susilo, W., Wong, D. S., Yang, G., et al. (2015). A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. *Future Generation Computer Systems, 52,* 95–108.
15. Tysowski, P. K., & Hasan, M. A. (2013). Hybrid attribute-and re-encryption-based key management for secure and scalable mobile applications in clouds. *IEEE Transactions on Cloud Computing, 1,* 172–186.
16. Yang, K., & Jia, X. (2014). Expressive, efficient, and revocable data access control for multi-authority cloud storage. *IEEE Transactions on Parallel and Distributed Systems, 25,* 1735–1744.
17. Fabian, B., Ermakova, T., & Junghanns, P. (2015). Collaborative and secure sharing of healthcare data in multi-clouds. *Information Systems, 48,* 132–150.
18. Patranabis, S., Shrivastava, Y., & Mukhopadhyay, D. (2017). Provably Secure Key-Aggregate Cryptosystems with Broadcast Aggregate Keys for Online Data Sharing on the Cloud. *IEEE Transactions on Computers, 66,* 891–904.
19. Jia, X., Wang, D., Nie, D., Luo, X., & Sun, J. Z. (2019). A new threshold changeable secret sharing scheme based on the Chinese Remainder Theorem. *Information Sciences, 473,* 13–30.
20. Prasetyo, H., & Guo, J.-M. (2019). A note on multiple secret sharing using Chinese remainder theorem and exclusive-OR. *IEEE Access, 7,* 37473–37497.
21. Yan, X., Lu, Y., Liu, L., Liu, J., & Yang, G. (2018). Chinese remainder theorem-based two-in-one image secret sharing with three decoding options. *Digital Signal Processing, 82,* 80–90.
22. Wang, Q., Li, J., Xia, W., Kruus, E., Debnath, B., & Lee, P. P. (2020) Austere Flash Caching with Deduplication and Compression. In *2020 {USENIX} Annual Technical Conference ({USENIX} {ATC} 20)*, (pp. 713–726).
23. Yan, Z., Ding, W., Yu, X., Zhu, H., & Deng, R. H. (2016). Deduplication on encrypted big data in cloud. *IEEE Transactions on Big Data, 2,* 138–150.
24. Abdel-Basset, M., Mohamed, M., Elhoseny, M., Chiclana, F., & Zaied, A. E.-N. H. (2019). Cosine similarity measures of bipolar neutrosophic set for diagnosis of bipolar disorder diseases. *Artificial Intelligence in Medicine, 101,* 101735.
25. Wei, G. (2017). Some cosine similarity measures for picture fuzzy sets and their applications to strategic decision making. *Informatica, 28,* 547–564.
26. Hur, J., Koo, D., Shin, Y., & Kang, K. (2016). Secure data deduplication with dynamic ownership management in cloud storage. *IEEE Transactions on Knowledge and Data Engineering, 28,* 3113–3125.
27. Jiang, T., Chen, X., Wu, Q., Ma, J., Susilo, W., & Lou, W. (2017). Secure and Efficient Cloud Data Deduplication With Randomized Tag. *IEEE Transactions on Information Forensics and Security, 12,* 532–543.
28. Liu, J., Wang, J., Tao, X., & Shen, J. (2017). Secure similarity-based cloud data deduplication in Ubiquitous city. *Pervasive and Mobile Computing*.
29. Bhagwat, D., Eshghi, K., Long, D. D., & Lillibridge, M. (2009). Extreme binning: Scalable, parallel deduplication for chunk-based file backup. In *Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on*, (pp. 1–9).

B. Rasina Begum Associate Professor in Department of Computer Science and Engineering at Mohamed Sathak Engineering College, Keelakarai, Tamil Nadu 623806. India. My correspondence mail address: rasinabegumphd@yahoo.com

P. Chitra Professor in Department of Computer Science and Engineering at Thiyagarajar College of Engineering,Madurai, Tamil Nadu 625 015. India