Cluster Computing (2020) 23:3057–3067
https://doi.org/10.1007/s10586-020-03069-3

Elliptic curve Diffie–Hellman cryptosystem in big data cloud security

E. K. Subramanian[1] · **Latha Tamilselvan**[1]

Received: 30 October 2019 / Revised: 23 January 2020 / Accepted: 1 February 2020 / Published online: 8 February 2020

Abstract

Big data require cloud that provides dynamically expanding data storage accessed through the Internet. The outsourcing data in the cloud for storing makes the user data management easier and reduces the cost of maintaining data. Still organizations are not confident to store their data in the cloud, because of security and privacy concerns. However, existing encryption methods are able to protect data confidentiality, but it has some drawbacks of access patterns can also leak sensitive information. The proposed system uses an Elliptic curve with Diffie–Hellman (ECDH) algorithm for encryption and decryption of data to improve the data security in the cloud. This algorithm reduced the computational complexity and encrypted data efficiently. In experimental analysis, the performance of proposed ECDH is calculated using evaluation parameters such as encryption time, decryption time, computation overhead and key generation time. The proposed ECDH algorithm has approximately 70% better performance in terms of encryption time than existing methods such as RSA, MRSA and MRSAC.

Keywords Big data · Cloud computing · Cloud storage · Cryptography · Elliptic curve Diffie–Hellman

1 Introduction

Cloud is a distributed computing model, that can host as well provide the customers with various internet services. The popularity of cloud computing has increased recently, because of the many advantages provided by the cloud [1, 2]. Cloud computing is widely used in the commercial field for data storage and online applications. The main advantage of online services is that users can access data from multiple locations any time. The cloud storage services relieve the online service providers from storage complexity and high maintenance cost [3–5]. However, the complete trust on the cloud service provider is not possible for the reason that the modification of data may happen at non-trusted cloud servers. Therefore, the data security is the first concerns in cloud storage services [6, 7]. To protect the sensitive data, users need to perform encryption of data before sending to the storage cloud and then enforce access control mechanism by cryptographic methods [8]. Many techniques are available to keep the data in a secure manner and mainly cryptographic algorithms are very helpful [9].

Recently, there are many cloud computing encryption techniques under research in industrial and academic field. Moreover, securing files in the cloud and protecting private information is the significant task. The privacy preservation is the process of protecting the sensitive data in the cloud [10]. To perform privacy, preserve task, several security approaches such as key generation, encryption, decryption etc. are required. Therefore, various privacy preservation methods were employed in the existing research works to protect the sensitive data [11]. The big data storage consists of several challenging functions such as multi-user access, maintenance issue, cost efficiency, and optimized storage. In addition, it does not allow any parallel computing techniques for integrating the big data with cloud computing [12, 13]. In cloud, big data security is very significant, because it suffers from several problems like huge size of sensitive or confidential data from different domains may be hacked by malicious intruders [14]. Sometimes the malicious users monitor the different organization's data to

✉ E. K. Subramanian
eksdeal@gmail.com

Latha Tamilselvan
latha.tamilselvan94@gmail.com

[1] Department of Information Technology, B S Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamilnadu, India

steal the data. Many researchers are working on developing the architecture for big data security and frameworks to secure large volume of data in the cloud [15]. ECDH based privacy-preserved query retrieval system is proposed to overcome these limitations. ECDH algorithm is used in encryption and decryption function to improve the cloud security. This algorithm improves the cloud storage security and enables faster access to data from the cloud.

The organization of the paper is given here. The latest research works on how to secure big data specifically in the cloudlike environment are described in Sect. 2. The proposed ECDH method based encryption and decryption of cloud storage data are explained in Sect. 3. The experimental analysis of our proposal and existing methods are explained in Sect. 4. Finally, Sect. 5 explains the conclusion as well as future enhancement of the present research work.

2 Literature review

Researchers have suggested a number of techniques for security of big data in the cloud like background. Brief discussion on some of the significant contributions from the available literatures are given below.

Yang et al. [16] presented IoT and Cloud based cloud service systems. The proposed method provided a good platform to reduce the complexity between cloud service provider and users. An advanced encryption method protects the cloud data and avoided the leakage of personal information. The proposed IoT and cloud-based protocol minimized the computational cost incurred through the application of bilinear pairings. This proposed protocol only considered the users belongs to the same groups and the number of users must be more than two.

Stergiou and Psannis [17] presented the survey of both cloud and Big Data technology and on the security and privacy challenges. The method combined the functionality of the two technologies with an aim to examine the benefits in security while integration. The big data cloud storage system uses new algorithm, namely Advanced Encryption Standard (AES) that provided more security in CC's and provided more privacy of data. The proposed AES method bit lags in data security and privacy due to no user verification.

Thangavel and Varalakshmi [18] presented an improved ElGamal based cryptosystem, which is asymmetrically employed to overcome key handling problems on cloud. Also, provided high security for transferring key file between owner and user. The proposed technique provided better authentication and performance against attacks. An advanced method supports both asymmetric and symmetric cryptosystem and improves the security, cloud storage performance as well as data retrieve the cloud data securely. However, the time taken for encryption and decryption increase as the character count increases.

He et al. [19] proposed a Privacy-Preserving Certificate Less Provable Data Possession (PP-CLPDP) scheme for certificate management and to ensure privacy protection. The PP-CLPDP scheme uses the public cloud, providing data integrity for big data. Certificate Management and key escrow problem has been solved in this method. The data integrity of the big data has been preserved in this method. The RSA method has been used to secure the data in the cloud storage and proof challenge is verified. The proposed PP-CLPDP scheme's computational cost was similar to the existing CLPDP model, so little bit improvement is required.

Song et al. [20] presented the retrieval of full text in the cloud storage system with high privacy. The proposed method used bloom filter, to scatter the storage issues in tree index format. The bloom filter based tree index system shows the similarity among the encrypted documents and query document through membership entropies of index words. The ranking algorithm was used to find the most queries related words in membership index. Also, maximizing the storage space for new document increases storage cost and it is relatively high.

Gnanaprakasam and Rajivkannan [21] proposed the double encryption technique of RSA and Optimal Elliptic Curve Cryptography (OECC) algorithm to encrypt the user document. The optimal key is selected for ECC based on the cuckoo search algorithm. Once the document is encrypted in the cloud and this is stored with the access structure that shows which types of user are allowed to access the document. The information is split and stored in the two separate servers to reduce the computation complexity. The greedy selection is applied to select an optimal key from the two servers. The cuckoo search method has not escaped from the local optimum solution and slow rate of convergence; this affects the computation time of the developed method. The proposed ECDH method involves in the combination of Elliptic Curve and Diffie–Hellman to reduce the computation complexity. Since the tags has been created in secure retrieval index table generation, the challenge time is reduced.

Tewari and Gupta [22] developed an authentication protocol using bitwise operations to reduce the communication cost and storage. As the method is ultra-lightweight, the computation overhead of the method is low. The developed method is analyzed and this shows that the method is untraceable in the function. The strength of the developed method is high and process the IoT data for the authentication. The tag, reader and database server are the three most important entities used in the developed method. The developed method doesn't able to withstand the Denial of Service (DoS) attacks.

John and Thomas [23] analysis the various adversary attacks that were used against the malware detection classifiers. The study shows that the existing defensive mechanism like simple retraining does not act as a defense in the malware detection. The adversarial retraining method has the disadvantage is that adversarial samples crafted by methods such as Jacobian based or FGSM are outside the training distribution in case of malware.

Olakanmi and Dada [24] developed a semi-honest model that allows the client to perform privacy preserving validation in the cloud without re-computing the computation. The morphism method was used by the client to effectively perform the proof of correctness in the cloud data. The method provides the anonymity for the client to check the integrity of the data in the cloud. Modified Paillier encryption is used by the developed method to increase the security of the data. The developed method has lower computation overhead in the performing the correctness of data in the cloud. The developed method is semi-supervised technique and doesn't withstand the attacks.

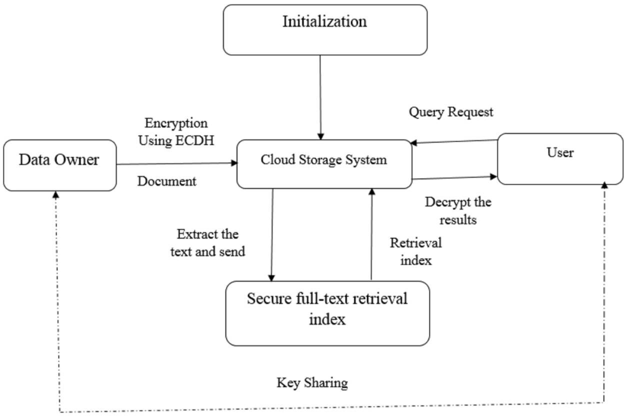


Fig. 1 Proposed architecture using big data security in cloud

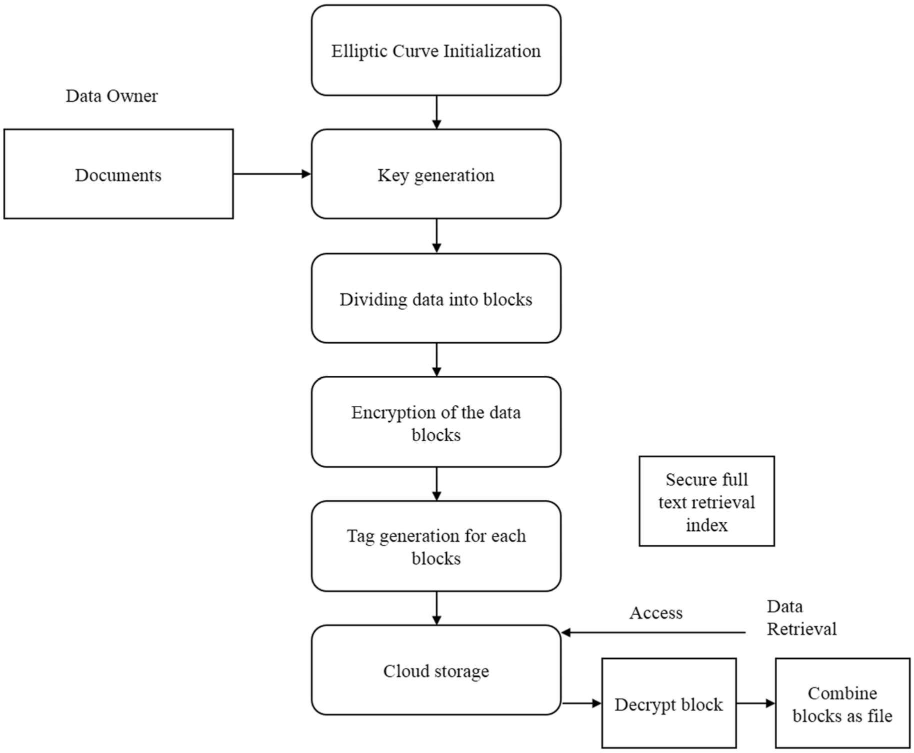
Azad and Navimipour [25] developed the combination of cultural and ant colony optimization to provide the optimized parameter for the make span and energy consumption optimization in the task scheduling problem. The experimental result shows that the proposed method has higher performance compared to the other existing method. The communication overhead of the method needs to be minimized.

2.1 Problem definition

The existing method in the data encryption technique has the limitations of process in the cloud. The problems in the existing data encryption techniques were discussed below.

1. The complexity of the existing data encryption technique is high and required to be reduced. The

Fig. 2 The proposed ECDH algorithm block diagram


Elliptic Curve Initialization
Data Owner
Documents
Key generation
Dividing data into blocks
Encryption of the data blocks
Secure full text retrieval index
Tag generation for each blocks
Access
Data Retrieval
Cloud storage
Decrypt block
Combine blocks as file

complexity of the method affects the execution time of the method.

   a. Solution: ecliptic curve is the simple encryption, which is applied for the data encryption. The complexity of the proposed ECDH method is low and the execution time of the method is reduced.

2. Existing encryption techniques doesn't secure the pattern access information and this involves in affects the privacy of the cloud.

   a. Solution: the pattern access information also encrypted in the proposed ECDH method. This increases the security against the attacks and protect the data in the cloud.

3 Proposed methodology

The 'big data' is referred as the large collection of complex distributed data created from all the digital sources available today. The research on big data facilitates the growth of scientific discovery and innovation. This paper proposed ECDH algorithm that provides high security for cloud storage environment. The ECDH uses symmetric encryption method for efficient data encryption. An elliptic curve function based DH algorithm is used in cloud data security.

Figure 1 represents the cloud based big data security model using ECDH algorithm. The proposed architecture consists of majorly four steps such as data owner, encryption process, decryption process, and text retrieval process. The major responsibility of proposed big data based security model is key generation and share the keys securely. The major benefit of using ECDH algorithm in key generation function that use the smaller keys for encryption. Also, the secured key procedure is employed for shared secret key generation. The proposed ECDH algorithm pseudo code is described in the following sections. The detailed description of the proposed block diagram is shown in Fig. 2.

The proposed ECDH algorithm initially creates the client–server private key and public key pairs. After initialization, securely share the keys between client and server using ECDH. The data encryption and decryption process employed for storing and retrieval of the user data. An input data is partitioned into several blocks. For each block, creates the secrete keys as S_K and placed in the cloud. While accessing the data, encrypted data blocks are decrypted and then merge all the block into a single document. The challenge and proof generation process is to verify the blocks and provide the permission to authorized user for data modification.

```
Key Generation: (input, output : $S_k$)
    Generate the private key and public key pairs for user and server
    User key $\leftarrow d_A, Q_A$
    Server Key $\leftarrow d_S, Q_S$
    Share the public key between user and server
    Compute the elliptic curve point $S_K = d_A Q_S = d_S Q_A$
Encryption: (Input: Document, Output: Encrypted blocks)
    Choose the document to be uploaded
    Divide Document into data blocks
    Encrypt the each block by secret key $S_k$
    Generate the tag index for each block
    Store the data in cloud storage
Decryption: (Input: Encrypted document, Output: Decrypted Document)
    Select the document to be retrieved from cloud storage
    Retrieve all blocks of that file from the multiple cloud storage
    Decrypt each blocks with secret key $S_k$
    Combine all blocks and download as single file
    Save the data
Modify Data: (Input: -, Output :-)
    Select the user account to attack
    Select which file to modify
    Select the block of data to be modified
    Replace the encrypted data with new data
    Save the changes
Challenge and Proof Verification: (Input: Document, Data block number, Output: Integrity check, Recover original data)
    Choose the file to verify
    Select the block and enter the metadata length
    Challenge the block for modification
    Proof generation
    function: proof verification
        if (cipher text symbol missing)
            return modified
        else
            not modified
        if modify
            recover the original data from backup cloud storage
    end function
End
```

The pseudo-code and proposed architecture component description is explained in below sections.

3.1 Data owner

The enterprise or an individual that has a large volume of private data is referred as data owner. The server configuration of data owners includes the restricted storage space due to a maximum number of computations and data storing facilities are available in the cloud. The communication module provides a single combined database management by connecting all cloud databases. The user query submitted is categorized into read only and read–write queries by SQL analyzer. The SQL distributor performs the query process by selecting the most appropriate load balancing technique. The entire database is synchronized by database management system while modifying the query request for efficient resource utilization.

3.2 Dataset description

The Enron corpus database is used in this research work. The database includes four tables; the entities are messages, employees, reference information and recipients. The dataset of Enron Email consists of 200,399 messages belonging to 158 users. The several emails are randomly selected to build an experimental dataset from the Enron Email.

3.3 Initialization (set up)

Setup (1^k): An implicit security parameter k; MPK is the output public parameter; MSK is the master key. A large prime number p is selected by CA, a bilinear group $(G, G1)$ with order p, a generator $g \in G, h \in G, y \in_{\Re} Z_p$ and $t_{ij} \in Z_p(i \in [1,n], j \in [1,n_i])$. The CA calculates $Y = e(g,h)^y$ and $T_{i,j} = g^{t_{i,j}}(i \in [1,n], j \in [1,n_i])$. The public key and master keys are initialized in Eq. (1),

$$\begin{cases} MPK = \left(e, g, h, Y, T_{i,j}(i \in [1,n], j \in [1,n_i])\right) \\ MSK = \left(y, t_{i,j}(i \in [1,n], j \in [1,n_i])\right) \end{cases} \quad (1)$$

where Z_p is the group of large prime order p. Assume that t and t' is the two different universal hash function in random oracle which maps $\{0,1\}^* \times \{0,1\}^* \rightarrow Z_p$ such that $t_{i,j}$ / $= t'_{i,j}$ is known to CA. The cloud server creates index nodes by inter linking them and provides the storage services.

3.4 Key sharing

The key sharing process involves the key generation process (KeyGen) performed by using a key generation algorithm. Then key sharing algorithm is run through the Central Authority (CA) and takes input from the CA.

Consider MSK and attribute list L of user u, CA generates $r \in_{\Re} Z_p$ and calculate the SK for user "u" as in Eq. (2).

$$SK_L = \left\{h^{y+r}, \forall v_{i,j} \in LD_{i,j} = \left(T_{i,j}\right)^r, g^r, L\right\} \quad (2)$$

where $v_{i,j}$ indicates a set of all possible attributes and $D_{i,j}$ represents the generated key. The secret key is shared for both owner and the user of data. The key generated is used to encrypt and decrypt data [26].

3.5 Data encryption using elliptical curve Diffie–Hellman

The ECDH algorithm is proposed encryption and decryption process for message. The ECDH algorithm is fully homomorphic method of encryption within a secure channel. It contains the pair of keys. One is public key, which encrypts data. The other one is private key for decrypting the data. The public key is used to derive message sharing directly and the private key used for decryption. Further successive data transaction uses the keys derived and ensured among the agreed parties in the channel.

The receiver's public key G^d is learned by the sender, where d is the private key of the receiver itself. The sender then generates a new ephemeral value y and associated value G^y. The sender then calculates the symmetric key k with the help of Key Generation function. The Key generation function (KGF) is described in Eq. (3),

$$k = KGF\left(G^{dy}\right) \quad (3)$$

The ECDH algorithm is executed in step-by-step procedure for data transaction between the sender (S) and receiver (R). At first, elliptic curve parameters of all kinds are generated. In the next step, every party should select the pair of keys, a private key (d) and a public key (Q). It's derived in Eq. (4),

$$Q = dG \quad (4)$$

whereas the curve generator is indicated as G.Consider, the (d_A, Q_A) represents sender key and $(d_B, Q_{\mathbf{B}})$ indicates receiver key. The public key is indicated as Q and it's shared with others during communication. The input messages in ECDH system are denoted as points (i.e. elliptic curve (x, y)). The receiver estimates the points like (x, y) and query is decrypted through the product $Q_B d_A$ or $Q_A d_B$. Equation 5 represents the symmetric features of the ECDH encrypted process.

$$Q_B = d_A d_B G = d_B d_A G = d_B G \quad (5)$$

Data owner store the files in the cloud storage system and files are encrypted immediately for security. The ECDH encryption method consists of several parameters such as, M is the message, MPK is the public parameter, A is the access structure of attributes, and DK is the data encryption key. The ECDH algorithm combined with the key generation parameter decreases the cost of communication and overheads. Equation (6) shows the encryption process,

$$CT = (A, E = Enc_{DK}(M)) \tag{6}$$

where the variable CT represents Cipher Text, which is the encrypted data. It verifies that to decrypt the query, the encrypted data must have valid attributes set and satisfy the access policy. The method assumes that the access structure contained in CT implicitly. An Elliptic Curve (EC) algorithm uses the session key negotiation function in both ends of a communication with less amount of data exchange for high security basis. The ECDH algorithm is more secure than the EC algorithm. ECDH employs smaller key length, minimum resource utilization and high computation speed. The ECDH algorithm helps to generate the key of the cloud data.

3.6 Secure full text retrieval index

The retrieval process of encrypted cloud storage data is described in this section. Initially the storage related services are generated by the server in the cloud. For retrieval process, the global system parameters are (H, m, k, p) initialize the cloud storage system. The hash function is denoted as H and $H_1, H_2, \ldots, H_h, H_i : \{0,1\}^* \to [1, m] (1 \le i \le h)$ is the hash arbitrary strings, which is an integer in the range 1 *to m*. The owner of data encrypts the files with the help of symmetric encryption before outsourcing, to maintain the data security. The owner has a key pair $(key_{doc},\ key_{index})$ stored locally. Consider that the data owner can allocate the $keys(key_{doc},\ key_{index})$ to the users who are authorized via safe channels. All words from the document d is extracted and d is encrypted with key_{doc} using ECDH algorithm by the data owner, before outsourcing a document d. Equation (7) represents the decryption process of user retrieved document.

$$M = Dec_{DK}(E) \tag{7}$$

where E is the encrypted key. Users receive the decrypted message. The user sends a query request to the cloud storage for decrypting the documents sent by the data owner. The decryption algorithm executions depend on the public parameter MPK. The CT includes the access structure A and the attribute set S depends on secrete key SK. Decrypt the CT if S satisfies the access tree and provide the M (message) otherwise "Ø". The decrypted result again sent to the user from the cloud storage system. When the huge volume of data variety (structured or unstructured form) stored in a cloud storage, is termed as Big Data Cloud (BDC). In BDC, huge volume of data is shared between the cloud users. So, ECDH algorithm is used to improve the cloud security. This algorithm provides maximum data security in the cloud and achieves encryption, decryption and key generation time. An experimental analysis of proposed ECDH and existing method's performance is described in the following sections.

4 Experimental result and discussion

The simulation experiment was performed using CloudSim 3.0 PlanetLab on a PC with 3.2 GHz i5 processor. The experimental data were taken from Enron Email Dataset which consists of a total 200,399 of messages belonging to 158 users. Multiple different number of emails were randomly selected from the Enron Email to form an experimental dataset. Every set of input keywords randomly were generated through the user. After that, the cloud server performs data search from the database and extracted the relevant qualified files. To find the efficiency of the algorithm proposed, several metrics such as, key generation time, encryption time, decryption time, and computation overhead were used in this research work.

- Computation Overhead: The overhead of computation in the auditing phase divides into the generation of challenge, generation of proof and verification of proof. The computation overhead is due to generation of private keys. The calculation of computation overhead is shown in Eq. 8,

$$computation\ overhead = n(2\exp_{G1} + Mul_{G1} + Hash_{G1}) \tag{8}$$

  where n represents the blocks in the common files. Mul_{G1} represents one multiplication operation in $G1$; $\exp_{G1}$ refers one exponentiation operation in $G1$; $Hash_{G1}$ whereas q represents one multiplication operation in Z_q^*; $Hash_{G1}$ refers to one hash operation in $G1$.
- Challenge generation: when user monitor any modification happened in the stored blocks of data or files without user authentication it will send a challenge. For a received challenge, the storage will generate a proof messages and forward it to user.
- Proof generation: In proof generation, F represents input files, an auditing challenge, a set of corresponding authenticators, and generates a proof P. The Proof P is used to prove that whether the cloud accurately stores the files or not.

- Proof verification: In this step, the inputs are the data proof *P* and public parameters of the system. For "valid proof" it returns "success"; or "failure", otherwise.

Table 1 represents the key generation time of the ECDH and existing cryptographic methods. The proposed ECDH algorithm takes minimum time for generating keys compare to the other existing methods. The existing RSA, MRSA, and MRSAC algorithm take maximum time for key generation. The proposed ECDH method provides a small encrypted key that requires small computation power and time for processing the message. Therefore, the proposed method has a lower computation time of 781 ms while the existing method has 8925 ms for 2048-bit key length.

Figure 3 depicts the encryption time of proposed and existing methods. The *x*-axis represents the key length in (bit) and *y*-axis represents the key generation time in (ms). If number of key length increase, then the key generation time also increases. Compared to the existing methods proposed ECDH algorithm shows minimum key generation time with respect to different key length. The proposed ECDH method has a lower computation time for various numbers of key length due to key generation for the encrypted message is low. The proposed ECDH method The proposed method ECDH algorithm takes minimum time in key generation due to its reduce the computational overhead. Therefore, encryption and decryption keys are generated in minimum time.

Tables 2 and 3 represents the encryption and decryption time of input and output data respectively. An existing and proposed method's performance is calculated in terms of different key lengths such as 100, 128, 256, 512, 1024, 2048, and 4096 in bits. Compared to the existing methods proposed ECDH algorithm shows better results. The proposed ECDH algorithm takes minimum encryption and decryption time. The proposed ECDH is a key exchange protocol, which exchange the key between client and server quickly. The existing method depends on the RSA requires a number key for the encryption and in-turn requires more time for computation. The proposed ECDH method generates less number of keys for the encryption and the computation time is reduced. The developed ECDH method has the decryption time of 187 ms for the 4096 key length, while the existing method has 10,957 ms. The computation time is much reduced due to the usage of less number of key generation. Moreover, the ECDH reduces the information delay between the client and server. Also, faster than existing methods such as RSA, MRSA, and MRSAC.

Figures 4 and 5 represents the performance of encryption time and decryption time respectively. The *X*, *Y* axis indicates the number of key lengths in bits and times in ms respectively. Compared to the existing methods, the proposed ECDH method takes minimum encryption and decryption time. Since the ECDH algorithm uses the key generation function, it reduces the information loss and securely exchange the key. By comparing the proposed and existing methods in the encryption and decryption time, shows that the developed method has a lower computation time in the encryption and decryption process compared to the other existing method. The computation is significantly reduced by reducing the number of key generation for the encryption and decryption process. The developed method generates a lower number of key generation than other existing method. Hence the developed method has lower computation than other existing methods.

Table 4 depicts the performance of computation overheads with respect to various numbers of blocks. Compared to the other two phases, the proof verification procedure takes longer time and challenge generation procedure takes the shortest time than other two procedures [27, 28]. Compare to the existing technique, the ECDH algorithm takes minimum challenge generation and proof generation time. For challenge generation, homomorphic tags are generated in the challenge process. Since tags has already been created in the secure retrieval index table generation process it fetches only the tag from the table during the challenge process. So it will fetch it in lesser time, which

Table 1 Key generation time of proposed and existing methods

Key length (in bit)	Proposed (ms)	RSA [26]	MRSA [26]	MRSAC [26]
100	78	72	110	158
128	79	92	144	244
256	94	469	484	584
512	110	140	172	192
1024	391	469	625	1625
2048	781	2453	8125	8925
4096	7637	91,542	93,899	123,899

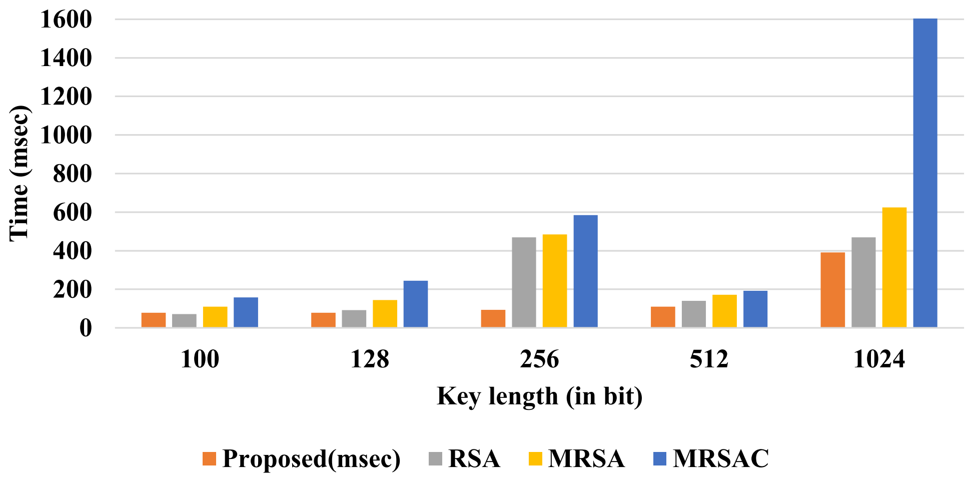


Fig. 3 Key generation time

Table 2 Encryption time performance of proposed and existing methods

Key length (in bit)	Proposed ECDH (ms)	MRSA	MRSAC
100	10	222	188
128	12	205	305
256	15	329	409
512	19	1672	2762
1024	30	11,625	13,625
2048	50	99,891	10,880
4096	92	110,907	21,887

cannot be measured in milliseconds results in challenge generation time to be zero for all blocks.

Figure 6 represents the computation overhead performance of proposed and existing methods. The plot contains a number of challenged blocks in *X*-axis and the computation overhead in *Y*-axis. This graphical representation describes the proof generation, proof verification and challenge generation of proposed ECDH method. The proposed ECDH algorithm achieved better results in terms of encryption time, description time and key generation.

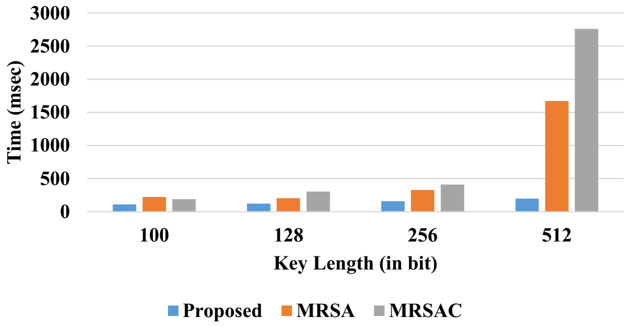


Fig. 4 Encryption time

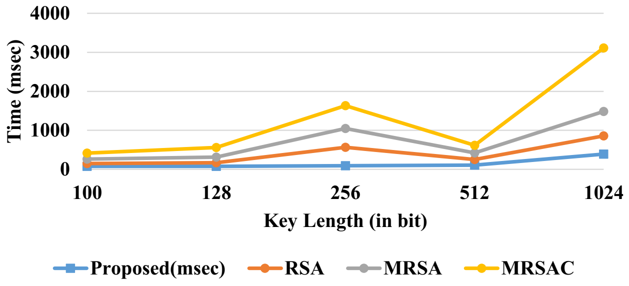


Fig. 5 Decryption time

Table 3 Decryption time of proposed and existing methods

Key length (in bit)	Proposed ECDH (ms)	RSA	MRSA	MRSAC
100	16	88	107	212
128	31	188	122	188
256	47	62	156	203
512	63	218	968	688
1024	78	1453	6938	7038
2048	109	15,203	53,609	83,709
4096	187	18,381	10,957	10,957

Table 4 Computation overhead

Number of blocks	Proof verification	Proof generation	Challenge generation
10	0.94	0.063	0
20	0.78	0.047	0
30	0.78	0.047	0
40	0.94	0.063	0
50	0.94	0.062	0
60	0.78	0.046	0
70	0.78	0.047	0
80	0.78	0.047	0
90	0.78	0.047	0

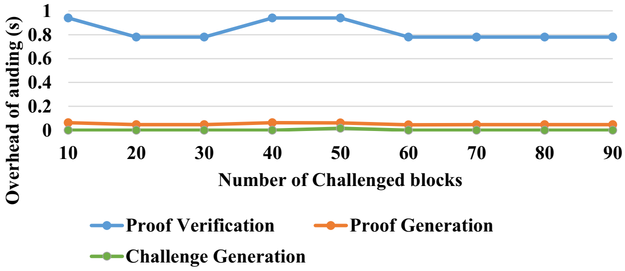


Fig. 6 Computation overhead

5 Conclusion

The cloud computing paradigm has become popular recently, because, it has the ability to store huge data and flexible computation. To reap the benefit of these advantages, many data owners outsource their data and data analysis operations (e.g., data queries, data insertion, modifying and so on) in the cloud. For security concerns, a data owner may like to encrypt data before outsourcing. In this paper, ECDH algorithm is used for cloud data security. The user data encrypted using algorithm is stored in the cloud storage. The data stored is retrieved using decryption function based on the user query. The proposed ECDH method is capable of processing the data with larger key size faster than existing algorithms. An experimental evaluation of ECDH algorithm performance is measured using different evaluation metrics such as encryption time, decryption time, key generation time and computation overhead. The execution time of ECDH algorithm is around 70% better performance than the existing cryptographic methods. In future, a secure relevant data retrieval mechanism can be incorporated into the cloud storage.

References

1. Salem, M.Z., Sabbeh, S.F., Tarek, E.L.: An efficient privacy preserving public auditing mechanism for secure cloud storage. Int. J. Appl. Eng. Res. **12**(6), 1093–1101 (2017)
2. Bhatt Agarwal, R.: A technological review on scheduling algorithm to improve performance of cloud computing environment. Int. J. Innov. Technol. Explor. Eng. (IJITEE) **8**(6), 166–172 (2019)
3. Zhou, L., Varadharajan, V., Hitchens, M.: Trust enhanced cryptographic role-based access control for secure cloud data storage. IEEE Trans. Inf. Forensics Secur. **10**(11), 2381–2395 (2015)
4. Wang, Q., Wang, C., Ren, K., Lou, W., Li, J.: Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst. **22**(5), 847–859 (2011)
5. Panwar, N., Negi, S., Rauthan, M.S., Aggarwal, M.A.Y.A.N.K., Jain, P.: An enhanced scheduling approach with cloudlet migrations for resource intensive applications. J. Eng. Sci. Technol. **13**(8), 2299–2317 (2018)
6. Aggrawal, M., Kumar, N., Kumar, R.: Optimized cost model with optimal disk usage for cloud. In: Aggarwal, V.B., Bhatnagar, V., Mishra, D.K. (eds.) Big Data Analytics, pp. 481–485. Springer, Singapore (2018)
7. Song, W., Cui, Y., Peng, Z.: A full-text retrieval algorithm for encrypted data in cloud storage applications. In: Li, J., Ji, H., Zhao, D., Feng, Y. (eds.) Natural Language Processing and Chinese Computing, pp. 229–241. Springer, Cham (2015)
8. Cheng, Y., Wang, Z.Y., Ma, J., Wu, J.J., Mei, S.Z., Ren, J.C.: Efficient revocation in ciphertext-policy attribute-based encryption based cryptographic cloud storage. J. Zhejiang Univ. Sci. C **14**(2), 85–97 (2013)
9. Srisakthi, S. Shanthi, A.P.: Design of a secure encryption model (SEM) for cloud data storage using hadamard transforms. Wirel. Pers. Commun. **100**, 1727–1741 (2018)
10. Wang, W., Chen, L., Zhang, Q.: Outsourcing high-dimensional healthcare data to cloud with personalized privacy preservation. Comput. Netw. **88**, 136–148 (2015)
11. He, X.M., Wang, X.S., Li, D., Hao, Y.N.: Semi-homogenous generalization: improving homogenous generalization for privacy preservation in cloud computing. J. Comput. Sci. Technol **31**(6), 1124–1135 (2016)
12. Liu, H., Ning, H., Xiong, Q., Yang, L.T.: Shared authority based privacy-preserving authentication protocol in cloud computing. IEEE Trans. Parallel Distrib. Syst. **26**(1), 241–251 (2015)
13. Kanna, G.P., Vasudevan, V.: A fully homomorphic–elliptic curve cryptography based encryption algorithm for ensuring the privacy preservation of the cloud data. Clust. Comput. **22**, 9561–9569 (2019)
14. Wang, Z., Cao, C., Yang, N., Chang, V.: ABE with improved auxiliary input for big data security. J. Comput. Syst. Sci. **89**, 41–50 (2017)
15. Yang, K., Han, Q., Li, H., Zheng, K., Su, Z., Shen, X.: An efficient and fine-grained big data access control scheme with privacy-preserving policy. IEEE Internet Things J. **4**(2), 563–571 (2017)

16. Yang, C.Y., Huang, C.T., Wang, Y.P., Chen, Y.W., Wang, S.J.: File changes with security proof stored in cloud service systems. Pers. Ubiquit. Comput. **22**(1), 45–53 (2018)
17. Stergiou, C., Psannis, K.E.: Efficient and secure BIG data delivery in cloud computing. Multimed. Tools Appl. **76**(21), 22803–22822 (2017)
18. Thangavel, M., Varalakshmi, P.: Enhanced DNA and ElGamal cryptosystem for secure data storage and retrieval in cloud. Clust. Comput. **21**, 1411–1437 (2018)
19. He, D., Kumar, N., Wang, H., Wang, L., Choo, K.K.R.: Privacy-preserving certificate less provable data possession scheme for big data storage on cloud. Appl. Math. Comput. **314**, 31–43 (2017)
20. Song, W., Wang, B., Wang, Q., Peng, Z., Lou, W., Cui, Y.: A privacy-preserved full-text retrieval algorithm over encrypted data for cloud storage applications. J. Parallel Distrib. Comput. **99**, 14–27 (2017)
21. Gnanaprakasam, T., Rajivkannan, A.: Optimal Ecc based dual encryption technique for data security in cloud. Int. J. Adv. Eng. Technol. **VII**(II) (2016)
22. Tewari, A., Gupta, B.B.: Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. J. Supercomput. **73**(3), 1085–1102 (2017)
23. Gupta, B.B., Agrawal, D.P.: Handbook of research on cloud computing and big data applications in IoT. In: IGI Global. (2019). https://doi.org/10.4018/978-1-5225-8407-0
24. Olakanmi, O.O., Dada, A.: An efficient privacy-preserving approach for secure verifiable outsourced computing on untrusted platforms. Int. J. Cloud Appl. Comput. (IJCAC) **9**(2), 79–98 (2019)
25. Azad, P., Navimipour, N.J.: An energy-aware task scheduling in the cloud computing using a hybrid cultural and ant colony optimization algorithm. Int. J. Cloud Appl. Comput. (IJCAC) **7**(4), 20–40 (2017)
26. Anbuchelian, S., Sowmya, C.M., Ramesh, C.: Efficient and secure auditing scheme for privacy preserving data storage in cloud. Clust. Comput. (2017). https://doi.org/10.1007/s10586-017-1486-z
27. Zhang, Y., Yu, J., Hao, R., Wang, C., Ren, K.: Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. IEEE Trans. Depend. Secure Comput. (2018). https://doi.org/10.1109/TDSC.2018.2829880
28. Yu, J., Ren, K., Wang, C.: Enabling cloud storage auditing with verifiable outsourcing of key updates. IEEE Trans. Inf. Forensics Secur. **11**(6), 1362–1375 (2016)

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

E. K. Subramanian, B.Tech., M.E. is Ph.D. research scholar in the department of Information Technology of BS Abdur Rahman Crescent Institute of Science and Technology in Chennai, Tamilnadu, India. He is currently pursuing his Ph.D. in big data security.

Latha Tamilselvan, M.E., Ph.D. is working as professor in the department of Information Technology of BS Abdur Rahman Crescent Institute of Science and Technology in Chennai, Tamilnadu, India. Her research area of interest are cloud computing, network security and wireless networks. She is guiding many Ph.D. scholars on cloud computing, big data and computer networks.