International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 4, August 2023, pp. 4734~4742
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i4.pp4734-4742

Cloud security: literature survey

Amruta Gadad, Devi Anbusezhiyan
School of Computer Science and Applications, REVA University, Bangalore, India

Article Info

Article history:

Received Jul 6, 2022
Revised Sep 18, 2022
Accepted Oct 1, 2022

Keywords:

Cloud computing
Cloud data security
Decryption
Encryption
Rivest, Shamir and Adleman

ABSTRACT

Today, the growth of digitalization has made the ease for livelihood for all the organizations. Cloud computing the storage provider for all the computer resources has made it easy for accessing the data from anywhere anytime. But at the same time the security for cloud data storage is the major drawback which is provided by various cryptographic algorithms. These algorithms convert the data into unreadable format, known as cipher text, Rivest, Shamir and Adleman (RSA) one of the most popularly used asymmetric algorithm. This paper gives detailed review about such different cryptographic algorithms used for the cloud data security. The comparison study is also made for the size of data and to analyze the encryption time and decryption time, which concludes that to enhance the cloud data security some addon techniques are to be used along with these cryptographic algorithms. To increase the security level and to increase the transmission speed of plaintext, integrated method will be proposed by encoding the plaintext to intermediate plaintext and then intermediate plaintext will be compressed using any one of the compression techniques to increase the compression ratio, lastly the compressed file is encrypted to further enhance the security level.

Corresponding Author:

Amruta Gadad
School of Computer Science and Applications, REVA University
Rukmini Knowledge Park, Bangalore, Karnataka, India
Email: amgabsc@gmail.com

1. INTRODUCTION

The growth of technology increased the use of digital world which in turn increased the use of storage devices from the small compact disc (CD), digital video disc (DVD), universal serial bus (USB) drives, and hard disk to the latest cloud storage. Today each and every data we store in cloud with the aim of accessing it from all the places we are in. Hence, we can say the Cloud storage as the father of all the storage devices. But as the applications of the cloud storage has increased, the types of threats like insider threats, viruses, worms, botnets, phishing attacks, denial of service attacks and many more also increased.

Cloud data security has become of prime importance as the loss of data may lead to major losses. To secure the cloud data we need to use different cryptographic algorithms where the data is converted into non readable format known as cipher text. This helps to secure the data from being read by unknown individual. To enhance this data security various cryptographic algorithms have been applied by the researcher’s protecting the cloud data. Cloud computing provides a web-based platform to variety of applications running on more than a large number of personal computers (PCs) and servers to simultaneously work together giving additional low-spending administrations, refreshes programming consequently, raises joint effort among clients and administration sellers [1]. Cloud also provides many advantages such as more fault tolerance and multi-factor authentication to secure the information in the cloud trying to maintain confidentiality and integrity of data [2].

Classification of cryptographic algorithms is based on the number of keys used and the methods used for encryption and decryption of data [2]. The number of keys used means, it either uses a single key for encryption and as well as for decryption basically known as symmetric cryptography or it uses two separate keys, open key for encryption and private key for decryption or vice versa depending on the algorithm used. There are different key generation algorithms defined by various key management servers where the private and the public keys are generated by key distribution centers.

The work of many researchers is concentrated on either giving robust security or improve the speed of data transfer. Security is many times achieved by using the Rivest, Shamir and Adleman (RSA) algorithm which gives resistance to attacks, advance encryption standard (AES) and data encryption standards (DES) have also proved the level of security due to its number of rounds, the data is transmitted through, to achieve security against attacks. To increase the speed of data transfer rate some of the compression algorithms are also used like Huffman Coding, Run Length Coding and many more where the size of the data is reduced and increases the transfer rate of data from client to server or cloud. Most of the researchers used a combination of these two techniques cryptographic and compression but the level of data security can also be enhanced by including any of the encoding techniques such as Fibonacci, Lucas encoding, prime numbers and many more. The main motivation to write this article is that there is no proper survey paper that shows the proper study about securing the text data in the cloud.

This paper reviews about the different methodologies proposed for cloud data security. In the first section we have discussed each of the methodology proposed by different researchers to enhance the data security of cloud which is further followed by the discussions of all the methodologies in which the results and their achievements were discussed. In the last section the comparison study is made for all the considered proposed methodologies. Finally, we summarize the paper with the conclusion and our future work.

The major contribution in this survey paper is on gathering knowledge about the different methodologies used by the researchers to secure the cloud data wherein they are trying to achieve the confidentiality, data integrity, authentication, non-repudiation, replay attacks and also to increase the speed of data transfer what are the different methods applied using compression techniques. This literature survey also contributes to information of different encoding methods used to secure data before sending for cloud.

2. LITERATURE SURVEY

Tariq and Agarwal [1] showed how the data can be accessed securely using the keyword which is stored in the cloud. This accessing is securely done using the fuzzy keyword searching wherein the keyword is one which is saved in the cloud during the file uploading. The fuzzy keyword search algorithm works efficiently by giving the results if there are minor spelling mistakes. Here during the data transfer from the client to the cloud the data is encrypted using AES symmetric algorithm with secret key. The encrypted file in the server is again encrypted using the asymmetric RSA providing double encryption and more security. During downloading the file from the server, the keyword is matched and then it is decrypted using RSA and AES decryption algorithms. The results are good for keyword search but the encryption and decryption time are more compared to the conventional AES algorithm. Das [2], concentrated on security and confidentiality of cloud data, using the padding and hybrid encryption RSA algorithm. The user data is first encoded by padding and then the hybrid encryption RSA algorithm is applied. The Homomorphic encryption is done for the encrypted message to allow multi-party computation without the need for decryption which results in maintaining security and confidentiality with moderate overheads.

William *et al.* [3] wrote the article where the scholars suggested the combination of symmetric and asymmetric algorithm which is also processed through the hashing algorithm. The proposed methodology first converts the given data into cipher text using AES algorithm of key size as 128, 192 or 256 bits. The key of AES is again processed through the elliptical curve cryptography (ECC) algorithm for encryption. The encrypted text is again fed through the secure hash algorithm (SHA) 256 algorithm to generate the message digest. The encrypted message and the encrypted AES key both are sent over the network, wherein first the encrypted AES key is decrypted using ECC decryption algorithm and then using this obtained key the AES decryption is carried out so that the original plain text is obtained. The digest of the SHA is used for data integrity check. The results are calculated for both textual and image data sets. Gupta and Namasudra [4] proposed a scheme where a network is developed which consists of Master Servers and slave servers. One target server is selected among all slave server. One target server is selected among all slave server which is used when any faulty node is found. All the pages of virtual machine are transferred to the Target Server which starts responding for the master slave. This process is repeated by frequently searching for Target Servers depending on its available resources. The three proposed algorithms perform different operations, namely the host selection migration time algorithm which is used to find out the faulty node where in the master server tracks faulty node by sending broadcast message to all slave servers. Then the virtual machines

(VM) reallocation migration time is required to find the target server which the algorithm finds by finding the server that has max capacity and sufficient resources. In the VM reallocation bandwidth usage the migration takes place from faulty node to the target server. It also takes care that users request not lost during migration. The comparison analysis is made by taking different virtual machines and calculating their CPU core and migration time.

Sharma *et al.* [5] are securing the health care files of the hospitals using the identity-based encryption. This health care data is collected in the form of files with the help of block chain-based internet of things (IoT) architecture, during this collection procedures the files are processed through different layers of the architecture. Firstly, in the application layer patient’s data is stored that are collected with the help of IoT devices. The next layer is the blockchain layer where the development is based on the Etherew platform which is basically designed to perform all smart contract functionalities such as registration, logic uploading and integrity check. The third layer is security layer which manages security and authentication. The files are first encrypted and then uploaded, the integrity checking facility is also provided for the users. Lastly in the physical layer Swarm storage is used where the data is stored after encryption of the file using the AES. Seth *et al.* [6] proposed the hybrid encryption algorithm which is constructed to make competent and safe data transfer over the cloud. The plain text is split into n fragments and for each fragment the Paillier Encryption homomorphic algorithm is applied and the encrypted data is sent over the cloud. This encrypted data is again encrypted in the cloud using Blowfish algorithm which is nothing but double encryption. The segments are also processed for HashMap to maintain the data integrity. During the decryption process each fragment is subjected to blowfish first and then for the Paillier decryption algorithm. During this decryption if all the Hash mapped segments are matched then the merging of fragments takes place. The proposed hybrid methodology results are compared with that of the hybrid algorithm of RSA and AES. Abid *et al.* [7] proposed a plan to provide higher security by using 4 keys for encryption and decryption that are generated through RSA algorithm where 4 prime numbers are used instead of 2 prime numbers, the two open keys are used for decryption and two private keys are used for decryption. Further the Chinese remainder theorem-RSA (CRT-RSA) encryption process is carried out using homomorphic encryption by scrambling the data with transposition, shuffling and XOR operations. The comparison study is made between different cryptographic algorithms and proposed homomorphic encryption-CRT RSA (HE–CRT RSA) algorithm.

El-Attar *et al.* [8] suggested the hybrid automated algorithm to maintain the confidentiality and obtain high efficiency during encryption of large files. The proposed method consists of random key generation using the RSA algorithm for generating the private keys and encrypted keys. The data to be uploaded are divided into blocks of random size wherein for each block the automated sequential cryptography and automated random cryptography is applied and then the encrypted blocks are stored in the cloud. Both sequential and random algorithms majorly work on AES and DES approaches. Both the automated approaches resulted in high level of security and also achieved high efficiency during encryption and decryption. The results are also compared with improved automated random cryptography based on S-Box generator. This improved algorithm resulted in more efficient results compared to both the previous methods. Tajammul *et al.* [9] mainly concentrated on generating a key on sensing data automatically. Here the file which is uploaded by the user is first passed through the digital signatures algorithm (DSA) which generates key and also encrypts the data simultaneously. The encrypted data is further compressed hence the output of this algorithm is both encrypted and compressed file which helps in occupying less space in cloud. As the generated keys are stored in local server, they are used during the decompression and decryption process. The result shows the comparison study analyzing encryption and decryption time that is made using the different file size and the time is obtained before the compression and after the compression. This also resulted in saving cloud space resulting in cost efficiency. Thabit *et al.* [10] described about a new light weight cryptographic algorithm by generating 5 separate keys in each round of Feistel based encryption algorithm with substitution and permutation technique which is further used to encrypt and decrypt the data. The results show the high level of security along with which high flexibility is achieved. Pavani *et al.* [11] stated the problems and solutions for data protection issues such as safety or interlinked defense. The proposed algorithm allows for build, edit, uninstall and store info of file for the users. Here the authentication algorithm generates the key, encrypts the data by calculating $E = [p + k + i]$ where p is the plain text, k is the shared link and i is the attribute letters place and then send it to cloud storage server along with the master key and all other keys for cloud users. The decryption is done using $([c - k - i] + 256)$ where c is the cipher text, and I is the attribute+ location in cipher text. The efficiency is calculated for key generation the larger the key size the more time to generate the key, file upload and download time is calculated for varying file sizes. Hence the confidentiality is maintained using the efficient key management method.

Tajammul and Parveen [12] proposed to encrypt the plaintext and then upload the file to the cloud. To encrypt the file, the algorithm uses 6 different matrices to generate a key matrix and also to encrypt the data. The generated matrices are also kept in user desktop to decrypt the data. The six different matrices are used such as F matrix to store the frequency of each letter of plain text, R matrix is used for reduced frequency value of each letter, character matrix is produced by converting R values into characters by summing, S matrix is used for symbol matrix having 36 symbol, M matrix is used for corresponding entries. N matrix is the transpose of M matrix with the help of all these matrices the key is randomly generated.

Kaffah *et al.* [13] have built the crypto mail sending software which takes care of the security. Before sending the mail to the client the file is attached and the content of mail are encrypted by AES algorithm followed by Huffman compression. These encrypted files are sent directly through simple mail transfer protocol (SMTP) along with the key which is used for decryption at the sender's side. The algorithm also calculated the level of accuracy by using the following formula [11].

$$Accuracy = (Total\ no\ of\ successful/(Total\ amount\ succeeded + Failed)) * 100\%.$$

Kumaresan and Shanmugam [14] presented the comparison study of their previous method of Attribute based flexible delegation and the proposed method of time variant attribute based multitype encryption algorithm. For 'A' attributes with 'K' keys that has to be given for n users. Hence for each user a taxonomy is created and the access for services is given only to those users whose taxonomy evaluation is cleared. The data encryption is done using padding scheme which changes dynamically. Data manipulation access is given only for the authorized access service users. The taxonomy of the user gives a clear detail about each user who are allowed for which type of service access. Namasudra *et al.* [15] worked on deoxyribonucleic acid (DNA) computing to provide security for big data in cloud environment. Here 1024-bit DNA based password is used as a key. The user after registration needs to login and then request for data access wherein the public key of the data owner is searched from the database and provide the public key for only those clients whose certificate and the secret key authenticity is confirmed. Post of which the data owner sends the generated DNA based secret key and certificate for the user which helps the user to access and store the data. In this method the results are calculated for number of users v/s key generation time and encryption and decryption time. Patel and Dadhania [16] proposed the article where the authors used both the symmetric and asymmetric algorithms such as AES and RSA. The MD5 Hash algorithm is also used for encryption. The given text or images is initially converted into a single vector from the formed row column matrix. This vector is divided into two blocks such as A and B, the block A is substituted to AES plus Hash for encryption and the block B is subjected to RSA plus Hash. The key obtained from both the blocks are XOR'ed and form a new key C which is again used for encryption. The same is used for decryption and continued obtaining plaintext by reverse process.

Sivakumar *et al.* [17] used Heroku cloud as a cloud platform where data uploaded by the user is processed through advance encryption standard (AES) and the generated key is given for the user. Here majorly performance is calculated based on assess delay. The size of the text file used here are of 3, 5, 7, 10 and 15 MB which clearly shows that as there is increase in file size and hence the delay time is also increased simultaneously. Singh and Sharma [18] tried to enhance the cloud data security along with improving the efficiency by using various cryptographic techniques. Here the data is split into different modules and all the modules are stored in different cloud platforms where in applying AES and secure hash algorithm (SHA) for encryption. The comparison study is also made between the split algorithm and different standard symmetric algorithms for analyzing the security level, number of keys used and efficiency of the algorithms. The results are better for the split algorithm giving the highest efficiency using a single key resulting in high level of data security.

Makkaoui *et al.* [19] worked on cloud data confidentiality and efficiency. Cloud-RSA and multiprime RSA are used to Encrypt the message whereas decryption is processed using multiprime RSA along with CRT to obtain high efficiency. The performance is calculated by comparing the results of encryption and decryption time for multiprime cloud RSA, cloud RSA and Cloud-ElGamal multiprime cloud RSA is also used for security analysis, it shows the good efficiency results compared to other methods. Lavanya and ThamizhThendral [20] presented the articles by the self-designing the algorithm known as deep substitution encryption method. Here the given plain text is passed through five different substitution technique i.e., each character has 6 cipher text. The plain text is first substituted with American Standard Code for Information Interchange (ASCII) code, after obtaining the ASCII codes the periodic elements of the periodic table are substituted. These periodic vales are further substituted with flower names. The 4th substitution phase i.e., hypertext markup language (HTML) color names given for the flower names of 3rd phase. This color names are finally substituted as HEXCODE which is considered as cipher text. The cipher text is decrypted back to the original plaintext by reversing all the five substitution phases in reverse order. To break this algorithm the hackers, need to have a vast knowledge about the entire periodic table and name of flowers.

Miri and Rashid [21] proposed a method for data duplication, which is preserved from access of semi secured cloud service providers. The given plaintext is processed through Burrows wheeler transformation (BWT) encoding in which the block of plaintext is sorted using lexicographic sort which is further subjected to bzip2 function applying 'move to front transformation and Huffman coding. The transformation created a L vector which is used to decompress the encoded text. To use this L vector the user is asked question for verification. Failing of verification will lead to access of only compressed file. The compressed files of the user are compared to avoid deduplication.

Devi and Mani [22] applied double compression before encrypting the file to enhance the data security. Burrows wheeler transformation (BWT) technique is used for which move to front transformation is applied. The obtained results are further subjected to run length encoding (RLE) compression to reduce the complexity of redundancy. Which is finally dealt with modified RSA algorithm to encrypt the file, the RSA algorithm is modified by converting the obtained 'n' value into binary values which are further applied to find the cipher text. Singh *et al.* [23] tried to apply a simple and most secured encryption algorithm by using binary sequence for the given plain text. Firstly, the given plain text is converted into the binary bits and then it appended to make km/2 bits where Km is the mirror key and Kr is the rotational key. Applying mirroring to the appended sequence and then processed further for rotation leading to cipher text. The decryption process is carried out in reverser by rotational operation and then sequence mirroring and finally removing the right most bits which will generate the final original plain text. This algorithm is proved to be secure against the brute force attack also.

Abdullah *et al.* [24] here the plain text is divided into 2 parts, in which for the first part of the plain text in encrypted using the AES algorithm of key size 128 bits. The encrypted text is followed by Lempel-Ziv-Welch (LZW) compression. Then the second part of the plaintext is encrypted using 2,048 bits key of RSA which is also further subjected to LZW compression. Due to the use of both the AES and RSA algorithms the security level and robustness is high without comprising with the efficiency. The results of the proposed hybrid cryptographic algorithm (HCA) are compared with that of several other algorithms proving the results of HCA algorithms as best.

Mani and Devi [25] could enhance the data security by applying pre-processing before encryption. The pre-processing involves encoding the given plain text by using Lucas and Fibonacci series obtaining first level of security and then the encoded text is further compressed with Huffman encoding resulting in second level of security and finally it is subjected to the RSA public key cryptographic algorithm resulting in cipher text which is third level of security. The cipher text is converted back to the plain text in reverse process by decrypting the cipher text then decompressed and finally decoded obtaining original plain text. Namasudra and Roy [26] in this scheme the cloud service provider maintains the separate table for storing the data. The table basically consists of four columns wherein the first column is the group number (Gp. No), second column is the data owners ID which is the ID number given for each data of cloud service provider. The third column is the data size column in which the data is moved to the particular group, i.e. the data owner sends the data to the cloud service provider and this data will be sent to particular group according to its size. This helps in reducing the access and search time of the data. During the process of file uploading, the data is first encrypted using the secret key and then it is again encrypted using the private key of the data owner. The data owners encrypt the data and the certificate using the cloud service providers public key and finally makes a bundle of encrypted data. The fourth column of the cloud service provider table consists of the date and time of the file which the data owner uploaded. While accessing the data the cloud service provider is trying to access and search the file only in that section. The data owners are also asked to register before sending any file to the cloud service provider. Mani and Devi [27] tried to enhance the data security by generating different key streams using Pythagorean triplets. Any two positive prime integers are used to find the triplets such that the gcd of the triplets is always 1. Barning tree is constructed using this triplet which are further used for used as a keystream. Out of the 8-bit keystream 1 bit is used for parity checking. DES algorithm is modified for encrypting the data file by using 8 characters from the primitive pythagorean triples (PPT) during first round of DES and is continued for all the 16 rounds of data encryption standard (DES). The modified DES results in good security and the generation of keystreams using PPT will enhance the security levels.

3. DISCUSSION

Table 1 helps to analyse the study of different techniques that are used to frame the proposed models which are intern enhancing the cloud data security and also the discussion is made about their limitations and challenges faced during the process. Considering all these proposed methods, it could be analysed that the RSA and AES algorithms along with Fiestal cipher model played a vital role in cloud data security. There are many methods been proposed with a hybrid combination of cryptographic and compression algorithms to enhance data security.

Table 1. Analyzing the various techniques used to enhance the cloud data security

Ref No.	Algorithm used	Discussion
[1]	AES, RSA, fuzzy keyword search	The proposed algorithm proved with good encryption time and decryption time along with authentication but failed to obtain the results of security level for the designed methodology.
[2]	Optimal asymmetric encryption padding (OAEP)-homomorphic encryption (HE)-RSA for encryption	The result analysis of achieved security level are not mentioned and also leads in limitations of homomorphic encryption.
[3]	AES with ECC	This method proves the security for medical data exchange which ensures authentication and integrity using SHA and confidentiality of sensitive data.
[5]	Symmetric AES algorithm	The analysis is done for throughput (total packet received/time frame) and during data upload data access and integration checking. The throughput and delay calculations are done using JMeter. The comparison study is made between the methods and hence proved that the proposed method takes less time to encrypt and decrypt the health care files.
[6]	Paillier and Blowfish without compression, Paillier and Blowfish with Compression, RSA and AES	The methodology proves 28% more efficient than the combined RSA and AES approach which also maintains the confidentiality, integrity and accessibility against anticipated attacks.
[7]	CRT–RSA, HE-CRT-RSA	The results obtained for encryption and decryption time of the proposed method are compared with the conventional RSA algorithm and could find that the proposed method result in good output but the level of security and authentication is not analyzed.
[8]	ASC, ARC, IARC	The encryption and decryption time for different file sizes have been calculated comparing the results with the conventional algorithms achieving results for the proposed methodology.
[9]	DSA, GZIP, AES	The total time saving factor is calculated for encryption and decryption before and after compression but the security levels and authentication is compromised.
[10]	NLCA	Along with the proposed algorithm the experimental results are also compared with different cryptographic algorithm and proved that the proposed method has high security achieving all the threats of data security
[11]	ASCII characters, mathematical formulas are generated for encryption and decryption	Comparison study is made between the proposed method and Attribute based encryption (ABE) method achieving high confidentiality, but the results are not computed security levels.
[12]	Algorithm used 6 Matrix such as F matrix, R matrix, C matrix, S matrix, M Matrix, N Matrix	The algorithm analysis is made for different file sizes from 500KB to 5 MB resulting in same encryption and decryption time. The algorithm fails to analyze the space complexity along with authentication and integrity.
[13]	AES and Huffman Encoding	The designed SMTP software gives a good accuracy level of 90.62% and also calculating the encryption and decryption time.
[14]	Attribute based encryption, Time variant and attribute-based encryption algorithm	The comparison study is analyzed between the two methods of ABE and TAM depending on the number of services, calculating throughput performance and security levels.
[15]	DNABDS	Failed to analyze the transmission speed and security levels.
[16]	AES, RSA, AES+Fiestal+Hash, AES-Hash+RSA-Fiestal	This methodology results are analyzed for both textual and image data concentrating on security level and data integrity neglecting the authentication consequences
[17]	Heroku cloud and AES	The results did not specify the security level and integrity check is also to be made.
[19]	Cloud RSA, Multiprime Cloud RSA, Cloud ElGamal	The algorithm gives the best comparative results during encryption and decryption which is done using addon Chinese Remainder Theorem algorithm. But the security levels and authentications measures are avoided.
[20]	Deep substitution encryption method (DSEM)	To break this algorithm the hackers, need to have a vast knowledge about the entire periodic table and name of flowers. On an average to crack the code for 12 characters two centuries are needed. The proposed methodological implementation and results are not analyzed.
[21]	Burrow wheel transform encoding (BWT), bzip2 and lexicographic sort	The comparison study is made before and after compression, i.e., for a file size of 36KB the encoded size is 36KB but after compression the file size is 16KB, which is the major change in saving the space.
[22]	RSA after using modified BWT, RLE – BWT, RLE – EBWT	Here the comparison study is made between the proposed modified BWT method and the convention RSA, showing the results of efficiency, compression ratio and security levels of modified BWT.
[23]	Encryption using translation, mirroring and rotation of Binary sequence	The article lags in mentioning the computational results of security level which is the prime concept. The authentication and data integrity are not considered for secure transmission.
[24]	HCA	The results of the proposed HCA are compared with that of several other algorithms proving the results of HCA as best. The analysis of security levels is not mentioned.
[25]	RSA, RSAFIB, RSALUC, Huffman Coding	The proposed methodology compromises with the encryption time and decryption time, achieving high security levels and good compression ratio.
[26]	Size based secure access control model for cloud computing (SzSBAC)	Here the data searching time is minimized because of which the users can pay less money for using the cloud services
[27]	Primitive Pythagorean triplets, data encryption standards, hamming distance	The hamming distance is calculated between two consecutive rounds to analyze the strength of the keys generated using 56 bits.

4. RESULTS

From the above review following results were taken to make the proper analysis of the different methodologies used. The efficiency of the algorithms using encryption and decryption time for different file size is analyzed and converted the different file size formats i.e., from bits, bytes, megabyte and gigabytes to one file size formats as Kilobytes, similarly the different time formats like sec and minutes are converted to one time file format as milliseconds. This conversion is basically made to analyze the difference in the encryption and decryption time of different methodologies designed by different researchers and it is shown in the Table 2. Some of the papers used for review showed the comparative study of all the parameters such as confidentiality, efficiency and security level that are related to data security.

Table 2. Analyzing the encryption time and decryption time for various methodologies

Ref. No.	File size (KB)	Algorithm used	Encryption time (ms)	Decryption time (ms)
[1]	311.9462	AES	102	-
[1]	311.9462	AES with RSA	137	-
[3]	150	AES with ECC	88	113
[3]	150	AES	1,815	2,305
[6]	7.01171	Paillier and Blowfish without compression	6,133.4	6,133.4
[6]	7.01171	Paillier and Blowfish with compression	2,140,520	2,140,520
[6]	7.01171	RSA & AES	237,070	237,070
[7]	0.064	CRT–RSA	0.019	0.02
[8]	2306867.2	Automated sequential cryptography (ASC)	127,370	465,630
[8]	2306867.2	Automated random cryptography (ARC)	162,760	486,220
[8]	2306867.2	Improved automated random cryptography (IARC)	12,702,620	14,195,370
[9]	512	Data sensitive algorithm (DSA)+GZIP	5,000	3,000
[9]	512	AES	10,000	7,000
[10]	51200	New lightweight cryptographic algorithm (NLCA)	2,342	-
[12]	5120	Algorithm used 6 matrix such as F matrix, R matrix, C matrix, S matrix, M matrix, N matrix	74,000	74,000
[13]	5.05371	AES and Huffman encoding	20.531	1.349
[15]	375000	Deoxyribonucleic acid-based data security (DNABDS)	9,000	1,000
[16]	0.00391	AES	0.06	0.0068
[16]	0.00391	RSA	0.055	0.0456
[16]	0.00391	AES+Fiestal+Hash	0.04	0.0358
[16]	0.00391	AES-Hash+RSA-Fiestal	0.032	0.0021
[19]	0.025	Cloud RSA	5.75	451.15
[19]	0.025	Multiprime cloud RSA	5.75	800.54
[19]	0.025	Cloud ElGamal	742.67	451.22
[22]	16384	RSA after using modified BWT	48,957	48,925
[22]	16384	BWT-RSA	42,836	42,422
[22]	16384	RLE–BWT–RSA	39,155	3,912
[22]	16384	RLE–EBWT–RSA	34,652	34,647
[24]	25.01464	Subasree	2,063	1,085
[24]	25.01464	Ren	1,432	821
[24]	25.01464	Zhu	998	713
[24]	25.01464	Two-phase hybrid cryptography algorithm (THCA)	998	713
[24]	25.01464	HCA	548	636
[25]	16384	RSA after compression	48,939	48,923
[25]	16384	RSA based Fibonacci encoding (RSAFIB) after compression	41,993	41,986
[25]	16384	RSA based Lucas encoding (RSALUC) after compression	40,011	40,025

5. CONCLUSION

This literature review gives a clear understanding of the different methodologies used for cloud data security. It is used to understand the importance of different cryptographic algorithm. The analysis of all the methodologies used to find the efficiency levels of different algorithms and also the security levels. The RSA and AES algorithms plays a vital role in data security with good efficiency. The cloud security can be enhanced using different compression algorithms such as Lempel–Ziv–Welch (LZW), arithmetic coding or any other lossless compression algorithm which intern reduces cloud storage space and improves cloud security levels. The survey gives a good future to design better algorithms to enhance a cloud data security.

REFERENCES

[1] H. Tariq and P. Agarwal, "Secure keyword search using dual encryption in cloud computing," *International Journal of Information Technology*, vol. 12, no. 4, pp. 1063–1072, Dec. 2020, doi: 10.1007/s41870-018-0091-6.

[2] D. Das, "Secure cloud computing algorithm using homomorphic encryption and multi-party computation," in *2018 International Conference on Information Networking (ICOIN)*, Jan. 2018, pp. 391–396, doi: 10.1109/ICOIN.2018.8343147.

[3] P. William, A. Choubey, G. S. Chhabra, R. Bhattacharya, K. Vengatesan, and S. Choubey, "Assessment of hybrid cryptographic algorithm for secure sharing of textual and pictorial content," in *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, Mar. 2022, pp. 918–922, doi: 10.1109/ICEARS53579.2022.9751932.

[4] A. Gupta and S. Namasudra, "A novel technique for accelerating live migration in cloud computing," *Automated Software Engineering*, vol. 29, no. 1, May 2022, doi: 10.1007/s10515-022-00332-2.

[5] P. Sharma, N. R. Moparthi, S. Namasudra, V. Shanmuganathan, and C. Hsu, "Blockchain-based IoT architecture to secure healthcare system using identity-based encryption," *Expert Systems*, vol. 39, no. 10, Dec. 2022, doi: 10.1111/exsy.12915.

[6] B. Seth *et al.*, "Secure cloud data storage system using hybrid Paillier-blowfish algorithm," *Computers, Materials and Continua*, vol. 67, no. 1, pp. 779–798, 2021, doi: 10.32604/cmc.2021.014466.

[7] R. Abid *et al.*, "An optimised homomorphic CRT-RSA algorithm for secure and efficient communication," *Personal and Ubiquitous Computing*, Sep. 2021, doi: 10.1007/s00779-021-01607-3.

[8] N. E. El-Attar, D. S. El-Morshedy, and W. A. Awad, "A new hybrid automated security framework to cloud storage system," *Cryptography*, vol. 5, no. 4, Dec. 2021, doi: 10.3390/cryptography5040037.

[9] M. Tajammul, R. Parveen, N. K. Gaur, and S. D, "Data sensitive algorithm integrated with compression technique for secured and efficient utilization of cloud storage," in *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, Sep. 2021, pp. 1–9, doi: 10.1109/GUCON50781.2021.9573648.

[10] F. Thabit, A. P. S. Alhomdy, A. H. A. Al-Ahdal, and P. D. S. Jagtap, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 91–99, Jun. 2021, doi: 10.1016/j.gltp.2021.01.013.

[11] V. Pavani, P. S. Krishna, A. P. Gopi, and V. L. Narayana, "Secure data storage and accessing in cloud computing using enhanced group based cryptography mechanism," in *Materials Today: Proceedings*, Dec. 2020, pp. 1-5, doi: 10.1016/j.matpr.2020.10.262.

[12] M. Tajammul and R. Parveen, "Auto encryption algorithm for uploading data on cloud storage," *International Journal of Information Technology*, vol. 12, no. 3, pp. 831–837, Sep. 2020, doi: 10.1007/s41870-020-00441-9.

[13] F. M. Kaffah, Y. A. Gerhana, I. M. Huda, A. Rahman, K. Manaf, and B. Subaeki, "E-mail message encryption using advanced encryption standard (AES) and huffman compression engineering," in *2020 6th International Conference on Wireless and Telematics (ICWT)*, Sep. 2020, pp. 1–6, doi: 10.1109/ICWT50448.2020.9243651.

[14] S. Kumaresan and V. Shanmugam, "Time-variant attribute-based multitype encryption algorithm for improved cloud data security using user profile," *The Journal of Supercomputing*, vol. 76, no. 8, pp. 6094–6112, Aug. 2020, doi: 10.1007/s11227-019-03118-8.

[15] S. Namasudra, D. Devi, S. Kadry, R. Sundarasekar, and A. Shanthini, "Towards DNA based data security in the cloud computing environment," *Computer Communications*, vol. 151, pp. 539–547, Feb. 2020, doi: 10.1016/j.comcom.2019.12.041.

[16] U. Patel and P. Dadhania, "Multilevel data encryption using AES and RSA for image and textual information data," in *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Mar. 2019, pp. 1–5, doi: 10.1109/i-PACT44901.2019.8960227.

[17] P. Sivakumar, M. NandhaKumar, R. Jayaraj, and A. S. Kumaran, "Securing data and reducing the time traffic using AES encryption with dual cloud," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, Mar. 2019, pp. 1–5, doi: 10.1109/ICSCAN.2019.8878749.

[18] A. Singh and S. Sharma, "Enhancing data security in cloud using split algorithm, caesar cipher, and vigenere cipher, homomorphism encryption scheme," in *Advances in Intelligent Systems and Computing*, vol. 841, Springer Singapore, 2019, pp. 157–166.

[19] K. El Makkaoui, A. B. Hssane, and A. Ezzati, "MultiPrime cloud-RSA: a fast homomorphic encryption scheme for data confidentiality protection in clouds," *International Journal of Intelligent Enterprise*, vol. 6, 2019, doi: 10.1504/IJIE.2019.101128.

[20] B. Lavanya and V. ThamizhThendral, "A novel data ciphering method for secure cloud storage," in *2019 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2019, pp. 1–6, doi: 10.1109/CCST.2019.8888439.

[21] A. Miri and F. Rashid, "Secure textual data deduplication scheme based on data encoding and compression," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct. 2019, pp. 207–211, doi: 10.1109/IEMCON.2019.8936222.

[22] A. Devi and K. Mani, "Enhancing security in RSA cryptosystem using burrows-wheeler transformation and run length encoding," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, no. 1, pp. 687–696, 2018.

[23] V. P. Singh, P. Sharma, D. Kumar, and N. Jaiswal, "A new symmetric key encryption algorithm based on jumbling binary sequence of message," in *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Jun. 2018, pp. 143–146, doi: 10.1109/ICACCE.2018.8458064.

[24] K. M. Abdullah, E. H. Houssein, and H. H. Zayed, "New security protocol using hybrid cryptography algorithm for WSN," in *2018 1st International Conference on Computer Applications and Information Security (ICCAIS)*, Apr. 2018, pp. 1–6, doi: 10.1109/CAIS.2018.8442003.

[25] K. Mani and A. Devi, "Enhancing security in cryptographic algorithm based on LECCRS," *Electronic Government, an International Journal*, vol. 13, no. 1, 2017, doi: 10.1504/EG.2017.083940.

[26] S. Namasudra and P. Roy, "A new table-based protocol for data accessing in cloud computing," *Journal of Information Science and Engineering*, vol. 33, no. 3, pp. 585–609, 2017, doi: 10.6688/JISE.2017.33.3.1.

[27] K. Mani and A. Devi, "Modified DES using different keystreams based on primitive pythagorean triples," *International Journal of Mathematical Sciences and Computing*, vol. 3, no. 1, pp. 38–48, Jan. 2017, doi: 10.5815/ijmsc.2017.01.04.

BIOGRAPHIES OF AUTHORS

Amruta Gadad received her Master’s degree in Computer Science from University of Mysore, Mysore, Karnataka, India, in 2012 and Qualified the State Level Eligibility Test (KSET) in the 2020. Currently, she is a Research Scholar at the School of Computer Science and Applications, REVA University, Bangalore, Karnataka. She is working as an Assistant Professor in The Oxford College of Science, Bangalore, Karnataka. During 2017-2021, she was working as a Guest faculty in Government First Grade College, Karnataka. During 2013 -2017, she has worked as Hardware Technician, Vcompumatiks, Belgaum, Karnataka. Her research interests include cryptography, data compression, cloud computing and cloud data security. She can be contacted at email: amgabsc@gmail.com.

Devi Anbusezhiyan received her MCA and M.Phil. from Bharathidasan University, Trichy, India in Computer Science Applications. During 2004-2016 (April), she had been with the Department of Computer Science at the Lowry Memorial College, affiliated to Bangalore university, Karnataka, India where she was working as an Associate Professor. During 1998-2001, She was working as a programmer in different software companies. During 2017-2019, she was working as an Asst. Professor in Karnataka College, affiliated to Bangalore university, Karnataka, India. She is currently working as an Associate Professor in REVA University, Bangalore, Karnataka, India. She completed her PhD in Cryptography with “Efficient Encoding and Key Generation Techniques to Enhance the Security of Compressed Cryptosystems”, Bharathidasan University, Trichy, India. She published and presented around 11 research papers at international journals and conferences. She published 4 patents. She can be contacted at email: devi.a@reva.edu.in.