International Journal of Advanced Research in Engineering and Technology (IJARET)
Volume 10, Issue 5, September-October 2019, pp. 154-165 Article ID: IJARET_10_05_016
Available online at http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=10&IType=5
ISSN Print: 0976-6480 and ISSN Online: 0976-6499
© **IAEME** Publication

TIME CONSERVING SECURED CLOUD DATA STORAGE SOLUTION BASED ON KECCAK AND ELLIPTIC CURVE CRYPTOGRAPHY

B. Jyoshna
Research Scholar, KL University, Computer Science and Engineering,
Keshav Memorial Institute of Technology, Hyderabad, India

Dr. K. Subramanyam
Associate Dean, Computer Science and Engineering,
KL University, Vaddeswaram, India

ABSTRACT

The cloud computing environment offers numerous benefits and different flavours of data storage patterns, which completely relieves the users from the annoying processes of data management, storage equipment upgradation and so on. However, the major concern of the users is the data security and to combat with it several security-based solutions are presented in the literature. This article presents a time conserving cloud data storage solution based on keccak and Elliptic Curve Cryptography (ECC) with minimal storage overhead. The proposed work applies LZ77, which is a lossless compression algorithm on the data to be outsourced and then the proposed algorithm is applied. This work does not employ any third-party agent for tracking the data security, as the third party is not completely trustworthy. Hence in this approach, the cloud users apply the proposed algorithm before the process of outsourcing and this approach withstands data analysis and data tamper attacks. Additionally, the proposed work consumes minimal time for execution, which in turn reduces the energy consumption as well. The performance of the proposed work is effective, when compared to the existing algorithms.

Keywords: Cloud data security, data compression, energy conservation, data outsourcing

Cite this Article: B. Jyoshna and Dr. K. Subramanyam, Time Conserving Secured Cloud Data Storage Solution Based on Keccak and Elliptic Curve Cryptography, *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 10 (5), 2019, pp 154-165.
http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=10&IType=5

1. INTRODUCTION

Cloud computing is a promising technology that has grabbed the attention of numerous researchers. The cloud computing technology renders boundless benefits to the cloud service users and it is extremely simple for the users to exploit the cloud services. The only demand

placed by the Cloud Service Providers (CSP) to the Cloud Service Users (CSU) for the service access is the network connectivity. The CSU can enjoy all the services provided by the CSPs provided the CSU has got network connectivity. There are three important entities involved in the concept of cloud computing and they are CSP, CSU and the cloud service.

The CSP provides a wide range of services to the users and the services are categorized under three classes. They are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The IaaS and PaaS are usually utilized by the technical group of users, while the SaaS can be utilized even by the novice users. The cloud services provide even the memory and processing capability to the users. All the users need is a computing device connected to the network and the computing device can be of any type such as laptop, smart phones, tablets and so on.

The CSU utilizes the cloud services provided by the CSP for a standard charge mentioned in the Service Level Agreement (SLA). Both the CSP and CSU should adhere to the SLA, wherein all the terms and conditions are presented. Irrespective of the presence of SLA and numerous attractive features of cloud model, the cloud users feel disinclined to utilize the cloud services, especially data storage. The cloud data storage services make the users free from the hassles such as data management, storage equipment upgradation and so on however, the users feel insecure that the outsourced data may get leaked or misused.

The CSP may gain access over the outsourced data, which makes the client unwilling to share the data. Certain cloud servers provide encryption for data security, however all the keys involved are also stored in cloud storage. This is a major drawback and the cloud users feel it unsafe. In order to deal with this issue, the cloud users could make the data in unintelligible format and the security-based information may be stored on the user side itself. In this case, the CSP cannot gain access to the data easily.

Recognizing the benefits of data outsourcing and security, numerous works are presented in the existing literature for ensuring data security. Yet, there is a constant demand for effective secured data storage solution with minimal computational, space and time overheads. This article presents a solution for secure data storage, which is based on Keccak hash tree and Elliptic Curve Cryptography (ECC). Keccak hash tree roots on the hashing process and the data is organized in tree like structure, which makes it easy to manage. The keccak is proven to be strong against several security threats and is a candidate of SHA-3 (Secure Hash Algorithm), which is released in the year 2015. The main contributions of this work are as follows.

- The data to be outsourced are encrypted in the client side, such that all the key based informations are completely dealt with by the clients.
- Initially, the data to be outsourced is compressed with the help of a lossless compression algorithm, in order to reduce storage and computational overhead.
- The utilization of keccak tree helps in easy retrieval and location of the required data, while ensuring better data organization.
- The inclusion of ECC algorithm provides even more security to the complete system.
- Dynamic key is employed for encryption with ECC algorithm, which further tightens the security.
- The proposed approach withstands data analysis and data tampering attacks.
- The security provided by the proposed work is analysed with respect to avalanche effect and is satisfactory.

The remainder of this paper is organized as follows. Section 2 discusses the related review of literature with respect to cloud data security. The proposed security solution for data storage in cloud servers is presented in section 3. The performance of the proposed work is evaluated in section 4 and the final section 5 concludes the article.

2. RELATED LITERATURE

This section presents details about the security solutions meant for the cloud data storage in the existing literature.

In [3], a trust-based access control system is presented for the cloud data storage. This work builds several trust models and cryptographic role-based access control mechanisms. The trust models decide the roles of the cloud users based on their trustworthiness. A data security model for cloud environment is presented on the basis of semi-trusted third party is presented in [4]. This work focuses on the issues such as key management, access control and file deletion. Shamir's threshold scheme is employed for key management. Numerous key managers are employed by the system to manage the key shares and it eliminates the issue of single point of failure as well.

A lightweight secure auditing scheme for cloud storage is presented in [5]. This work proposes a lightweight secure auditing scheme for shared data in cloud storage based on hashgraph and a third-party medium. The third-party medium takes care of the groups, which forms a group by combining the TCP sliding window concept and other interlinked functions to improve agent security.

In [6], a security model is presented to ensure privacy of medical data in healthcare cloud based on pairing based cryptography. This work secures the private health data in the cloud with the help of fog computing. Tri-party single round authenticated key agreement protocol is presented on the basis of bilinear pairing cryptography for establishing the session key.

A secure multiple keywords-based retrieval system that works over encrypted cloud data is presented in [7]. This paper deals the data privacy issues in cloud with the help of Searchable Symmetric Encryption (SSE). The server-side ranking with Order Preserving Encryption (OPE) leads to data privacy leakage. This issue is handled by a two-round searchable encryption, which provides top-k multiple keywords retrieval. This work employs vector space model and homomorphic encryption.

In [8], an identity-based data outsourcing with comprehensive auditing is presented for cloud computing environment. This work prompts the users to authorize dedicated proxies for data upload to the cloud server. The proxies are detected and authorized by eliminating complex certificate management in secure distributed systems. The proposed comprehensive auditing does audit the origin, type and consistency of the data.

In [9], security is presented as a service model for cloud computing environment. This work provides a baseline security to the cloud provider for securing the cloud infrastructure and several security functionalities are provided for cloud tenants. An identity based proxy-oriented data uploading and remote data integrity check is presented in [10]. This work follows identity based public key cryptography, which enables proxy oriented data upload and remote data integrity checking in public cloud. This work relies on bilinear pairings and Diffie-Hellman problem.

A secure and efficient data integrity verification scheme is presented for cloud Internet of Things (IoT) in [11]. This work states that the data integrity schemes based on Rivest Shamir Adleman (RSA) and Boney Lynn Shacham (BLS) signatures involve computational overhead and batch signature inefficiency respectively. To address this issue, a short signature algorithm is presented, which preserves privacy and ensures public auditing.

In [12], a security assessment model is presented for IaaS cloud. In this work, a cloud architecture reference model is built up and it involves numerous security controls and cloud security assessment model. The security of four multi-tenant cloud architectures is discussed with several security controls.

A multiple replica Merkle hash tree for public auditing for dynamic cloud bigdata storage is proposed in [13]. This work presents an authenticated data structure, which is on the basis of Merkle hash tree and it supports dynamic data updates and authentication. In [14], a technique to ensure public auditability and data dynamics for storage security is presented. The security problems with respect to dynamic data updates are discussed and the verification scheme is proposed. This work utilizes merkle hash tree for block tag authentication. Additionally, bilinear aggregate signature is employed for handling numerous auditing tasks.

In [15], a block design based key agreement for group data sharing is presented for cloud computing environment. This work is based on block design based key agreement protocol, which can support multiple users. The group data sharing model is employed for producing the conference key and the computational complexity is measured.

Motivated by the existing data security solutions for cloud computing environment, the proposed solution intends to keep the data secure from the user side itself, such that the need to rely on a third party system is not the case. The proposed cloud data security solution is elaborated as follows.

3. PROPOSED CLOUD DATA SECURITY SOLUTION BASED ON KECCAK AND ECC

Though there are several research issues associated with cloud computing, cloud data security is the most researchable topic since its advent. The mid and small scale business communities are attracted by the cloud services, which ensures numerous advantages such as flexibility, scalability, elasticity, pay for utilized services, service on demand and on. The main concern of the cloud users is the data security and the users distrust or doubt the CSP for the data misuse or leakage, due to the bitter experience in real world scenarios. For instance, Amazon S3 and Gmail are accounted for data deletion and system failure [16,17]. The overall flow of the work is depicted in figure 1.

Hence, the cloud users cannot rely on the CSPs completely and some precautions should be made even at the user side. One of the most effective precautions to avoid data misuse is the data encryption, which makes the data in unintelligible format. The CSP can gain access to the data only upon performing some manipulations and the ease of data access depends on the effectiveness of the encryption algorithm. The more efficient the encryption algorithm, the harder it is to break the security line and understand the data.

Understanding the potential of encryption algorithm, the proposed data security solution presents a secured algorithm based on keccak hash tree and ECC algorithms. The keccak algorithm is based on the concept of hashing and follows a tree data structure. This helps in attaining effective data organization and makes the process of data retrieval easier. Initially, the data to be outsourced is compressed by a lossless compression algorithm for minimizing the storage, time and computational complexities.

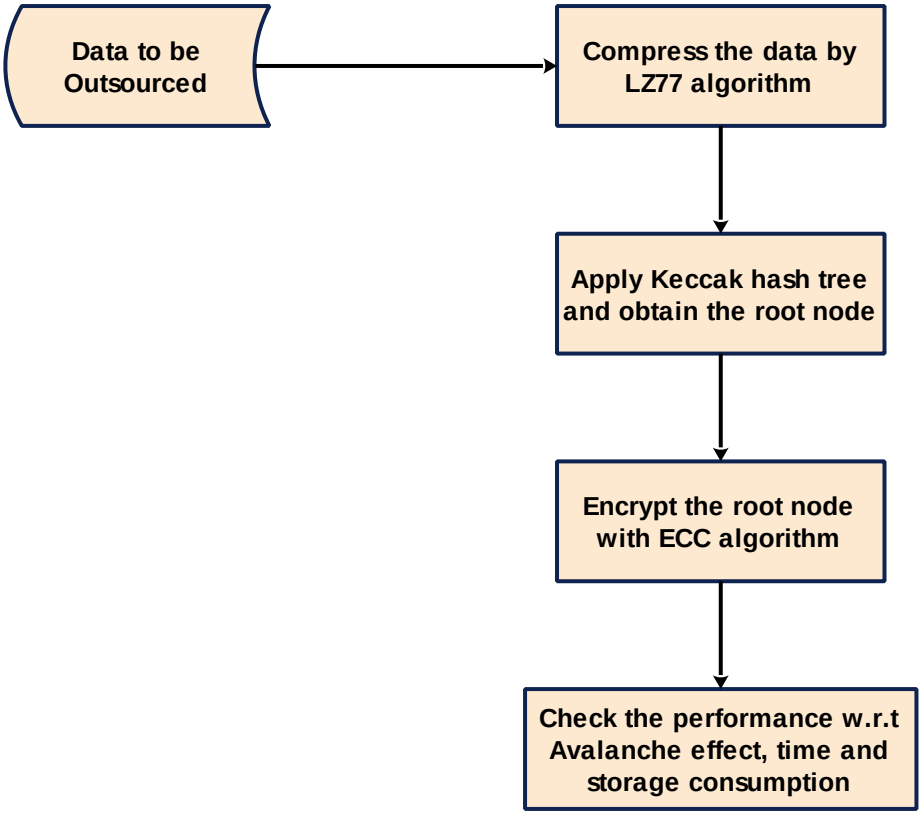

Figure 1 Overall work flow of the proposed work

The so compressed data is segregated into numerous data blocks and the hash codes are computed for the same. The data is organized in bottom-top approach, such that the root node carries the hash code of the entire tree. Thus, the root node is the most important part of the tree and hence, the hash code of the root node is again encrypted by ECC algorithm. This idea ensures better security and can effectively combat against data analysis and data tamper attacks. As this work relies on hash code, tampering data and analysing the data are quite difficult, owing to the hash code and the standard organization of data. All the phases involved in the proposed work are explained in the following section.

3.1. Data Compression

The proposed cloud data security solution compresses the data to be outsourced by employing Lempel-Ziv 77 (LZ77) algorithm [18,19], which is a popular lossless compression algorithm and is used in various commercial applications. One of the best sample applications that uses LZ77 compression algorithm is the ZIP. On applying this algorithm, the volume of the input data is minimized about forty percent. Compressing the data before the process of performing encryption reduces the data volume and thus the storage space. This in turn reduces the memory requirement and speeds up the entire process. Besides, this work employs lossless compression algorithm, such that the data is compressed without any alteration and reduces the time, space and computational complexities. The following section presents the encryption algorithm by combining the keccak and ECC algorithms.

3.2. Proposed Algorithm by Combining Keccak and ECC

As stated earlier, the Keccak hash function is a successful candidate of SHA-3 algorithms and it relies on sponge formation. The sponge formation is based on two important parameters such as bit rate (br) and capacity (c) [20]. Based on the br and c values, the sponge formation is

carried out and the summation of br and c fixes the permutation of the keccak's width. The working principle of keccak involves three important phases, which are initialization, absorption and squeeze. The working principle of the algorithm is shown in figure 2.

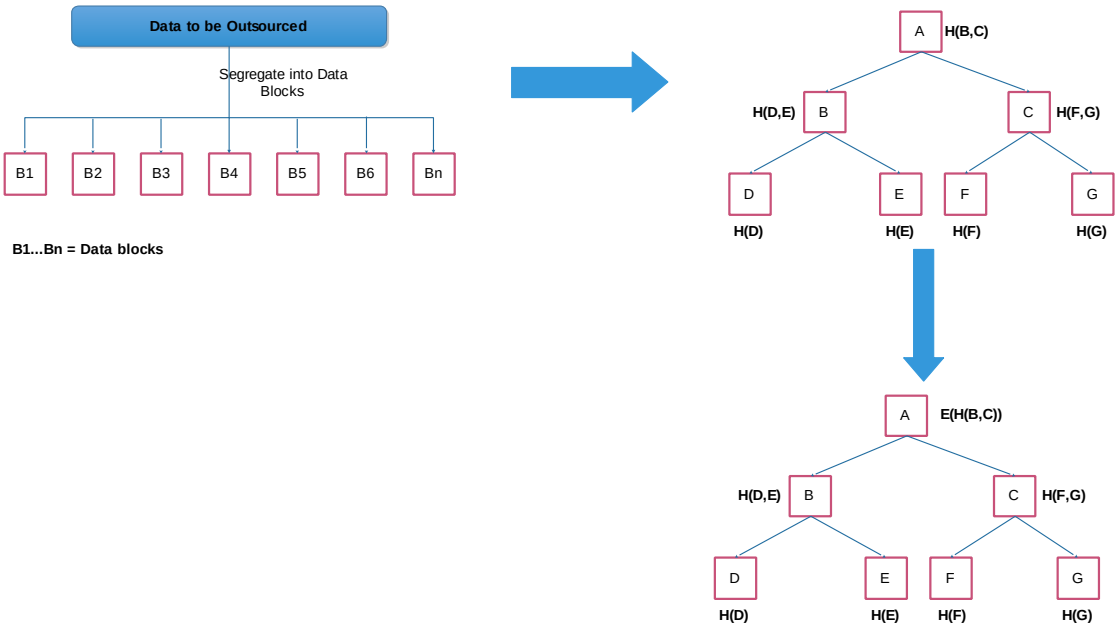

Figure 2 Proposed time conserving security-based solution for cloud data storage

In the initialization phase, the input data is divided into numerous data chunks and the keccak hash function is applied. During the second phase, the data chunks with fixed standard bits are performed XOR operation with the current matrix and this process is repeated for 24 rounds of permutations. When all the input data are processed by this way, the squeeze phase is initiated, in which the final matrix is truncated to get the required hash value. Suppose, when the required hash length is not attained, the process is repeated until the required hash length is attained. All these processes are involved in sponge formation and are summarized as follows.

The sponge formation of keccak hash tree works on bits and it follows a principle that the sum of br and c must be to the maximum of 1600. The reason for this value fixation is to withstand bit attacks. The proposed work involves the output length of 256 bits. Let the data to be outsourced is represented as follows.

$$Data = \{Data_{bt1}, Data_{bt2}, Data_{bt3}, \ldots, Data_{btn}\}$$

All the operations performed by the keccak is carried out in byte based operations. Now all these divided blocks are arranged in tree format and the sponge formation is done. Now, all the bits are arranged at the bottom of the tree and the hash value is computed by the nodes in previous level. The hash value is calculated by the keccak hash technique and the hash tree is presented as follows.

Here, the root node of the tree contains the hash code of all the data being present in the tree. Instead of encrypting all the nodes involved in the tree, this work encrypts the root node alone with the help of ECC algorithm, which is explained as follows.

Keccak Sponge Formation

Input : Compressed data CD;

Output : Formed sponge;

Begin

ST[a,b]=0 for all (a,b) in 0 to 4;

BL = Divide CD into bytes;

// Absorption

For all BL_i in BL

Do

ST [a,b]=ST[a,b] $\oplus$ BL [a+5b]

$ST = keccak - f[br + c](ST)$

End do

End for;

//Squeeze

Q = Empty String;

Q=Q || ST[a,b] //String concatenation

ST=$Keccak - f\ [br + c](ST)$

Return Q;

End;

The ECC is a public key based algorithm, which could produce a key of minimal length [21]. The elliptical curve is established by the following equation.

$$b^2 = a^3 + xa + y \tag{1}$$

In the above equation, x and y are the integers. The ECC algorithm completely relies on the base point through which the elliptic curve and the maximal limit of prime number are framed. The base point of the curve is denoted as P and let R be the random number lies in between 1 and $m - 1$, which can be represented by

$$PK = R \times P \tag{2}$$

Where PK is the public key and private key is denoted by R. The potential of any cryptographic algorithm relies on the effectiveness of key generation. The hash code of the root node is encrypted by passing the data to be encrypted along with the private key, followed by which the public key is generated by the generator function (GF). The encrypted text is returned by means of a four digit randomly generated number RD in association with GF. The data encryption algorithm is as follows.

Root node encryption algorithm

Input: Code of root node (RC), private key R

Output: Encrypted code

Begin

Produce PK randomly by GF;

$PK = R \times GF$

Produce Ciphers $CP = RD \times GF$

Encrypt code by $EC = (RD \times PK) + (RC, P)$

Return EC;

End

The decryption process of the algorithm involves the encrypted data along with the secret key. The decryption is carried out with the help of P and the secret key. As all these operations are carried out in the client side, there is no need for any panic with respect to key security. This work weeds out the possibility of data sniffing by the CSP by employing three measures such

as data encryption on user side, data compression and effective data encryption. The proposed algorithm is presented as follows.

Proposed Time Conserving Data Security Algorithm

Input : Data to be outsourced

Output : Encrypted data

Begin

Compress the data by LZ77 compression algorithm;

Divide the data into several data blocks;

Pass the data blocks into keccak hash tree;

Repeat finding hash values for all the nodes;

Extract the hash code of root node;

Apply ECC algorithm on it;

Obtain encrypted hash code;

End;

This work does not rely on any third party to ensure data security and all the data security related operations are carried out on the user-side itself. This increases the reliability of the work and the data is compressed before being treated by encryption algorithm. This idea boosts up the memory and time conservation. The data encryption is carried out with the help of keccak hashing and the root node of the tree is again encrypted by ECC algorithm. Hence, an effective data storage security solution is provided at the cost of minimal time and space complexities, which in turn conserves more computational energy. The performance of the proposed work is analysed in the following section.

4. RESULTS AND DISCUSSION

The performance of the proposed work is analysed by simulating the proposed work in Java on a stand alone computer with 8 GB RAM and 1 TB hard disk. The performance of the proposed work is analysed in terms of avalanche effect, time consumption and space overhead. The performance of the proposed work is compared against standard algorithms such as RSA and Blowfish. Initially, the avalanche degree of the proposed work is compared against RSA and blowfish algorithms. The avalanche degree (AD) is computed by the following formula.

$$AD = \frac{T_{cb}(ED)}{T_b(ED)} \tag{3}$$

In the above equation, ET is the encrypted text, $T_{cb}(ET)$ is the total number of changed bits in the encrypted data and the $T_b(ED)$ are the total count of bits in the encrypted data. The avalanche degree determines the change in output data, when the input data is changed. Hence, the greater the AD, the better is the performance of the encryption algorithm and it lies in between 0 and 1. Different sizes of data are considered for analysis and the original data is modified in three positions one by one. The results attained by the proposed algorithm are compared against RSA and Blowfish algorithms.

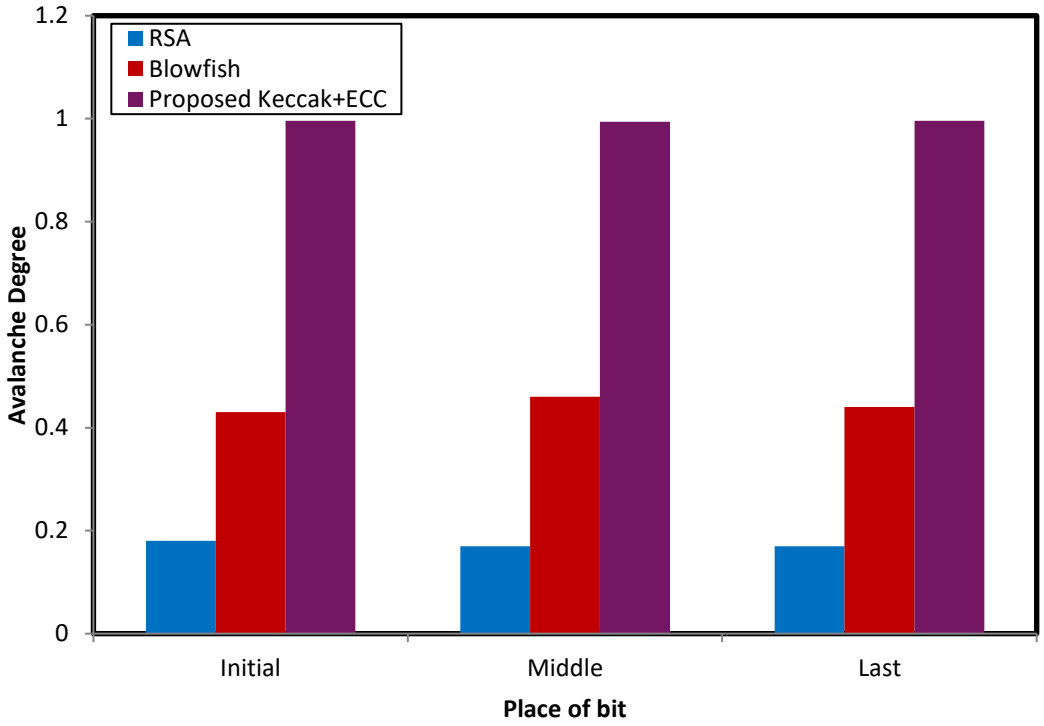

Figure 3 Avalanche effect analysis of the proposed algorithm

From the experimental results, it is proven that the avalanche degree of the proposed encryption algorithm is greater, which means that the proposed algorithm is stronger enough and provides better security. The following graph analyses the time consumption of the algorithms.

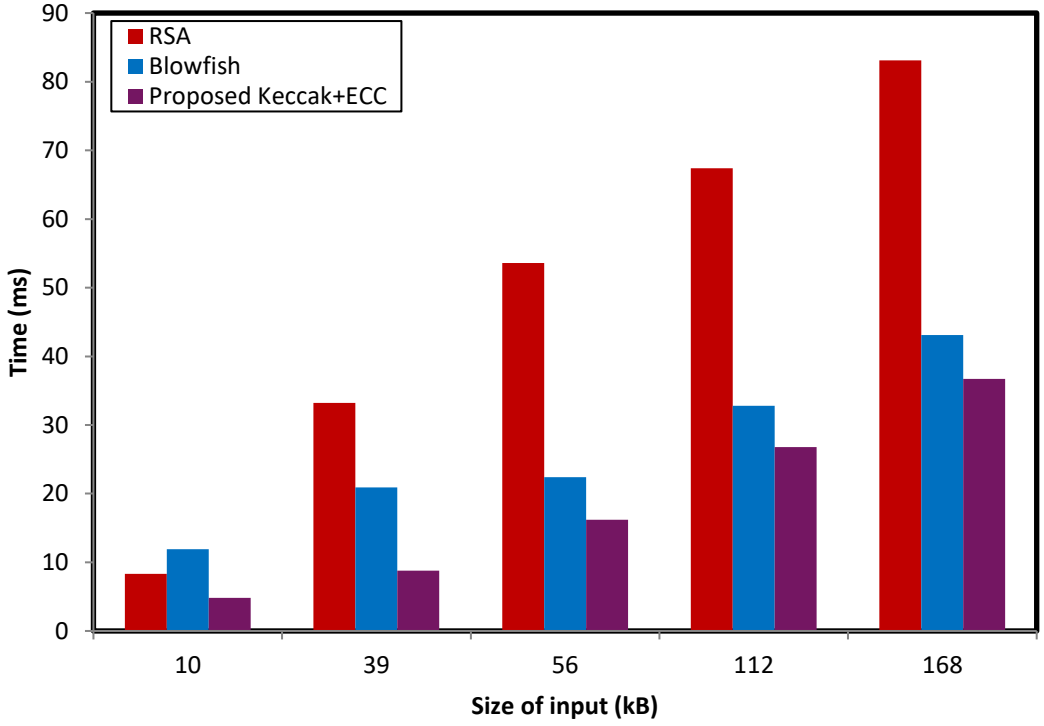

Figure 4 Average time consumption analysis of encryption algorithms

The average time consumption of the proposed and other comparative algorithms is presented in figure 4. Different text blocks with varying sizes are processed and the average time consumption is computed. The time consumption of RSA algorithm is greater than the blowfish and the proposed algorithm. The main reason for more time consumption of RSA is the increased key length, which increases the computational complexity and thereby increased time consumption. The following graph indicates the power of data compression, which is performed prior to the proposed encryption algorithm.

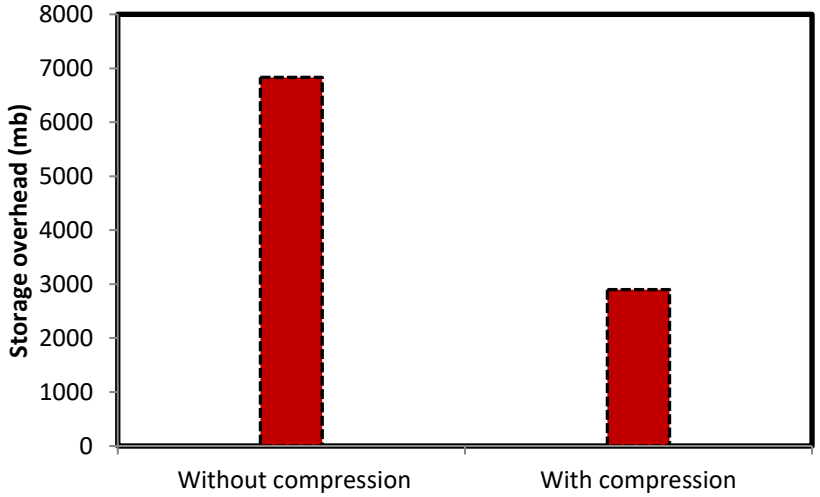

Figure 5 Storage overhead analysis with and without compression on the proposed algorithm

From the experimental analysis, it is clearly evident that the storage overhead of the proposed algorithm with compression is far minimal than the proposed work without compression. As stated earlier, the application of LZ77 compression algorithm reduces about 40 percent of the data. This compressed form of data helps in reducing the storage space, time and computational complexity as well. Hence, this section proves the security provisioning of the proposed algorithm by means of avalanche degree.

The time and space overheads are the main metrics, which mostly result in trade-off. However, due to the inclusion of the compression algorithm and the effective organization of data, the time consumption is considerably reduced. The space overhead is minimized by the application of compression algorithm, without affecting the performance of the algorithm. Hence, the proposed algorithm is good enough for better security, yet with minimal storage and time requirements. The following section concludes the article.

5. CONCLUSIONS

This article presents a time conserving secured cloud data storage solution, which is based on keccak hash tree and ECC algorithms. The goal of this work is to achieve minimal time consumption and memory overhead however, without compromising the security of the proposed algorithm. The volume of the input data is minimized by applying a lossless compression algorithm LZ77, followed by which the proposed algorithm is applied. The compressed data is dealt with the proposed combination of keccak and ECC algorithms. The performance of the proposed work is analysed with respect to Avalanche effect, time consumption and memory overhead. The attained results are compared with the analogous algorithms and the proposed algorithm proves better performance. In future, this work is planned to be extended to make it work with digital images.

REFERENCES

[1] Rimal, B. P., Choi, E., & Lumb, I. (2009, August). A taxonomy and survey of cloud computing systems. In 2009 Fifth International Joint Conference on INC, IMS and IDC (pp. 44-51). IEEE (2009).

[2] Zhou, M., Zhang, R., Xie, W., Qian, W., & Zhou, A. (2010, November). Security and privacy in cloud computing: A survey. In 2010 Sixth International Conference on Semantics, Knowledge and Grids (pp.105-112). IEEE. (2010)

[3] Zhou, L., Varadharajan, V., & Hitchens, M. (2015). Trust enhanced cryptographic role-based access control for secure cloud data storage. IEEE transactions on information forensics and security, 10(11), 2381-2395. (2015)

[4] Ali, M., Malik, S. U., & Khan, S. U. (2015). DaSCE: Data security for cloud environment with semi-trusted third party. IEEE Transactions on Cloud Computing, 5(4), 642-655. (2015)

[5] Tian, J., & Jing, X. (2019). A Lightweight Secure Auditing Scheme for Shared Data in Cloud Storage. IEEE Access. (2019)

[6] Al Hamid, H. A., Rahman, S. M. M., Hossain, M. S., Almogren, A., & Alamri, A. (2017). A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography. IEEE Access, 5, 22313-22328. (2017)

[7] Yu, J., Lu, P., Zhu, Y., Xue, G., & Li, M. (2013). Toward secure multikeyword top-k retrieval over encrypted cloud data. IEEE transactions on dependable and secure computing, 10(4), 239-250. (2013)

[8] Wang, Y., Wu, Q., Qin, B., Shi, W., Deng, R. H., & Hu, J. (2016). Identity-based data outsourcing with comprehensive auditing in clouds. IEEE transactions on information forensics and security, 12(4), 940-952. (2016)

[9] Varadharajan, V., & Tupakula, U. (2014). Security as a service model for cloud environment. IEEE Transactions on network and Service management, 11(1), 60-75. (2014)

[10] Wang, H., He, D., & Tang, S. (2016). Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud. IEEE Transactions on Information Forensics and Security, 11(6), 1165-1176. (2014)

[11] Zhu, H., Yuan, Y., Chen, Y., Zha, Y., Xi, W., Jia, B., & Xin, Y. (2019). A Secure and Efficient Data Integrity Verification Scheme for Cloud-IoT Based on Short Signature. IEEE Access, 7, 90036-90044. (2019)

[12] Gonzales, D., Kaplan, J. M., Saltzman, E., Winkelman, Z., & Woods, D. (2015). Cloud-trust—A security assessment model for infrastructure as a service (IaaS) clouds. IEEE Transactions on Cloud Computing, 5(3), 523-536. (2015)

[13] Liu, C., Ranjan, R., Yang, C., Zhang, X., Wang, L., & Chen, J. (2014). MuR-DPA: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud. IEEE Transactions on Computers, 64(9), 2609-2622. (2014)

[14] Wang, Q., Wang, C., Ren, K., Lou, W., & Li, J. (2010). Enabling public auditability and data dynamics for storage security in cloud computing. IEEE transactions on parallel and distributed systems, 22(5), 847-859. (2010)

[15] Shen, J., Zhou, T., He, D., Zhang, Y., Sun, X., & Xiang, Y. (2017). Block design-based key agreement for group data sharing in cloud computing. IEEE Transactions on Dependable and Secure Computing. (2017)

[16] T.Armerding, The 15 worst data security breaches of the 21st century, in : COS Security and Risk, Csoonline, 2012.

[17] G.M. Stevens, Data Security Breach Notification Laws, in, Congressional Research Service, 2012.

[18] Sayood, K.: 'Introduction to data compression' (Elsevier, USA, 2012)

[19] Blelloch, E.: 'Introduction to data compression'. Available at: https://www.cs.cmu.edu/guyb/realworld/compression.pdf, January 2013

[20] G .Bertoni, J. Daemen, M. Peeters, G. V. Assche, "The KECCAK SHA-3 Submission version 3," pp.1-14, 2011. Article (CrossRef Link).

[21] Hankerson, D., & Menezes, A. (2011). Elliptic curve cryptography (pp. 397-397). Springer US.

[22] Zhou, X., & Tang, X. (2011, August). Research and implementation of RSA algorithm for encryption and decryption. In Proceedings of 2011 6th International Forum on Strategic Technology (Vol. 2, pp. 1118-1121). IEEE.

[23] Mandal, P. C. (2012). Superiority of Blowfish algorithm. International Journal of Advanced Research in Computer Science and Software Engineering, 2(9).

[24] Payel Guria, A Quadruplex Data Encryption Scheme to Secure Cloud Data. International Journal of Computer Engineering and Technology, 9(5), 2018, pp. 70-77.

[25] K.Subramanian and M.Mohamed Sirajudeen, An Architectural Framework for Secure Cloud data Storage Management by using Orthogonal Handshaking Authentication Mechanism (OHSAM), International Journal of Mechanical Engineering and Technology, 9(3), 2018, pp. 791–799

[26] Toa Bi Irie Guy-Cedric and Suchithra R, Implementation of a Novel Algorithm Secure Double Encryption Standard (Sdes-384bit) to Prevent Side-Channel Attack for Cloud Data Center, International Journal of Mechanical Engineering and Technology, 9(9), 2018, pp. 1118–1126.

[27] A. Punitha and Dr. Nancy Jasmine Goldena, Resource Allocation Planner for Disaster Recovery (RAP-DR) Based on Preeminent Responsive Resource Allocation Using Parameter Selection of Virtual Machines or Cloud Data Server. International Journal of Computer Engineering and Technology, 9(5), 2018, pp. 96-108.