Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Multi Encryption Approach to Provide Security for Cloud Integrated Internet of Things

Rubika Walia
Research Scholar, Department of Computer Science & Engineering,
Maharishi Markandeshwar Engineering College, Mullana
MMICT&BM (A.P)
M.M. (Deemed to be University), Mullana
Email id:ahluwalia.rubika@gmail.com
Prachi Garg
Associate Professor, Department of Computer Science & Engineering,
Maharishi Markandeshwar Engineering College, Mullana
M. M. (Deemed to be University), Mullana
Ambala, Haryana, India
Email id:Prachigargji@gmail.com

Article Info
Page Number: 1538-1544
Publication Issue:
Vol. 71 No. 3s2 (2022)

Article History
Article Received: 28 April 2022
Revised: 15 May 2022
Accepted: 20 June 2022
Publication: 21 July 2022

Abstract
It is said that cloud computing is continually evolving, and cloud transition is becoming easier and easier. Cloud computing adoption is hampered by a number of security and privacy issues. An organizational entity, its resources, and its assets must be protected by a highly secure system. In this article, we propose and put into practice a framework that uses dual encryption, combining Chacha20 encryption on the client side with AES256 cryptographic technique on the server side for authorization. On an Oracle virtual machine running under Virtual Box 6.1.36 and using Python 3.7.4 on the Ubuntu 18.04 platform, simulations and analysis were carried out.

Keywords: Cloud Service Provider, Advance Encryption System, Secure Hash Algorithm, One Time Pad, Rivest Shamir-Adleman

1. Introduction

The flexibility, control and ease of use of cloud computing comes with a variety of problems. Particularly, it has been widely noted that security issues are the key barriers to cloud adoption. A company will be exposed to security breaches in the area of Vista even if it claims to offer world-class security but does not regularly update its security procedures. Because it offers a security mechanism for web-based applications, network security has gained significant relevance over the last few years. Protection is essential to reducing the impact of contemporary attacks.Attack mitigation and confidentiality concerns are the only significant characteristics of recent attacks [14, 15]. By using an Integrated Encryption technique, we want to introduce a novel method to address the security issues in this manuscript. The plan can be realized by developing a secure and fully automated information storage and transfer protocol between CSPs and users who have access to a variety of cloud storage volumes to store their information. Two phases can be used to implement the protocol:

1) Upload phase
2) Retrieve phase

SHA512 is the hashing method that is used to ensure data integrity on the client side. Before uploading the file to the cloud, a hash map is created on the server. The checksums are updated when the file has been downloaded. The assessment of checksums carried out at the metadata put at the conclusion of the files at the client end ensures the integrity of data [12, 13]. Double encryption and SHA512 were used to encode the metadata. The checksums that were recently computed and those that had previously been stored are compared.The checksums will not match if there is any breach. An authenticator for the ChaCha20 stream cipher uses cryptographic techniques. The key benefit of ChaCha20 is that it is built using the ARX (Addition, Rotation, and XOR) cryptography technique, which offers faster performance, less complexity, and prevents timing attacks. We make a lightweight algorithm (ChaCha20), which is characterized by high speed and low implementation resources, more secure by converting it into a hash function.

2. Related Work

To validate the data or file saved at a cloud data center, Deswarte et al. [1] designed an RSA-based hash algorithm. That plan allows clients to undertake numerous challenges on the same Meta data. Due to the high computational complexity of this approach, all of the blocks should be exponentiated into files at the server location.A technique to ensuring that data saved remotely across many sites is secure was described by Schwarz et al. in [2] Based on algebraic signature. This system stores data fingerprints and verifies them during the authentication procedure. Despite this advantage, the computational complexity of this approach is significant at the client and server sites have security-related problems as well.For static authentication of huge data files, Juels et alPOR.'s (Proofs of Retrievability) technique was presented [3]. In addition to special blocks called sentinels that are randomly enclosed into files for detection, this system includes spot checking and error correcting codes to secure data ownership and Retrievability. A hash tree is developed to keep the out of order of these sentinels' blocks. POR is primarily used for private information, not for public information. Because sentinels' nodes are present, dynamic updating is not allowed, thus there is no public data authentication in this approach.Takabi et al. [4] present acomprehensive cloud safetythreatvaluation. For the purpose of protecting virtual machines at cloud centres, Zhang et al. [5] present the cloud visor concept. By utilizing the Merkle hash tree methodology, the author of the cloud visor method has proposed a symmetric key based encryption strategy with AES -128 bit to secure data storage at cloud data centres. The AES -128 bit encryption strategy by side channel scheme and differential cryptanalysis of the MD5 Hash function are the limitations of this scheme.To guarantee the integrity of data kept at data centres and to maintain the security of sensitive data, Pardeshi et al. [6] suggested solution uses a Merkle hash tree with the sha-1 hash algorithm. This method, which is based on the SHA-1 hash function, is used by Merkle Hash Tree (MHT) to retain the hash values of all data blocks at cloud data centres. Nevertheless, despite these advantages, this technique suffers from side channel attacks and differential cryptanalysis of the SHA-1 hash function.A solution based on dynamic auditing protocol that can carry out operations on data in cloud

data centres was introduced by Erway, C. Chris, et al. [7] In this plan, a third party auditor checks the cloud data center's dynamic sensitive data for data change and integrity. For the purpose of verification, the suggested technique calls for the linear combining of data blocks by an outside auditor at the server location. However, because linear combinations of data blocks are needed for verification, it will reveal sensitive information to the auditor. A lightweight hash functions strategy is proposed by Hussain H et al. [11] that combine the ChaCha20 data encryption method with chaotic maps to keys generation. Using a chaos-based random number generation approach has strengthened the already-existing ChaCha20 encryption technology.

3. Proposed Work

We have successfully implemented a chained symmetric encryption using two distinct ciphers and two passwords. The objective was to offer a collection of brief scripts that may be used to execute symmetric encryption twice with reliable, long-lasting results. In order to implement Chach20 and AES256 encryptions, we used Oracle VM Virtual Box 6.1.36 and Ubuntu 18.04. For encryption techniques, Python 3.7.4 is utilized as a scripting language. We start with Chacha20 and move on to AES256.

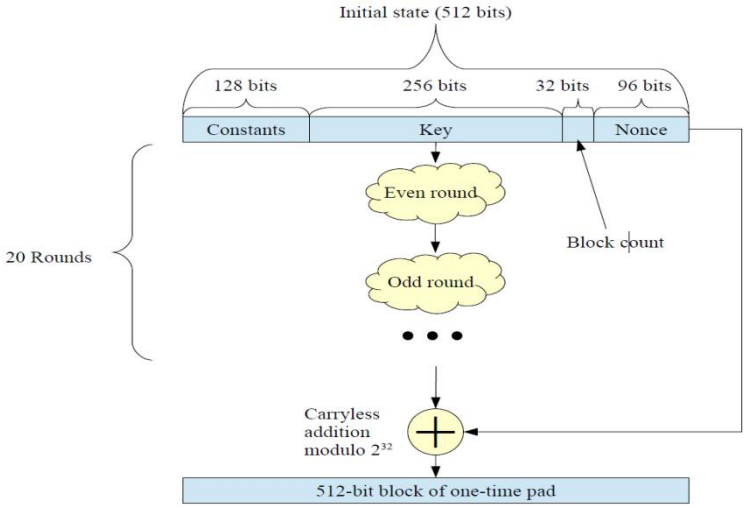

Figure 3.1: Chacha20 Algorithm

Step 1:

First the initial 512-bit state of ChaCha20 is formed. This state consists of a few fields as:

Constant (128 bits): It is initialized with the ASCII values of thecharacters of the following string.

Key (256 bits): The secret key.

Block count (32 bits): A counter that starts from 0 and is incrementedto generate the next chunks of OTP.

Nonce (96 bits): A unique number that can be changed togenerate a new OTP with the same key.

Step 2:

Second, the initialstate is transformed by 20 round functions of two types: even and odd. Each roundfunction converts its 512-bit input into a 512-bit output.

Step 3:

Finally, the initial stateis added to the result of the last round function. To do the summation stage, both operandsare viewed as arrays of 32-bit integer numbers with 16 elements. Thesummation modulo 2^{32} is done element-wise between the arrays. The result of thesummation is used to do encryption as follows:

$C_i = P_i$X-OR PAD_i

Where C_i is the i^{th} bit of the cipher text; P_i is the i^{th} bit of the plaintext; and PAD_i is the i^{th} bit of ChaCha20 result.

Step 4:

To generate the next 512-bits of OTP, the block count field of the initialstate is incremented and the process repeats (Step 1-Step 3).

4. Results and Discussion

4.1 Encryption

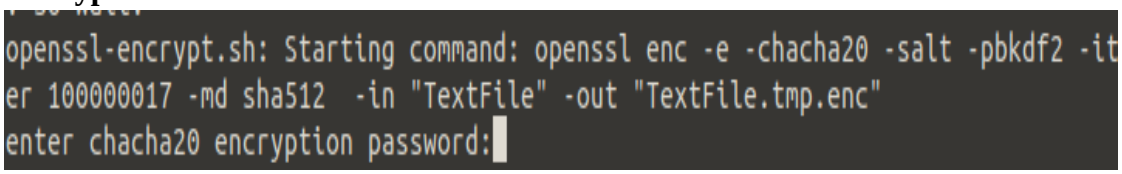

Figure 4.1: Client side Encryption

As shown in figure 4.1 TextFile is given as input and TextFile.tmp.enc is the encrypted file generated by Chach20 encryption.

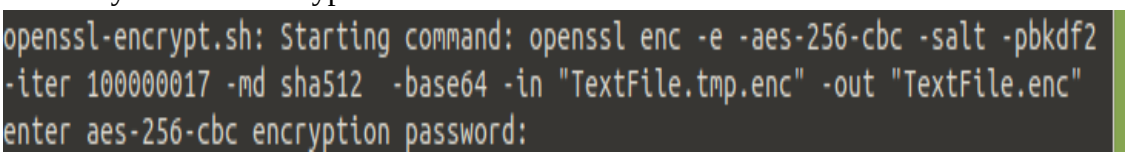

Figure 4.2: Server side Encryption

As shown in figure 4.2 TextFile.tmp.enc is given as input and TextFile.enc is the encrypted file generated by AES256 encryption.

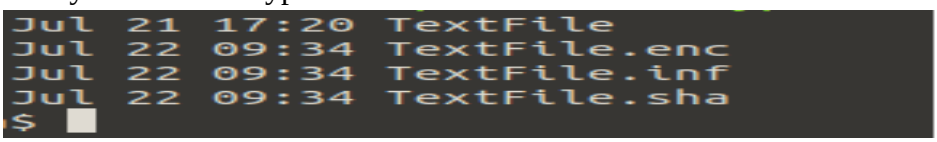

Figure 4.3: Files stored at Server

As shown in figure 4.3 encrypted file TextFile.enc and hash map of encrypted file TextFile.sha generated by Secure Hash algorithm 512 is stored at server.

4.2 Decryption

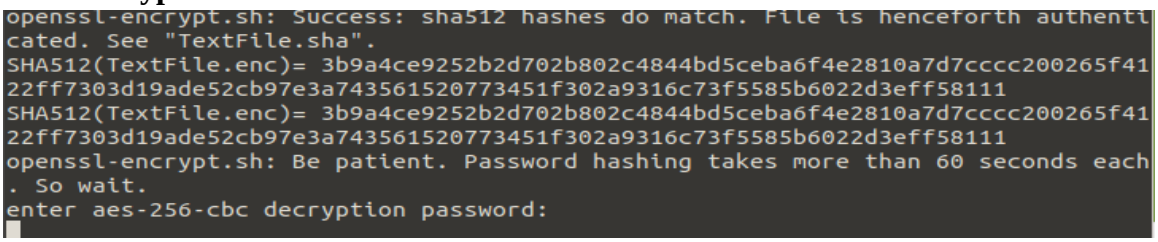

Figure 4.4: Server side Decryption

As shown in figure Sha 512 hash is checked and if it matches then it asks for AES 256 password for decryption.

Figure 4.5: Client side Decryption

As shown in figure Sha 512 hash is checked and if it matches then it asks for Chacha 20 password for decryption.

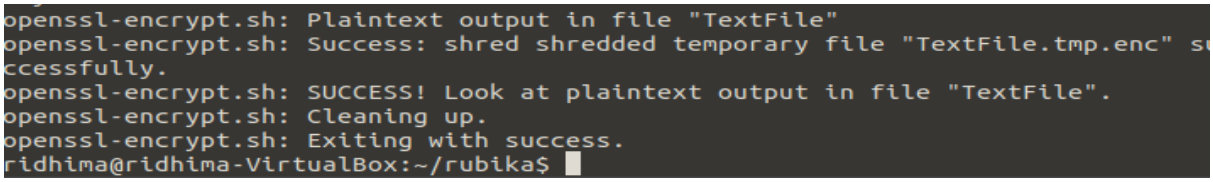

Figure 4.6: Plaintext File at Client side

As shown in figure 4.6 Plaintext output file TextFile recovered at client side.

Conclusion

In order to develop and expand security tools for cloud computing, we describe a fresh architectural concept for giving protection in this article. In this paper, we introduced two-tier security architecture, one on the client and one on the server. The article supported security requirements including Integrity and Authorization and provided an explanation of security domination. In order to lessen the difficulties of enforcing the protection of data for information security in the cloud, we have projected a secure-computation protocol through this document. To the best of our knowledge, it is the most significant endeavor that jointly makes use of two encryption approaches for data storage security.

References

1. Deswarte, Yves, Jean-Jacques Quisquater, and AydaSaïdane, "Remote integrity checking", Proceedings ofIICIS 140 (2004): pp. 1-11.
2. Schwarz, Thomas SJ, and Ethan L. Miller, "Store, forget, andcheck: Using algebraic signatures to check remotely administered storage", Distributed Computing Systems, 2006, ICDCS 2006, 26th IEEE International Conference on IEEE, 2006, DOI: 10.1109/ICDCS, 2006.80

3. Juels, Ari, and Burton S. Kaliski Jr, "PORs: Proofs of Retrievability for large files", Proceedings of the 14th ACM conference on Computer and communications security, Acm, 2007, DOI: 10.1109/DSN, 2006.56
4. Deepak Mathur, N. K. V. . (2022). Analysis & Prediction of Road Accident Data for NH-19/44. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(2), 13–33. https://doi.org/10.17762/ijrmee.v9i2.366
5. Takabi, Hassan, James BD Joshi, and Gail-JoonAhn, "Security and privacy challenges in cloud computingenvironments", IEEE Security & Privacy 6 (2010): 24-31, DOI: 10.1109/MSP.2010.186
6. Zhang, Fengzhe, et al., "CloudVisor: retrofitting protection ofvirtual machines in multi-tenant cloud with nestedvirtualization, "Proceedings of the Twenty-Third ACMSymposium on Operating Systems Principles. ACM, 2011, DOI:10.1145/2043556.2043576
7. Pardeshi, Poonam M., and Deepali R. Borade, "EnhancingData Dynamics and Storage Security for Cloud Computingusing Merkle Hash Tree and AES Algorithms", InternationalJournal of Computer Applications 98.21 (2014). DOI: 10.5120/17304-7734
8. Erway, C. Chris, et al, "Dynamic provable data possession", ACM Transactions on Information and System Security (TISSEC) 17.4 (2015): 15. DOI: 10.1145/2699909
9. Feng Y, Liu Y, Zhao G, Xia M , "An improved AES encryption algorithm based on theHénon and Chebyshev Chaotic Map. Int J Simul SystSciTechnol 17(48):25.1–25.8. https://doi.org/10.5013/ijssst.a.17.48.25 15, 2016
10. De SF, Schauer A, Sigl G,"ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications",Autom Test Eur 692–697. https://doi.org/10.23919/date.2017.792 7078, 2017
11. Bulla, P. . "Traffic Sign Detection and Recognition Based on Convolutional Neural Network". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 4, Apr. 2022, pp. 43-53, doi:10.17762/ijritcc.v10i4.5533.
12. McLaren P, Buchanan WJ, Russell G, Tan Z, "Deriving ChaCha20 key streams fromtargeted memory analysis. J InfSecurAppl 48:102372. https://doi.org/10.1016/j.jisa.2019.102372, 2019
13. Hussain H. Alyas and Alharith A. Abdullah, "Enhancement the ChaCha20 EncryptionAlgorithm Based on Chaotic Maps", https://doi.org/10.1007/978-981-16-0666-3_10, Springer Nature Singapore Pte Ltd. 2021
14. A. Gupta, T. Gulati and A. K. Bindal, "WSN based IoT applications: A Review," 2022 10th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-22), 2022, pp. 1-6, doi: 10.1109/ICETET-SIP-2254415.2022.9791495.
15. Varun, B. N. ., S. . Vasavi, and S. . Basu. "Python Implementation of Intelligent System for Quality Control of Argo Floats Using Alpha Convex Hull". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 5, May 2022, pp. 60-64, doi:10.17762/ijritcc.v10i5.5554.
16. Oberoi, N., Sachdeva, S., Garg, P., Walia, R "Dynamic extraction and analytics of big data from cloud and social media integrated platforms" https:// DOI:10.37418/amsj.9.6.48 Advances in Mathematics: Scientific Journal, 2020, 9(6), pp. 3703–3711

17. Walia, R., Garg, P.Cryptography: Analysis of SYN and UDP Attacks using wire shark http://10.1109/ICCMST54943.2021.00039 Publisher: IEEE ,2021.
18. Garg, D. K. . (2022). Understanding the Purpose of Object Detection, Models to Detect Objects, Application Use and Benefits. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(2), 01–04. https://doi.org/10.17762/ijfrcsce.v8i2.2066
19. V. Bhatt and A. K. Bindal, "Smart Hardware Development under Industrial IOT (IIOT) 4.0: A Survey Report," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 262-265, doi: 10.1109/ISPCC53510.2021.9609399.
20. M. S. Kiran and P. Yunusova, “Tree-Seed Programming for Modelling of Turkey Electricity Energy Demand”, Int J Intell Syst Appl Eng, vol. 10, no. 1, pp. 142–152, Mar. 2022.