DOI: http://dx.doi.org/10.26483/ijarcs.v14i3.7007

ISSN No. 0976-5697

Volume 14, No. 3, May-June 2023

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

AN ENHANCED SOFTWARE AS A SERVICE (SAAS) ARCHITECTURAL MODEL FOR CLOUD BASED SECURITY USING HYBRID SYMMETRIC ALGORITHM

Afiesimama Dimabo Joshua & Onate Egerton Taylor
Department of Computer Science
Rivers State University of Science Technology
Port Harcourt, Rivers State, Nigeria.

Abstract: Protecting cloud data from security breaches, preventing and ensuring safe cloud data is concealed from unauthorized access or infiltration by unwanted users are all part of cloud infrastructure security. This proposed technique is based on ensuring data confidentiality by adding extra-layer of security from the client side, developing a framework that implements the AES 256-bit key and Fernet 128-bits algorithms, which encrypt and decrypt files via random algorithm selection during uploads and downloads. When a file is uploaded to the cloud, the decryption key is recorded in the local server's file log. The software use randomized algorithm selection to provide security to ensure that an attacker having a prior link cannot reset the user account since the attacker cannot guess the token. Furthermore, invalidating this token ensures that the link cannot be used more than once if it was previously logged anywhere. To propose and verify the efficiency of the model the results generated(in quantitative values) are frequency (monobit) testing, with a minimum bit size of 64, given the P-values derived from the above study (i.e., 0.8838, 0.6187, 0.3768, 0.2817, 0.5843, 0.4167). The findings reveal throughput for encryption of 0.91531 and for decryption of 0.4854, space complexity of 0.4854, and entropy of 7.942616667.This research work studies the design of a robust, adaptable, and non-deterministic SaaS application that can be secure for enterprises and subsequent research studies will be conducted in the development.

Keywords: extra-layer. Monobit, non-deterministic. AES 256-bit key and Fernet,Data Confidentiality, Throughput, entropy

1. INTRODUCTION

SaaS has the best chance of succeeding among cloud options. After all, as more businesses adopt SaaS solutions for a wide range of business processes, the entire growth of the SaaS market will stay consistent over the next several years, extending beyond the early SaaS fields of core engineering and commercial applications. (Kidd, 2020).
There are several challenges to its adoption, which has become a serious issue. The public cloud, for example, poses a significant danger since data from several organizations kept in the same location, adjacent each other.

Clients in SaaS systems rent and obtain software capabilities from internet providers. As a result, the cloud client has only one layer, the User Layer, which often contains a web browser and/or the ability to access the providers' online services. This comprises data integration and display, for example. Distribution Layer, Presentation Layer, Business Service Layer, Application Service Layer, Data AccessLayer, Data Storage Layer, and Supporting Service Layer are typical levels of SaaS providers (Tekinerdogan, 2013).

Cloud computing's confidentiality and security issues are one of the major roadblocks to its adoption, because data in cloud storage is not directly in the customer's control, there is a risk of data authenticity, classification, and other issues.

The basic CIA (confidentiality, integrity, and availability) paradigm, which in the cloud might be read as follows, may be used to classify security issues.

Confidentiality is used to scramble data such that it is incomprehensible to everybody but those who are allowed to see it. To provide privacy, cryptographic method and mode of operation must be created and implemented in such a way that an unauthorized party is unable to discover the keys associated with the encryption.

Data integrity means that there has been no addition, deletion, or substitution of material. Digital signatures, also known as message authentication codes, are cryptographic tools that may be used to identify both unintended changes that may occur due to equipment failure or transmission difficulties, as well as deliberate changes that may be done by an attacker.

Availability demands the ability to describe and check that providers meet the requirements expressed in service level agreements (SLAs) established between information owners / customers and providers. The concerns to be addressed, the obstacles to be met, and the specific guarantees to be provided to ensure the fulfillment of the security qualities listed above are determined by the features of the personnel.

Cloud computing weaknesses include data lock-in secrecy, data availability and auditability, data transmission bottlenecks, performance unpredictability, scalable storage, vulnerabilities, and software licensing. The expense of cloud computing is tied to security issues. Cloud computing security issues may be separated into three categories: security categories, security in service, and security dimensions. The following concerns are addressed by the research work:

Inadequate design and algorithms to secure file streaming

Over time, a cryptography key pattern may be predicted using a single method.

Entrusting entire security to a third party when data are costly to compromise

Uncertainty over trustworthy cloud service certification across cloud service providers in the geo-locational repository.

I. Related literature review

Ahsan *et al*. (2018) investigated the difficulty of searchable encryption in a data sharing setting, as well as its vulnerability to keyword guessing and statistical assaults, in their article. They introduced RanSCrypt, a revolutionary searchable encryption technique that incorporates randomization in a modified term. At the heart of the proposed method. The work established two random keyword constructs and allow the public to verify their resemblance without exposing anything about the two keywords. [2]

Sweta and Jayasimha(2020) built a model that contained unique and previously tracked down patterns and signatures that identify that a website is not a genuine website, as well as all of the website catalogs that are being recognized as phishing websites. This method discovers and monitors URLs that turn out to be phishing websites. Those websites are blocked, and a warning will be sent to the user's email address. [10]

Abroshan (2021) improved security and performance, they merged Blowfish and elliptic curve approaches. This has little effect on speed while calculation speed is crucial in cloud computing, they revealed that complicated cryptographic techniques are unsuccessful. Future study should put the technique to the test in a variety of infrastructures, including cloud computing with large datasets. [1]

In Eswari and Manikandan (2016) an algorithm called Key Factor Authentication and Access Control (KFAAC) has been proposed. KFAAC verifies clients using the Key Encryption Scheme to verify their uniqueness. It ensures the privacy for both the service provider and the customer, as well as the cloud environment. [6]

A randomization encryption encrypts a data by selecting a cryptographic hash at random from a collection of ciphertexts. Such procedures may achieve better cryptography than their deterministic contemporaries. They increase the size of the message space and eliminate the risk of chosen plaintext attacks (Rivest& Sherman, 1983). [8]

According to Aleem *et al*. (2021), a quantitative survey was designed and carried out to discover critical architecture criteria for a better and more effective SaaS service. The findings of an empirical analysis show that vendors and developers must address critical architecture considerations in order to compete in today's market. It focuses primarily on elements found through a rigorous literature review. [3]

II. Cloud Computing Architecture

The two basic components of cloud computing architecture are the frontend and backend. The frontend functions as a user, connecting to the backend via an internet. The front-end is accessible to authorized users in the cloud computing architecture. Requests are routed from the frontend to the backend via the middleware. The backend protects the data and processes information from the frontend. The backend accounts for a bigger share of the entire cloud computing architecture: (Borah, 2021). [5]

2. METHODOLOGY

I. The Proposed System Analysis

The proposed system will able to encrypt file data and decrypt file data via a Flask-based web app. Before a file is uploaded the system will randomly select one of the algorithms to encrypt the file and generate a decrypting key for the recipient. The web app displays a form in which the user must select the file to be encrypted. The file`s type is automatically determined, and the encryption algorithm is applied to the file type before it is posted to the public cloud storage. This ensures that the user has sole access to their public cloud storage on login. Users are allowed to create a cryptographic key that is unique to them or a generated key can be used. The symmetric encryption algorithm is used to encrypt the file using this key.

II. Algorithm Implementation

To avoid message tampering, the Fernet Algorithm is a standard protocol with ready implementations in a number of computer languages. It combines AES CBC encryption with version information, a timestamp, and an HMAC signature. The AES encryption method is a symmetric block cipher algorithm that uses keys ranging from 128 to 256 bits in size. A symmetric algorithm encrypts and decrypts plain text and cipher text using the same cryptographic keys.

Proposed System

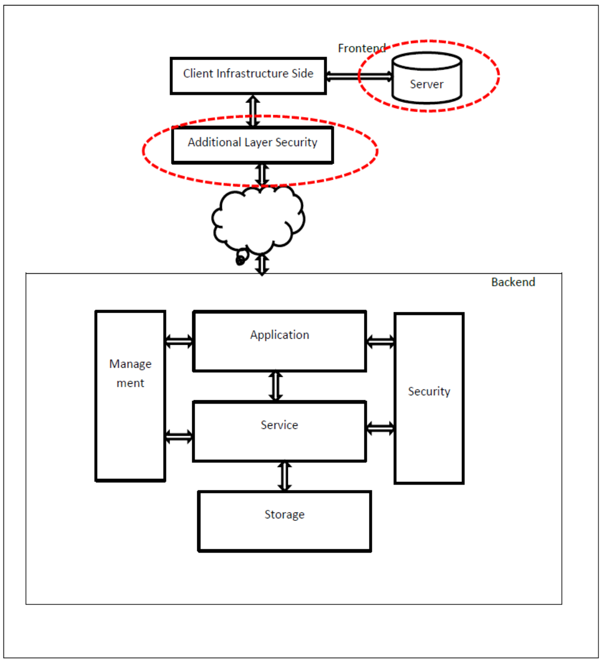

Figure 1: Proposed System Architecture

The Block Diagram of Random Selection of AES &Fernet algorithms can be explained using diagram below.

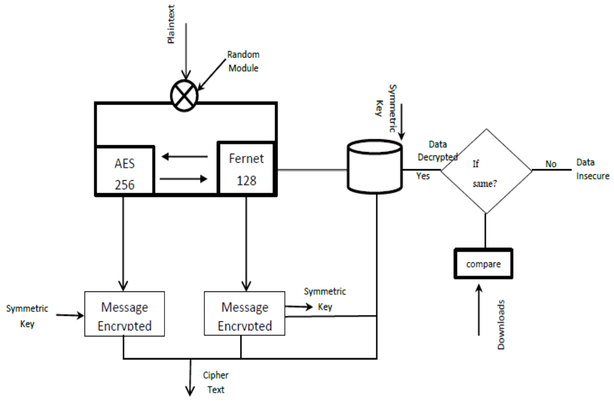

Figure 2: Security layer

Figure 2 depict how the process of encryption and decryption is carried out in the proposed architecture.The proposed architecture adds an extra layer of security in by integrating client-side encryption straight into the applications layer of the OSI model. This security layer is called an hybrid symmetric encryption using randomized selection of algorithm, This hybrid encryption consist of AES algorithm 256 bits and Fernet Algorithm 128 bits which is randomly select an encryption algorithm to encrypt files before transmission via the web and decryption when downloading from the web. Where the encryption keys for the hybrid is stored in the local server which mapped ciphertext downloading from the web. The server is secure with a USB token to assure confidentiality.

III. How the Algorithm Works

Data/Files is uploaded from our local machine to the application, the application activates random selection between two (Fernet and AES) algorithms. One algorithm between the two is picked and that algorithm is used to encrypt the file. This Encrypted file is then stored on the cloud data base. When a file on the other is to be downloaded through the interface from the cloud, the data is decrypted using the encryption key used in encrypting the data which was stored in the local server.

IV. Proposed Model Implementation
AES and Fernet Encrypting Simulation

To test the system that will generate data set for the analysis, Google colab as IDE and python programming language.Several parameters factors will be used to assess the key's strength and the cryptography's unpredictability.

By using input as text for the dataset for measuring cryptographic key performance, the study adopt AES 256-bit, 14 rounds, and Fernet 128-bits algorithms.

3. RESULT/ DISCUSSION

I. Evaluation of Parameters

We test the ciphertext generated by the suggested technique for randomness. Only binary sequences are suitable for these tests. In the case of statistical testing of cryptographic algorithms, samples are taken from the algorithm's output including plain text for evaluation, the ciphertext is translated into binary. The table 4.1 display the data set that is generated from the simulation.

Table 1: Encryption Simulation Result

Plaintext Files Name	Cryptography keys
testdata	gMn7CB5V
testdata2	lJdAn0E=
test_data3	hb6OXVE=
testdata	7c1RMo30wADiSVfdvioKO6YLmHyAu64xoCvjptkKTcU=
testdata2	lVzWu1tKpvJk7w04DPRVgpThZtVg7W4mX5X6I6fnMOc=
testdat3	nK2Ims8znzRyCCDqMCzIh5KZ0UAz_Xm29RtsDlXyQIg=

Table 2: Frequency (Monobit) Test P-value

Frequency (Monobit) test P-value

Plaintext	Proposed		Existing
Yello	0.8838	0.3917	0.4375
Hello	0.6187	0.5834	0.3281
Mello	0.3768	0.4167	0.1857

Table 3: Cryptographic Algorithm Evaluation with Different File Text

Time Evaluation for Algorithms in (ms)

File size in btye	Proposed System Encryption(ms)	Proposed System Decryption(ms)	Existing System Encryption(ms)	Existing System Decryption(ms)
File.txt1(7kb)	7.6445	8.4804	7.4567	8.7643
File.txt2(5kb)	5.4639	6.3431	5.4759	6.4785
File.txt3(5kb)	5.4639	6.3432	5.4759	6.4786

Table 4: Throughput for Proposed and Existing system

Total Average Time Evaluation and Throughput of the Proposed and Existing Systems

	PROPOSED SYSTEM(ms)	EXISTING SYSTEM
Total Average Time Encryption	6.1908	6.1305
Total Average Time decryption	7.0556	7.3405
Throughput(kb/sec)	0.91531	0.9244
Decryption Throughput(kb/sec)	0.8032	0.7826

Table 5: Entropy for Proposed System

File Name	Proposed	
	Plaintext	Ciphertext
Testdata	5.3659	7.8996
testdata2	5.3499	7.8899
test_data3	5.3633	7.8999
Testdata	5.3875	7.9898
testdata2	5.3875	7.9876
testdat3	5.3865	7.9889
Average	5.373433333	7.94261667

Table 6 Performance Evaluation Result

Plaintext	Bits sequence	P-value	Throughput	Entropy
testdata	01100111 01001101 01101110 00110111 01000011 01000010 00110101 01010110	0.8838	Encryption 0.91531	7.8996
testdata2	01101100 01001010 01100100 01000001 01101110 00110000 01000101 00111101	0.6187	Throughput Decryption	7.8899
testdata3	01101000 01100010 00110110 01001111 01011000 01010110 01000101 00111101	0.1768	0.8032	7.8999
testdata4	00110111 01100011 00110001 01010010 01001101 01101111 00110011 00110000 01110111 01000001 01000100 01101001 01010011 01010110 01100110 01100100 01110110 01101001 01101111 01001011 01001111 00110110 01011001 01001100 01101101 01001000 01111001 01000001 01110101 00110110 00110100 01111000 01101111 01000011 01110110 01101010 01110000 01110100 01101011 01001011 01010100 01100011 01010101 00111101	0.2917		7.9898
testdata5	01101100 01010110 01111010 01010111 01110101 00110001 01110100 01001011 01110000 01110110 01001010 01101011 00110111 01110111 00110000 00110100 01000100 01010000 01010010 01010110 01100111 01110000 01010100 01101000 01011010 01110100 01010110 01100111 00110111 01010111 00110100 01101101 01011000 00110101 01011000 00110110 01001001 00110110 01100110 01101110 01001101 01001111 01100011 00111101	0.0843		7.9876
testdata6	01101110 01001011 00110010 01001001 01101101 01110011 00111000 01111010 01101110 01111010 01010010 01111001 01000011 01000011 01000100 01110001 01001101 01000011 01111010 01001001 01101000 00110101 01001011 01011010 00110000 01010101 01000001 01111010 01011111 01011000 01101101 00110010 00111001 01010010 01110100 01110011 01000100 01101100 01011000 01111001 01010001 01001001 01100111 00111101	0.4167		7.9889

We proposed a hybrid model non-deterministic pattern that randomly select algorithm for encrypting and decrypting files through uploading and downloading either from the local server to the cloud or from the cloud to their local server. The goal is to mitigate the risk of predicting the pattern of encryption and decryption. The proposed framework that was written in Python programming language. Three datasets were primary data created for study as a result of the simulation procedure. Testdata.txt, testdata txt1, testdata txt2, and testdata txt3 are the names of the files. Table 4.1 shows how the input data was encrypted during the saving process, which resulted in the generation of an encryption key.

The presented result demonstrates the advantage of the proposed algorithm over existing algorithms in terms of execution time, with little variation in encryption and decryption throughput when compared to values 0.9153 and 0.9244 in table 4, which displayed the trend of various file sizes in relation to encryption and decryption time. The first input value was 7 KB, as shown in the table 3, and when the KB size was lowered, the trend shifted downward. This might be related to the efficiency of the cipher AES-XBC-SHA256, which reduces the hash value that will check encryption keys and algorithms before decrypting the file for usage.

Accepting Ho if the P-value is then deemed random, which Ho accepts. The null hypothesis is accepted if the P-value is, suggesting that the sequence seems to be random. The null hypothesis is rejected if P-value, suggesting that the sequence is not random. The parameter represents the likelihood of a Type I error. Usually, is chosen in the [0.01] range. Given the P-values derived from the above study (i.e., 0.8838, 0.6187, 0.3768, 0.2817, 0.5843, 0.4167). The moderate level of P-values, which varies between 0.8 and 0.2, has good dependability and can be used. . In the Frequency (Monobit) test, the statistics value, or level of significance is 0.01. Conclusively is that the sequence is random.

4. CONCLUSION

In the Open system model, the security layer is tied to the application via the network layer. This allows the client to have his own security to avoid compromise from cryptanalysis or network provider outage. The security layer employs a hybrid encryption approach in which encryption keys are chosen at random to encrypt files uploaded to the cloud.

Mathematical characteristics are used to create all of the keys. The more bits in a key, the more calculation time it takes to encrypt the data. Fernet Algorithm uses CBC AES, is a symmetric technique that ensures that the encrypted communication cannot be modified or read without the key.

The random module contains a number of methods for selecting elements at random. The chosen () function in the random module returns a single random item from a list, tuples, or strings when used. We use pip install cryptography to install the library, then import the libraries for AES 256 and Fernet 128.

A simulation was done utilizing the primary dataset used to evaluate the study. A frequency (monobit) test was done, and the estimated value, which varies between 0.8 and 0.2, was met. When compared to the indicated system, the study produced an excellent outcome with a small deviation in throughput. The recommended method has a very high ciphertext entropy of 7.942616667..

The hybrid approach outperforms the existing technique in terms of frequency' p-value and entropy, the values produced by both methods are almost comparable. The algorithm satisfies the statistical tests, it does will be resistant to all potential attacks.

Randomization is one method for 'building' security into your code, making it resilient, and making your applications non-deterministic. This approach is selection of algorithm, each file have a different encryption keys. There are many libraries that provide randomization it can be applied in most high level language. And would work with any country cloud apps in the market, regardless of their capacity to collaborate.

This study might be taken in a new direction in the future by defining new trajectories in confidentiality of data.

ACKNOWLEDGEMENT

The Authors would like to thank the reviewers for their insightful comments.

REFERENCES

[1]. Abroshan, H. (2021). A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms. International Journal of Advanced Computer Science and Applications, 12(6), 12-23.

[2]. Ahsan, M., Idna B. I. M., Bin A. W., Ali, A. I., Khan, N., Al-Garwi, M. & Rahman, A. (2018). Searching on Encrypted E-Data Using Random Searchable Encryption (RanSCrypt) Scheme. Symmetry, 10(5), 99-161

[3]. Aleem, S., Ahmed, F., Batool, R. &Khattak, A. (2021). Empirical Investigation of Key Factors for SaaS Architecture. Institute of Electrical and Electronics Engineers Transactions on Cloud Computing, 9(3), 1037–1049.

[4]. Anirudk, V. K. (2019). What Is Cloud Computing Architecture: Front-End & Back-End Explained. Toolbox.

[5]. Borah, R. (2021). Cloud Computing Architecture: What is Front End and Back End? www.clariontech.com.

[6]. Eswari, S. &Manikandan, S. (2016). Key Factor Authentication and Access Control for Accessing Cloud Computing Services. International Journal of Computer Applications, 151(4), 35–41.

[7]. Kidd, C. (2020). SaaS in 2020: Growth Trends & Statistics. BMC Blogs.

[8]. Rivest, R. L. & Sherman, A.T. (1983). Randomized Encryption Techniques. Springer Link. Webeduclick. (2021, October 22). Types of Ciphers in Cryptography. Webeduclick.Com.

[9]. Paschal Uchenna, C. (2018). Security of Cloud Virtualized Resource on a SaaS Encryption Solution. Science Journal of Energy Engineering, 6(1), 8.

[10]. Sweta, M. &Jayasimha, S. R. (2020). Detection of Phishing Attacks using Content Analysis in the Cloud. International Journal of Recent Technology and Engineering,9(1), 2277-3878.