CAHIERS MAGELLANES-NS
Volume 06 Issue 2
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

BABY GIANT PUBLIC CRYPTOGRAPHY AND GOLDWASSER HOMOMORPHIC CUCKOO HASH SECURED DATA STORAGE IN FOG ENVIRONMENT

T. RAJKUMARAN
Research Scholar, PG & Research Department of Computer Science, Gobi Arts & Science College, Gobichettipalayam, Tamil Nadu, India

Dr. T. K. SHANMUGAM
Research Supervisor, Associate Professor & Head Department of Computer Science, Gobi Arts & Science College, Gobichettipalayam, Tamil Nadu, India

Abstract

With increasing scale of big data in IoT application, fog computing is a contemporary computing prototype that is broadening Cloud Computing (CC) in connection with the edge of network in the field. Also there is a huge number of storage resources positioned on the edge of the network to manifest a geographical distributed storage as far as fog computing system (FCS) is concerned. It is utilized in storing big data acquired by fog computing nodes and to minimize the costs involved in management for stimulating data to the cloud. However, the fog node storage at the network edge features several types of attacks. Because of this, applying a high security cloud storage service is considered to be a must in today's scenario. In this work, a method called, Baby Giant Public Cryptography and Goldwasser Homomorphic Cuckoo Hash (BGPC-GHCH) secured data storage with higher data confidentiality rate in fog environment is proposed. The BGPC-GHCHmethod performed key generation, encryption and decryption for increasing the security performance. The proposed method is split into three steps namely, user registration, key generation and encryption/decryption. Initially, in BGPC-GHCH method, user registration is performed where the details of the corresponding users are registered to the fog server. Upon successful completion of user registration for registered user, in the key generation step, fog server generates public key and private key using Baby Giant Step Public

Cryptography-based Key Generation. After key pair (i.e., public key and private key) generation, user encrypts the data with public key and transmits the encrypted data to the fog server for secured data storage employing Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing. Cuckoo Hash Function is used by fog server for encrypted user data storage with their hash value. The stored information gets retrieved by user through decryption. This in turn, the data gets stored in more secured manner using BGPC-GHCHmethod. Experimental evaluation is carried out on factors such as encryption time, data confidentiality rate and data integrity with respect to number of data and number of fog users.

Keywords: Cloud Computing, Fog, Baby Giant, Public Cryptography, Goldwasser Homomorphic Cuckoo Hash

1. Introduction

Fog computing is considered to be one of the types of cloud architecture that is positioned in between data and cloud. The fog computing design is contemplated as the distributed design in which

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

fog nodes accumulate data packets from numerous edge devices. This empowers data to be transmitted in an expeditious manner, augmenting boosting comprehensive network performance and efficiency. With the aid of fog computing, both data management and data storage can be significantly rationalized. Contemporary decades have witnessed considerable evolution and application of digital technology. This technological evolution give rises to an extensive amount of sensitive data that must be preserved. Hence security of confidential data is one of the major issues as far as fog computing is concerned. Owing to this, attaining a reliable security level in the fog computing environment is decisive.

Chaotic-Map Based Encryption was introduced in [1] for preserving the privacy of 3D point and mesh fog data. The data coordinates involved in the fog design were transformed by way of sequence initiated by chaotic behavior. In addition the bifurcation analysis was executed with enhanced scope of map. Finally, the chaotic system was evaluated by means of two distinct functions namely, Lyapunov exponent and approximate entropy, therefore reducing the attack considerably. Nevertheless, the complexity involved in the computation process was not reduced by this designated designed approach.

Continuous Delivery Continuous Verifiability (CD/CV) method was designed in [2] for consistent data IoT flows in edge–fog–cloud. Initially, the CD model was outlined based on the extraction, transformation and load (ETL) mechanism for both performing verification of applications and execution of the same in edge–fog–cloud infrastructure. Moreover, the designed method generated sequences of execution and integrity of digital assets. In addition, the CV model metamorphosed ETL and DAG into business model in private blockchain for performing registration of transaction in a transparent manner in edge–fog–cloud workflows, therefore improving the attack rate considerably. However, the data confidentiality was not improved by CD/CV method.

A lightweight authentication protocol was designed in [3] to produce proper access in devices. Here, the authentication protocol was introduced with focus on scalability, movement, user registration and device registration. Moreover, a three-layered model was used along with the cloud, fog, and edge devices. Also to ensure attack detection rate, the pre-existing cipher suites were measured employing post-quantum cryptography. Finally, a fail-safe mechanism was employed where without the intervention of human efforts IoT devices organized services in a significant manner. However, the authentication time was not reduced by designed protocol.

The fog-based IoT network was constructed in [4] by creating platform that secure endpoints via public-key encryption. Moreover, the servers were allowed to mask data packets distributed within network. Also, permissioned blockchain was employed with the purpose of tracking encryption process with proposed access via keys. Moreover, with the aid of security immutable and automated data structure was provided for the network along with the inclusion of handshake mechanism with public key pair for each data transaction. However, the computational complexity was not minimized by handshake mechanism.

1.1 Contributions of the work

In order to overcome the existing issues, a novelBaby Giant Public Cryptography and Goldwasser Homomorphic Cuckoo Hash (BGPC-GHCH) secured data storage method is introduced with the following novel contributions.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

- To improve data confidentiality and data integrity in fog computing, a novel BGPC-GHCH method has been designed.
- To minimize storage costs, the BGPC-GHCH method performs user registration, key generation, encryption and data storage using the homomorphic property.
- To improve data confidentiality and data integrity, the BGPC-GHCH method employs Baby Giant Step Public Cryptography-based Key Generationfor a corresponding data packet and group feature the generator values are obtained for several fog users. This model enables the identification of authorized fog users and malicious fog attackers with higher confidentiality and integrity.

- To reduce the encryption time, Goldwasser–Micali Homomorphic cryptosystem-based Encryption is applied that using Goldwasser Micali Homomorphic cryptosystem property encrypts data packet based on random number in a significant manner and ensures secured storage by means of Cuckoo Hashing function.
- Finally, comprehensive experiment assessments are carried out to estimate the performance of the BGPC-GHCHmethod along with the various metrics, i.e., data confidentiality, data integrity, storage cost and encryption time.

1.2 Organization of the work

The rest of the article is categorized into different sections as follows. Section 2 reviews the related works in fog computing for secure data storage. Section 3 provides a concise description of the proposed BGPC-GHCH method with the aid of pseudo code representation. Section 4 describes the experimental setup, followed by implementation details in Section 5, and performance analysis of the proposed BGPC-GHCH method, comparing it with existing methods using various metrics in Section 6. Finally, conclusion of paper is presented in Section 7.

2. Related works

To improve software services security re, several research studies have been presented for both comprehending and classifying the mechanisms for evaluating security. Owing to change in the demand of users', security objectives must also be changed in a timely manner. The main objective of ensuring security remains in security the software from different types of attacks. An encryfuscation technique for both obfuscating data and location was presented in [5].

One of the materializing techniques for delivering healthcare data to remote users is tele- health. On the other hand, tele-health depends on IoT that in turn present with the weakest security. In [6], elliptic curve cryptography (ECC) over the conventional RSA model was designed. Here, with the aid of effective modular multiplication resulted in saving energy consumption and latency considerably. Yet another fuzzy Analytic Hierarchy Process was designed in [7] to achieve appropriate security solutions with convenience.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

Cloud computing (CC) refers to the on-demand accessibility of different types of resources more precise data storage without the requirement of human involvement. Improving the security necessitates certain amount of drawbacks like, access time, utilization of resource and constrained storage. To address on these aspects, a Non-Dominated Sorting Genetic Algorithm Access Control was proposed in [8]. The main focus here remained in minimizing the energy consumption and also security was improved considerably. Moreover, a stream cipher algorithm (SCA) was also employed in encrypting input data on the basis of the water tank parameters acquired from sensors.

In spite of several features supported by the cloud environment, it is also not said to be free from several issues. This is due to the reason that for data users may not entirely depend on a cloud environment that is owned by a third party and hence security remains a major factor to be focused. Several materials and methods have been recently researched to address the data security issue while handling data storage. Nevertheless, there is a lack of existing methods that tackle the data security issue when it is stored in a cloud environment. A precise security method while data being shared and stored in the cloud was presented in [9] to lessen security assaults. Despite security aspects improved data confidentiality and data integrity were not analyzed in detail.

In the incessant advancement of the computer age, increased insistence of Internet of Things (IoT) devices and the cloud has seeked a middleware. To meet to the demand, fog computing has materialized to ensure fast and secure services by catering existence objects used on a daily basis. Quantum cryptography was applied in [10] that not only collected the data packets but also transmitted the same to the nearby fog servers for further analysis. Yet another privacy preservation mechanism using machine learning was designed in [11] based on fog computing. Here also by using secure data normalization method communication and computation overhead were reduced considerably. Though communication and computation overhead were reduced, however the encryption time was not focused.

The issue of conventional data integrity is low data security and constrained communications efficiency. To solve these issues, a data integrity audit method using data blindingwas presented in [12]. Also the security proof was validated employing computational Diffie-Hellman (CDH) assumptions that in turn enhanced the security of data audit considerably. Various cryptographic algorithms were investigated in [13] to improve the encryption efficiency. Though encryption efficiency was improved the storage cost was not focused.

Implementing CC cloud computing warrants or delegates several paths for web-based service offerings to meet different requirements. Nevertheless both data security and privacy has become a censorious issue that constrains several cloud applications. A method called, Security Aware Efficient Distributed Storage model was proposed in [14] for ensuring secured distribution via intelligence cryptography. Yet another secure cloud fog computing employing matchmaking attribute based encryption was presented in [15]. Also a detailed security analysis was made against real world security threats.

Yet another mixed linear and non linear spatiotemporal chaotic system ensuring privacy protection for fog computing and IoT were designed in [16]. A fog enabled privacy preservation model employing convolutional neural network with bidirectional LSTM was designed in [17]. A review of applications in fog computing environment was investigated in [18]. A dual mechanism covering

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

admission control policy and key agreement were designed in [19] by fostering cloud provider trust significantly. A fog based multipurpose IoT for ensuring data security and providing privacy mechanism was designed in [20].

Motivated by the above materials, in this work a novel method called, a secured data storage cryptography for fog computing environment called, Baby Giant Public Cryptography and Goldwasser Homomorphic Cuckoo Hash (BGPC-GHCH) is proposed. The detailed description of the BGPC-GHCH method is provided in the following sections.

3. Methodology

Fog computing being a distributed decentralized system organizes information forwarded to fog server for performing significant storage processing. Also fog computing provides flexible on-demand services for both individuals and business establishments in storing their data on fog server. With the extensive evolution of fog computing, storage services assists fog users for lessening their local storage menaces. These storage services also assist the users in accessing their data anytime and anywhere. In fog computing environment, the major issue is that the user data should be kept secret from being misused by malicious users. Hence, it is mandatory to ensure secure storage while protecting user data from unauthorized users, so secure fog data storage is an emerging service where not only user confidentiality should be gain but should also be flexible enough in accessing data as far as cloud users are concerned. Despite several techniques being developed for secure data storage, still there requires many issues to be addressed like, high data confidentiality, high data integrity at the cost of low storage complexity. In this work, a novel Baby Giant Public Cryptography and Goldwasser Homomorphic Cuckoo Hash (BGPC-GHCH) secured data storage with higher data confidentiality and integrity rate for efficient secured data storage is introduced rate in the fog environment. Figure 1 shows the structure of BGPC-GHCH method.

CAHIERS MAGELLANES-NS
Volume 06 Issue 2
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

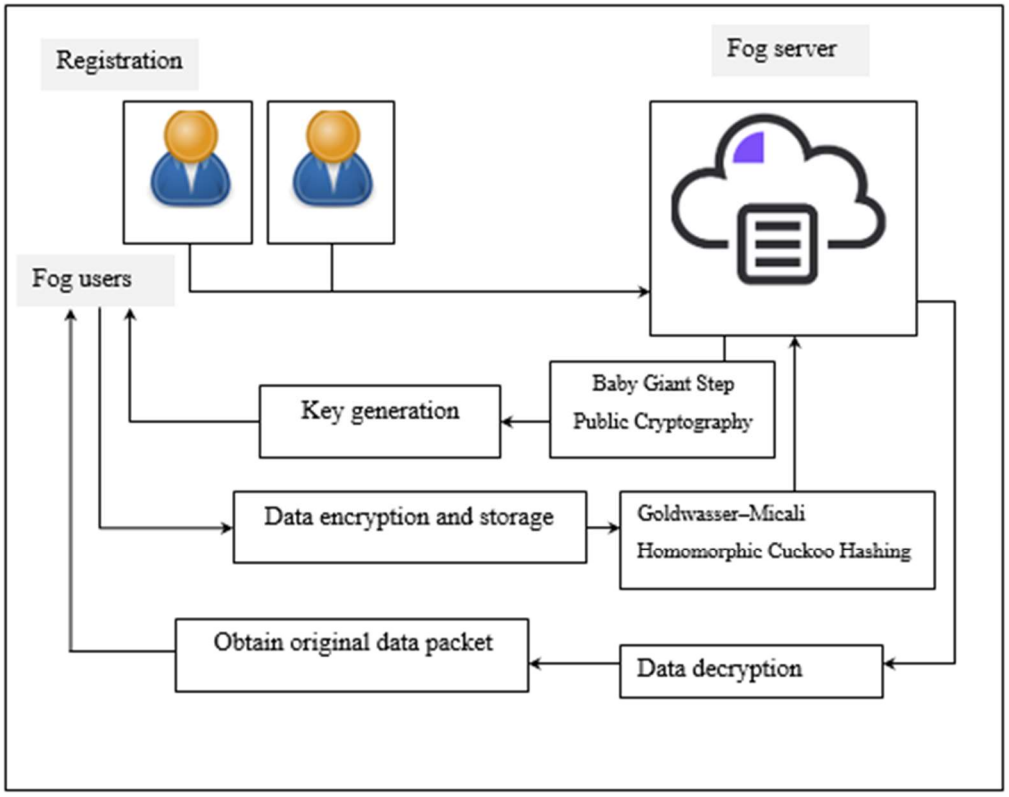

Figure 1 Structure of BGPC-GHCH method

The process of BGPC-GHCH method is illustrated in above figure 1 to achieve secure data storage in fog environment. The fog based architecture consists of three entities like, fog users '$FU_1, FU_2, FU_3, \ldots\ldots FU_n$', fog server '$FS$' and data owner '$DO$' for ensuring secure data storage. Here, fog user stores distinct numbers of data packets with each packet holding information of each fog user features '$DP_1, DP_2, \ldots DP_l$' obtained from WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) into fog server.

The proposed BGPC-GHCHmethod includes various processes namely registration, key generation, data encryption and storage, and finally decryption. Initially, distinct numbers of fog users log into the fog server with the purpose of registering their details in the form of data packets. By considering users information in the form of data packets, fog server registers individual user. Upon successful registering of fog users, key generation process is carried out by the fog server by employing Baby Giant Step with the purpose of generating a pair of keys like, private and public keys that assists in encrypting or decrypting the data. The keys generated are employed in storing fog user data packets on fog server in a secure manner. Following which encryption is performed using Goldwasser–Micali Homomorphic cryptosystem.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

Finally, for efficient data storage, data encryption is performed on each data based on the receiver's public key employing Cuckoo Hash Storage that in turn assists in safeguarding sensitive information and enhance secure communication between fog users and the fog server. Following which the encrypted data packets are sent to the fog server. Before storing encrypted data, digital signature is obtained in the form of hash value by applying Cuckoo Hash Function. Then, the fog server stores encrypted data along with digital signature and hash value with minimum memory and encryption time. Whenever fog user accesses the data packet from a fog server, they perform decryption and signature verification to regenerate the original data packet. Finally, the new signature is generated to verify fog user to access data packet from storage service correspondingly. Hence, proposed BGPC-GHCHmethod is designed to provide higher data confidentiality and integrity in fog secure data storage.

3.1 Dataset Description

In order to perform secure data storage in fog computing environment, WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research is acquired from https://www.cse.wustl.edu/~jain/ehms/index.html. This dataset has been generated employing a real-time Enhanced Healthcare Monitoring System (EHMS) for patients. The information pertaining to health from distinct patients is collected by means of IoT devices attached to their bodies and sent to the fog server via internet. In this scenario, an attacker may intercept the data packets prior it arrives at the fog server. As a result, the method is responsible for verifying user authenticity and detecting different types of attacks, to name a few being, man-in-the-middle attacks like, spoofing and data injection, during data transmission of patient between users and the fog server. The dataset consists of an overall of 44 features and 16000 instances or records. Out of 44 features, 35 features are said to be network flow metrics, eight patients' biometric features like, Heart Rate, Respiration Rate, ST segment, Systolic blood pressure, Diastolic blood pressure, Blood oxygen, pulse rate and temperature. Finally, the feature labeled for the output label (0 or 1), where samples with attacker or unauthorized users are labeled as 1 and on contrary the samples with genuine or authorized users are labeled as 0. For experimental purposes, the samples are measured in terms of patients or fog users. To conduct fair comparison, ten iterations are performed for each method, (i.e., both proposed and existing methods), with the input ranging between 1000 users and 10000 users in fog computing environment.

3.2 Baby Giant Step Public Cryptography-based Key Generation

In the proposed BGPC-GHCH Method, the fog user 'FU' (i.e., patient) registers their detail (i.e., an overall of 44 features) 'F' to the fog server 'FS' for carrying out secured data storage. Here, a sample instance of a fog user with their corresponding feature details are said to be in a data packet 'DP'. In other words, each fog user possesses a data packet holding the corresponding 44 feature information for further processing. This is mathematically expressed as given below.

$$UR \rightarrow DP_i(i = 1,2, \dots, l) \rightarrow \begin{bmatrix} FU_1F_1 & FU_1F_2 & \dots & FU_1F_n \\ FU_2F_1 & FU_2F_2 & \dots & FU_2F_n \\ \dots & \dots & \dots & \dots \\ FU_mF_1 & FU_mF_2 & \dots & FU_mF_n \end{bmatrix} \quad (1)$$

As given in the above equation (1), the user registration 'UR' (i.e., fog user or patient 'FU_m'

CAHIERS MAGELLANES-NS
Volume 06 Issue 2
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

with 'm' samples) is said to be performed with respect to 'n' features 'F_n' respectively in the form of 'l' data packets. The fog user registers their detail (i.e., data packets) to the fog server for further processing. Upon successful completion of the user registration process, the fog server generates public key and private key for the registered user employing Baby Giant Step Public Key Cryptography. Prior to the public key '$PubK$' and private key '$PrivK$' generation, discrete log problem is said to be one of the elementary significance to the domain of public key cryptography. This is because the more the complexity the discrete log is the more the difficulty remains in computing the same and vice versa.Several of the most frequently used cryptography systems are designed on the basis of the assumption that discrete log is tremendously laborious to measure, hence the more laborious it is, the more security it is said to be provided during data transfer. One method to shoot up the complicatedness of the discrete log function is to base the cryptosystem on a larger group and is performed in our work by means of Baby Giant Step Public Key Cryptography. Figure 2 shows the structure of Baby Giant Step Public Cryptography-based Key Generation model.

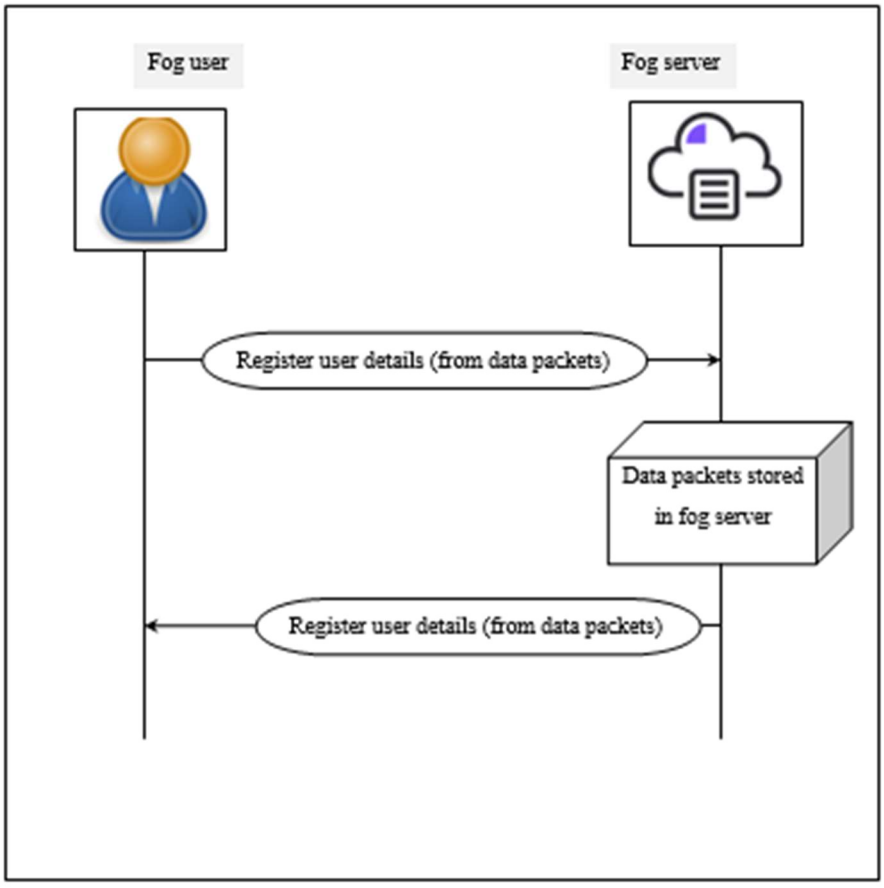

Figure 2 Structure of Baby Giant Step Public Cryptography-based Key Generation model

As illustrated in the above figure, given a monogenous group 'MG' of order 'O', a generator

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

'Gen' of the group for a corresponding data packet and group feature '$GenF$', the issue is to identify an integer 'K' such that

$$Gen^K = GenF \quad (2)$$

Then, the Baby Giant Step Public Key Cryptography function is based on rephrasing 'K' as given below.

$$K = iq + j \quad (3)$$

$$q = \left[\sqrt{O}\right] \quad (4)$$

$$0 \leq i < q;\ 0 \leq j < q \quad (5)$$

Therefore from the above generator 'Gen' of the group for a corresponding data packet and group feature '$GenF$' we have

$$Gen^K = GenF \quad (6)$$

$$Gen^{iq+j} = GenF \quad (7)$$

$$Gen^j = GenF(Gen^{-q})^i \quad (8)$$

From the above equations (6), (7) and (8) results the 'Gen^j' values, are recalculated for several 'j' values (i.e., for several fog users) respectively. Following which the public key and private key for each fog users are generated by the fog server by initializing four prime numbers 'PN_i', 'PN_j', 'PN_k' and 'PN_l'. With the initialized prime numbers the threshold is evaluated as given below.

$$Th = PN_i * PN_j * PN_k * PN_l \quad (9)$$

$$Phi(W) = (PN_i - 1) * (PN_j - 1) * (PN_k - 1) * (PN_l - 1) \quad (10)$$

Next, three random numbers 'RN_i', 'RN_j' and 'RN_k' are obtained via random number generator and base modulus is obtained as given below.

$$V = (RN_i * RN_j * RN_k) \quad (11)$$

With the above base modulus and threshold, the public key '$PubK$' and private key '$PrivK$' are obtained as given below.

$$PubK = [Phi(W) * V] \quad (12)$$

$$PrivK = [(PrivK * PubK) mod Phi(W) * V\ ] = 1 \quad (13)$$

The private key '$PrivK$' as given above in equation (13) is selected in such a manner that its value is '1'. The pseudo code representation of Baby Giant Step Public Cryptography-based Key Generation is given below.

Input: Dataset 'DS', Samples '$S = \{S_1, S_2, \dots, S_m\}$', Fog User (i.e., patient) '$FU = \{FU_1, FU_2, \dots, FU_n\}$', Data packets '$DP = \{DP_1, DP_2, \dots., DP_l\}$'
Output: delay minimal key generation
1: **Initialize** 'm', 'n', 'l' 2: **Initialize** prime numbers 'PN_i', 'PN_j', 'PN_k' and 'PN_l' 3: **Begin** 4: **For**each Dataset 'DS' with Samples 'S', Fog User (i.e., patient) 'FU' and Data packets 'DP' **//User registration** 5: Perform user registration 'UR' as given in equation (1) **//Discrete log to base cryptosystem**

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

6: Perform discrete log function to base the cryptosystem on a larger group as given in equations (2), (3), (4) and (5)
7: Formulate the generator for a corresponding data packet as given in equations (6), (7) and (8)
//Public key and private key generation
8: Formulate the threshold as given in equations (9) and (10)
9: Obtain base modulus as given in equation (11)
10: Generate public key as given in equation (12)
11: Generate private key as given in equation (13)
12: **Return** public key '$PubK$', private key '$PrivK$'
13: **End for**
14: **End**

Algorithm 1 Baby Giant Step Public Cryptography-based Key Generation

As given in the above algorithm with the objective of improving the delay in addition to data confidentiality and data integrity, a public key cryptography mechanism using Baby Giant Step is employed during the key generation process. Not only the data packets should be received by the authorized receiver but also it should not be modified so that both data confidentiality and data integrity can be achieved. With this objective, first discrete log to base cryptosystem is applied to the registered users so that it is highly complicated to compute and hence it provides a data transfer in a secured manner, therefore corroborating the objective of reception of the data packets to the intended recipient (i.e., high data confidentiality). In addition, public key and private keys are said to be generated employing prime numbers via random number generator and base modulus that in turn ensures that the information in data packets are not modified (i.e., high data integrity).

3.3 Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing secured data storage

In this section with the generated key pair (i.e., public key '$PubK$' and private key '$PrivK$'), the fog user encrypts the data using Goldwasser–Micali Homomorphic cryptosystem and transmits the encrypted data packets to the fog server for carrying out secured data storage. Here, cuckoo hash function is employed by the fog server for performing encrypted user data storage with their corresponding hash value. The stored information gets retrieved by the user via decryption. Figure 3 shows the structure of Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing secured data storage model.

CAHIERS MAGELLANES-NS
Volume 06 Issue 2
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

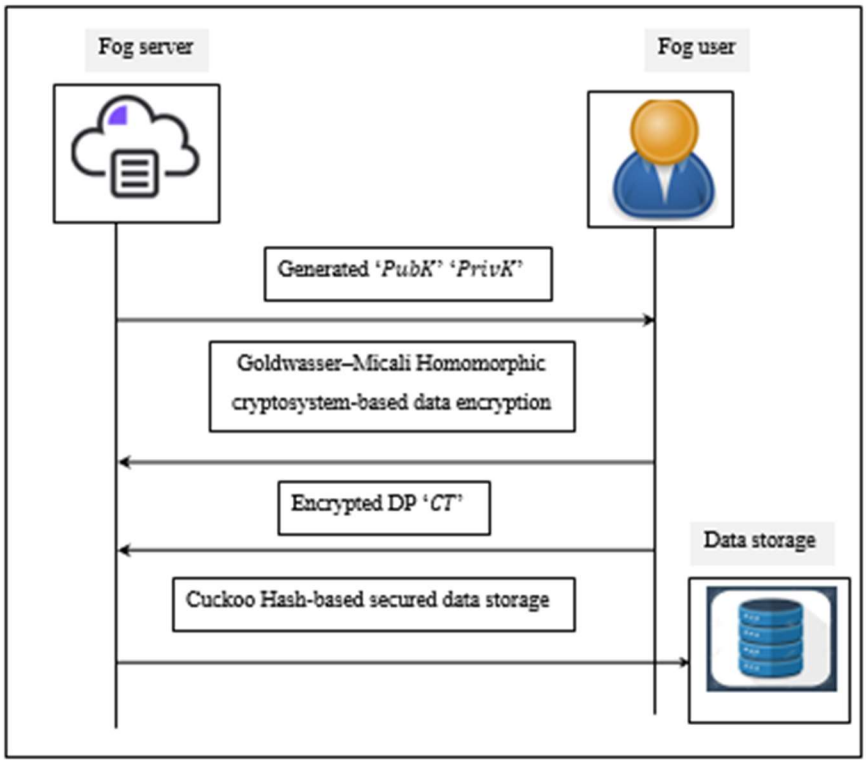

Figure 3 Structure of Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing secured data storage

As illustrated in the above figure, with 'n' numbers of fog users in queue to perform encryption via fog server and feature information stored in the form of 'l' data packets 'DP' the private key is subjected to Goldwasser–Micali Homomorphic cryptosystem property. Then by XORing the public and private key, cipher text is generated. Storage is performed using Cuckoo Hashing to ensured security. In the Goldwasser Micali Homomorphic cryptosystem property, let us consider the public key to be modulus '$mod\ PubK$' and the private key be '$PrivK$', then the encryption of a data packet 'DP' is mathematically represented as given below.

$$Enc(DP) = Priv^{DP} RN^2 mod\ PubK \quad (14)$$

From the above equation (14), the encryption property is generated based on certain random number 'RN'. Following which the homomorphic property is formulated as given below.

$$Enc(DP_1).Enc\ (DP_2) = PrivK^{DP_1} RN_1 PrivK^{DP_2} RN_2 mod\ PubK \quad (15)$$

$$Enc(DP_1.DP_2) = PrivK^{DP_1+DP_2} (RN_1 RN_2)^2\ mod\ PubK \quad (16)$$

$$CT = Enc(DP_1 \oplus DP_2) \quad (17)$$

From the above equations (15), (16) and (17), with the aid of homomorphic property encrypted results are obtained for further processing. Following which the encrypted data is subjected to Cuckoo Hashing for ensuring secured data storage. The Cuckoo Hashing consists of two tables 'T_1', 'T_2' with table size being 'p' and two hash functions 'H_1', 'H_2' respectively. Now for an encrypted data packet 'CT', the fog server pass it to 'H_1' to obtain its index in the first table. If the subsequent row is vacant, then the encrypted data packet 'CT' is stored there. This is mathematically stated as given below.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

$$CT_{1,old} = T_1[H_1(CT)] \quad (18)$$

On contrary, if another encrypted data packet 'CT_{old}' is present then it is removed and the new encrypted data packet 'CT' is stored there. Also, the old encrypted data packet 'CT_{old}' is checked by the fog server and pass it to 'H_2' to obtain its index in the second table. This is mathematically stated as given below.

$$CT_{2,old} = T_2\left[H_2\left(CT_{1,old}\right)\right] \quad (19)$$

If the subsequent row is vacant, then the old encrypted data packet 'CT_{old}' is stored and stops the process. This procedure is said to be repeated for all the encrypted data packets of 'n' corresponding fog users. In this manner, secure data storage is said to be ensured. Finally, the stored information gets retrieved by user through decryption. The pseudo code representation of Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing secured data storage is given below.

Input: Dataset 'DS', Samples '$S = \{S_1, S_2, \dots, S_m\}$', Fog User '$FU = \{FU_1, FU_2, \dots, FU_n\}$', Data packets '$DP = \{DP_1, DP_2, \dots., DP_l\}$'

Output: secured data storage

1: **Initialize** public key '$PubK$', private key '$PrivK$', random number 'RN'
2: **Initialize** tables 'T_1', 'T_2', table size 'p', hash functions 'H_1', 'H_2'
3: **Begin**
4: **For** each Dataset 'DS' with Samples 'S', Fog User 'FU, Data packets 'DP', public key '$PubK$', private key '$PrivK$'
//Encryption
5: Formulate encryption for certain random number as given in equation (14)
6: Obtain encrypted data (i.e., packet) as given in equations (15), (16) and (17)
7: Return encrypted data or cipher text 'CT'
//Secured data storage
8: Formulate cuckoo hashing 'H_1' as given in equation (18)
9: **If** '$CT \neq \emptyset$'
10: **Then** '$T_1[H_1(CT)] = CT$'
11: **End if**
12: Formulate cuckoo hashing 'H_2' as given in equation (19)
13: **If** '$CT_{1,old} \neq \emptyset$'
14: **Then** '$T_2\left[H_2\left(CT_{1,old}\right)\right] = CT_{1,old}$'
15: **End if**
//Decryption
16: New cipher text 'CT'' is generated by fog server and send to fog user
17: **If** '$CT' = CT$' then
18: Cipher text is valid
19: Decrypt data packet using private key '$PrivK$'
20: **Else**

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

21: Cipher text is invalid
22: Decryption is not performed
23: **End if**
24: **End for**
25: **End**

Algorithm 2 Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing secured data storage

As given in the above algorithm with the objective of ensuring secure data storage in fog computing environment Goldwasser–Micali Homomorphic cryptosystem to perform encryption and Cuckoo Hashing for secure data storage is done separately. First, with the generated public and private key as input, Goldwasser–Micali Homomorphic cryptosystem is formulated to obtain the encrypted data packet or cipher text. The advantage of using this Goldwasser–Micali Homomorphic cryptosystem remains in achieving semantic security against passive malicious fog users under the hypothesis that puzzling out the quadratic residuosity problem is exorbitant. In this way security is said to be ensured. Next secure data storage is implemented by means of Cuckoo Hashing function wherein it enables storing encrypted data packets or cipher text utilizing small space hence ensuring small failure probability as it directly relates to privacy guarantees in an extensive manner.

4. Experimental setup

This section describes the evaluation campaign of the proposed BGPC-GHCH method over the Chaotic-Map Based Encryption [1], and Continuous Delivery/continuous verifiability (CD/CV)[2]. The performances of Baby Giant Public Cryptography and Goldwasser Homomorphic Cuckoo Hash (BGPC-GHCH) secured data storage is implemented in Java with FogSim simulator. To ensure fair comparison same dataset WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research acquired from https://www.cse.wustl.edu/~jain/ehms/index.htmlis used or ensuring secured data storage in fog environment. With the help of storage service, fog data is stored and used for future purposes. For experimental considerations, the number of fog user data is measured from users. To evaluate performance, ten iterations are performed with the input data between 1000and 10000 user data in a fog environment. To analyze the performance of BGPC-GHCH method, the results are evaluated in terms of data confidentiality, data integrity, encryption time and storage complexity.

5. Implementation details

In this section the implementation details of proposed Baby Giant Public Cryptography and Goldwasser Homomorphic Cuckoo Hash (BGPC-GHCH) method for secured data storage is provided. The proposed BGPC-GHCH method is split into four different steps.

- First, user details registration are performed by obtaining the raw data from WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research acquired from https://www.cse.wustl.edu/~jain/ehms/index.html.

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

- The features present in the dataset are considered as data packets for each for user and follows with the process of user registration where each fog user registers their corresponding data packets with the fog server for further processing.
- Second upon successful completion of user registration, the fog server generates the public key and private key by employing Baby Giant Step Public Key Cryptography. Followed by which the generated public and private keys are sent to the corresponding fog user.
- Third with the obtained public and private keys, fog user encrypts the data packets using Goldwasser–Micali Homomorphic cryptosystem-based Encryption.
- Once data encryption is performed by the corresponding fog user, Cuckoo Hashing function is applied by the fog server to ensure secure data storage.
- Finally, decryption is said to be performed in the other end to accomplish the overall objective.

6. Discussion

6.1 Performance analysis of data confidentiality

While designing secure data stroage in fog computing environment, one of the most paramount performance factors is the data confidentiality rate. It is evalauted as the percentage ratio of the number of data that are received by the authorized for user. The data confidentiality rate is evaaluted as given below.

$$DC = \sum_{i=1}^{m} \frac{FU_{IR}}{S_i} * 100 \quad (20)$$

From the above equation (20), the data confidentiality 'DC' is measured based on the sample data obtained by the fog user (i.e., intended recipient 'FU_{IR}' with respect to the total number of sample data obtained as input 'S_i' for performing simulation purpose. The data confidentiality is evaluated in terms of percentage (%). The comparison table of the proposed BGPC-GHCH method with existing methods of Chaotic-Map Based Encryption [1] and CD/CV [2] in terms of data confidentiality is given below. Table 1 given below lists the data confidentiality of the proposed BGPC-GHCH method for 1000 to 10000 data sizes compared with the existing methods.

Table 1 Comparison of data confidentiality

Number of data samples	Data confidentiality (%)		
	BGPC-GHCH	Chaotic-Map Based Encryption	CD/CV
1000	97.5	94	91.5
2000	96.35	91.15	86.25
3000	95	90	86
4000	94.15	89	85.15
5000	93	88.35	86
6000	93.25	88.85	86.15
7000	94	90.35	87.35
8000	94.55	91	88
9000	95.35	91.25	88.35
10000	96.25	91.55	89

CAHIERS MAGELLANES-NS
Volume 06 Issue 2
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

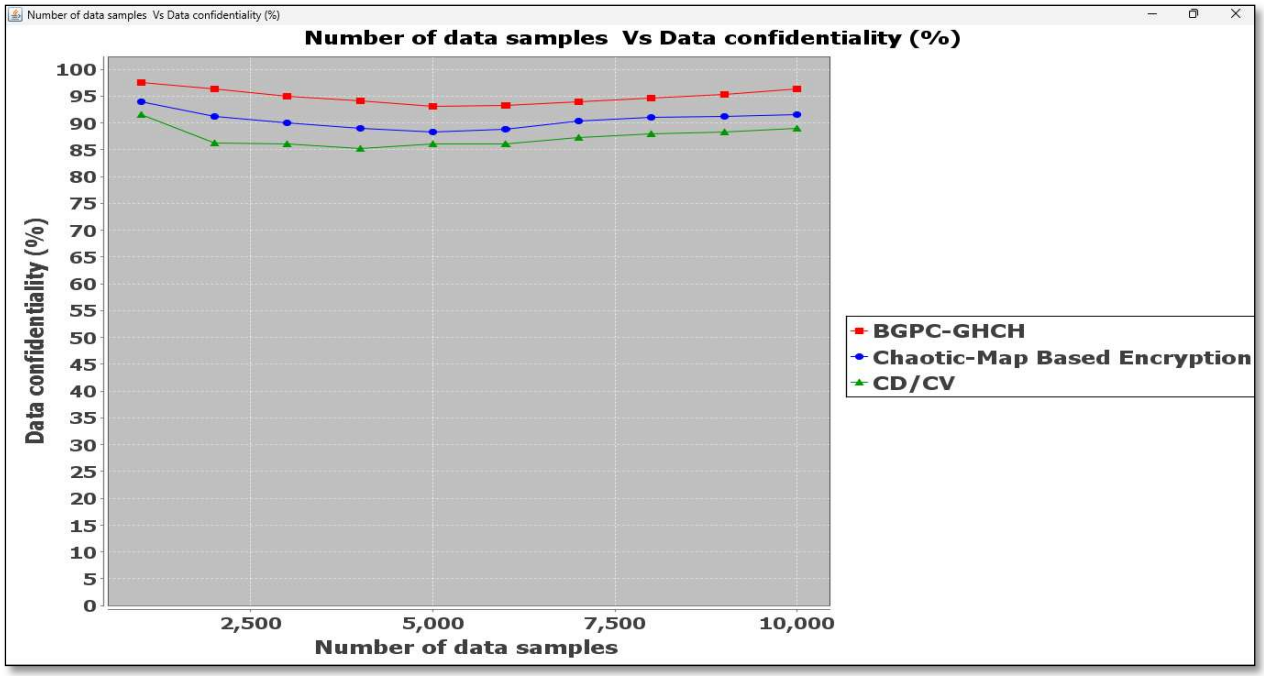

Figure 4 Graphical representation of data confidentiality

Figure 4 given above illustrates the data confidentiality using the three methods, BGPC-GHCH, Chaotic-Map Based Encryption [1] and CD/CV [2] respectively. From the above figure it is inferred that that increasing the number of data samples, neither increase of decreasing trend is observed using all the three methods. However, comparative analysis between the three methods showed good results using the BGPC-GHCH method than [1] and [2]. This is inferred from the simulation results where 97.50% of results were arrived using the BGPC-GHCH method, 94% using [1] and 91.5% using [2] for 1000 sample data. The reason for the improvement was owing to the application of the Baby Giant Step Public Cryptography-based Key Generation. By applying this key generation model, initially, discrete log to base cryptosystem was applied for each registered fog users with the purpose of forming highly complicated to compute it. With this data transfer was said to be ensured in a secured manner, therefore establishing the objective of data packet reception to the intended fog users in a confidential manner. Also, the Baby Giant Step Public Key Cryptography function was applied for the set of for a corresponding data packet and group feature to perform key generation. With this, the data packet received by the intended fog users were found to be enhanced using the BGPC-GHCH method and therefore substantial improvement was observed in data confidentiality rate also. With an average of 10 simulation runs, the data confidentiality using the BGPC-GHCH method was found to be improved by 5% when compared to [1] and 9% upon comparison with [2] respectively.

6.2 Performance analysis of data integrity

In this section to examine the integrity of data, the data integrity rate is measure. The data integrity rate is evaluated as the percentage ratio of number of data that are not modified by any malicious fog users to the number of sample data involved in the overall simulation process. The data

CAHIERS MAGELLANES-NS
Volume 06 Issue 2
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

integrity rate is measured as given below:

$$DI = \sum_{i=1}^{m} \frac{S_{NA}}{S_i} * 100 \quad (21)$$

From the above equation (21), the data integrity 'DI', is measured by considering into factor the overall sample data 'S_i' and the sample data that were not changed by the malicious fog users 'S_{NA}'. It is measured in terms of percentage (%). The results of the data integrity rate are tabulated in table 2.

Table 2 Comparison of data integrity

Number of data samples	Data integrity (%)		
	BGPC-GHCH	Chaotic-Map Based Encryption	CD/CV
1000	3.5	5	7.5
2000	3.85	5.25	8.25
3000	4	5.85	9
4000	4.35	6.35	10.35
5000	4.85	7	11
6000	5	7.25	11.23
7000	5.55	8	12
8000	6	8.35	12.45
9000	6.85	9	13
10000	7.35	9.55	13.35

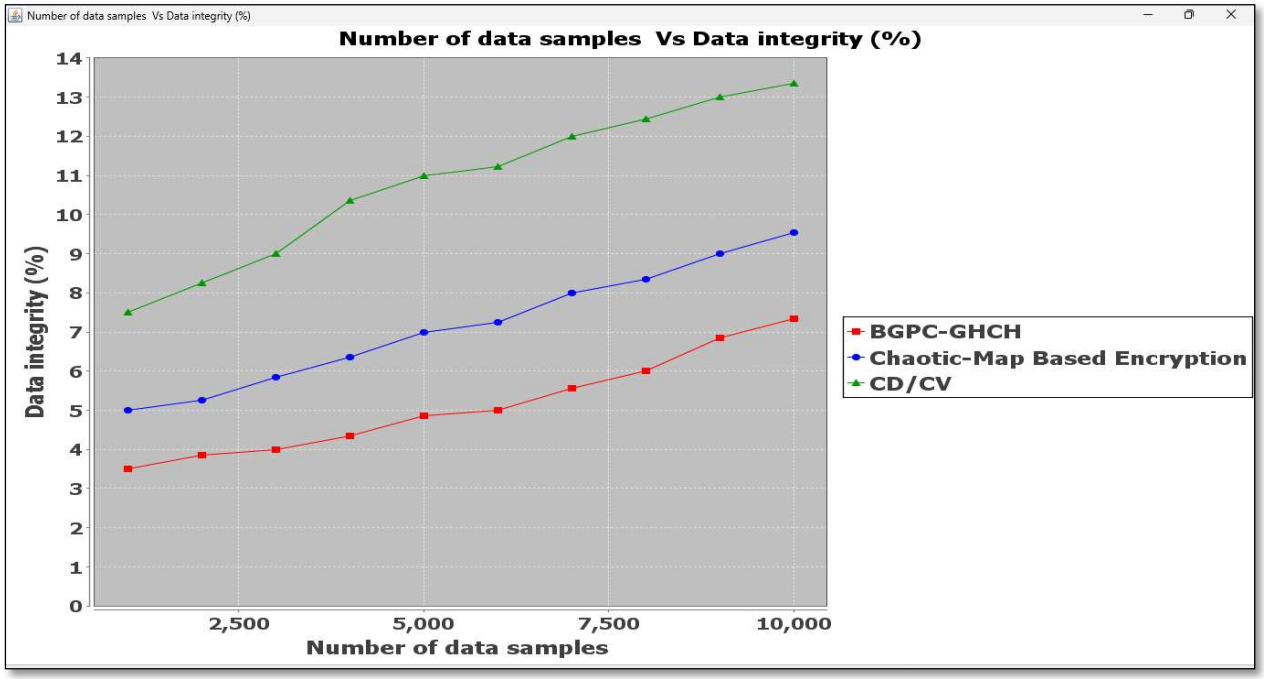

Figure 5 Graphical representation of data integrity

Figure 5 given above shows the data integrity rate using the three methods, BGPC-GHCH, [1]

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

and [2]. To conduct a fair comparison similar number of data samples were employed from the given dataset WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research for the three methods. An average of 10 simulations runs was performed to measure the data integrity. From the above figure an increasing trends were inferred using all three methods. However, a considerable amount of improvement was found when applying BGPC-GHCH upon comparison to [1] and [2]. The reason behind the improvement was owing to the application of Baby Giant Step Public Cryptography-based Key Generation algorithm that in turn selects a number arbitrarily using prime numbers via random number generator and base modulus, following which obtains the public and private key for a specified time instance with respect to the for a corresponding data packet and group feature. This in turn assisted in reducing the numbers of sample data not to be altered by the malicious fog users, therefore enhancing the data integrity rate. Also learning the communication patterns between fog users using four prime numbers in the cloud environment improves the data integrity rate using the BGPC-GHCHmethod by 29% compared to [1] and 53% compared to [2].

6.3 Performance analysis of encryption time

Encryption timeis defined astheamount of time consumed in performing the overall data encryption process for the fog user upon receipt of the generated keys from the fog user to perform secure data storage on fog server.The encryption time is evaluated as given below.

$$Enc_{time} = DP_i * T\ [Enc(DP)] \tag{22}$$

From the above equation (22), encryption time 'Enc_{time}' is measured by taking into consideration the number of sample data or data packets to be sent 'DP_i' and the actual time consumed in encrypting the data packet'$T\ [Enc(DP)]$'. The overall encryption time is measured in terms of milliseconds (ms).

Table 3 Comparison of encryption time

Number of data samples	Encryption time (ms)		
	BGPC-GHCH	Chaotic-Map Based Encryption	CD/CV
1000	250	330	450
2000	285.35	385	485
3000	315.35	445.15	535
4000	350	535.35	585.15
5000	385.55	600.15	655.35
6000	435	655.35	725.55
7000	485.15	700.35	815.35
8000	525	724.15	885.25
9000	600	800	935.15

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

10000	635.15	895.35	1025.15

Table 3 given above provides the performance results of encryption time for different numbers of fog user sample data. For performing simulation, the fog user sample data ranging between 1000 and 10000 were considered from dataset. For simulation the experiment was conducted for 1000 numbers of sample fog user data, and encryption time of secure data storage using BGPC-GHCH method was found to be 250 ms, whereas encryption time using [1] and [2] were observed to be 330ms and 450ms respectively. As a result, the BGPC-GHCH method is compared to be better than the observed results of existing methods.

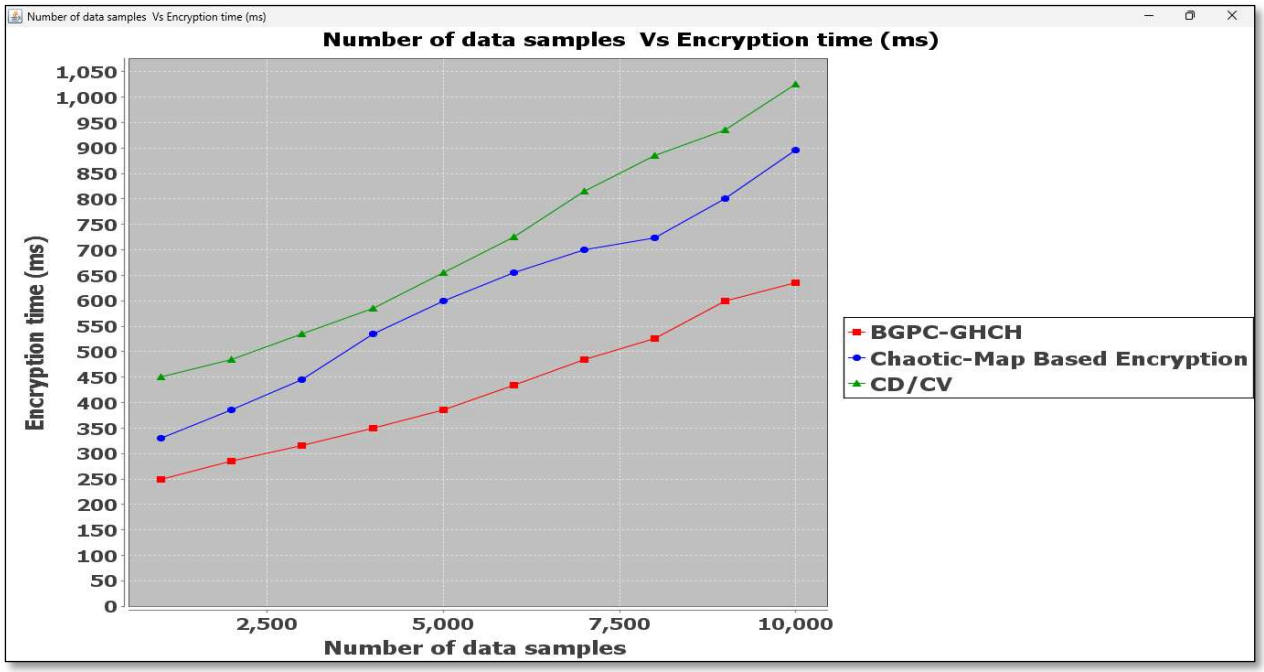

Figure 6 Graphical representation of encryption time

Figure 6 given above shows the comparative analysis of encryption time for 10000 distinct numbers of fog users' data. From above figure, x axis refers to the number of distinct data samples and y axis refers to the encryption time using the three methods. Here, the proposed BGPC-GHCH method provides lesser time utilization as compared to Chaotic-Map Based Encryption [1] and CD/CV [2]. Moreover, while increasing the number of fog user's sample data, encryption time for ensuring secure data storage is also found to be increased using all three methods. But relatively, the encryption time using BGPC-GHCH method is minimal as compared to other existing methods. This is owing to the reason that the fog server in our work performs the data storage via three distinct processes namely user registration, key generation and data encryption and hash generation for secure data storage. For each registered user initially, the fog server generates public and private key with minimum time. Followed by, Goldwasser–Micali Homomorphic cryptosystem applied to encrypt data into cipher text. Finally, Cuckoo Hashing Function is carried for each encrypted data to generate hash value for secure data storage. The encrypted data with hash value is stored in the fog server. The fog server uses the dual hash

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

table system for storing the multiple user data with minimum time and provides secure data access. Overall, the average value of comparison results refers to that the encryption time using BGPC-GHCH method was reduced by 30% and 40% upon comparison to [1] and [2] respectively.

6.4 Performance analysis of storage cost

Finally storage complexity or storage cost is measured. Storage cost refers to the amount of storage space or memory consumed by the fog server for storing fog user data packet in the fog server's database. This is mathematically evaluated as given below.

$$SC = \sum_{i=1}^{n} DP_i * Mem[storage\ ] \quad (23)$$

From the above equation (23), the storage cost 'SC' is measured by taking into consideration the data packets 'DP_i' involved in the storage process and the actual memory consumed for storing the data packets '$Mem[storage\ ]$' for the corresponding fog user via fog server respectively. It is measured in terms of Megabytes (MB).

Table 4 Comparison of storage cost

Number of data samples	Storage cost (MB)		
	BGPC-GHCH	Chaotic-Map Based Encryption	CD/CV
1000	50	75	90
2000	65	90	115
3000	73	105	130
4000	90	118	145
5000	105	125	165
6000	115	140	180
7000	127	155	215
8000	135	175	225
9000	140	195	240
10000	155	210	285

Table 4 given above illustrates the simulated values of storage cost for proposed BGPC-GHCH method with existing methods, [1] and [2]. For performing the experimental work, number of fog user data ranging between 1000 and 10000 is considered. From above table values, proposed method attains minimum storage while performing data access between fog users via fog server. For example with 1000 numbers of fog user sample data in vogue, proposed BGPC-GHCH method attains 50 MB of storage complexity where as existing [1] and [2] achieves 75 MB and 90 MB of storage complexity. Hence, it is significant that the storage cost using complexity using BGPC-GHCH method is lower for secure data storage as compared to other methods.

CAHIERS MAGELLANES-NS
Volume 06 Issue 2
2024

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

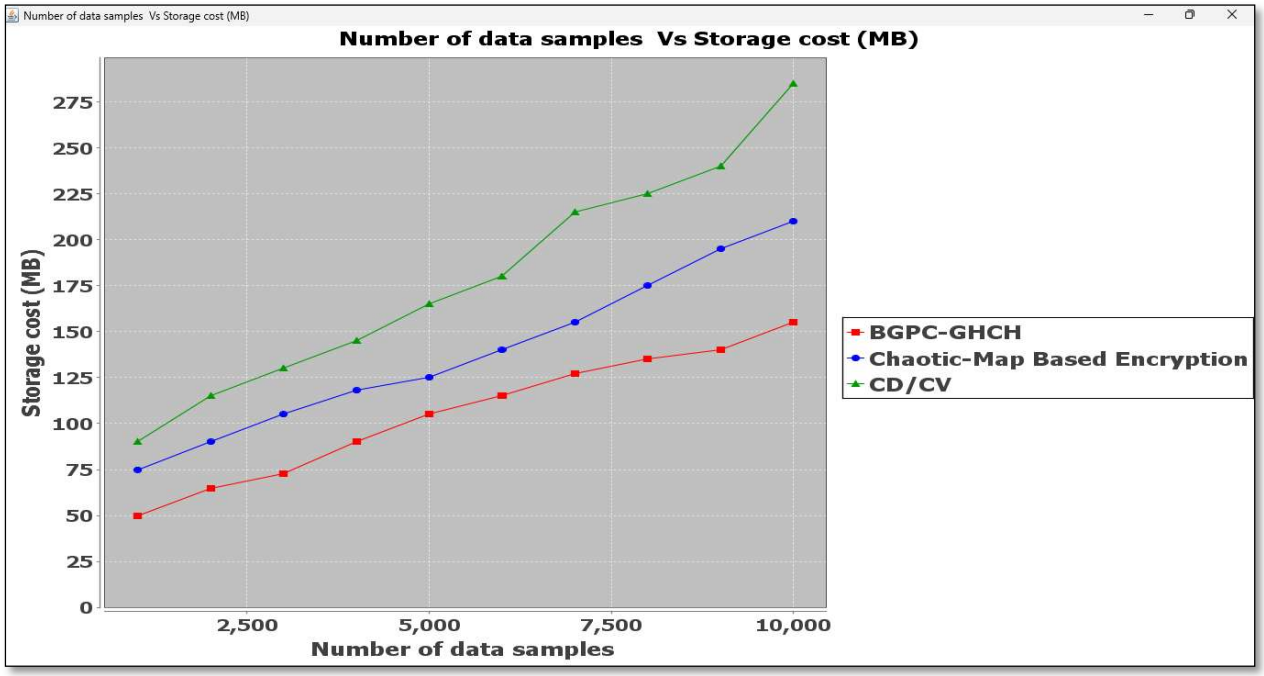

Figure 7 Graphical representation of storage cost

Finally, figure 7 given above demonstrate the measure of storage cost for both proposed BGPC-GHCH and existing methods, [1] and [2]. As shown in the above figure, x axis represents a number of fog users sample data and the y axis actually represents the storage cost. The performance result shows that the proposed BGPC-GHCH achieved better result upon comparison to the existing methods [1] and [2]. This is due to the application of Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing secured data storage algorithm. The homomorphic cryptographic technique performs key-pair generation, encryption, and decryption. During encryption, data packet is encrypted into cipher text and stored in fog server. Before storing encrypted data, digital signature is obtained in the form of hash value by applying Cuckoo Hash Function. by generating hash value. As a result, the fog user encrypted data or cipher text with hash value is stored in fog server with minimum storage space. Owing to this, the storage cost using proposed BGPC-GHCH is minimized by 24% and 41% when compared to existing [1] and [2].

7. Conclusion

A high security cloud storage service and efficient cryptography technique known as Baby Giant Public Cryptography and Goldwasser Homomorphic Cuckoo Hash (BGPC-GHCH) method is designed for secure data storage in fog environment with minimum time. The secure data storage for each fog user via fog server is performed by applying Baby Giant Step Public Cryptography-based Key Generation and Goldwasser–Micali Homomorphic cryptosystem-based Encryption and Cuckoo Hashing. Initially, fog user registers their details to the fog server for performing secured data storage. Following which the fog server generates public and private key for every registered user by means of Baby Giant Step Public Cryptography-based Key Generation. Next, the data encryption process was performed by means of Goldwasser–Micali Homomorphic cryptosystem-based Encryption and finally

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

secured data storage was done using Cuckoo Hashing function. To be more specific prior to the storing of encrypted data or cipher text, the hash value for each data is generated using Cuckoo Hashing function. Finally, the stored information gets retrieved by the authorized fog user via data decryption process. With this verification result, the cipher text was decrypted to generate the original fog user data. The observed quantitative results confirm that proposed BGPC-GHCHmethod provides improved performance of secure data storage with high data confidentiality, data integrity by minimizing the encryption time and storage cost considerably upon comparison to existing methods.

References

[1] K. R. Raghunandan, Radhakrishna Dodmane, K. Bhavya, N. S. Krishnaraj Rao and Aditya Kumar Sahu, "Chaotic-Map Based Encryption for 3D Point and 3D Mesh Fog Data in Edge Computing", IEEE Access, Volume 11, December 2022, Pages 3545 – 3554 [Chaotic-Map Based Encryption]

[2] Cristhian Martinez-Rendon, J.L. González-Compeán, Dante D. Sanchez-Gallegos and Jesus Carretero, "CD/CV: Blockchain-based schemes for continuous verifiability and traceability of IoT data for edge–fog–cloud",Information Processing and Management, Elsevier, Volume 60, Issue 1, January 2023, Pages 1-15 [Continuous Delivery/continuous verifiability (CD/CV)]

[3] Kumar Sekhar Roy, Subhrajyoti Deb, and Hemanta Kumar Kalita, "A novel hybrid authentication protocol utilizing lattice-based cryptography for IoT devices in fog networks", Digital Communications and Networks, Elsevier, 2023, Pages 1-18

[4] Marc Jayson Baucas, Petros Spachos, and Konstantinos N. Plataniotis, "Public Key Reinforced Blockchain Platform for Fog-IoT Network System Administration", IEEE Internet of Things Journal, Volume 9, Issue 22, 15 November 2022, Pages 22366 – 22374

[5] Jasleen Kaur, Alka Agrawal, Raees Ahmad Khan, "Encryfuscation: A model for preserving data and location privacy in fog based IoT scenario", Journal of King Saud University – Computer and Information Sciences, Elsevier, Mar 2022

[6] Atef Ibrahim, Fayez Gebali, "Compact modular multiplier design for strong security capabilities in resource-limited Telehealth IoT devices", Journal of King Saud University – Computer and Information Sciences, Elsevier, Jun 2022

[7] Alka Agrawal, Mamdouh Alenezi, Suhel Ahmad Khan, Rajeev Kumar, Raees Ahmad Khan, "Multi-level Fuzzy system for usable-security assessment", Journal of King Saud University – Computer and Information Sciences, Elsevier, Aug 2019

[8] K.S. Saraswathy S.S. Sujatha, "Secure data storage and access for fish monitoring in cloud environment", Measurement: Sensors, Elsevier, Feb 2023

[9] Rishabh Gupta, Deepika Saxena, Ashutosh Kumar Singh, "Cryptography Approach for Secure Outsourced Data Storage in Cloud Environment", Researchgate, Jun 2023

[10] Cherry Mangla, Shalli Rani, and Henry Kwame Atiglah, "Secure Data Transmission Using Quantum Cryptography in Fog Computing", Wireless Communications and Mobile Computing, Wiley, Jan 2022

[11] Ruoli Zhao, YongXie, Hong Cheng, Xingxing Jia, and Syed Hamad Shirazi, "ePMLF: Efficient and Privacy-Preserving Machine Learning Framework Based on Fog Computing", International Journal of Intelligent Systems, Wiley, Feb 2023

ISSN:1624-1940
DOI 10.6084/m9.figshare.26310784
http://magellanes.com/

[12] Genqing Bian, Yanru Fu, Bilin Shao, Fan Zhang, "Data Integrity Audit Based on Data Blinding for Cloud and Fog Environment", IEEE Access, Aug 2022
[13] Hassan Noura, Ola Salman, Ali Chehab, Raphael Couturier, "Preserving Data Security in Distributed Fog Computing", Ad Hoc Networks, Elsevier, Jun 2019
[14] Yibin Li, Keke Gai, Longfei Qiuc, Meikang Qiub, Hui Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing", Information Sciences, Elsevier, Sep 2016
[15] Shengmin Xu, Jianting Ning, Yingjiu Li, Yinghui Zhang, Guowen Xu, Xinyi Huang, and Robert H. Deng, "Match in My Way: Fine-Grained Bilateral Access Control for Secure Cloud-Fog Computing", IEEE Transactions on Dependable and Secure Computing, Sep 2022
[16] Yanhui Liu, Jianbiao Zhang, Jing Zhan, "Privacy protection for fog computing and the internet of things data based on blockchain", Cluster Computing, Springer, Feb 2021
[17] Syed Atif Moqurrab, Noshina Tariq, Adeel Anjum, Alia Asheralieva, Saif U. R. Malik, Hassan Malik, Haris Pervaiz, Sukhpal Singh Gill, "A Deep Learning-Based Privacy-Preserving Model for Smart Healthcare in Internet of Medical Things Using Fog Computing", Wireless Personal Communications, Springer, Aug 2022
[18] Saad Khan, Simon Parkinson and Yongrui Qin, "Fog computing security: a review of current applications and security solutions", Journal of Cloud Computing: Advances, Systems and Applications, Sep 2017
[19] D. Paulraj, S. Neelakandan, M. Prakash and E. Baburaj, "Admission control policy and key agreement based on anonymous identity in cloud computing", Journal of Cloud Computing: Advances, Systems and Applications, Mar 2023
[20] Theo Zschörnig,Jonah Windolph, Robert Wehlitz, Yann Dumont, Bogdan Franczyk, "A Fog-Based Multi-Purpose Internet of Things Analytics Platform", Computer Science, Springer, Apr 2022