International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 5 Issue: 8
ISSN: 2454-4248
83 – 93

Semantically Secured Non-Deterministic Blum–Goldwasser Time-Based One-Time Password Cryptography for Cloud Data Storage Security

Kavitha K
Department of Computer Applications
Dr. N.G.P. Arts and Science College
Coimbatore,India
e-mail: kavitha.k@drngpasc.ac.in

Saravanan V
Department of Information technology
Hindusthan College of Arts And Science
Coimbatore,India
e-mail: vsreesaran@gmail.com

***Abstract*—** The security level of outsourced data is significant in cloud storage. Few research works have been designed for secured cloud data storage. However, the data security level was lower because the authentication performance was not effective. In order to overcome such drawbacks, a Semantically Secured Non-Deterministic Blum–Goldwasser Time-Based One-Time Password Cryptography (SSNBTOPC) Technique is proposed. The SSNBTOPC Technique comprises three steps, namely key generation, data encryption and data decryption for improving cloud data storage security with lower cost. Initially, in SSNBTOPC Technique, the client registers his/her detail to the cloud server. After registering, the cloud server generates the public key and secret key for each client. Then, clients in cloud encrypt their data with the public key and send the encrypted data to the cloud server for storing it in the database. Whenever the client needs to store or access the data on cloud storage, the client sends the request message to the cloud server. After getting the requests, cloud server authenticates the clients using their secret key and Time-based One-Time Password (TOTP). After the verification process, SSNBTOPC Technique allows only authorized clients to get data on cloud storage. During data accessing process, the client data is decrypted with their private key. This helps for SSNBTOPC Technique to improve the cloud storage security with a minimal amount of time. The SSNBTOPC Technique carried outs the experimental evaluation using factors such as authentication accuracy, computational cost and data security level with respect to a number of client and data. The experimental result shows that the SSNBTOPC Technique is able to increases the data security level and also reduces the computational cost of cloud storage when compared to state-of-the-art works.

Keywords- *Clients; Cloud Data Storage Security; Non-Deterministic; Random Bits; Semantically Secured; Time-Based One-Time Password (TOTP)*

******

I. INTRODUCTION

Cloud computing provides different scalable applications at any time and anywhere such as storage and platforms as service to companies, individuals and governments. Security is an essential concern for preserving the data in the cloud. Storage security is the process of preserving the stored data from the illegal user access. Many authentication techniques were designed in conventional works to make sure the identity of users on cloud storage. However, existing methods failed to improve the authentication accuracy and computational cost of existing methods was higher. In order to improve the security level during the data storage with minimal cost, SSNBTOPC Technique is designed in this research work.

Conditional identity-based broadcast proxy re-encryption (CIBPRE) was presented in [1] to keep data confidentiality in the cloud environment. However, the computational cost was higher. Security-Aware Efficient Distributed Storage (SA-EDS) model was designed in [2]. The authentication accuracy of the client was not solved.

A public auditing scheme was presented in [3] for secure storage. However, authentication efficiency was lower. A relative analysis of cryptographic mechanisms was carried out in [4] to address protection of outsourced data in cloud infrastructures. But, the processing complexity was not reduced.

Secure, flexible and efficient data storage system was presented in [5] with aim of enhancing the efficiency and security in cloud. But, time complexity involved during secure data storage was an open issue. Security-Aware Efficient Distributed Storage (SA-EDS) model was intended in [6] for secure distributed big data storage in cloud computing. However, computation cost was not solved.

An e-Stream Cipher-Based Secure and Dynamic Updation Policy was intended in [7] for providing the security to the user's sensitive data at cloud data center. An alternative approach was designed in [8] for effective and feasible to protect the big data for cloud tenants. However, the data security level was lower.

An energy-efficient block-based sharing scheme was introduced in [9] for achieving higher confidentiality and integrity services in the cloud environment. But, time employed for obtaining security during cloud storage was more. A cipher text-policy attribute-based encryption (ABE) scheme and a proxy re-encryption scheme were presented in [10] for a secure P2P storage cloud with minimal computation overheads.

 ISSN: 2454-4248
83 – 93

In order to resolve the above mentioned existing limitations in cloud data storage, SSNBTOPC Technique is designed. The main contributions of SSNBTOPC Technique are presented in below.

- To increase the cloud data storage security level through client authentication performance as compared to state-of-the-art works, SSNBTOPC Technique is developed. The SSNBTOPC Technique is proposed by combining the Time-based One-time Password (TOTP) in Non-Deterministic Blum–Goldwasser Data Encryption/ Decryption.

- To enhance the authentication performance of clients while access cloud storage when compared to conventional works, TOTP is employed in SSNBTOPC Technique on the contrary to existing works. The TOTP is a single-use password which is utilized for verifying the clients. In SSNBTOPC Technique, TOTP is no longer valid which is changed for every 30 seconds. This assists for SSNBTOPC Technique to increases the accuracy of authentication in cloud data storage.

The remaining structure of the paper is formulated as follows. Section 2 describes the literature review. Section 3 explains proposed SSNBTOPC Technique with the support of architecture diagram. Section 3 and Section 4 explains the simulation setting and comparative result analysis. Finally, Section 6 portrays the conclusion of the paper.

II. RELATED WORKS

Homomorphism encryption scheme (HES) was designed in [11] to resolve the security requirement of cloud storage. Secure Encryption Model (SEM) was presented in [12] for user verification and increasing the security.

A role-based encryption (RBE) scheme was introduced in [13] to attain user-centric secure information stored in a cloud computing environment. High-Efficiency Video Coding (HEVC) was intended in [14] for secure data exchange between the mobile users and the media clouds. A secure disintegration protocol (SDP) was developed in [15] for enhancing privacy on-site and in the cloud.

A review of various techniques developed for securing cloud storage by application of different cryptographic techniques was analyzed in [16]. A threshold proxy re-encryption scheme was introduced in [17] for robust data storage on cloud. A novel technique was presented in [18] to secure data access from cloud data center.

A reliable and secure distributed cloud data storage schema was introduced in [19] with the application of Reed-Solomon codes. Short Comparable Encryption scheme based on sliding window method (SCESW) was intended in [20] to minimize computational complexity during cloud storage. However, the ratio of number of clients that are correctly verified is lower.

III. SEMANTICALLY SECURED NON-DETERMINISTIC BLUM-GOLDWASSER TIME-BASED ONE-TIME PASSWORD CRYPTOGRAPHY TECHNIQUE

The Semantically Secured Non-Deterministic Blum–Goldwasser Time-Based One-Time Password Cryptography (SSNBTOPC) Technique is designed with the objective of enhancing the security of cloud data storage with minimal computational cost. The SSNBTOPC Technique is introduced with the application of Time-based One-time Password algorithm in Non-Deterministic Blum–Goldwasser Data Encryption/ Decryption. On the contrary to existing works, Non-Deterministic Blum–Goldwasser Data Encryption/ Decryption is employed in SSNBTOPC Technique as it is a semantically secure cryptosystem which represents only negligible information regarding the client data can feasibly extracted from the cipher text. Besides to that, Non-Deterministic Blum–Goldwasser Data Encryption/ Decryption is applied in SSNBTOPC Technique provides perfect secrecy on the contrary to existing works which denote that the cipher text reveals no information at all about the client data. In addition to that, SSNBTOPC Technique is a non-deterministic cryptography on the contrary to state-of-the-art works where the encryption of the same client data under the same public key returns a dissimilar ciphertext. From that, SSNBTOPC Technique increases the security of cloud data storage as compared to conventional storage models.

Further, Time-based one-time passwords in SSNBTOPC Technique give additional security, because even if a user's secret key is stolen or compromised, an attacker cannot gain access without the TOTP, which changes every 30 seconds. In SSNBTOPC Technique, the Time-based One-time Password algorithm generates a onetime password at the current time and sent it corresponding client in cloud environment. When the secret key and TOTP of the client is correct, SSNBTOPC Technique allows the user to obtain data from cloud storage. Thus, SSNBTOPC Technique improves the authentication performance of clients who access data on cloud storage on the contrary to existing works. The overall architecture diagram of SSNBTOPC Technique is presented in Figure 1.

Figure 1 explains the overall processes of SSNBTOPC Technique to attain cloud data storage security. As exposed in the above figure, the client at first encrypts their data using a public key with the application of Non-Deterministic Blum–Goldwasser Data Encryption (NBDE) algorithm. Then, the encrypted data is stored in a cloud server with aim of improving the security of data on cloud storage. While the client transmits a request to access the stored data, SSNBTOPC Technique ensures their secret key and then formulates TOTP if and only if the secret key is valid.

 ISSN: 2454-4248
83 – 93

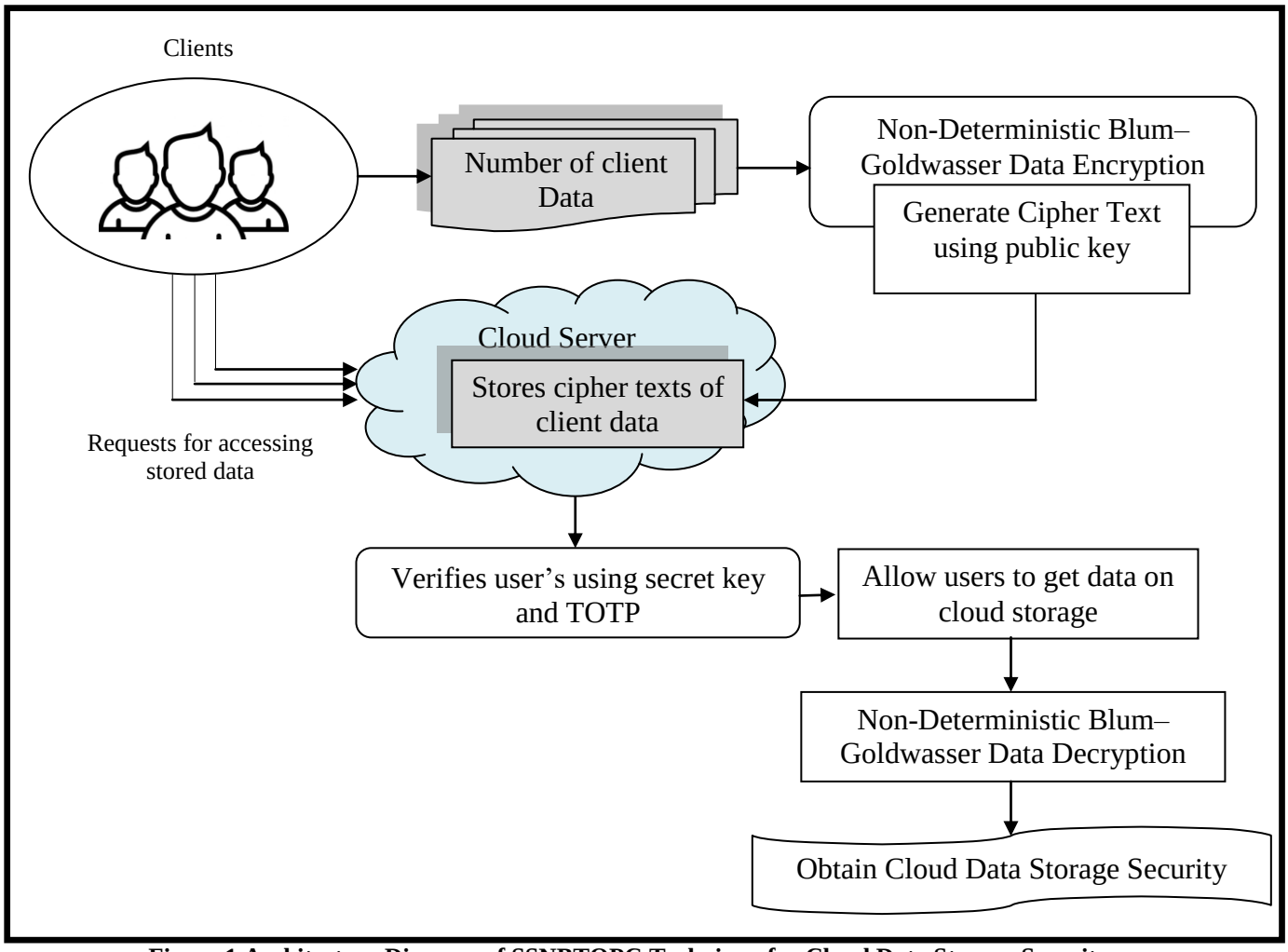

Figure 1 Architecture Diagram of SSNBTOPC Technique for Cloud Data Storage Security

This TOTP is transmitted to the corresponding client. After that, SSNBTOPC Technique authenticates TOTP of that client and permits Non-Deterministic Blum–Goldwasser Data Decryption to get data from cloud storage. Therefore, SSNBTOPC Technique enhances the security performance of cloud data storage as compared to existing works.

The SSNBTOPC Technique contains three main processes as below,

1) Key generation
2) Non-Deterministic Blum–Goldwasser Data Encryption (NBDE)
3) Non-Deterministic Blum–Goldwasser Data Decryption (NBDD)

The exhaustive processes of SSNBTOPC Technique are described in following subsections.

A. Key Generation

Generating and managing keys is a significant cryptographic process. The designed SSNBTOPC Technique is an asymmetric key cryptography. In SSNBTOPC Technique, the client registers his or her detail such as name first name, last name, date of birth (DOB), gender, mobile number, and mail-ID to the cloud server. After registering, the cloud server generates the public key and secret key for each client. Therefore, public key and a secret key are created for each client in cloud. In SSNBTOPC Technique, the public key is made public to anyone, while the private key must known only by the client who will decrypt the data encrypted with the public key. The public key of the client is used to securely store their data on cloud storage through Non-Deterministic Blum–Goldwasser Encryption. Besides, the secret key of the client is employed to ensure their identity when accessing stored data on cloud and also to decrypt the ciphertext.

Let us assume a number of clients in a cloud environment are represented as ' $C_i = C_1, C_2, \dots C_N$'. For each client in cloud, SSNBTOPC Technique produces the public and secret key. The SSNBTOPC Technique randomly selects two large prime numbers 'x'and 'y' '$x \neq y$' which independents of each other clients in cloud. Thus, the public key of client is mathematically constructed as,

$$\beta_{key} = xy \quad (1)$$

From (1), public key 'β_{key}' is generated for each client in cloud. Subsequently, secret key of client is mathematically formulated as,

 ISSN: 2454-4248
83 – 93

$$\$_{key} = f(x, y) \quad (2)$$

From (2), secret key '$\$_{key}$' is created for each client in cloud whereas '$f(x, y)$' denotes the factorization of '(x, y)'. The constructed key pair is provided to clients in cloud environment. The algorithmic steps of key generation process are presented in below.

// Key Generation Algorithm

Input: Number of Client '$C_i = C_1, C_2, \dots C_N$'

Output: Key pair '$(\beta_{key}, \$_{key})$'

Step 1: Begin

Step 2: **For** each client 'C_i'

Step 3: Randomly picks two large prime numbers 'x' and 'y'

Step 4: Generate public key 'β_{key}' using (1)

Step 5: Create private key using '$\$_{key}$' (2)

Step 6: Sent 'β_{key}' and '$\$_{key}$' to client '$C_i$'

Step 7: **End for**

Step 8:End

Algorithm 1 Key Generation

Algorithm 1 depicts the step by step algorithmic processes of Key Generation in SSNBTOPC Technique. By using the above algorithmic process, SSNBTOPC Technique makes a public and secret key for each client in cloud with a lower amount of time complexity.

B. Non-Deterministic Blum–Goldwasser Data Encryption

The storage security on cloud is essential to protect client data against illegal access. In existing works, few cryptographic techniques were designed to encrypt the data before storing it on cloud storage. However, the security level of cloud data storage was not sufficient as where may be user's secret key is stolen or compromised. In order to solve this drawback, Non-Deterministic Blum–Goldwasser Data Encryption/ Decryption is designed in SSNBTOPC Technique by integrating the Time-based one-time passwords in existing Blum–Goldwasser cryptosystem.

The Non-Deterministic Blum–Goldwasser Data Encryption (NBDE) algorithm is designed in SSNBTOPC Technique is a non-deterministic cryptography. The Non-deterministic is a specific type of encryption in which client data frequently encrypted with the same public key yields diverse cipher text. This helps for SSNBTOPC Technique to secure data on cloud storage form an unauthorized client for finding client data by comparing them to a dictionary of known ciphertexts. Then, proposed NBDE algorithm semantically secure because of the intractability of the factorization process involved key generation. Besides, NBDE algorithm is better in terms of memory space utilized for cloud data storage as compared to state-of-the art works because of constant-size ciphertext expansion. Also, the computational cost of NBDE algorithm is very lower than existing RSA. Hence, NBDE algorithm is designed in SSNBTOPC Technique for efficient cloud data storage with minimal computational cost and achieving higher security. The processes of NBDE algorithm is shown in below.

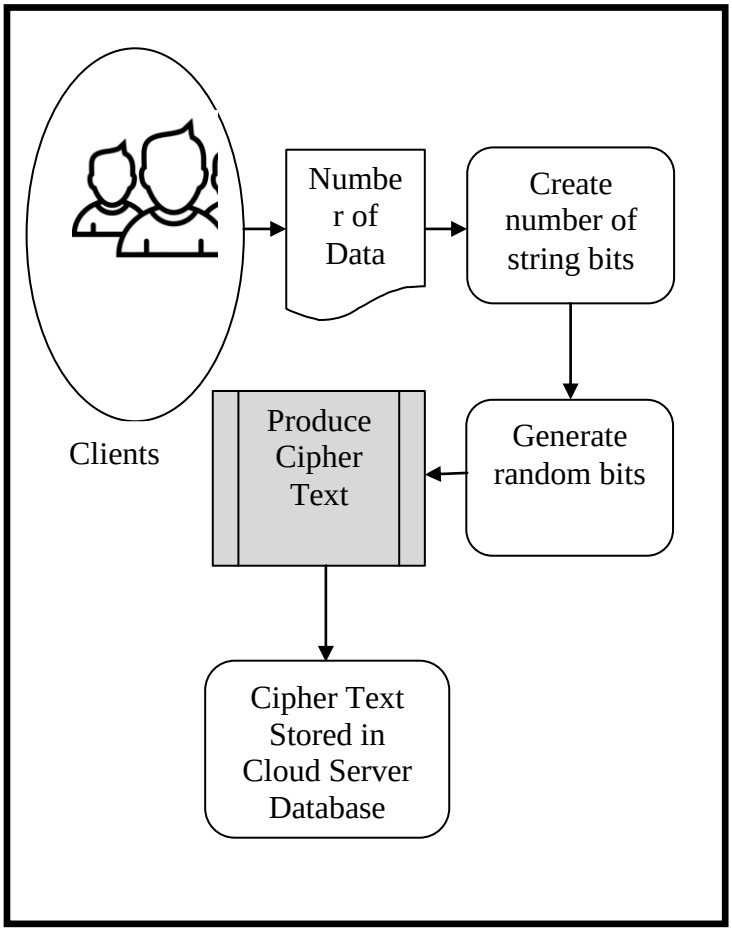

Figure 2 Processes of NBDE algorithm

Figure 2 illustrates the flow processes of the NBDE algorithm. As presented in the above figure, NBDE algorithms initially take a number of client data as input and then formulate number of string bits. After that, NBDE algorithms create the random bits for generated strings of client data. Finally, the data bits of client are XORing with random bits to create ciphertext. The generated ciphertext is securely stored in cloud server database with a minimal amount of time. Therefore, SSNBTOPC Technique attains lower computational cost for secured data cloud storage as compared to existing works.

Let us consider a number of client data denoted as '$\alpha_i = \alpha_1, \alpha_2, \dots \alpha_n$'. The NBDE at first encodes clients data is to be stored 'α' into a number of strings of 'm' bits using below mathematical expression,

$$\alpha = (\alpha_0, \alpha_1, \dots, \alpha_{m-1}) \quad (3)$$

 ISSN: 2454-4248
83 – 93

Subsequently, NBDE select a random number 'ρ' i.e. '$1 < \rho < N$' and then determines the below,

$$\gamma_0 = \rho^2 \, mod \, \beta_{key} \quad (4)$$

From (4), 'β_{key}' indicates the public key of client. Next, the NBDE employs the Blum Blum Shub (BBS) pseudo-random number generator to construct random bits '$\vec{\sigma} = \sigma_0, \sigma_1, .. \sigma_{m-1}$'. For each formulated random bits 'i' to 'm', then NBDE assigns 'σ_i' equal to least significant bit of 'γ_i' that mathematically performed as,

$$\gamma_i = (\gamma_{i-1})^2 mod \, \beta_{key} \quad (5)$$

$$\overrightarrow{\sigma_i} = LSB(\gamma_i) \quad (6)$$

From (5) and (6), '$LSB(\gamma_i)$' represents the least significant bit of 'γ_i'. Consequently, the NBDE creates the cipher text bits with the help of bits 'σ_i' from the BBS. From that, the data bits of client are XORing with random bits in order to construct cipher text which mathematically accomplished using below expression,

$$\varphi_i = \vec{\alpha} \oplus \vec{\sigma} \quad (7)$$

From (7), '$\vec{\alpha}$' denotes the data bits of client and '$\vec{\sigma}$' represents random bits whereas 'φ_i' refers to the cipher text. The algorithmic steps of NBDE is explained in below,

// Non-Deterministic Blum–Goldwasser Data Encryption Algorithm

Input: Number of Client '$C_i = C_1, C_2, \ldots C_N$'; Client Data '$\alpha_i = \alpha_1, \alpha_2, \ldots \alpha_n$'

Output: Improved Cloud Data storage Security with minimal cost

Step 1: Begin

Step 2: **For** each 'C_i'

Step 3: **For** each 'α_i'

Step 4: Generate 'm' number of string bits using (3)

Step 5: Choose 'ρ' and create '$\vec{\sigma}$' with client 'β_{key}' using (6)

Step 6: Produce ciphertext 'φ_i' using (7)

Step 7: **End for**

Step 8: **End For**

Step 9: End

Algorithm 2 Non-Deterministic Blum–Goldwasser Data Encryption

Algorithm 2 portrays the step by step processes of Non-Deterministic Blum–Goldwasser Data Encryption. With the algorithmic processes of NBDE algorithm, SSNBTOPC Technique produces ciphertext for each client data and then stored it in cloud server database to enhance the cloud data storage security level with minimal computational cost.

C. Non-Deterministic Blum–Goldwasser Data Decryption

Whenever the client wants to get the data which is stored on a cloud server, the client sent requests to the cloud server. In order to validate the identity of a client who accesses the data on cloud storage, at first the SSNBTOPC Technique verifies the secret key and then generates the time-based one-time passwords (TOTP) when the secret key of client is correct. Each TOTP is generated for use by only one client. The generated TOTP is valid for a particular period of time, and it's invalid after the logs in. Further, TOTP used in SSNBTOPC Technique cannot be easily duplicated. Thus, SSNBTOPC Technique provides an additional layer of security to ensure the identity of clients and to reduce the risk of illegal access on cloud data storage as compared to existing works with the application of TOTP.

In SSNBTOPC Technique, SSNBTOPC Technique combines a secret key with the current timestamp with help of a cryptographic hash function to make a TOTP and sent it to the corresponding client through mobile number or mail-ID. Then, the cloud server authenticates the client entered TOTP. When both the client entered and cloud server generated TOTP is matched, the client is considered as authentic and allowed for securely access the data from cloud storage. The process of NBDD is depicted in below figure.

Figure 3 depicts the flow processes of NBDD to secure the data which is stored on cloud storage from illegitimate access. As presented in the above figure, at first client sent requests to cloud server. After receiving client requests, cloud server ensures the identity of client with aid of secret key and TOTP. If both secret key and TOTP is correct, then NBDD permit the client to acquire data from cloud storage through decryption process. If both secret key and TOTP is not valid, then client is allowed to obtain data from cloud storage. From that, SSNBTOPC Technique increases the authentication performance of clients with lower time as compared to existing works.

Let us consider a number of clients represented as '$C_i = C_1, C_2, \ldots C_N$' and their requests to cloud server is denoted as '$CR_i = CR_1, CR_2, \ldots . CR_n$'.

For each client request, NBDD at first verifies the secret key using below,

$$z = \begin{cases} If \; (\$_{key} == \$^*_{key}), & then \; ToTP \; is \; generated \\ else & ToTP \; is \; not \; generated \end{cases} \quad (8)$$

ISSN: 2454-4248
83 – 93

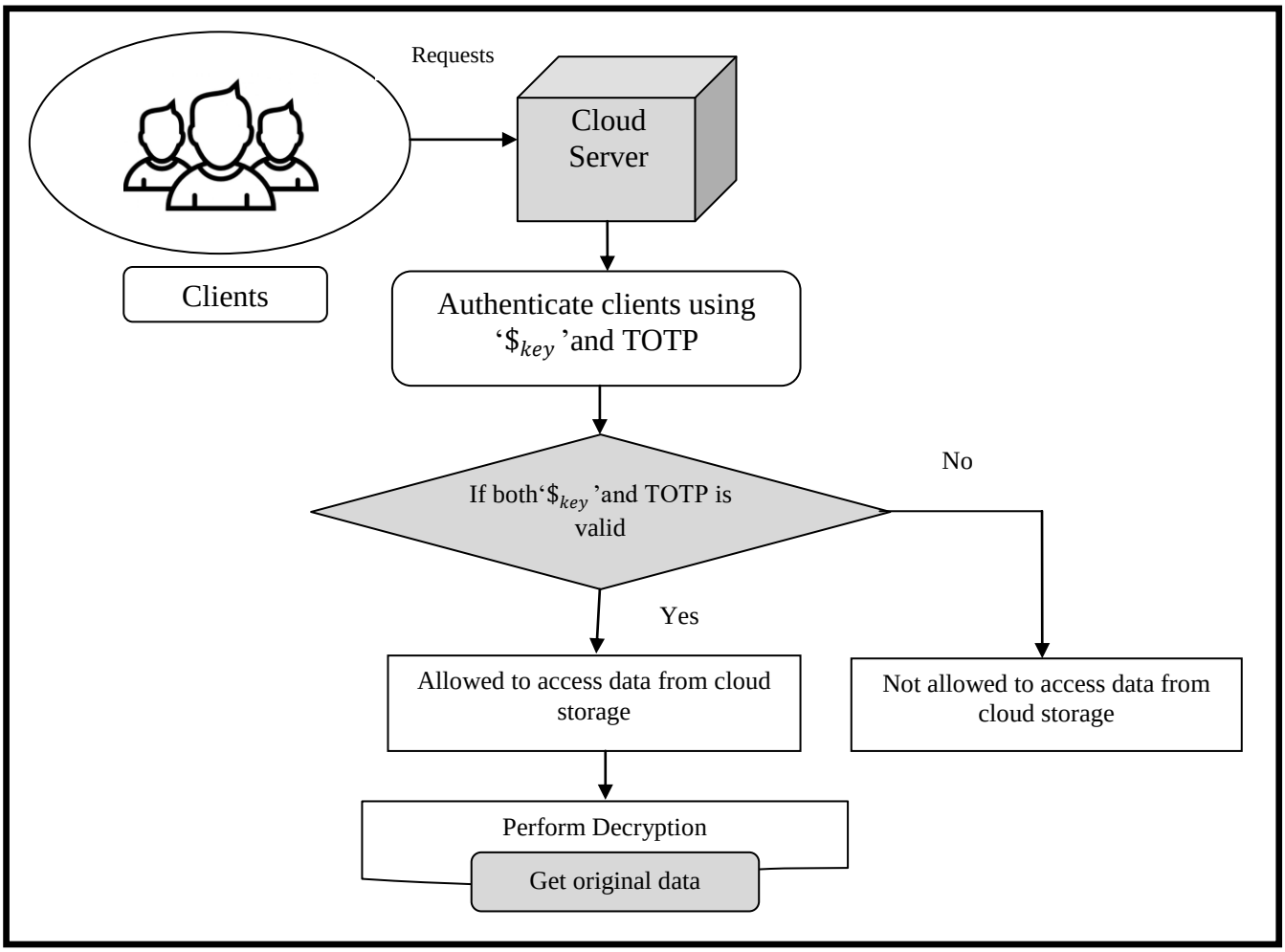

Figure 3 Flow processes of NBDD

From (8), 'z' returns the secret key authentication result where '$\$_{key}$' denotes the client entered secret key at the time of login and '$\$^*_{key}$' refers the secret key of corresponding client is stored in cloud server database. If both the client entered secret key '$\$_{key}$' and secret key that stored in cloud server ''$\$^*_{key}$' is identical, NBDD generates the TOTP.

The SSNBTOPC Technique employs HOTP (HMAC-based One-time Password) algorithm to produce TOTP. Thus, TOTP of client is mathematically evaluated using below expression,

$$TOTP = HOTP(\$_{key}, tc)) \quad (9)$$

$$HOTP\ value = HOTP\ mod\ 10^d \quad (10)$$

From (9) and (10), TOTP is generated for every time step where '$HOTP$' represents the HMAC-based One-time Password algorithm whereas '$\$_{key}$' denotes the secret key of client and 'tc' indicates the time counter. Here, 'd' point outs the desired number of digits of TOTP. The created TOTP of client is changed for every 30 seconds.

In SSNBTOPC Technique, the time counter is an integer that counting the number of durations. The time counter measures the differentiation between the current Unix time and some epoch. The time counter 'tc' is mathematically measured using below formulation,

$$tc = (t - t_o / t_s) \quad (11)$$

From (11), 't' represents the current time and 't_o' start of an epoch (T0) and 't_s' denotes the time step. While the both secret key and TOTP is legitimate, the SSNBTOPC Technique considers the client as authentic to securely access the data from cloud storage.

If the client in cloud computing environment is authentic, SSNBTOPC Technique allowed to Non-Deterministic Blum–Goldwasser Data Decryption (NBDD) process. The client acquires original data with help of their secret key '$\$_{key}$'. In order to decrypt the cipher text, initial seed 'a_0' of client data is obtained using below,

$$a_0 = ((y(y^{-1}\ mod\ x)\rho_x + xx{-}1\ mod\ y\rho y\ mod\ \$key \quad (12)$$

 ISSN: 2454-4248
83 – 93

From (12), 'ρ_x' and 'ρ_y' is evaluated with aid of prime factorization '(x, y)' as follows,

$$\rho_x = \gamma_L{}^{((x+1)/4)^m} mod\ x \quad (13)$$

$$\rho_y = \gamma_L{}^{((y+1)/4)^m} mod\ y \quad (14)$$

From (13) and (14), then bit-vectors $\vec{\sigma}$ of client data are re-determined with application of BBS generator. Followed by, client original data is obtained by XORing random bits '$\vec{\sigma}$' with cipher texts '$\overrightarrow{\varphi_i}$'. From that, the data decryption is performed mathematically as,

$$\alpha_i = \overrightarrow{\varphi_i} \oplus \vec{\sigma} \quad (15)$$

From (15), original data of client 'α_i' is re-generated. The algorithmic steps of NBDD are described in below.

// **Non-Deterministic Blum–Goldwasser Data Decryption Algorithm**

Input: Number of Clients '$C_i = C_1, C_2, \ldots C_N$'; Client Request '$CR_i = CR_1, CR_2, \ldots . CR_n$'; Cipher Text '$\varphi_i = \varphi_1, \varphi_2, \ldots . \varphi_{m-1}$'

Output: Improved authentication accuracy for Cloud Data Storage Security

Step 1: Begin

Step 2: **For** each 'CR_i' // **Authentication**

Step 3: Verify 'C_i' with their '$\$_{key}$' using (8)

Step 4: **If** '$\$_{key}$' is valid, **then**

Step 5: Generate 'OTP' for secret key '$\$_{key}$' using (9), (10), (11)

Step 6: Sent 'OTP' to the corresponding client

Step 7: **If** 'C_i' entered 'OTP' is valid, **then**

Step 8: 'C_i' is allowed for obtaining data from cloud storage

Step 9: Data decryption process is permitted

Step 10: **Else**

Step 11: 'C_i' is not allowed for getting data from cloud storage

Step 12: Data decryption process is not allowed

Step 13: **Endif**

Step 14: **End If**

Step 15: **End For**

Step 16: **For** each ciphertext 'φ_i' // **Data Decryption**

Step 17: Find 'a_0' with '$\$_{key}$'of client using (12)

Step 18: Re-evaluate '$\vec{\sigma}$' using (13) and (14)

Step 19: Original client data 'α_i' using (15)

Step 20: **End for**

Step 21: End

Algorithm 3 Non-Deterministic Blum–Goldwasser Data Decryption

Algorithm 3 shows the step by step algorithmic processes of NBDD to get enhanced authentication performance when acquiring the data from cloud storage. For each client request made to a cloud server, initially, NBDD performs authentication where secret key and TOTP of client is ensured. Then NBDD allows the client to obtain the data from cloud storage when the secret key and TOTP is authentic. Thus, NBDD increases the authentication accuracy of client who desires to access the data on cloud storage with a lower amount of time. When the client is authorized person, decryption is permitted to decrypt the ciphertext of client data with their secret key. As a result, SSNBTOPC Technique obtains improved data security level for cloud storage with minimal computational cost.

IV. EXPERIMENTAL SETTINGS

The SSNBTOPC Technique is implemented in Java Language with cloudsim simulator using personal cloud dataset [21] in order to estimate the proposed performance. The personal cloud dataset comprises of 17 fields namely file size (i.e. client data), operation_time_start, operation_time_end, bandwidth trace, node_ip, node_name, quoto_total (storage capacity). For conducting the experimental evaluation, SSNBTOPC Technique considers the different number of client data. The experimental result of SSNBTOPC Technique is compared against with two state-of-the-art works namely conditional identity-based broadcast proxy re-encryption (CIBPRE) [1] and Security-Aware Efficient Distributed Storage (SA-EDS) model [2]. The experimental performance of SSNBTOPC Technique is estimated in terms of authentication accuracy, computational cost, and data security level.

V. RESULT AND DISCUSSIONS

In this section, the performance result analysis of SSNBTOPC Technique is discussed. The efficiency of SSNBTOPC Technique is compared with existing namely conditional identity-based broadcast proxy re-encryption (CIBPRE) [1] and Security-Aware Efficient Distributed Storage (SA-EDS) model [2] using metrics below.

A. Performance Measure of Authentication Accuracy

In SSNBTOPC Technique, authentication accuracy determined as the ratio of a number of clients that are correctly verified as authorized or unauthorized to the total number of clients. The authentication accuracy 'AA' is mathematically evaluated using below,

$$AA = \frac{\mu_{CA}}{N} * 100 \quad (16)$$

From (16), the authentication performance of SSNBTOPC Technique is measured with respect to a diverse number of clients. Here, 'μ_{CA}' denotes the number

 ISSN: 2454-4248
83 – 93

of clients correctly authenticated as authorized or unauthorized and 'N' denotes the total number of clients. The authentication accuracy is estimated in terms of percentages (%).

Sample Calculation:

- **Existing CIBPRE:** number of clients that are correctly ensured as authorized or unauthorized is 19 and the total number of clients is 25. Then authentication accuracy is calculated as follows,

$$AA = \frac{19}{25} * 100 = 76\ \%$$

- **Existing SA-EDS:** number of clients accurately verified as legal or illegal is 17 and the total number of clients is 25. Then authentication accuracy is evaluated as follows,

$$AA = \frac{17}{25} * 100 = 68\ \%$$

- **Proposed SSNBTOPC:** number of clients accurately authenticated is 21 and the total number of clients is 25. Then authentication accuracy is obtained as follows,

$$AA = \frac{21}{25} * 100 = 84\ \%$$

In order to evaluate the accuracy of client's verification during data access process from cloud storages, SSNBTOPC Technique is implemented in Java language with a different number of clients in the range of 25-250. When taking 150 numbers of clients to conduct experimental process, SSNBTOPC Technique attains 91 % authentication accuracy where conventional CIBPRE [1] and SA-EDS model [2] acquires 83 % and 81% respectively. From these result, it is expressive that the authentication accuracy for secured cloud data storage using SSNBTOPC Technique is higher than other state-of-the-art works. The comparative result analysis of authentication accuracy is shown in below.

Figure 4 presents the impact of authentication accuracy versus a number of clients using three methods namely CIBPRE [1] and SA-EDS [2] and proposed SSNBTOPC technique. As depicted in above figure, proposed SSNBTOPC technique achieves enhanced authentication accuracy in order to ensure the identity of clients who access the cloud data storage as compared to existing CIBPRE [1] and SA-EDS [2].

This is due to the application of TOTP in Blum–Goldwasser cryptography in SSNBTOPC technique on the contrary to existing works. TOTP applied in SSNBTOPC technique for authenticating clients. This TOTP is valid for limited time duration. Once expired, the TOTP is no longer valid. When the client enters a valid TOTP into a login form together with his username and secret key, SSNBTOPC technique accurately verifies the client as a legitimate person.

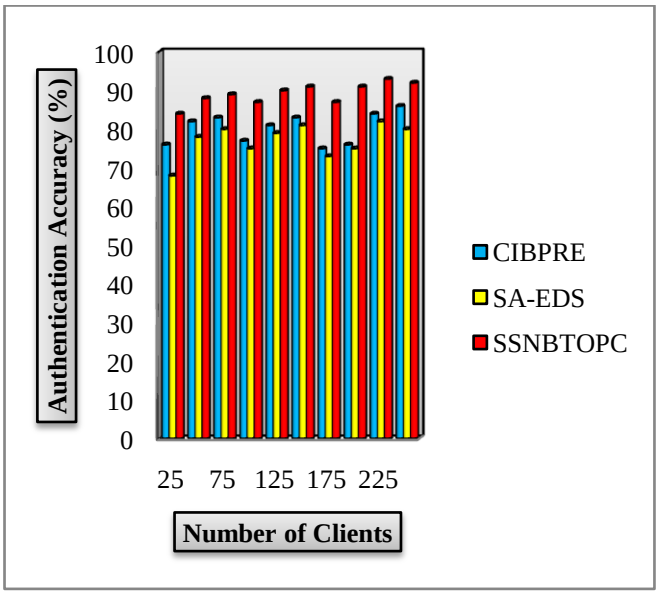

Figure 4 Measurement of Authentication Accuracy versus Number of Clients

This helps for SSNBTOPC technique to enhance the ratio of a number of clients that are correctly verified as authorized or unauthorized as compared to state-of-the-art works. Hence, SSNBTOPC technique improves the authentication accuracy of cloud data storage by 11 % and 16 % as compared to conventional CIBPRE [1] and SA-EDS [2] respectively.

B. Performance Measure of Computational Cost

Computational Cost measures the amount of time needed to attain secure cloud data storage. The computational cost is 'CC' mathematically obtained as,

$$CC = n * t\ (SCDS) \qquad (17)$$

From (17), computational cost of SSNBTOPC Technique is determined with respect to different number of client data where 'n' denotes number of client data and '$t\ (SCDS)$' refers the time taken for securely storing data on cloud storage. The computational cost is determined in terms of milliseconds (ms).

Sample Calculation:

- **Existing CIBPRE:** time required for secured cloud data storage is 1.1 ms and the total number of client data is 20. The computational cost is formulated as follows,

$$CC = 20 * 1.1 = 22\ ms$$

- **Existing SA-EDS:** time employed for secured cloud data storage is 1.25 ms and the total number of client data is 20. Then, computational cost is measured as follows,

$$CC = 20 * 1.25 = 25\ ms$$

- **Proposed SSNBTOPC:** time used for secured cloud data storage is 0.9 ms and the total number of

 ISSN: 2454-4248
83 – 93

client data is 20. Then, computational cost is calculated as follows,

$$CC = 20 * 0.9 = 18\ ms$$

The SSNBTOPC Technique is implemented in Java language by considering a various number of client data in the range of 20-200 to measure the computational cost involved during secured cloud data storage. When employing 160 numbers of client data to carry out experimental work, SSNBTOPC Technique gets 46 ms computational cost where state-of-the-art works CIBPRE [1] and SA-EDS model [2] obtains 54 ms and 77 ms respectively. From the above results, it is significant that the computational cost of secured cloud data storage using SSNBTOPC Technique is lower than other conventional works. The tabulation result analysis of computational cost is described in below.

Number of Client Data	Computational Cost (ms)		
	CIBPRE	SA-EDS	SSNBTOPC
20	22	25	18
40	32	38	24
60	36	45	21
80	35	48	31
100	40	55	36
120	46	76	40
140	50	69	43
160	54	77	46
180	58	81	49
200	60	80	54

Table 1 Tabulation for Computational Cost

Table 1 depicts the impact of computational cost versus the number of client data using three methods namely CIBPRE [1] and SA-EDS [2] and proposed SSNBTOPC technique. As exposed in the above table, proposed SSNBTOPC technique attains lower computational cost to accomplish secured cloud data storage as compared to existing CIBPRE [1] and SA-EDS [2]. This is because of the application of NBDE algorithm in SSNBTOPC technique on the contrary to conventional techniques where it takes client data as input and then encrypts the data with aid of public key of the client. The NBDE algorithm generates the ciphertext through XORing the data bits of client with random bits constructed. Then, the created ciphertext is securely stored on cloud server database with a lower amount of time utilization. This supports for SSNBTOPC technique to utilize the minimum amount of time for secure cloud data storage. Thus, SSNBTOPC technique minimizes the computational cost involved during secured cloud data storage by 17 % and 39 % as compared to conventional CIBPRE [1] and SA-EDS [2] respectively.

C. Performance Measure of Data Security Level

The data security level is measured as the ratio of number of data on cloud storage that are obtained only by authentic clients to the total number of client data. The data security level 'DSL' is mathematically obtained as,

$$DSL = \frac{\mu_{AAC}}{n} * 100 \quad (18)$$

From (18), data security level is determined with respect to a various number of client data. Here, 'μ_{AAC}' point outs the number of data get only by authorized clients and 'n' denotes the total number of client data. The data security level is estimated in terms of percentages (%).

Sample Calculation:

- **Existing CIBPRE:** number of data obtained only by legitimate clients is 15 and the total number of client data is 20. Then data security level is estimated as follows,

$$DSL = \frac{15}{20} * 100 = 75\ \%$$

- **Existing SA-EDS:** number of data accessed only by genuine clients is 12 and the total number of client data is 20. Then data security level is calculated as follows,

$$DSL = \frac{12}{20} * 100 = 60\ \%$$

- **Proposed SSNBTOPC:** number of data obtained only by authentic clients is 17 and the total number of client data is 20. Then data security level is measured as follows,

$$DSL = \frac{17}{20} * 100 = 85\ \%$$

For determining the security level of data on cloud storage, SSNBTOPC Technique is implemented in Java Language with help of a varied number of client data in the range 20-200. When considering 180 numbers of client data to accomplish experimental evaluation, SSNBTOPC Technique acquires 92 % data security level where state-of-the-art works CIBPRE [1] and SA-EDS model [2] achieves 89 % and 88 % respectively. From the results obtained, it is clear that the security level on cloud data storage using SSNBTOPC Technique is higher than other conventional works. The experimental result analysis of data security level is presented in below.

Figure 5 depicts the impact of data security level versus a number of client data using three methods namely CIBPRE [1] and SA-EDS [2] and proposed SSNBTOPC technique. As shown in the above figure, proposed SSNBTOPC technique obtains improved data security level

 ISSN: 2454-4248
83 – 93

on cloud storage when compared to existing CIBPRE [1] and SA-EDS [2]. This is owing to the application of NBDD algorithm in SSNBTOPC technique on the contrary to existing works where it ensures the secret key and TOTP of each client who sent the requests to the cloud server.

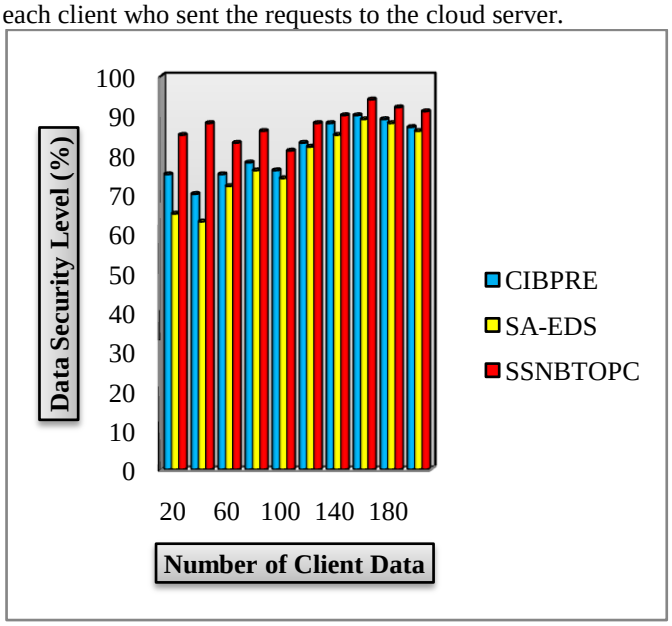

Figure 5 Measurement of Data Security Level versus Number of Client Data

The NBDD algorithm permits only authorized clients to get the data from cloud storage when the secret key and TOTP is legitimate. Furthermore, NBDD algorithm in SSNBTOPC technique allows data decryption process if the client is the authoritative person in cloud. From that, SSNBTOPC Technique significantly avoids the unauthorized access data on cloud storage as compared to conventional works. As a result, SSNBTOPC technique increases the security level of cloud data storage by 9 % and 14 % as compared to conventional CIBPRE [1] and SA-EDS [2] respectively.

VI. CONCLUSION

An effective SSNBTOPC technique is designed with the goal of increasing the cloud data storage security level with lower computational cost via client authentication. The goal of SSNBTOPC technique is obtained with aid of Non-Deterministic Blum–Goldwasser Data Encryption/ Decryption algorithm and TOTP. By using the algorithmic process of NBDE and NBDD, SSNBTOPC technique significantly increases the data security level of cloud storage with a reduced amount of time when compared to state-of-the-art works. Furthermore, with the application of TOTP and NBDD algorithm, the developed SSNBTOPC technique enhances the authentication accuracy in order to preserve the data on cloud storage from unauthorized user access. As a result, SSNBTOPC technique attains enhanced data security level for cloud storage. The effectiveness of SSNBTOPC technique is evaluated in terms of authentication accuracy, computational cost and data security level and compared against with state of the art works. The experimental result demonstrates that SSNBTOPC technique provides better performance for cloud storage with an enhancement of data security level and minimization of computational cost as compared to state-of-the-art works.

REFERENCES

[1] Wei Wang, Peng Xu, Laurence T. Yang, “Secure Data Collection, Storage and Access in Cloud-Assisted IoT”, IEEE Cloud Computing, Volume 5, Issue 4, Pages 77 – 88, 2018

[2] Li Yibin, Keke Gai, Longfei Qiuc Meikang Qiu, Zhao Hui, “Intelligent cryptography approach for secure distributed big data storage in cloud computing”, Information Sciences, Elsevier, Volume 387, Pages 103-115, May 2017

[3] Hui Tian, Yuxiang Chen, Chin-Chen Chang, Hong Jiang, Yongfeng Huang, Yonghong Chen and Jin Liu “Dynamic-Hash-Table Based Public Auditing for Secure Cloud Storage”, IEEE Transactions on Services Computing, Volume 10, Issue 5, Pages 701 – 714, September-October 2017

[4] Nesrine Kaaniche and Maryline Laurent, “Data Security and Privacy preservation in Cloud Storage Environments based on Cryptographic Mechanisms”, Computer Communications, Elsevier, Volume 111, Pages 120-141, October 2017

[5] Jun-Song Fu; Yun Liu; Han-Chieh Chao; Bharat K. Bhargava; Zhen-Jiang Zhang, “Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing”, IEEE Transactions on Industrial Informatics, Volume 14 , Issue 10 , Pages 4519 – 4528, 2018

[6] Dharavath Ramesh, Rahul Mishra, Damodar Reddy Edla, “Secure Data Storage in Cloud: An e-Stream Cipher-Based Secure and Dynamic Updation Policy”, Arabian Journal for Science and Engineering, Springer, Volume 42, Issue 2, Pages 873–883, February 2017

[7] Hongbing Cheng; Chunming Rong; Kai Hwang; Weihong Wang; Yanyan Li, “Secure big data storage and sharing scheme for cloud tenants”, China Communications, Elsevier, Volume 12 , Issue 6 , Pages 106 – 115, June 2015

[8] Abdul Nasir Khan, M. L. Mat Kiah, Mazhar Ali, Sajjad A. Madani, Atta ur Rehman Khan, Shahaboddin Shamshirband, “BSS: block-based sharing scheme for secure data storage services in mobile cloud environment”, The Journal of Supercomputing, Springer, Volume 70, Issue 2, Pages 946–976, November 2014

[9] Heng He; Ruixuan Li; Xinhua Dong; Zhao Zhang, “Secure, Efficient and Fine-Grained Data Access Control Mechanism for P2P Storage Cloud”, IEEE Transactions on Cloud Computing, Volume 2, Issue 4, 2014

[10] Jian Zhang,Yang Yang, Yanjiao Chen, Jing Chen, Qian Zhang, “A general framework to design secure cloud storage protocol using homomorphic encryption scheme”, Computer Networks, Elsevier, Volume 129, Pages 37-50, December 2017

[11] S. Srisakthi, A. P. Shanthi, “Design of a Secure Encryption Model (SEM) for Cloud Data Storage Using Hadamard Transforms”, Wireless Personal Communications, Springer, Volume 100, Issue 4, Pages 1727–1741, June 2018

 ISSN: 2454-4248
83 – 93

[12] Lan Zhou; Vijay Varadharajan; Michael Hitchens, "Enforcing Role-Based Access Control for Secure Data Storage in the Cloud", The Computer Journal, Volume 54, Issue 10, Pages 1675 – 1687, 2011
[13] Muhammad Usman, Mian Ahmad Jan, Xiangjian He, "Cryptography-based secure data storage and sharing using HEVC and public clouds", Information Sciences, Elsevier, Volume 387, Pages 90-102, May 2017
[14] Bharat S. Rawal, V. Vijayakumar, Gunasekaran Manogaran, R. Varatharajan, Naveen Chilamkurti, "Secure Disintegration Protocol for Privacy Preserving Cloud Storage", Wireless Personal Communications, Springer, Pages 1–17, 2018
[15] Lan Zhou, Vijay Varadharajan, Michael Hitchens, "Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage", IEEE Transactions on Information Forensics and Security, Volume 8, Issue 12, Pages 1947 – 1960, 2013
[16] Yong Peng, Wei Zhao, Feng Xie, Zhong-Hua Dai, Yang Gao, Dong-Qing Chen, "Secure cloud storage based on cryptographic techniques", The Journal of China Universities of Posts and Telecommunications, Elsevier, Volume 19, Supplement 2, Pages 182-189, October 2012
[17] Hsiao-Ying Lin, Wen-Guey Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding", IEEE Transactions on Parallel and Distributed Systems, Volume 23 , Issue 6 , Pages 995 – 1003, June 2012
[18] Sultan Ullah and Zheng Xuefeng, "TCLOUD: A Trusted Storage Architecture for Cloud Computing", International Journal of Advanced Science and Technology, Volume 63, Pages 65-72, 2014
[19] Haiping Xu and Deepti Bhalerao, "Reliable and Secure Distributed Cloud Data Storage Using Reed-Solomon Codes", International Journal of Software Engineering and Knowledge Engineering, Volume 25, Volume 09, Issue 10, Pages 1611-1632, 2015
[20] Qian Meng, Jianfeng Ma, Kefei Chen, Yinbin Miao, and Tengfei Yang, "Comparable Encryption Scheme over Encrypted Cloud Data in Internet of Everything", Security and Communication Networks, Volume 2017, Article ID 6430850, Pages 1-11, 2017
[21] Personal Cloud Datasets: http://cloudspaces.eu/results/datasets