Secured Cloud Data Migration Technique by Competent Probabilistic Public Key Encryption

M. G. Aruna[1,*], K. G. Mohan[2]

[1] Associate Professor, Department of Computer Science and Engineering, M S Engineering College, Affiliated to VTU, Bengaluru, Karnataka, 562110, India

[2] Professor and Head, Department of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, 560064, India

The corresponding author, email: arunamg110@gmail.com

Abstract: Cloud computing, a recently emerged paradigm faces major challenges in achieving the privacy of migrated data, network security, etc. Too many cryptographic technologies are raised to solve these issues based on identity, attributes and prediction algorithms yet; these techniques are highly prone to attackers. This would raise a need of an effective encryption technique, which would ensure secure data migration. With this scenario, our proposed methodology Efficient Probabilistic Public Key Encryption (EPPKE) is optimized with Covariance Matrix Adaptation Evolution Strategies (CMA-ES). It ensures data integrity through the Luhn algorithm with BLAKE 2b encapsulation. This enables an optimized security to the data which is migrated through cloud. The proposed methodology is implemented in Open Stack with Java Language. It achieves better results by providing security compared to other existing techniques like RSA, IBA, ABE, PBE, etc.

Keywords: Luhn algorithm; encryption; efficient probabilistic public key encryption (EPPKE); covariance matrix adaptation evolution strategies (CMA-ES); trusted third party (TTP)

Received: Dec. 17, 2018
Revised: Sep. 6, 2019
Editor: Bo Cheng

I. INTRODUCTION

Cloud computing, the recent technology has its primary goal as to enable security, information accumulation and net processing, administration, with all registering resources envisioned as administrations and finally conveyed over the Internet [1] [2]. . In order to maximize the potentiality of cloud computing, top computing cloud service providers initiate the global association. Thereby it enhances the energy efficiency of data centers by minimizing the environmental impact caused with high energy consumption of cloud infrastructures [3].

End-users utilize the services offered by the cloud service providers without exact knowledge about where the resources of such services are located, possibly in other legislative domains. This becomes the root-cause for several issues when disputes occurs [4]. To deal with these issues, cloud computing provides three delivery models, such as SaaS (Software as a Service), Paas (Platform as a Service) and IaaS (Information as a Service). Amid SaaS is a dominant delivery model to meet with the requirements of enterprise IT services. Yet, lack of visibility made the enterprises to feel

uncomfortable with the SaaS model [5]. Companies that deal with cloud computing hand over their data to third-party service providers, who store and process such data in the cloud and whose physical location is hidden and is placed anywhere in the world. This becomes possibly a problem [6]. In order to appropriately identify and assess the risks which are introduced to an organization while on using cloud computing, the Economist's Business Risk model uses four key things: access, availability, infrastructure, and integrity [7].

In cloud federation, the restrictions on data storage and its access differs by states within the same country, or between countries. Hence it is an impossible task to fully harmonize the privacy and data protection rules internationally [8]. A trusted third party, tasked with assured specific security characteristics within a cloud environment is deployed with the integration of Public Key Infrastructure (PKI), Lightweight Directory Access Protocol (LDAP) and Single-Sign-On (SSO) mechanisms to ensure the authentication, integrity and confidentiality of involved data and communications [8].

Dealing such issues during distribution of information to web users in an efficient and cost-effective manner is a challenging task [9]. In order to deal with the challenges faced with cloud computing, a polynomial-time optimal offline algorithm is used. It reduces the cost required for the migration of geo-dispersed big data to the cloud [10].

Moreover, heterogeneities caused with resource mapping are considered as another challenging issue while dealing with the Virtual Machines (VMs) and the Physical Machines (PMs). The deployment of skewness concept offer several solutions to avoid the heterogeneity problem by measuring the uneven utilization of multi-dimensional resources of a server [11]. Deploying an autonomous system in a cloud infrastructure to effectively provision the resource is a challenging task because of the unpredictable consumer demands, software and hardware failures, heterogeneity of services, power management, and conflicting signed SLAs between consumers and service providers [12]. When considering the medical field, due to the availability of high valued sensitive Personal Health Information (PHI), the third-party storage servers are often the targets of various malicious behaviors which may lead to exposure of the PHI [13]. Conversely, the encryption schemes used with cloud services made cloud storage more prone to attackers. As the issues with encryption are overwhelmed with homomorphic encryption strategy, the fully homomorphic encryption schemes too becomes inefficient due to its complexity in limiting the data size of a program [14].

A vulnerability is cloud specific if it is natural or common in a central cloud computing innovation or has its main driver in one of NIST's key cloud qualities or is brought about when cloud developments make attempted and tested security controls troublesome or difficult to execute, or is predominant in setting up the best in class cloud offerings [15]. Conventional security systems, for example, personality, validation, and approval are no more enough for mists in their present structure [16]. Be that as it may, a need of trust between cloud clients and suppliers has upset the widespread acknowledgment of the mists as outsourced registering administrations. To advance multi-tenancy, we must outline the cloud computing system to be secure, reliable, and tried and true [17]. Shared and conveyed assets in the cloud frameworks make it difficult to build up a security model for guaranteeing the information security and protection [18]. But the security and protection assurance administrations can be accomplished with the assistance of secure cloud application administrations [19].

Thus several characteristics of cloud computing has its impact on security and privacy of data during migration which leads to some potential concerns. Prior methodologies have deployed many cryptographic techniques to prevent the data from outsider attacks and leakage. But all the efforts get fail, due to the weak encryption process. Moreover, the

The paper presented a secure data migration technique called EPPKE optimized with CMA-ES.

user-cloud registration process allows the attacker to leverage the data by offensive cloud models such as Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Similarly, the threat is also considered to be issue or threat occurs while migrating a data from cloud to cloud. The data migrated being the application data or any other data and this is done for reasons such as data center relocation, server updating, etc. Hence it is essential to develop an effective computing infrastructure with enhanced data encryption techniques ensuring secured data migration within the cloud environment. Hence in this paper a novel methodology for secure data migration in cloud computing infrastructure with a cryptographic method called Randomized Optimal Cryptographic Technique (ROCT) is innovated. Moreover, the key for encryption is generated with an Efficient Probabilistic Public Key Encryption (EPPKE) algorithm using CMA-ES. To check the authenticity of the cloud user, the algorithm called Luhn algorithm with BLAKE 2b is run to retrieve or modify the uploaded data by the user verifying the user ID.

1.1 Structure of paper

The remaining of the paper is structured as follows: ***Section 2*** discuss about the literature review of the secure data migration in cloud computing. Our proposed methodology is detailed in ***Section 3*** while in ***Section 4*** the implementation of our method and the experimental results are discussed and finally the ***Section 5*** concludes the paper.

II. Related Work

The very recent works related to secure data Migration are discussed as follows:

Feng Wang *et al*. [20] had explained a Cloud-Assisted Live Media Streaming (CALMS) framework for the migration purposes in a cost efficient manner in the cloud. The framework was allowed the cloud servers to house different dynamics of the user requests. They provided best solutions for the processes performed by the cloud servers in a real world platform. They stated that their method enabled numerous migrations of conventional streaming systems. They also developed some practical solutions for purposes such as user redirection and cloud server organization etc. They performed the simulation experiments on real data traces from both cloud service providers (Amazon EC2 and Spot Cloud) and a live media streaming service provider (PPTV). They showed that the framework tackles the cost associated with complete system deployment but still some latency occurred.

Xuanjia Qiu *et al*. [21] had depicted a dynamic control algorithm for the perfect placement of contents and dispatch requests in a hybrid cloud infrastructure comprises public cloud and private cloud which minimized the operational cost of the overall process by means of joint content placement and load distribution algorithm. They achieved this by efficient scheduling of the content Migration and dispatching with Lyapunov optimization theory. They have showed the ideality of their algorithm based on some theoretical analysis and with some prototype model. The results showed that the response times were elegantly bounded by the optimization algorithm. This process provides lacking in providing security.

Michael Menzel *et al*. [22] had claimed a framework to facilitate the migration of multi component web applications by extending the Cloud Genious framework. They identified the most important selection criteria, selection goals, and cloud service alternatives, considering the use case of migration on a web application cluster to public cloud services such as Amazon EC2 and Go Grid. They explained a hybrid decision making approach that combines multi-criteria decision making (AHP) and evolutionary optimization techniques (genetic algorithms (GAs)) for selecting best computation service and VM image. They also carried out a comprehensive experimental evaluation based on a realistic scenario for verifying the performance of the proposed decision making technique. They conducted experiments by implementing Cumulus Ge-

nius, a prototype of the selection algorithm and the GA deployable on Hadoop clusters. Experiments with Cumulus Genius give time complexities of the GA.

Yan Zhu *et al.* [23] had explained a method to construct an RBAC-compatible attribute-based data access control for cloud storage service to provide a user-friendly and easy-to-manage secure Attribute-Based Access Control (ABAC) mechanism. Similar to role hierarchies in RBAC, attribute hierarchies were introduced by using Attribute-Based Encryption (ABE) in order to define a seniority relation among all values of an attribute, whereby a user holding senior attribute values acquired permissions of his/her juniors. Based on these notations, they presented a new ABE scheme called Attribute-based Encryption with Attribute Hierarchies (ABE-AH) to provide an efficient approach to implement comparison operations between attribute values on a poset derived from an attribute lattice. By using bilinear groups of a composite order, they presented a practical construction of ABE-AH based on forward and backward derivation functions. Compared with prior solutions, their scheme offered a compact policy representation approach that could significantly reduce the size of private-keys and cipher texts. To demonstrate how to use the presented solution, they illustrate how to provide richer expressive access policies to facilitate flexible access control for data access services in clouds. The process is less expressive of security ensurement.

Mazhar Ali *et al*. [24] had depicted a Data Security for Cloud Environment with Semi-Trusted third party (DaSCE) which was the system developed to secure the data when the problem of leakage of data arouse. The system developed by them gives some functions such as (i) Management of key (ii) access control and (iii) file certain deletion. For the management of key they employed the Shamir's (k,n) threshold scheme and they generated the key with k out of n shares where they utilized more number of key managers to host one share of the key by each of the managers. The need for the multiple key managers is that the cryptographic key failure at any single points is avoided. Leakage of data occurs.

Alsalhi [25] described a novel scheme for QuBits steganography based on adaptive neural networks. Steganography based on qubits string along with the adaptive neural networks with the recycling of the modified particle swarm optimization algorithm, and using the enhanced general controlled NOT gate and NEQR representation model with the optimal target of the quantum ANNS (QANNs). In this scheme, the cover image is trained to be more accrued. Then in the obtained stego file, coefficients are classified based on their XORs. The suggested scheme avoids attacking of the sensitive data in a way that receiver can extract the information without any errors. Considering the preformed classification, secret qubits will not be revealed in the transferring process and then with the use of inverse extracting, stego file will be obtained. The most important features that our work obtained are good adaptation with human vision system and retrieval of data without getting error.

By the overall analysis, the work of Feng Wang *et al.* [20] and Xuanjia Qiu *et al.* [21] depicted the ideal solutions to minimize the cost for the processes performed by the cloud servers, but at the same time it fails to ensure security. Michael Menzel *et al.* [22] and Yan Zhu *et al*. [23] explained a framework to facilitate the migration of multi component web applications that illustrates about the access policies to facilitate flexible data access control in clouds. Though it ensures data access control, it generates time complexities. Finally Mazhar Ali *et al.* [24] had depicted a Data Security for Cloud Environment with Semi-Trusted third party (DaSCE) which was the system developed to secure the data, still the problem of leakage of data arouse here. However all these works focus on various demands, data security and time complexities are still questionable with data migration process. Thus a need for efficient and enhanced novel technology remains stable in the field of data

migration in cloud computing.

III. Secure Data Migration with Random Optimized Cryptographic Technique

Secured data Migration is considered as the major concern during when transmitting the data/information between the cloud computing infrastructures. To deal with the security issues an enhanced novel randomized ideal cryptographic technique is proposed in this work. It includes three major phases such as data encryption, authentication, and data retrieval, in order to ensure secured data migration.

With authentication phase, the cloud user is authenticated by the Trusted Third Party (TTP) using the Luhn algorithm [26]. In the second stage, data encryption is performed with the Randomized Optimal Cryptographic Technique (ROCT) using Efficient Probabilistic Public Key Encryption (EPPKE) scheme. Finally optimization is achieved with the Covariance matrix Adaptation Evolution Strategies (CMA-ES).

- While on migration, the EPPKE algorithm is used to perform data encryption between the clouds, which is followed by the generation of cipher texts for a single plain text. From amid the generated cipher text, the best cipher text pattern is chosen preferably with CMA-ES algorithm. Moreover the encryption process is carried out by considering the encryption quality, correlation coefficient and the different types of attack.
- Ensuing with the process of encryption, digest encapsulation is to be done with the hash function called BLAKE 2b, which is superior than the other types of hash functions. Having those selected cipher text and encapsulated digest, data migration is carried out as the third stage to the Cloud Service Provider (CSP) and the data will be stored there itself. Afterwards the CSP confirms the user by providing an ID to the cloud user through TTP.
- At last, the data is retrieved from the cloud by using the user ID and the required modifications are made by the cloud user through CSP based on user authentication. In addition, BLAKE 2b algorithm and the similar digests ensure the data Integrity

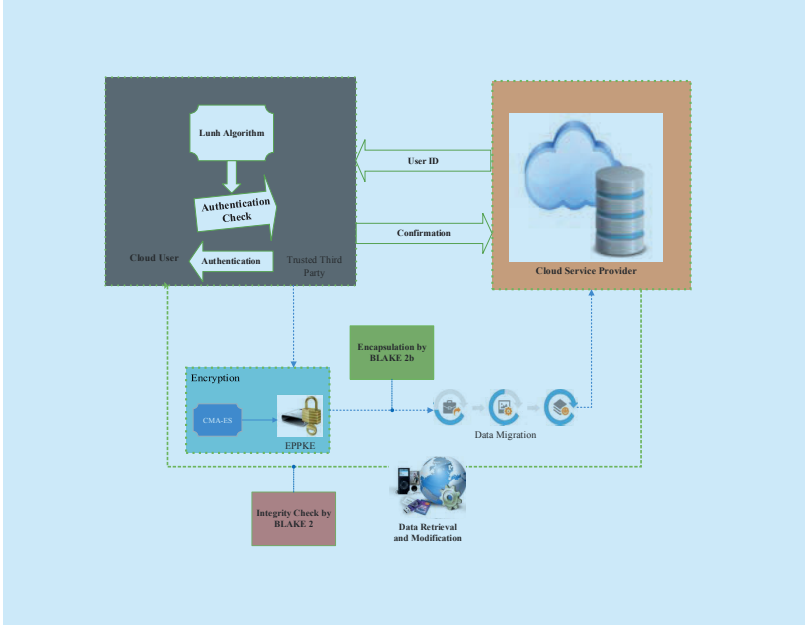

Fig 1. *Secure data migration in cloud computing with EPPKE and CMA-ES with integrity and privacy check.*

Figure 1 illustrates about the proposed methodology in detail. Each phases in the proposed work are explained briefly in the upcoming sections. The main entities involved in our proposed methodology are: cloud user, the person who is the member of the cloud as well the owner of data; CSP one who has large storage space to store enough information and large amount of resource pool to enable them and the trusted third party, the third party trusted by both the CSP as well as the cloud user.

3.1 Problem formulation

The characteristics features of cloud computing rises several security and privacy concerns of data in the cloud. Some of the major issues caused with cloud computing are: network security issues, attackers, privacy issues during data migration etc. In addition to that, threats like breaches, insecure software interfaces etc., leads to worsen the situations by unauthorized and illegal access of data during encryption causing data loss. Furthermore, the threats occurred during the user-cloud registration pro-

cess allows the attacker to leverage the data by offensive cloud models such as Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

Prior methodologies ensure security during data migration using some cryptographic techniques based on identity, attributes and prediction algorithms. But all those algorithms get fails while performing encryption on the data/information, which leads to the occurrence of threats or outsider attacks. Hence in our work, an enhanced cryptographic techniques called EPPKE optimized with CMA-ES is proposed to ensure both data security and privacy. In addition, data integrity is also achieved by the Luhn algorithm with BLAKE 2b encapsulation and the human intervention in this process is minimized to completely automate the system. The remaining process in our proposed methodology is explained through the following steps.

3.2 Cloud user authentication with luhn algorithm

In authentication phase, the Authenticity of the cloud user is validated with the help of Luhn algorithm. Luhn algorithm [26] is the one which is used to check the validity of the user through checksum calculation of their ID. Before explaining the process of the user Authentication by TTP the operation of Luhn algorithm is given.

3.2.1 Luhn algorithm

The Luhn algorithm acts as a guard against data entry errors in multi digit numbers for things like product codes. It does so through the creation of a check digit, which is an extra digit that is computed from the digits of the original product code and is appended to the end of product code. Because only one value of the check digit "matches" a particular product code, mistyping digits usually results in a product code that doesn't match its check digit. Thereof are two related processes: 1. generation of the check digit for the original number and 2. validation of a number (with check digit appended).

The algorithm is based on the idea that the sum of the digits in the product code should be a multiple of 10. That is, for the first process – picking a check digit – the algorithm will choose a value, that, when appended to the original number, will result in all of the digits equally a multiple of 10. Likewise, in the second process – validation – the algorithm will sum all the digits to see if they are a multiple of 10.

What makes these processes tricky is that every other digit is "doubled." The word "doubled" is in quotes because it's not a strict doubling if the result is a two-digit number. For example, if we "double" the digit 7, instead of simply multiplying 7 by 2 to get 14 and adding 14 into the sum, we instead put the two digits 1 and 4 into the sum. Thus the "doubling" of 7 results in adding 1 and 4, or a total of 5, into the overall sum.

To make the check digit generation process

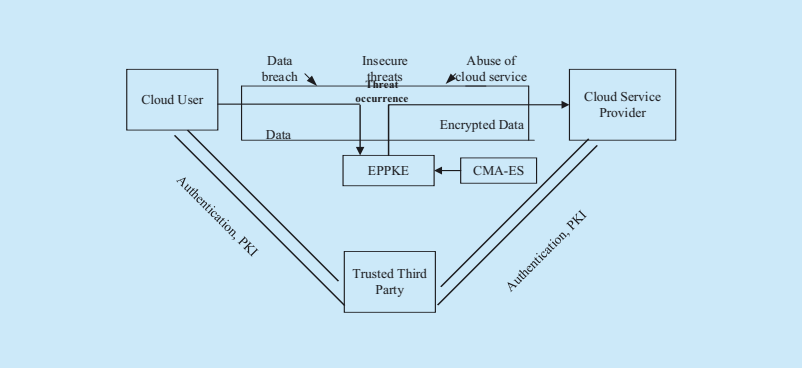

Fig. 2. *Threat model for security.*

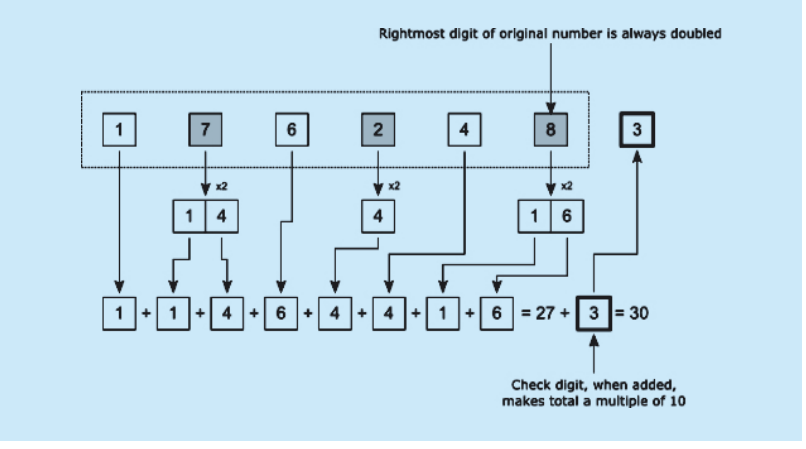

Fig. 3. *Luhn algorithm working diagram.*

clearer, here's an expansion of the example used in the book. We'll start with the number 176248 and determine the correct check digit.

The dotted outline shows the original number. The first thing to note is that every other digit is "doubled," starting with the rightmost digit of the original number and proceeding leftward. In this case, with six digits in the original number, the second, fourth, and sixth digit (shown as shaded boxes) are doubled – the digits with the values 7, 2, and 8. While we determine which digits to double starting with the rightmost digit of the original number, in this case, the 8 of 176248, we don't have to add the digits right-to-left; we can add them in any order. If we know how many digits are in the original number, for example, we'll know which digits have to be doubled.

Remember the rule about doubling the digits. The 7 is doubled to become 1 and 4; the 2 simply becomes 4; the 8 becomes 1 and 6. All of the digits, doubled or not, are summed to get 27. The only value for the check digit that results in a number that is a multiple of 10 is 3. With a check digit of 3, the sum of the entire number including the check digit is 30.

Algorithm 1. Luhn algorithm to generate valid ID

***Input*:** n-integer ($a_1a_2a_3a_4a_5a_6a_7$)

***Output*:** n-integer with check digit, x ($a_1a_2a_3a_4a_5a_6a_7x$)

//Doubling operation

From left to right double every

$(2n+1)^{th}$ digit.

For $a_n = ij$ (i-tens position, j-ones position)

//Addition operation

If $i \neq 0$ then $a_n = i + j$

Else $a_n = j$

End

Return $Adoub\left(a_1a_2a_3a_4a_5a_6a_7\right)$

// *Adoub* represents doubling and the addition operation

$$Sum = \left(a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7\right) \quad (1)$$

$M = Sum \ mod \ 10$

If $M \neq 0$, Mark a_7 as Check digit x

Else $x = 10 - M$

End

Return $a_1a_2a_3a_4a_5a_6a_7x$

Here's the validation process on the resulting number, 1762483.The algorithm explanation for the Luhn algorithm for to generate valid ID is given below in algorithm 1.

The process involved in checking the Authentication of cloud user by the TTP using Luhn algorithm is as follows.

Step 1: User inputs an alphanumeric value by employing the user name, nationality, and Date of Birth (DOB) to generate his peculiar key.

Step 2: This alphanumeric key is then converted into numerical value using the UNICODE scheme where each alphanumeric value has a numerical value.

Step 3: Then using Luhn algorithm check digit is added with the key and forwarded to the user as her ID.

Step 4: Upon Authentication check the TTP runs the Luhn algorithm and the generated checksum from the ID is divided by 10 means then the ID is valid and Authentication is given to them otherwise the ID is invalid and Authentication is prompted.

Thus Luhn algorithm is then utilized check the validity of the user by means of verifying the calculated checksum value of the peculiar key value which acts as first level of security and after that each digits of the peculiar key are made check sum which in turn generates other level of security by converting alphanumeric key to UNICODE which in turn providing ensured security. Also in addition of security providing the validity and the invalidity of the ID is also be found out by Luhn algorithm property by means of the mod function with 10. Thus by checking the validity of user and the ensured secured level process aids in upcoming the cryptographic security ensuring process is done with secured and authenticated output. In addition to that the thread type stated as insecure interface is get rid off by luhn algorithms authentication enhancement process. After authenticated by the TTP, the cloud user then encrypts his data using a secure and efficient encryption technique and here we proposed to use the ROCT with EPPKE method and the optimization of the cipher texts is

done with the help of CMA-ES optimization. This optimized encryption technique is discussed in detail in the following section.

3.3 Data encryption with optimized efficient probabilistic public key encryption (EPPKE)

Once the cloud user is authenticated by TTP, encryption is essential to perform secure data Migration. The prime numbers must be kept secret because the encrypted data may become suspect to security attacks such as chosen-cipher text attack. Here the attacker will input the number of known cipher texts into the cryptographic system, which generates corresponding plain texts from that results, so this will leads to the chance of predicting the secret key. Hence to protect the data from this type of attacks, the random hash function is added along with the prime numbers p and q during the encryption process. It results with random decryption oracle, which is the core of proposed EPPKE scheme. In RSA of OU98 scheme encryption, the user creates a public key based on two large prime numbers p and q, along with an auxiliary value n; where n is of the form p2q that provides more secured secrecy using the modulus switching method. The method doesn't fully refresh a cipher text (as the re-encryption algorithm does), but successfully limits the unwanted losses of data growth in the cipher text during homomorphic computations. Using a technique similar to "dimension reduction", the evaluator reduces the magnitude of the noise without knowing the secret key as in the other encryption schemes of RSA. Instead, the evaluator only needs to know the cipher text size in order to transform the cipher text, c modulo q into a different cipher text modulo p without sacrificing the correctness of the decryption procedure. As a result, this technique has small cipher text size as compared to RSA schemes. Hence this OU98 scheme is applied in our framework which provides better encryption with chosen-cipher text. It also avoids the threats such as data breaches and abuse to cloud service.

This encryption technique was originally developed by T. Okamoto, S. Uchiyama, and E. Fujisaki of NTT Labs in Japan [27]. As we have stated above the encryption technique is based on random oracle model which performs public key encryption with the original random hash function into a secure encryption technique. The variables and the steps involved in the EPPKE algorithm is given and are explained as follows.

3.3.1 Variables

In this section the variables associated with the EPPKE algorithm is given in table 1.

From the table 1, it is to be noted that $\kappa_{Len} = \rho_{Len}$, i.e., the length of the symmetric key and the plaintext (message) will have equal length always since the encryption is performed as a bit-wise operation and this can be understood in the following sections.

3.3.2 Key generation

The algorithm for key generation in EPPKE is represented as K. The input for this algorithm is the security parameter denoted as ς whose length become equal to ς_{Len} and this is the positive integer.

The output from the algorithm is the pair of public and private keys denoted as (p_κ, s_κ). In EPPKE, the public key has a tuple given as $(n, g, h, G, H, \varsigma_{Len}, h_{Len}, R_{Len}, r_{Len})$. In that tuple, $n = p^2 q$, $g, h \in Z$, G, H = Hash Functions, ς_{Len} length of ς and also for $p\&q$ and h_{Len} is size of H, R_{Len} is session keys and r_{Len} specifies size of random elements. Similarly, the secret key is represented as (p, g_p) and g_p is calculated as $g^{p-1} mod p^2$. The operation of the Key generation is illustrated through following steps.

Table I. *Encryption performance.*

Parameter	Representation
Symmetric key	κ
Symmetric key Length	κ_{Len}
Plaintext	ρ
Cipher text	c
Length of plaintext	ρ_{Len}

- Pick two prime values $p\&q$ of equal size ($size(p=q)=k$) and then calculate n as p^2q where $p-1=p'u$ and $q-1=q'v$. Similar to $p\&q$, the parameters $p'\&q'$ are also prime values and $|u|=|v|=\mathrm{O}(logk)$.
- Pick g randomly as stated above randomly like the order of $g_p=g^{p-1}modp^2$ is p. It is to be noticed that $gcd(p,q-1)=1$ and $gcd(q,p-1)=1$.
- Pick $h_0 \in Z$ randomly and calculate $h=h_0{}^n modn$.
- Let $\varsigma_{Len}=k$ and $r_{Len}=2k+c^0$ where c^0 is a constant and > 0. Fix R_{Len} as $R_{Len} \le k-1$.
- Choose the hash functions as $G:\{0,1\}^{R_{Len}} \to \{0,1\}^{\kappa_{Len}}$ and similarly $H:\{0,1\}^{3k+R_{Len}+\rho_{Len}} \to \{0,1\}^{h_{Len}}$.
- The parameter h can also be calculated as $g^n modn$ by setting $r_{Len}=2(k+1)$.

Next by generating the pair of public and private keys the encryption of the data is done and we have introduced the random optimization in the encryption process and this is explained in detail as follows.

3.3.3 Randomly optimized encryption

The encryption algorithm for EPPKE is represented as E and the input for the algorithm are the plaintext ρ, public key p_κ and the symmetric encryption technique $\ddot{\mathrm{E}}=\kappa \oplus \lambda.\rho$ (here $\lambda=[0,1]$ is the random parameter) whereas the output is the cipher text set as given by $c=(c_1,c_2,c_3)$. The process of the encryption technique is illustrated as follows.

- Pick $r \in \{0,1\}^{r_{Len}}$ also $R \in \{0,1\}^{R_{Len}}$ and calculate the symmetric key κ as $G(R)$.
- Calculate $c_1=g^R h^r modn$, $c_2=\ddot{\mathrm{E}}(\rho)$ and $c_3=H(c_1,R,\rho)$

Normally in EPPKE three different cipher texts are generated and among them two are ($c_1\&c_3$) generated with random parameters and to enhance more security to the cipher text c_2 we have included a random parameter denoted as λ varies from $0:1$ and with that parameter N different number of cipher texts are generated as

$$c_2=\{c_2^1,c_2^2,c_2^3,....c_2^N\}. \quad (2)$$

From this N number of cipher texts, the best one is obtained with the help of an optimization technique called Covariance matrix Adaptation Evolution Strategies (CMA-ES) with constraint to the certain parameters.

3.3.3.1 Random optimization with covariance matrix adaptation evolution strategies (CMA-ES)

The optimization of the cipher texts is done with the aid of the optimization technique called Covariance matrix Adaptation Evolution Strategy (CMA-ES) [28], which is similar to some basic concepts involved in Genetic Algorithm (GA) (Recombination) as well as Particle swarm optimization (PSO) (Population Based). The advantage on dealing with CMA-ES optimization is global convergence. CMA-ES is the recently developed evolution based optimization, commonly applied to the problem of electromagnetic field. Since its performance is better than GA and PSO algorithm in terms of convergence speed. The random generated cipher texts are optimized with constraint to the parameters so called Encryption Quality. The definitions and calculation of these parameters, the objective function formulation, and the optimization of the cipher texts are clearly illustrated as follows.

i) Encryption Quality

The Encryption Quality is denoted as Q_E, which is calculated for measuring the quality of Bitmap images encryption [29]. Q_E for the generated cipher texts is calculated as mentioned below.

- Measure the deviations between the plaintext and the cipher text, in which how many places they are differing and is calculated using the following equation (3).

$$d=|\rho-c_2^n|, n=1,2,.....N. \quad (3)$$

- Compute the average value of bits deviation as given in equation (4).

$$\bar{d}=\frac{1}{\rho_{Len}}(d). \quad (4)$$

- Calculate Encryption Quality Q_E as in equation (5).

$$Q_E=|d-\bar{d}|. \quad (5)$$

This is the first objective of our optimization technique and our aim is to maximize the Encryption Quality and hence the first part of objective function is given as in equation (6).

$$f_1 = max(Q_E). \quad (6)$$

The second objective function called correlation coefficient is measured and formulated as follows.

ii) Correlation coefficient

The correlation coefficient can be denoted as $r_{\rho c_2^n}$ and this is also one of the parameters used for measuring the correlation between pixels of the encrypted and the original image [30]. Similarly, the correlation between the cipher text and the plain text is calculated as follows.

- Calculate the mean of both ρ and c_2^n using the equations (7) and (8) given below

$$E(\rho) = \frac{1}{\rho_{Len}} \sum_{i=1}^{\rho_{Len}} \rho_i, \quad (7)$$

$$E\left(c_2^n\right) = \frac{1}{\rho_{Len}} \sum_{i=1}^{\rho_{Len}} c_{2i}^n. \quad (8)$$

Measure covariance between ρ and c_2^n using equation (9) as given below.

$$cov(\rho, c_2^n) = E\left[\left(\rho - E(\rho)\right)\left(c_2^n - E\left(c_2^n\right)\right)\right] \quad (9)$$

Then the standard deviations of ρ and c_2^n is calculated using the equations (10) and (11) as given below.

$$std(\rho) = \frac{1}{\rho_{Len}} \sum_{i=1}^{\rho_{Len}} (\rho_i - E(\rho))^2, \quad (10)$$

$$std\left(c_2^n\right) = \frac{1}{\rho_{Len}} \sum_{i=1}^{\rho_{Len}} (c_{2i}^n - E(c_2^n))^2. \quad (11)$$

Finally, $r_{\rho c_2^n}$ is calculated using equation (12) as given below.

$$r_{\rho c_2^n} = \frac{cov(\rho, c_2^n)}{\sqrt{std(\rho)}\sqrt{std\left(c_2^n\right)}}. \quad (12)$$

It is to be noted that depending on the value of $r_{\rho c_2^n}$, the relationship between ρ and c_2^n is decided for the encryption should be a successful, once if the relation is relatively low. The value of $r_{\rho c_2^n}$ and the relationship between the original and encrypted texts is given by the following condition in (13).

$$r_{\rho c_2^n} = \begin{cases} > 0, Strong\ positive\ relationship\ \rho\ and\ c_2^n \\ 0, No\ relationship\ \rho and c_2^n \\ < 0, Strong\ negative\ relationship \\ between\ \rho and c_2^n \end{cases} \quad (13)$$

As seen from the condition (13) the objective is that when $r_{\rho c_2^n}$ should be equal to zero then the second part of the objective function is given as in equation (14).

$$f_2 = min(r_{\rho c_2^n}). \quad (14)$$

The third objective function called Differential Attack is formulated as given below.

iii) Differential attack

Differential Attack is the one in which the attacker will try to observe the change in the encrypted data by modifying some bit values in the original text [31]. There are two measures, which are used to detect the impact of the single bit value on the whole encrypted image and this also suits for our proposed methodology where we analyze the impact of single bit change in the plaintext to that of the cipher text. The measures are (a) Information Entropy factor and (b) Avalanche Effect (AE) and the calculation of these measures are given in equations (15) and (16).

$$H(m) = -\sum\{0 \le i \le n-1\} p(m_i) log_2 p(m_i), \quad (15)$$

where $p(m_i)$ represents probability of m_i.

$$AE = \frac{HammingDistance}{FileSize}. \quad (16)$$

The measures given in equations (15) and (16) should also be maximum for information entropy factor and should be minimum for avalanche factor to avoid the differential attack and thus the final part of our objective function is formulated as given in equation (17).

$$f_3 = max(H(m)) + min(AE). \quad (17)$$

The overall objective function of the proposed method and the optimization of the best cipher text is given in the next section.

iv) Objective function Formulation and cipher text optimization

The formation of the objective function of our proposed methodology is thus given by

combining the equations given in (6), (14) and (17) and the overall objective function f is given in the equation (18).

$$\left.\begin{aligned} f &= f_1 + f_2 + f_3 \\ f &= max(Q_E) + min(r_{\rho c_2^n}) \\ &+ max(H(m)) + min(AE) \end{aligned}\right\}. \quad (18)$$

Based on the objective function given in equation (16) the optimization of cipher text is achieved with CMA-ES. Here the optimization is done to increase the quality of the text and to protect the text from the attackers. The steps involve how CMA-ES operates in producing the optimum cipher text is explained as below:

(a) Initialization

Based on the objective function choose the following parameters as λ (Number of population (c_2^n)), μ (Children selected), σ (step-size), $\langle x \rangle^{g=0}$ (Mean of distribution) and where g is number of iterations. Thus, the encryption process initially done for 1st iteration process and for each step size variation the values get iterated and the encryption process continues. Such that there is an increased need to analyze the step size, which plays an important role in encryption of the data with better encryption and decryption quality because by using step size updating it does not misses any data as possible.

Set the value for each parameter as $w_{m,1:\mu}$ (recombination weight of the m^{th} best child), μ_{eff} (Effective number of children with weighted average), l_σ (learning rate for step size control), l_c (Learning rate for rank one update of covariance matrix), d_σ (damping rate for step size control), l_{cov} (Learning rate of covariance matrix update).

Initialize B (Eigen vectors of the covariance matrix) =I, D (Eigen values of the covariance matrix) = I, C= BD^2B^T.

(b) Sampling and evaluation

For the evaluation of the parameter, assign the functions for each limit

Sample λ values from the population as:

$$x_{m,1:\lambda} \leftarrow N\left(\langle x \rangle, \sigma^2 C\right). \quad (19)$$

Evaluate the cost function as: $\text{cost}_m \leftarrow f_m$, where f_m is given by f in equation (19).

If termination occurs then stop the iteration or go to the selection phase.

(c) Selection

Select the parameter with the objective function and arrange $x_{m,1:\lambda}$ with respect to cost_m then select the μ best values as $x_{m,1:\lambda}$.

(d) Recombination

In the recombination phase, the values are taken to combination of the selection phase. Generate new mean $\langle x \rangle^{g=1}$ from $x_{m,1:\lambda}$ produced from the selection phase.

(e) Step size control

Update the values of l_σ, d_σ and σ.

(f) Adaptation of Covariance matrix

Update the values of l_c, l_{cov} and C.

Then go for next iteration at the end the optimum cipher text is produced and the cipher text set

$c = \left(c_1, c_{2opt}, c_3\right)$ is passed to the Decryption phase.

3.3.4 Decryption

After successfully encrypting ρ with randomly optimized EPPKE using CMA-ES, cipher text set is decrypted in the reverse manner at the receiver side and this is illustrated as follows. The input for the Decryption phase D, is the cipher text set $c = \left(c_1, c_{2opt}, c_3\right)$ and the secret key s_κ and the output become either the original plaintext or null string based on secret key.

- Calculate $c_p = c_1^{p-1} modp^2$,
  $R' = \dfrac{L\left(c_p\right)}{L\left(g_p\right)} modp$ where $L = x \rightarrow \dfrac{x-1}{p}$ for $x = 1 modp$.
- Calculate $K' = G\left(R'\right)$ as well as $m' = D_{K'}\left(c_{2opt}\right) = K' \oplus c_{2opt}$.
- Check whether the following condition is satisfied

$$c_3 = H(c_1, R', m'). \quad (20)$$

If the above condition is satisfied means the output will be the m' as the decrypted text or produce null string.

3.4 Data encapsulation by BLAKE 2b algorithm and secure data migration

Followed with the encryption of data using ROCT-EPPKE with CMA-ES algorithm, the data is encapsulated with the technique called BLAKE 2b [32], which is the variant of the BLAKE technique to produce digest on the cipher text and after encapsulation the data is migrated to the Cloud Service Provider. The background and the working principle of the BLAKE 2b algorithm is as follows.

3.4.1 BLAKE 2b Encapsulation

Background

BLAKE 2 is the cryptographic hash function in which the permuted replica of the input is XOR-ed with some constants called IV constants. There are two types of BLAKE-2 algorithm, they are: BLAKE 2b and BLAKE 2s where the former produce digests of size varied from 1 to 64 bytes and the latter one produces digests with size equal to 1 to 32 bytes. Similarly the first one is optimized to work in the 64 bit platform and the second one is optimized for 32 bit platform. Among these two types of cryptographic hash functions the first one is found to be faster than the second type and thus we employed the BLAKE 2b hash function to calculate the message digest for the purpose of ensuring the data Integrity. In the following section the working principle of BLAKE 2b algorithm and how it is applied in our proposed methodology is explained.

Working principle of BLAKE 2b

The BLAKE 2b algorithm works on any data having length in the range of $0 \le l \le 2^{B_L}$ (where, l = length of c_{2opt} and $B_L = 128$= Block Length) and hence the cipher text should also has length equal to or multiple of B_L. If this is not the case means then the padding operation is performed in which the null bytes are added with c_{2opt} and make it compatible for digest calculation with BLAKE 2b and hence the number of sequences are generated as $N_{seq} = \frac{l}{B_L}$. The number of blocks produced are $b^0, b^1, b^2, \ldots b^{N_{seq}-1}$ and each having 16-word length, after this process hashing of the padded blocks are performed as follows.

$$h^0 \leftarrow IV \oplus B_P, \tag{21}$$

where B_P=Parameter Block and this is specified as follows for our proposed methodology in Table 2.

For $i = 0,1,\ldots N_{seq} - 1$ perform hashing as $h^{i+1} \leftarrow compress(h^i, b^i, l^i)$ and finally return the value of $h^{N_{seq}}$. IV is the 64-bit words and can be specified as follows in Table 3.

In the compress operation in addition to the chain value h^i and message block b^i counter $(T = T_0, T_1)$ as well as finalization flags (F_0, F_1) are used as the input. The compress operation is performed as follows. The finalization flags $F_0 = ff\ldots\ldots ff$ and processing the last block otherwise $F_0 = 00\ldots\ldots 00$, similarly $F_1 = ff\ldots\ldots ff$ if $F_0 = ff\ldots\ldots ff$ and $F_1 = 00\ldots\ldots 00$ if $F_0 = 00\ldots\ldots 00$.

$$\begin{bmatrix} \upsilon_0 & \upsilon_1 & \upsilon_2 & \upsilon_3 \\ \upsilon_4 & \upsilon_5 & \upsilon_6 & \upsilon_7 \\ \upsilon_8 & \upsilon_9 & \upsilon_{10} & \upsilon_{11} \\ \upsilon_{12} & \upsilon_{13} & \upsilon_{14} & \upsilon_{15} \end{bmatrix} \leftarrow$$

$$\begin{bmatrix} h^0 & h^1 & h^2 & h^3 \\ h^4 & h^5 & h^6 & h^7 \\ IV_0 & IV_1 & IV_2 & IV_3 \\ T_0 \oplus IV_4 & T_1 \oplus IV_5 & F_0 \oplus IV_6 & F_1 \oplus IV_7 \end{bmatrix},$$

where $\upsilon_0, \upsilon_1, \ldots \upsilon_{15}$ are the internal states and transformed through a sequence of 12 rounds,

Table II. *Parameter Block, B_P Specification for BLAKE 2b.*

Parameter	Value
Digest Byte Length	35
Key Byte Length	20
Salt	All-33 string
Personalization	All-"ff" string

Table III. *IV Parameter specification.*

IV_0 = 6a09e667f3bcc908	IV_4 = 510e527fade682d1
IV_1 = bb67ae8584caa73b	IV_5 = 9b05688c2b3e6c1f
IV_2 = 3c6ef372fe94f82b	IV_6 = 1f83d9abfb41bd6b
IV_3 = a54ff53a5f1d36f1	IV_7 = 5be0cd19137e2179

where a round does as follows:

$G_0(\upsilon_0,\upsilon_4,\upsilon_8,\upsilon_{12})$, $G_1(\upsilon_1,\upsilon_5,\upsilon_9,\upsilon_{13})$,
$G_2(\upsilon_2,\upsilon_6,\upsilon_{10},\upsilon_{14})$, $G_3(\upsilon_3,\upsilon_7,\upsilon_{11},\upsilon_{15})$,
$G_4(\upsilon_0,\upsilon_5,\upsilon_{10},\upsilon_{15})$, $G_5(\upsilon_1,\upsilon_6,\upsilon_{11},\upsilon_{12})$,
$G_6(\upsilon_2,\upsilon_7,\upsilon_8,\upsilon_{13})$,$G_7(\upsilon_3,\upsilon_4,\upsilon_9,\upsilon_{14})$.

Here the G function is applied to all the columns and then all the diagonals in parallel manner and can be defined as follows.

$$a \leftarrow a + b + b_{\sigma_r(2i)} \tag{22}$$

$$d \leftarrow (d \oplus a) >>>> 32$$

$$c \leftarrow c + d \tag{23}$$

$$b \leftarrow (b \oplus c) >>>> 24$$

$$a \leftarrow a + b + b_{\sigma_r(2i+1)} \tag{24}$$

$$d \leftarrow (d \oplus a) >>>> 16$$

$$c \leftarrow c + d \tag{25}$$

$$b \leftarrow (b \oplus c) >>>> 63$$

The constants used in the G function are given as follows in the Table 4.

The calculations said above is performed for a number of rounds (such as, 12) and then the newly formed chain values will take the following form.

$$h'^0 \leftarrow h^0 \oplus \upsilon_0 \oplus \upsilon_8$$
$$h'^1 \leftarrow h^1 \oplus \upsilon_1 \oplus \upsilon_9$$
$$h'^2 \leftarrow h^2 \oplus \upsilon_2 \oplus \upsilon_{10}$$
$$h'^3 \leftarrow h^3 \oplus \upsilon_3 \oplus \upsilon_{11}$$
$$h'^4 \leftarrow h^4 \oplus \upsilon_4 \oplus \upsilon_{12}$$
$$h'^5 \leftarrow h^5 \oplus \upsilon_5 \oplus \upsilon_{13}$$
$$h'^6 \leftarrow h^6 \oplus \upsilon_6 \oplus \upsilon_{14}$$
$$h'^7 \leftarrow h^7 \oplus \upsilon_7 \oplus \upsilon_{15}$$

A compressed hash function is obtained finally as $h'^{i+1} \leftarrow compress\left(h'^i, b^i, l^i\right)$ which is stored at the user side and the data forwarded to the CSP as $c = \left(c_1, c_{2opt}, c_3\right)$. The Cloud Service Provider (CSP) is the one which has large number storage space to store the data accessed from different sources and upon migrating the data to the CSP the confirmation of the user is necessary and this is explained in the next section.

Table IV. *Permutations of [0-15] used in BLAKE 2 function.*

σ_0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
σ_1	14	10	4	8	9	15	13	6	1	12	0	2	11	7	5	3
σ_2	11	8	12	0	5	2	15	13	10	14	3	6	7	1	9	4
σ_3	7	9	3	1	13	12	11	14	2	6	5	10	4	0	15	8
σ_4	9	0	5	7	2	4	10	15	14	1	11	12	6	8	3	13
σ_5	2	12	6	10	0	11	8	3	4	13	7	5	15	14	1	9
σ_6	12	5	1	15	14	13	4	10	0	7	6	3	9	2	8	11
σ_7	13	11	7	14	12	1	3	9	5	0	15	4	8	6	2	10
σ_8	6	15	14	9	11	3	0	8	12	2	13	7	1	4	10	5
σ_9	10	2	8	4	7	6	1	5	15	11	9	14	3	12	13	0

3.5 Cloud user identity confirmation

The confirmation of cloud user identity is done in this stage and the process of User Confirmation is as follows.

- In this phase after the user successfully migrating his data to the CSP by the user ID, it will generate the corresponding ID for the user and sent that ID to the TTP.
- While receiving the user ID the TTP check that ID stored there and if the Authentication of the user is validated means confirmation will be sent to CSP.
- After receiving confirmation from TTP the migrated data is stored in CSP.

Thus the user data is securely migrated to the CSP after that if the user can retrieve his data and modify it and the process associated with this is illustrated in the next section.

3.6 Data retrieval and modification

After successfully storing the data to the CSP at certain point the user may want to retrieve, change or modify the contents of the data and this is done in the phase of Data Retrieval and Modification. The process involved in Data Retrieval and Modification is given through the following steps.

- User sends request to the TTP and the Authentication of the user is checked using the Luhn algorithm as explained in section 3.2.
- If the authenticity of the user results in success means the request is forwarded to the CSP and based on the verified user ID the data is forwarded to the user.
- After retrieved the data, the user performed the digest calculation as done in section 3.4.1 and digest at user side and the calculated digests are compared, if both are

same means Integrity is ensured and after that Decryption of the data is performed as in the same manner given in section 3.3.4. Then the user performed corresponding modifications on the data and again performed the secure data migration in the same manner and to verify the privacy of the data user once again sends the request to access the data and based on the data availability the response is provided by the CSP as positively or either as negatively.

- Moreover on migrating the data from one server to another there is the problem of Software Recovery (SR) in which even though the data is completely removed from the past CSP the data can be recovered using some SR tools and techniques. The only solution to this problem is that the manner in which the data is protected and this is confirmed by our proposed encryption technique and can be validated through the following mathematical proof.

3.6.1 Ensuring privacy of user data with optimized EPPKE - A mathematical proof

The data is retrieved from the CSP and the Integrity check is performed with the digest calculation as given in section 3.6. After that the privacy of the data is validated using the mathematical proof as following from the cipher texts generated by optimized EPPKE. The proof is initiated with the cipher text set $c = \left(c_1, c_{2opt}, c_3\right)$ retrieved from the CSP after data Integrity verification. Consider that, the retrieved data from the existing CSP_i is now migrated to another Service Provider CSP_j or otherwise stored elsewhere by the user. After retrieval of the data, it no longer presents in CSP_i but the SR tool λ is capable of retrieving the data $c = \left(c_1, c_{2opt}, c_3\right)$. Since the user data may be of a sensitive one then there is the chance of attacks over the data become rising and that would be possible from the adversaries present inside CSP_i.

As given in the section 3.3.4., to perform the decryption process the adversary has to calculate c_p, R' K' and m' where c_p, R' K' is to be calculated from c_1 and m' is calculated from c_{2opt} and that is possible if the adversary has the clear distinction between the three variables in the set c. Even though the distinction between those variables in the set c is possible means then the adversary try different keys to decrypt c_{2opt} and to produce m', he fails with his attempt. Because, the key used for decrypting the message is calculated from both c_p and the secret key and the produced text m' is not a valid one because we have optimized c_{2opt} with the optimization technique called CMA-ES. So in that case the user never comes to know that the text decrypted by him is the correct one. Hence the privacy of our data is ensured and verified through this proof. This mechanism is an automated proof for security scheme while data migration by the utilization of BLAKE 2b, which automatically analyses the data of various size and provide high efficieny in encryption. The Overall flow chart of the proposed methodology is given below in Figure 4.

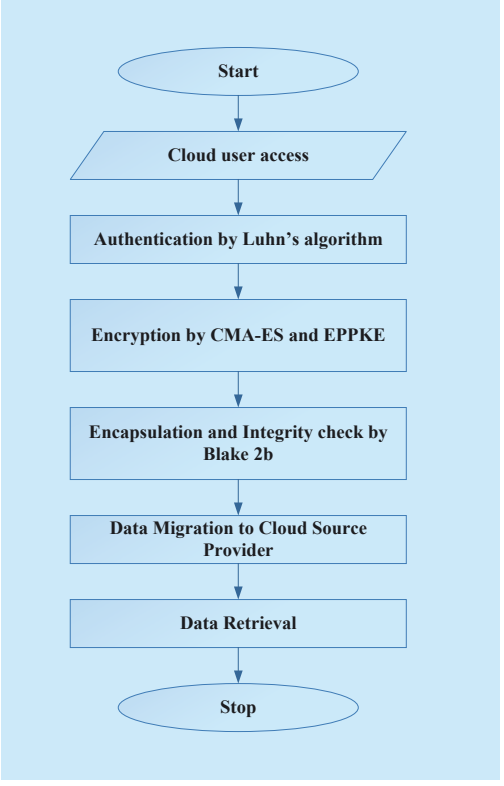

Fig. 4. *Overall flowchart of proposed methodology.*

The implementation setup of our proposed methodology, the obtained results, its performance evaluation and the discussion of its efficiency compared to the existing data migration techniques are given in the next section.

IV. Experimental Results and Discussion

The implementation of our proposed secure data migration with optimized EPPKE cryptographic technique is implemented in the Open Stack tool. The experimental set up used to implement the proposed methodology, the results produced with different performance measures and the efficiency of our proposed methodology on comparing with the existing techniques are presented in this section in detail.

4.1 Experimental setup

The proposed methodology is implemented with Open stack. In this tool there are three frameworks such as Horizon, Swift and keystone are used respectively to provide backend services, data storage and authentication purposes using Java Language. The size of the data to be migrated and the number of cloud user is considered as one in our proposed methodology. Later the number is varied to validate the efficiency of our proposed methodology on compared to other methods. The results generated by our proposed methodology with different performance measures are presented in the succeeding section.

4.2 Dataset

The dataset for the data migration is collected from the medical dataset [33], which is the Drug and Health plan data of the year 2015 obtained from the official US government site for medicine. The dataset contains Cost Benefit Report Structure, Geography, Local Contract Service Areas, Plan Cobrand Names, Plan Drugs Cost Sharing, Plan Drug Tier Cost, Plan Services, and Regional Contract Service Areas. Among those we used Plan Services for the migration from the cloud user side to the CSP and in addition to that the datasets such "2011 American Community Survey 1-Year PUMS Person File" which is a nationwide survey that collects information such as age, race, income, commute time to work, home value, veteran status, and other data. Data from the American Community Survey and the Puerto Rico Community Survey were collected during calendar year 2011 and the corresponding experimentation results are given in the following sections.

Table V. *Encryption quality Q_E with different file size.*

File Name	Size (KB)	Encryption Rate R_E	Decryption Rate R_D
F1	981	1.9469	1.8765
F2	14057	1.2432	1.1923
F3	2136	1.0848	1.0431
F4	11818	4.7223	4.2341
F5	8904	1.1913	1.1256
F6	36,051	6.0445	5.9754

4.3 Results of the proposed data migration scheme

In this section the results of our proposed methodology is taken by varying the amount of file size being migrated at each time from the cloud user to the CSP. The performance measures used in our paper for the evaluation of its efficiency are the Encryption Quality, Correlation coefficient Factor, Differential Attack Measures Information Entropy Factor (IEF), Avalanche Effect (AE) and Execution Time. The obtained results are tabulated as well as given in the form of graphs.

4.3.1 Encryption quality

The Encryption Quality Q_E is calculated using equation (3) which is one of the factors that helps to generate cipher texts randomly in CMA-ES. The results of Q_E obtained by the optimization process are presented in the table 5 with different file sizes.

In the figure 5, the encryption quality of different file size varies with encryption value. For the increase in file size, the encryption quality increases. The objective function for

random calculation of CMA-CS with the help of these factor shows that having the optimization should be done for the different file with varying sizes.

4.3.2 Correlation coefficient factor

The second performance measure is the correlation coefficient factor $r_{\rho c_2^n}$, which evaluates the correlation between the cipher-texts produced randomly. The results of $r_{\rho c_2^n}$ is presented with different file sizes in the table 6.

Correlation is a measure of the relationship between two variables of plain text and cipher text. It is calculated based on the covariance factor. From the correlation factor value, it is known that there is no occurrence of strong correlation and the zero correlation. Also there is no vast difference between the plain text and cipher text in accordance with different file size. Hence for the optimization of CMA-ES, it is easy to evaluate the minimum value with less operating time.

4.3.3 Information entropy factor

The third one among the performance measures is the Information Entropy Factor(IEF) calculated using equation (13) and this measures the number of bits differ between two cipher texts, the values are tabulated in table 7 and the results also presented in the figure 7 with varying file size.

IEF is calculated with number the difference of information entropy and the cipher text. Hence its unit is also a integer. It is a secure value and also seen that in figure 4, increase in the file size assures that information entropy has higher secure value. For the evaluation of the CMA-ES, the objective function of information entropy is taken maximum which indirectly gives the security.

4.3.4 Avalanche effect

The fourth performance metric is the Avalanche Effect (AE) percentage calculated using equation (14) and it gives the mean value of the number of bits changing in different cipher texts for different amount of file sizes. The result of AE is given in the table 8 and represented as the form of graph in figure 8.

In cryptography, the avalanche effect is the desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions, wherein if an input is changed slightly (for example, flipping a single bit), the output changes significantly (e.g., half the output bits flip). In the case of high-quality block ciphers, such a small change in either the key or the plaintext should cause a drastic change in the cipher text. From the figure the objective function avalanche effect is carried out with six different files of varying sizes 981, 14057, 2136, 11818, 8904 and 36051 KB. The average vale of avalanche factor is taken as 70.833 which is calculated in the evaluation of CMA-ES.

4.3.5 Execution time

The Execution time is the final performance measure to evaluate our proposed methodology in terms of different processes and the minimum amount of time required for each of the processes involved in our proposed methodology ensures its efficiency in migrating the data with minimum computational overhead also in a secure manner. Here the computational time is calculated for the processes such as for Authentication, Encryption, Decryption,

Table VI. *Correlation coefficient factor $r_{\rho c_2^n}$ with different file size.*

File Name	Size (KB)	Correlation coefficient Factor $r_{\rho c_2^n}$
F1	981	0.6199
F2	14057	0.6259
F3	2136	0.6744
F4	11818	0.6559
F5	8904	0.6306
F6	36,051	0.6314

Table VII. *Information entropy factor (IEF).*

File Name	Size (KB)	Information Entropy Factor(IEF)
F1	981	2.987
F2	14057	7.684
F3	2136	4.561
F4	11818	7.581
F5	8904	7.364
F6	36,051	7.984

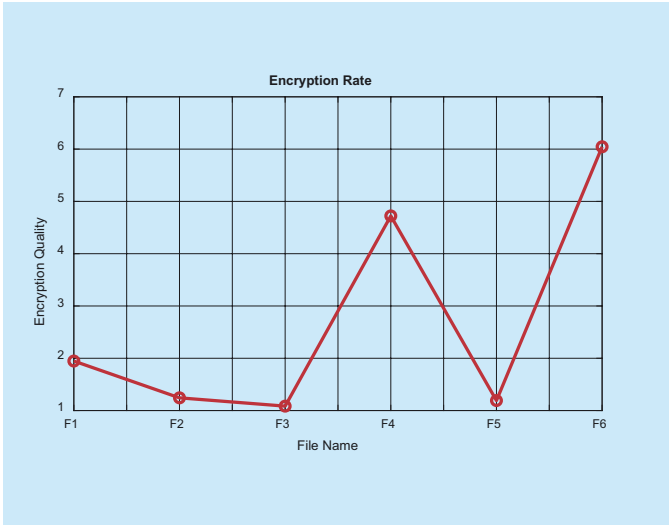

Fig. 5. *Encryption Quality Q_E.*

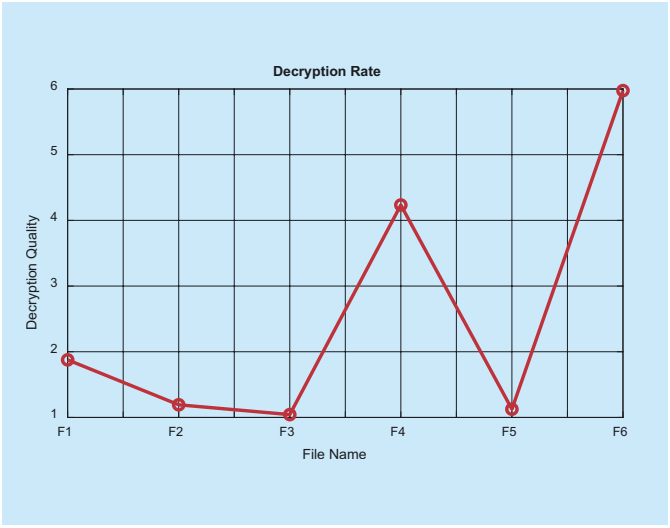

Fig. 6. *Decryption Quality Q_E.*

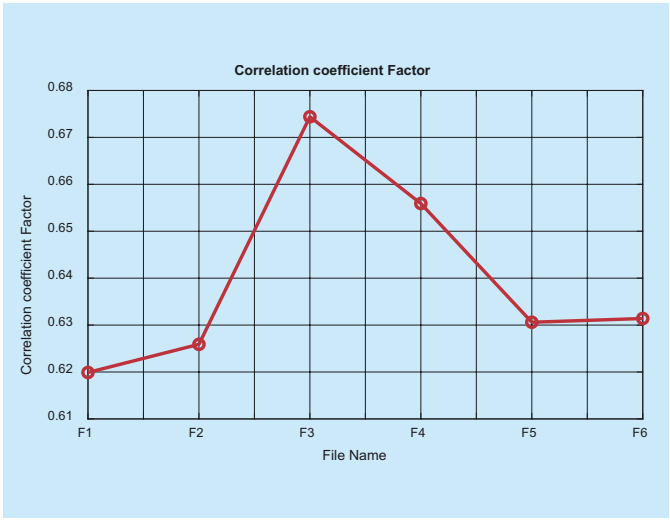

Fig. 7. *Correlation coefficient Factor $r_{\rho c_2''}$*

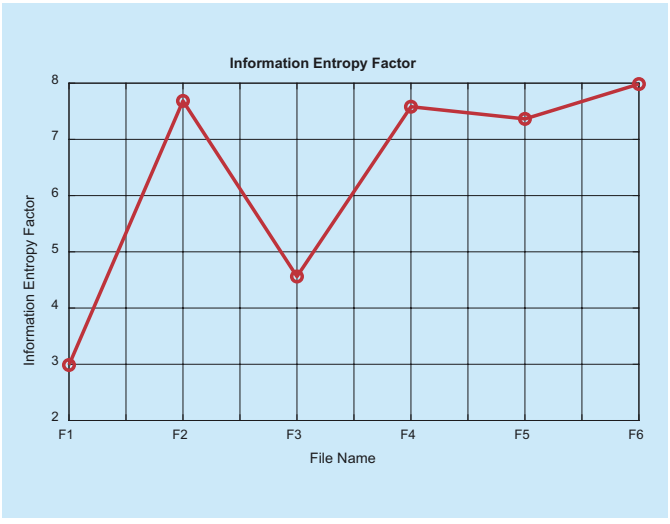

Fig. 8. *Information entropy factor (IEF).*

Table VIII. *Avalanche effect (AE) percentage.*

File Name	Size (KB)	Avalanche Effect(AE) (%)
F1	981	60
F2	14057	75
F3	2136	66
F4	11818	71
F5	8904	68
F6	36,051	85

and User Confirmation by CSP, Data Retrieval and Modification as well as for Integrity and Privacy Check. The results for this Execution Time is given in table 9 and in figure 9.

It is known that the time taken for authentication using the innovative Luhn algorithm takes the minimum time of 0.024secs. After that the encryption which is carried out by the ROCT with EPPKE takes 0.717secs with the decryption time of 0.335secs because of the optimizational use of CMA-ES. Finally the user confirmation is carried out with just 6.59secs and shows the proposed work is worth time saving scheme.

In the next section, efficiency of our system is validated by comparing it with other existing works and the obtained results are tabulated as well as plotted in graphs.

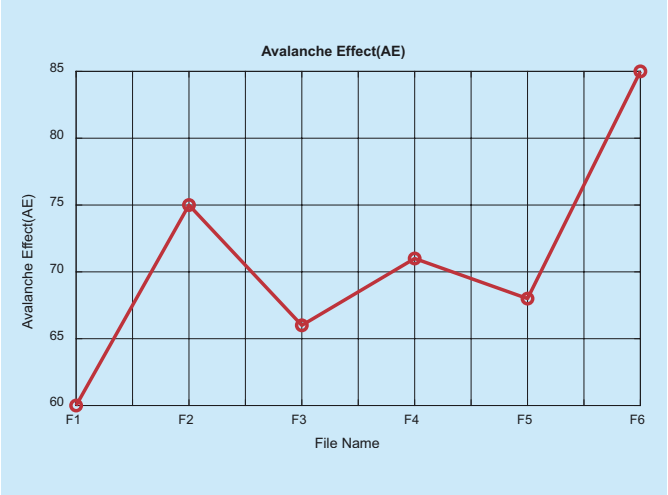

Fig. 9. *Avalanche effect (AE) percentage.*

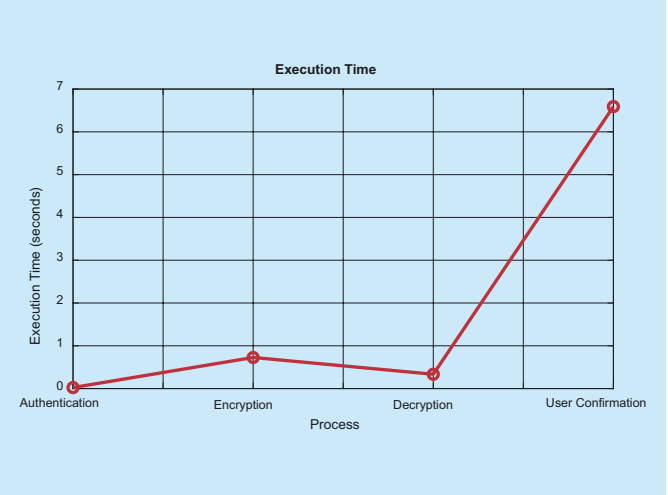

Fig. 10. *Graph for execution time of different processes.*

4.4 Performance evaluation

The efficiency of our proposed methodology can be proved further by comparing the results produced by our proposed one with other conventional techniques. We are performing the comparison in two phases, they are the encryption time of the data with different encryption techniques also the number of valid decryption time with different techniques. The existing techniques with which we are going to perform the comparison are discussed in following manner.

4.4.1 Encryption time

The encryption time of the data after encrypting it by the proposed optimized EPPKE technique is compared with the encryption techniques such as Attribute Based Encryption (ABE), Identity Based Encryption (IBE) and Password Based Encryption (PBE). In ABE user data is encrypted based on his attributes, in IBE based on the identity information provided by the user encryption is performed and similarly in PBE the encryption of the data is performed with the password as generated by the user. With the results produced by these encryption techniques with varying file sizes in our proposed one is compared and the results are given in table 10 as well as in figure 10.

Table IX. *Execution time of proposed methodology.*

Process	Execution Time (seconds)
Authentication	0.024
Encryption	0.727
Decryption	0.335
User Confirmation	6.59

Table X. *Encryption time.*

File Name	Size (KB)	Encryption Time(Sec)			
		Proposed	ABE	PBE	IBE
F1	981	0.402	0.463	0.475	0.498
F2	14057	0.801	0.824	0.864	0.876
F3	2136	0.443	0.473	0.497	0.505
F4	11818	0.725	0.764	0.789	0.814
F5	8904	0.564	0.594	0.618	0.645
F6	36,051	0.967	0.996	1.324	1.625

By the use of the ROCT with EPPKE, the plain text is converted into cipher text for the encryption process. If all the plain text is converted into the single cipher text then there is a chance to decrypt the text easily by the third party. In order to reduce that, the plain text is converted into three cipher text c1,c2,c3. For converting the plain text into the cipher text usually the existing methods like ABE, PBE, IBE takes the huge time of 0.685sec, 0.76 and 0.82 sec. While considering the proposed

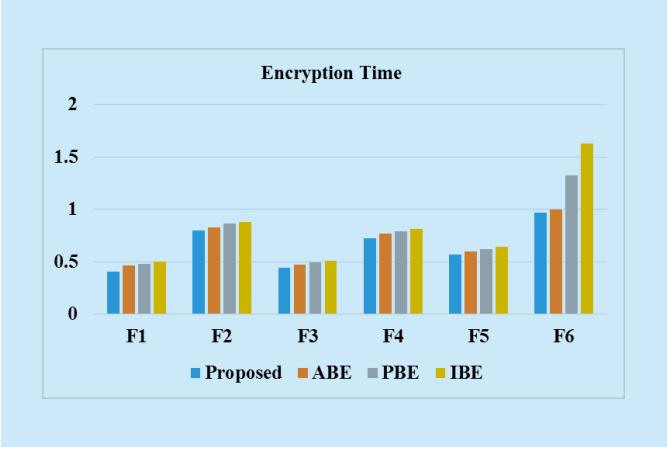

Fig. 11. *Comparison graph of encryption.*

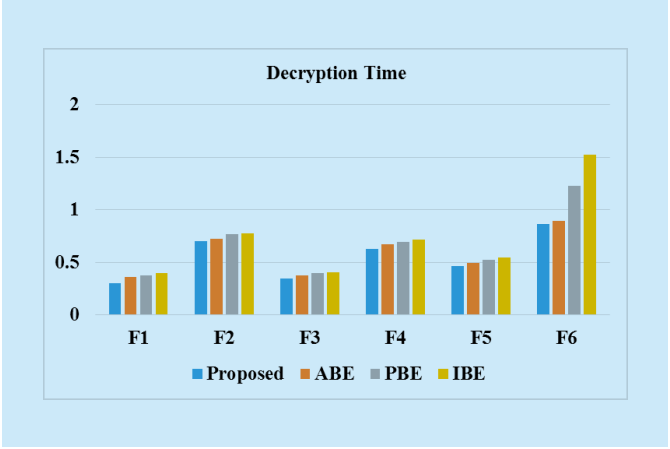

Fig. 12. *Comparison graph of decryption time.*

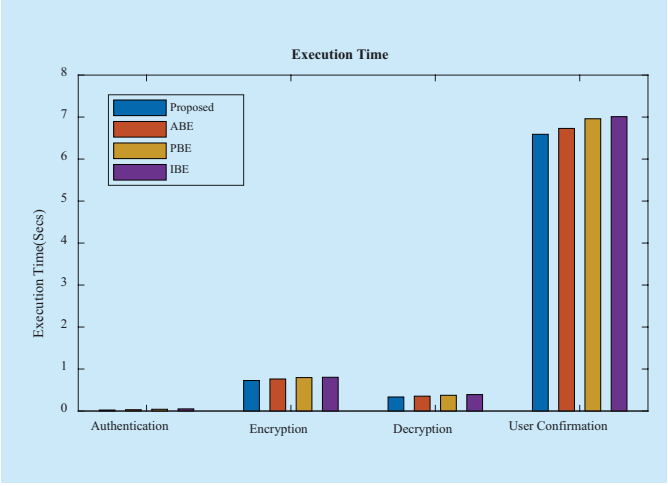

Fig. 13. *Comparison of execution time.*

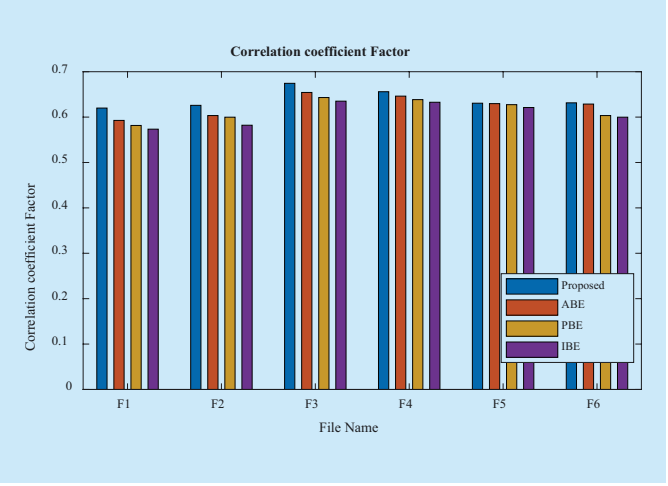

Fig. 14. *Comparison of correlation coefficient factor.*

Table XI. *Decryption time.*

File Name	Size (KB)	Decryption Time(Sec)			
		Proposed	ABE	PBE	IBE
F1	981	0.301	0.364	0.376	0.399
F2	14057	0.701	0.725	0.766	0.778
F3	2136	0.345	0.376	0.399	0.407
F4	11818	0.628	0.669	0.691	0.714
F5	8904	0.464	0.494	0.521	0.549
F6	36,051	0.867	0.896	1.228	1.527

Table XII. *Comparisons with other existing schemes in terms of delay.*

Delay Time(Sec)			
Proposed	ABE	PBE	IBE
0.0501	0.109	0.128	0.217

scheme, it takes the total encryption time for converting the plain text into cipher text with large number of files as an average of 0.65sec which is worth enough for fast encryption.

4.4.2 Decryption time

The decryption time of the data after decrypting it by the proposed technique is compared with the encryption techniques such as Attribute Based Encryption (ABE), Identity Based Encryption (IBE) and Password Based Encryption (PBE). In ABE user data is encrypted based on his attributes, in IBE based on the identity information provided by the user encryption is performed and similarly in PBE the

encryption of the data is performed with the password as generated by the user. The results are given in the table 11 as well as in the figure 11 as follows.

To break the cipher text for decryption, a random number should be given for each cipher text in encryption. This phase is carried out by randomly optimized encryption methods. From the result obtained in the figure 11, it is clear that the proposed method performs the encryption with less time of 0.51 sec by the use of the optimization algorithm. While other method of ABE, PBE and IBE shows the high decryption time of 0.58,0.66 and 0.72 sec.

By obtaining the low encryption and decryption time with the use of ROCT-EPPKE and the random selection optimization, the table 12 depicts the minor delay concern of 0.0501sec in the retrieval phase which effectually lowers the communication overhead of our proposed system when compared to the other techniques of ABE, PBE and IBE

The comparison for the significant proposed parameter such as Correlation Coefficient Factor", "Information Entropy Factor", "Avalanche Effect", and "Execution Time" with the techniques such as ABE,PBE and IBE and the values are show in figures and tables 13~16.

The scheme described in [33] is suitable for distributed mobile cloud services environment; however, it does not support user anonymity and user un traceability. Therefore, one of the design goals for the proposed scheme is to offer user anonymity and user un traceability to preserve user privacy. In order to evaluate security strength of a proposed authentication scheme, security analysis based on formal proof technique is usually conducted. From table 17, it is very obvious that only our scheme and the scheme proposed in [34] have conducted formal proof process in terms of security strength. Existing schemes introduced in [33] , [35] are also vulnerable to several security threats. For example, the schemes in [36] and [35] are vulnerable to replay attack, time synchronization problem, and forgery attack. The existing scheme is vulnerable to time synchronization problem and forgery attack.

Table XIII. *Comparisons with other existing schemes in terms of execution time.*

Process	Execution Time(Secs)			
	Proposed	ABE	PBE	IBE
Authentication	0.024	0.031	0.042	0.051
Encryption	0.727	0.763	0.797	0.804
Decryption	0.335	0.354	0.375	0.392
User Confirmation	6.59	6.73	6.96	7.01

Table XIV. *Comparisons with other existing schemes in terms of correlation coefficient factor.*

File Name	Size (KB)	Correlation coefficient Factor			
		Proposed	ABE	PBE	IBE
F1	981	0.6199	0.5927	0.5816	0.5734
F2	14057	0.6259	0.6034	0.5998	0.5821
F3	2136	0.6744	0.6543	0.6431	0.6351
F4	11818	0.6559	0.6462	0.6385	0.6327
F5	8904	0.6306	0.6297	0.6274	0.6211
F6	36,051	0.6314	0.6287	0.6034	0.5998

Table XV. *Comparisons with other existing schemes in terms of information entropy factor.*

File Name	Size (KB)	Information Entropy Factor			
		Proposed	ABE	PBE	IBE
F1	981	2.987	3.0133	3.1124	3.5671
F2	14057	7.684	7.874	7.945	8.013
F3	2136	4.561	4.742	4.876	5.043
F4	11818	7.581	7.765	7.976	8.084
F5	8904	7.364	7.543	7.653	7.872
F6	36,051	7.984	8.082	8.095	8.132

Table XVI. *Comparisons with other existing schemes in terms of avalanche effect.*

File Name	Size (KB)	Avalanche Effect(%)			
		Proposed	ABE	PBE	IBE
F1	981	60	65	72	75
F2	14057	75	78	82	85
F3	2136	66	68	74	78
F4	11818	71	74	77	80
F5	8904	68	71	75	82
F6	36,051	85	87	89	91

By the overall comparison, it is clear that the security gets increased in data migration by the use of Random Optimized Cryptographic Technique (ROCT) with the key for encryption by Efficient Probabilistic Public

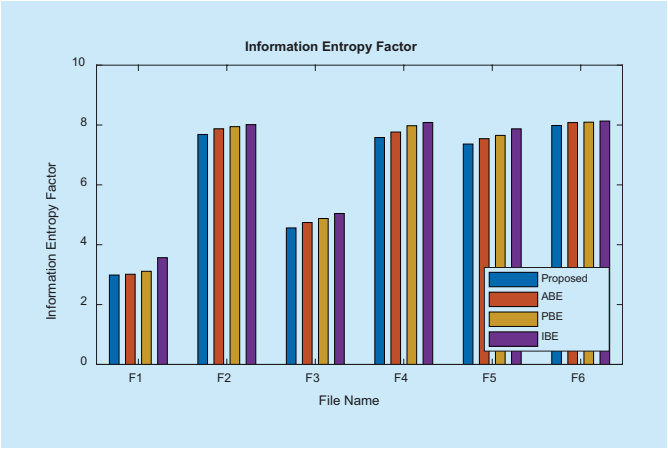

Fig. 15. *Comparison of information entropy factor.*

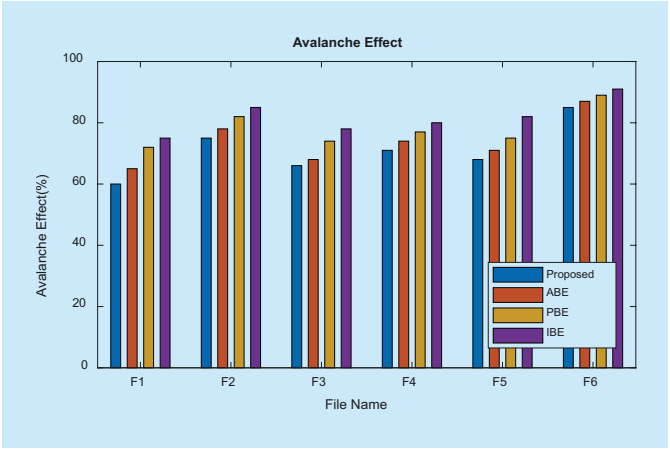

Fig. 16. *Comparison of avalanche effect.*

Table XVII. *Comparisons with other existing schemes in terms of security properties.*

S.No	Security properties	ECC based scheme [33]	Bi-linear pairing scheme [34]	IBA scheme [35]	Key agreement scheme [36]	Proposed scheme
1.	Resistance to reply attack	✓	×	✓	✓	✓
2.	Provision of user anonymity	×	×	×	×	✓
3.	Provision of user traceability	×	×	×	✓	✓
4.	Resistance to offline password attack	✓	✓	✓	✓	✓
5.	Resistance to time synchronization	×	×	✓	✓	✓
6.	Resistance to forgery attack	×	×	✓	✓	✓
7.	Suitability to multiple service provider environment	×	×	✓	×	✓
8.	Formal security proof	×	×	×	✓	✓

Table XVIII. *Efficiency of the proposed method.*

Methods	Time	Encryption	Decryption
ABE	0.109	0.68	0.58
PBE	0.128	0.76	0.66
IBE	0.217	0.82	0.72
PROPOSED	0.050	0.61	0.50

Key Encryption (EPPKE). Also the algorithm called Luhn algorithm is run by the TTP in which the authentication of the user is verified by means of the user ID and confirm with the TTP secured the data from the several attacks.

By the performance analysis it is established from table 18, on the use of cryptographic technique of Random optimisation, the efficiency of the work get increased by taking less delay of 0.0501sec although the other techniques like ABE, PBE and IBE takes varies difference in delay of 0.109, 0.128 and 0.217 sec. Owing to the work of EPPKE, the encryption time intakes in 0.61sec only by comparing with ABE of 0.685, PBE of 0.76 and IBE of 0.82 sec proving that the methodology works well with better efficiency. Considering the decryption time of 0.5sec showed that the proposed method of CMA-ES optimised the process with less computing time compared to other techniques of ABE, PBE and IBE of 0.58, 0.66 and 0.72 sec. Along with the use of Luhn algorithm with BLAKE 2b encapsulation method increases the efficiency of the whole process by doing as an automated system without human intervention. With the combined approach of all these shows the improvisation of security in data migration with high efficiency and computational heads. Overall, the results showed that the proposed Random Optimized Cryptographic Method for data migration technique has a very good per-

formance with the total execution of 7.676sec considering less delay of 0.0501 and more security from various attacks.

V. Conclusion

Data migration in cloud computing is a relatively new research due to privacy and security issues concerned with the public clouds. The paper presented a secure data migration technique called EPPKE optimized with CMA-ES. The multi secured attention includes in authentication check, secured encryption and data retrieval. Evaluating the efficiency and security analysis of the proposed technique with the most common encryption algorithms like ABE, PBE and IBE indicates high security and speed in encryption, decryption process with 0.61 and 0.5sec. Accordingly from the obtained results, it is visible that this techniques paves favourable proceedings for real time implementation in near future.

References

[1] K.V.K.M. Kumar, "Overview of cloud computing architecture: service delivery models, security & privacy issues and trust", *IJRET: International Journal of Research in Engineering and Technology*, vol. 2, issue 12, 2013, pp. 607-609.

[2] I.A.T Hashem et al., "The rise of 'big data' on cloud computing: review and open research issues", *Information Systems*, vol. 47, January, 2015, pp. 98-115.

[3] A. Beloglazov, et al., "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Generation Computer Systems*, vol. 28, no. 5, 2012, pp. 755-768.

[4] C. Rong et al., "Beyond lightning: a survey on security challenges in cloud computing", *Computers & Electrical Engineering*, vol. 39, no. 1, 2013, pp. 47-54.

[5] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing", *Journal of Network and Computer Applications*, vol. 34, no. 1, 2011, pp. 1-11.

[6] S. Marston et al., "Cloud computing - the business perspective", Decision Support systems", vol. 51, Issue 1, 2011, pp. 176-189.

[7] Ali O and Soar J, "Challenges and issues within cloud computing technology", *Proc. Fifth International Conference on Cloud Computing, Grids, And Virtualization*, 2014, pp. 55-63.

[8] C. Espositon et. al., "Encryption-based solution for data sovereignty in federated clouds", IEEE Cloud Computing, vol. 3, no.1, 2016, pp. 12-17.

[9] D. Zissis, and D. Lekkas, "Addressing cloud computing security issues", *Future Generation Computer Systems,* vol. 28, no.3, 2012, pp. 583-592.

[10] B. P. Rimal et al., "Architectural requirements for cloud computing systems: an enterprise cloud approach", *Journal of Grid Computing*, vol. 9, no. 1, 2011, pp. 3-26

[11] L. Zhang et al., "Moving big data to the cloud: an online cost-minimizing approach", *IEEE Journal on Selected Areas In Communications*, vol. 31, no.12,2013, pp. 2710-2721.

[12] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment", *IEEE Transaction on Parallel and Distributed Systems*, vol. 24, no. 6, 2013.

[13] Chung Y et al., "Utilization of workflow management system for virtual machine instance management on cloud", *Concurrency and Computation: Practice and Experience, vol.* 27, no.17, 2015, pp. 5350-5373

[14] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption", *IEEE Transactions on Parallel and Distributed Systems,* vol.24, no.1, 2013, pp. 131-143.

[15] M. D. Ryan, "Cloud computing security: the scientific challenge, and a survey of solutions", *Journal of Systems and Software*, vol. 86, no.9, 2013, pp. 2263-2268.

[16] B. Grobauer, T. Walloschek and E. Stöcker, "Understanding cloud computing vulnerabilities", *IEEE Security & Privacy,* vol.9, no. 2, 2011, pp. 50-57.

[17] K. Hashizume et al., "An analysis of security issues for cloud computing", *Journal Of Internet Services And Applications, vol.* 4, no. 1, 2013, pp. 1-13.

[18] S. Poongodi, P. Murugan and P. Kuppusamy, "Shared authority based privacy-preserving authentication protocol in cloud computing", *Cloud Computing*, vol. 19, 2015, pp 1-3.

[19] C. Modi, D. Patel, B. Borisaniya, A. Patel and M. Rajarajan. "A survey on security issues and solutions at different layers of cloud computing", *The Journal of Supercomputing*, vol. 63, no.2, 2013, pp. 561-592.

[20] F. Wang, J. Liu, M. Chen and H. Wang, "Migration towards cloud-assisted live media streaming", *IEEE/ACM Transactions on Networking*, 2015, pp. 1-11.

[21] X. Qiu, H. Li, C. Wu, Z. Li and F. Lau, "Cost-minimizing dynamic migration of content distribution services into hybrid clouds", *IEEE Transactions on Parallel and Distributed Systems*, 2015, vol. 26, no. 12, pp. 3330-3345.

[22] M. Menzel, L. Wang, S. U. Khan and J. Chen,

"Cloudgenius: a hybrid decision support method for automating the migration of web application clusters to public clouds", *IEEE Transactions on Computers*, 2015, vol. 64, no.5, pp. 1336-1348.

[23] Y. Zhu, D. Huang, C. Hu and X. Wang, "From Rbac To Abac: Constructing flexible data access". *IEEE Transactions on Services Computing,* 2015, vol. 8, no. 4, pp. 601-616.

[24] M. Ali, S. Malik and S. Khan, "Data security for cloud environment with semi-trusted third party", *IEEE Transactions On Cloud Computing*, 2016, pp.1-14.

[25] Y. Alsalhi, "An accurate and high-efficient qubits steganography scheme based on hybrid neural networks", *Multimedia Tools and Applications*, 2019, pp. 1-17.

[26] K. K. Tripathi and L. Ragha. "Hybrid approach for credit card fraud detection", *Parity*, vol.16, no. 2, 2013, pp. 8-11.

[27] T. Okamoto, S. Uchiyama and E. Fujisaki, "Efficient probabilistic public-key encryption", *Submission To IEEE P1363a*, 1998

[28] M. D. Gregory, Z. Bayraktar and D. H. Werner, "Fast optimization of electromagnetic design problems using the covariance matrix adaptation evolutionary strategy", *IEEE Transactions On Antennas And Propagation*, vol. 59, no. 4, 2011, pp. 275-1285

[29] K. Hashizume et al., "An analysis of security issues for cloud computing". *Journal of Internet Services and Applications*, 2013, vol. 4, no. 1, pp. 5.

[30] M. Amin, S. Hussain et al. "Profiling-based energy-aware recommendation system for cloud platforms", *In Computer Science and Its Applications*, 2015, pp. 851-859.

[31] M. Abd El-Wahed, S. Mesbah and A. Shoukry, "Efficiency and security of some image encryption algorithms", *Proc. The World Congress on Engineering*, 2008, 1, pp. 2-4.

[32] J. P. Aumasson et. al., "Simpler, smaller, fast as md5", *Proc. International Conference on Applied Cryptography and Network Security,* 2013, pp.119-135.

[33] C. Tien-Ho et al., "An advanced ECC dynamic id-based remote mutual authentication scheme for cloud computing", *Proc. 5th FTRA Int. Conf. Multimedia Ubiquitous Eng.*, 2011, pp. 155–159.

[34] S. Huang and W Ding, "Cryptanalysis of three dynamic id-based remote user authentication schemes using smart cards", *Proc. IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, 2016, pp. 44-52.

[35] H. Sun, Q. Wen, H. Zhang and Z. Jin, "A novel remote user authentication and key agreement scheme for mobile client–server environment", *Appl. Math. Inf. Sci.*, vol.7, no.4, 2013, pp. 1365–1374.

[36] J. Tsai and N. Lo. "A privacy-aware authentication scheme for distributed mobile cloud computing services", *IEEE Systems Journal*, vo. 9, no.3, pp. 805-815.

Biographies

M. G. Aruna, is a research scholar of Sai Vidya Institute of Technology, Affiliated to Visvesvaraya Technological University, Belagavi, India. She has completed her B.E and M.Tech from Bangalore University and Dr. MGR University in 2001 and 2006 respectively. Her research interests is in cryptography, DBMS, IOT, Machine Learning, data security in cloud computing. She has overall 14 years of experience as an academician and 5 years of research experience.

K. G. Mohan, has received PhD from Anna University in 2007 in the domain of Computer Architecture. His area of research interest includes low power architecture design, Cloud Computing, Wireless Sensor Networks, IoT, Network Security, etc. He has overall 30 years of experience as an academician and 14 years of experience in research. He has published paper in many international journal and conferences of reputed.