ISSN (e): 2250 – 3005 || Volume, 13 || Issue, 2|| Mar.-Apr. – 2023 ||
International Journal of Computational Engineering Research (IJCER)

Cloud Security and Homomorphic Encryption techniques For Identity Management in Cloud Computing.

Dr, Vineetha KR, *Associative Professor, Nehru College of Engineering and Research Centre.*
Harshil S, *Department of MCA, Nehru College of Engineering and Research Centre.*

ABSTRACT
Today, business organizations operate in a dynamic environment and thus need to keep adjusting in terms of size and resources. To meet these changing needs, firms are required to scale up their IT infrastructure (hardware, software, services etc.) by investing time, money and other scarce resources. This scaling up process, generally, is slow and expensive option as companies are required to make huge capital investments in land, building, hardware, software, employees etc. Further, owning a huge infrastructure merely is not a guarantee of meeting business requirements smoothly. In fact, most of the times; companies fail to utilize these resources fully, resulting in negative returns on such investments. In short, there are lots of challenges associated with these kinds of investments that discourage firms to own them.
Cloud computing is a new age technology that overcomes some of challenges associated with having own IT infrastructure. It is a paradigm shift in computing that is gaining acceptance in Information Technology industry now (Pring et al., 2009). This technology looks similar to its predecessors such as client/server computing, peer to peer computing, distributed computing, cluster and grid computing but is far superior in terms of performance and optimal utilization of resources. In this computing, data moves away from personal computer and desktops to large data storages called data centers to deliver applications as a service (Dikaiakos et al., 2009). It is the next step after Grid computing in the evolution of on-demand and pay-as-per-use model (Schubert and Jeffery, 2012).
This certainly, has come up as a boon for the all kinds of organizations. Be it small or big, profit or non-profit, govt. or private, all can get benefitted by this. Since cloud computing is in nascent stage, especially in India, therefore, there are many interesting challenges pertaining to its implementation & maintenance that provide research opportunities in this area. This research primarily deals with the systems for managing identity of users over the cloud.
Key words*: Cloud Security, RSA algorithm, Homomorphic encryption.*

Date of Submission: 10-03-2023 Date of acceptance: 23-03-2023

I. INTRODUCTION

Security is required to ensure the resources' Confidentiality, Integrity, and Availability (CIA).
Data and information can be encrypted and stored in the cloud.
If an operation needs to be done on the data, though, the data must first be decrypted. Nonetheless, attacks are al ways possible on encrypted data. The fact that everything is done by a third party instead of the data owner puts security and privacy as a top concern in cloud computing. All private or public data is accesSed and stored via re mote computers that are not within the data owner's control Data confidentiality is compromised because the clo ud server's management of the data is outside the confines of the data owner's trust.

Strictly speaking, cryptographers, who utilise techniques of cryptography to secure data, are concerned with data secrecy. The process of converting plain text into cypher text is known as cryptography. This method is typically used to transport data securely from one location to another by making sure that the data can only be accessed by receivers and users who have been verified.

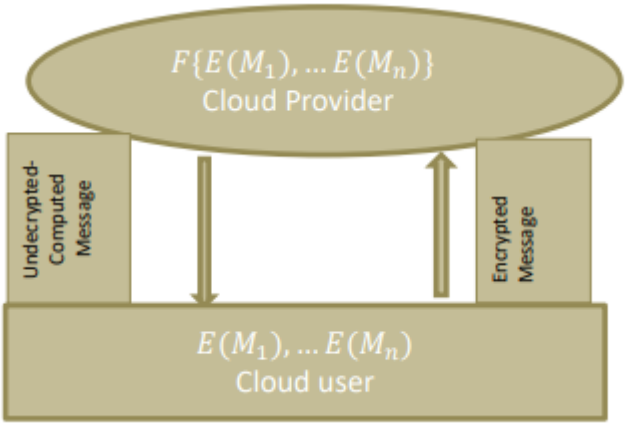

Figure 1: Cloud and User Interaction.

II. LITERATURE SURVEY

Survey on identity and access management in cloud computing: According to Pinki, Harsh Dhiman, federated identity has taken on a significant role in the federated identity management environment as a beneficial feature for Single Sign-on (SSO) and user management. Platform trustworthiness, managing various digital identities, and identity theft are a few issues that arise in a federated identity management context. The key principles in cloud authentication and federated environments are Security Assertion Markup Language (SAML), OAuth, and OpenID. In terms of cloud computing security, this paper discusses the Identity and Access Management (IAM) problem.

A Critical Analysis on cloud identity management: According to Temidayo Abayomi-Zannu and Isaac Odun-Ayo, a digital identity enables an entity to be represented in various types of information that enable the entity to be recognised within a specific framework [4]. Identity management (IDM) is a group of interconnected rules that enables the administration, maintenance, management, information interchange, discovery, and authentication processes used to identify an information with the aim of maintaining general security. [4]. Using cloud services, information may be accessible anywhere, at any time, over the Internet. As a result, an identity management strategy is required in order to confirm that a user is legitimate and to provide services based on such credentials. A procedure for managing identities attempts to protect the user and other processes in terms of confidential and sensitive data.

Privacy Strategies in cloud computig: Carlos Becker and Jorge Werner note that Since the advent of cloud computing, tens of thousands of users and several applications have attempted to connect and exchange sensitive data. Therefore, the use of models and tools is crucial for the secure administration of identities and to prevent data privacy issues while managing applications and resources. Federated identity management is addressed by models and tools, and it is crucial that they make use of privacy features to help them comply with the law as it is. In order to give a survey of privacy in cloud identity management, this article will present and contrast the key features and difficulties outlined in the literature. A discussion on the usage of privacy and potential future study directions is included at the conclusion of this article..

III. METHODOLOGY

Key security issues that the industry is now dealing with were addressed in Gleeson's 2009 study. An additive homomorphic method with the name Pailler cryptosystem was proposed by Pascal Pailler. This system can be used for many different purposes, including electronic voting, among others (ElGamal et al., 1985).

Simon and Carlos talked about the most current developments in homomorphic encryption methods. They conducted a survey on current developments in the algorithms for Somewhat Homomorphic Encryption (SWHE) and Fully Homomorphic Encryption (FHE) (Aguilar-Melchor et al., 2013). A multiplicative property-based algorithm was introduced by Taher Elgamal (Sakurai et al., 2002).

Key security issues that the industry is now dealing with were addressed in Gleeson's 2009 study. An additive homomorphic method with the name Pailler cryptosystem was proposed by Pascal Pailler. This system can be used for many different purposes, including electronic voting, among others (ElGamal et al., 1985).

Aldar C-F Chan focuses on partial homomorphism, which allows one to manipulate encrypted data. They have provided the Iterated Hill Ciper and Modified RSA additive homomorphic schemes (AC-F, 2009).

In a paper published in 2012, scientists suggested the use of multivariate keys and homomorphic encryption for cloud security. They have provided a thorough explanation of the Key Dependent Message

(KDM) encryption system that can be used to protect cloud data. Three homomorphic encryption algorithms—RSA, ElGamal, and Paillier—have been in-depth studied by Farah et al. (2012). They have assessed each of the three algorithms and displayed a comparison between them. The outcome demonstrates that RSA outperforms ElGamal and Paillier, while ELGamal outperforms Paillier.

1.Homomorphic Encryption(HE)

Using a technique called homomorphic encryption, it is possible to compute on encrypted data without first decrypting it. If the user then decrypts the result, which is in encrypted form, it returns the original result without requiring knowledge of the original plaintext (Micciancio et al., 2009).

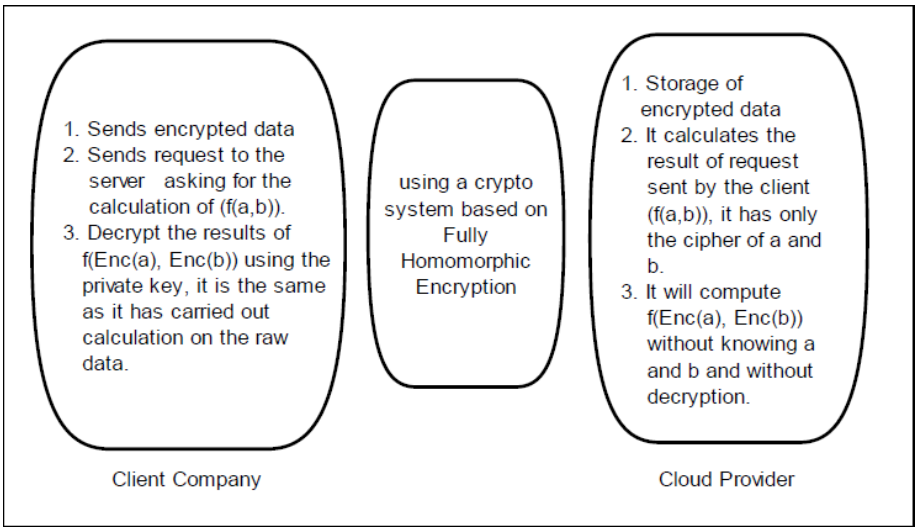

Figure 2: Homomorphic Encryption on Cloud (Liu, 2012)

- **Types Of Homomorphic Encryption**

There are three types of homomorphic encryption methods present in this literature:

A) Partially Homomorphic Encryption (PHE)

If an encryption method can only execute one operation on encrypted data, such as addition or multiplication but not both, it is referred to as partially homomorphic encryption (PHE) (Yang et al., 2014).Some examples of partially homomorphic cryptosystems are given below -

- RSA - multiplicative homomorphism
- ElGamal- multiplicative homomorphism
- Paillier- additive homomorphism

B) Somewhat Homomorphic Encryption (SWHE)

Both multiplicity and addition are possible for partial homomorphic structures, but not both. As a result, the plan cannot guarantee information confidentiality when computing in the cloud. In his 2009 dissertation at MIT, Craig Genry was successful in developing a new homomorphic system that can support addition and multiplication. In addition to the standard encryption and decryption techniques, the scheme also includes an evaluate function. Cascaded logic gates are used as the form of the evaluate functions (AND, OR, NOT, etc). Making the evaluate function run calculations on the encrypted data is the primary objective here. The evaluate function increases the amount of memory needed to use Craig's approach, which is a big drawback. The problem of the memory size led to the solution of arbitrary depth. Although using arbitrary depth sounds appealing, there is a problem with numerical noise when it is implemented.

C) Fully Homomorphic Encryption (FHE)

Fully homomorphic encryption (FHE), a type of encryption, can compute any operation and execute addition and multiplication (Yang et al., 2014). Craig Gentry created a lattice-based cryptosystem in 2009 that is the first (and only) of its kind. For the representation of keys and cypher text in the Gentry scheme, a complicated mesh of ideal lattices is used. A system's private key is made up of a matrix V that was created at random and a second matrix W that has the formula: V W c mod f(x), where c is a constant (Gentry, 2009). SWHE or PHE are typically the schemes that are being deployed on cloud systems.

- **Benefits Of HE**

Many advantages and uses can be found with homomorphic encryption. Enhanced privacy is one of these advantages. One of the objectives of cryptography in general is privacy, but homomorphic encryption can offer even greater privacy than conventional encryption techniques. The user can decrypt this cypher text to view various statistics, metrics, or whatever else they're interested in when the host performs computations and returns the encrypted output to them. Private information retrieval is a wonderful use case for homomorphic encryption. By using this method, the search engine would be unable to access any data while still providing the user with useful results. One of the main advantages of this application is that sensitive data may still be retrieved without ever disclosing even the type of the data, even for a user who lives in a place where privacy is valued as a luxury.Some common usage of HE are given as follows-

-Analysis of disease and find out its treatment without disclosing the details of patient.

-In the corporate clients and organization do not want to disclose their confidential information. By using this technique functions can be computed on data and data itself remain private.

- It also can be used for the protection of mobile agents by either using computation with encrypted function or computation with encrypted data.

- **Drawbacks of HE**

The intricacy of the systems is one of the main disadvantages. A much more complicated lattice-based cryptosystem is required for fully homomorphic encryption. Homomorphic systems occasionally perform poorly. Crypher texts that are produced after encryption are substantially larger than plaintexts. The communication channel needed more capacity as a result of sending these encryption texts. The very huge cypher text is used for all calculations. Thus, compared to computations performed on plain texts, computation time will become slow (Yang et al., 2014). when several users are actively involved at once. Each user who steals another user's private key has the ability to decrypt that user's data and violate their privacy. If the attacker has access to the cypher text in any way from the server, he can use it to generate new cypher texts. which, after decryption, might stand for another plaintext.

2.RSA ALGORITHM

At MIT, Ron Rivest, Adi Shamir, and Len Adeleman created the RSA algorithm in 1977. The procedure uses a block cypher method and a sizable positive integer n [14]. The scheme is an effective cryptosystem due to the enormous size of n (1024) and the substantial amount of computational labour necessary to break the encryption. In other words, it is almost impossible for a brute-force attack to succeed in a reasonable amount of time. The cipher-text C is represented as C = Me mod n given a message content block M. The formula for a decrypted message M is M = C d mod n. The value of n, which is the product of the two prime numbers p and q that are chosen at random, must be known by Alice and Bob (the sender and the recipient). The sum of (p-1) and (n) yields the Euler Totient function (n) (q-1). Alice chooses a value of e that is a coprime of and must be smaller than (n). Bob determines d using the formula de = 1 mod n. While Bob determines d, Alice is aware of the value of e. PU = e, n and PR = d, n, where PU and PR are the public key and private key, respectively, make up the public key encryption method known as RSA. Because the procedure only permits multiplications of ciphertexts and does not ensure addition, it is referred to as a partial homomorphic algorithm.

3.RSA AND HOMOMORPHIC ENCRYPTION

Two keys, one public and one private, are used in the RSA asymmetric encryption technique for encryption and decryption, respectively. In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman developed RSA. Prior to transferring data over a network, it has only ever been used to generate public and private keys and encrypt the data. Moreover, authors presented the idea of homomorphic encryption (Gentry, 2009; Rivest et al., 1978).

The two features of homomorphic encryption are as follows:

If Encrypt (P1 P2) = Encrypt (P1) Encrypt, then a particular cryptosystem is regarded as additive homomorphic encryption (P2)

If Encrypt (P1P2) = Encrypt (P1) Encrypt, a given cryptosystem is regarded as multiplicative homomorphic encryption (P2)

IV. RESULT ANALYSIS

The two characteristics of homomorphic encryption are as follows: A particular cryptosystem is regarded as additive homomorphic encryption if

The example that follows will help you grasp HE better.

Let m to be a simple text.
Operation (m) equals "decrypt" (or "operation (encrypt (m)")
Let R+ and R* be two sets of positive real numbers and their corresponding sets of logarithms; the addition of real numbers and the multiplication of logarithms are carried out using homomorphic operations (Micciancio et al., 2009).

Let x, y, and z be R+.
If x and y are equal, then log(x + y) equals log(z), or log(x + y) equals log(x * y).
The original value of z, or the result, is obtained by taking the antilog of the log (z).
In the example above, there are two approaches to find z: directly and by logarithms. We obtain the same outcome in both situations. Thus, it is more secure to conduct operations on encrypted data rather than plain text.
use it with encrypted data. Some more examples are given in following figures to understand Homomorphicencryption on Integers and Strings-

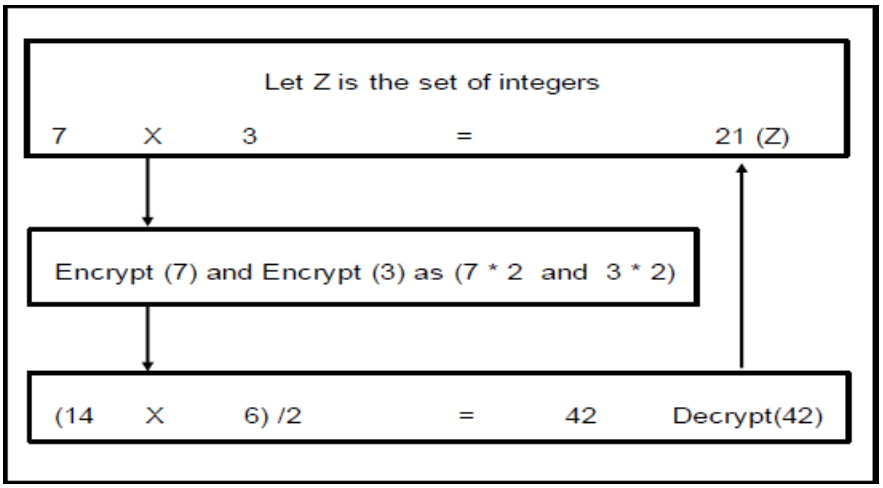

Figure 3: Homomorphic encryption on integers

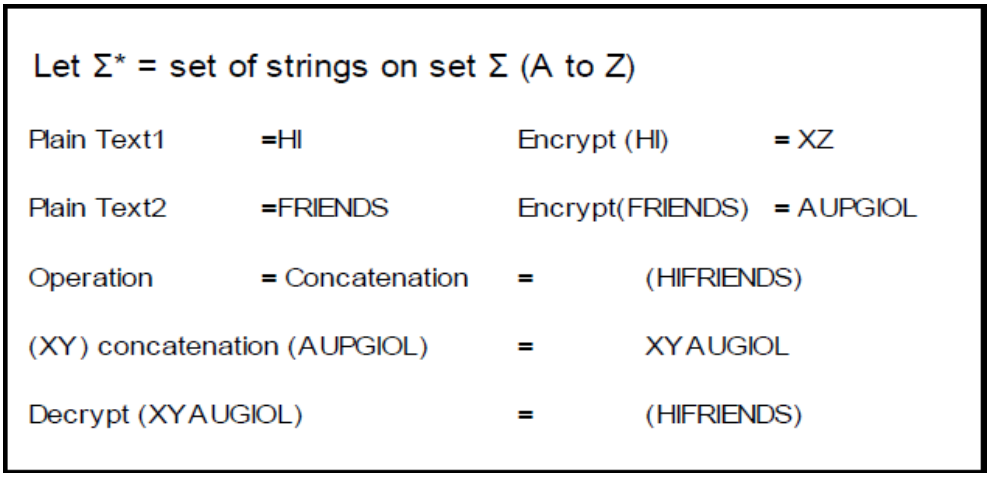

Figure 4: Homomorphic Encryption on Strings

Case study to implement RSA as Partial Homomorphic Encryption algorithm

We conducted an experiment to determine whether RSA has the homomorphic property by using the following data: - We used 100 pieces of land data that needed to be kept in the cloud. The information includes the name of the person, the shape of the land (S or R), and its length and width. The user wants to compute the area of the land whenever it is needed using encrypted data that has been saved in the cloud. Figure 1 shows the data flow diagram of area calculation on cloud and Figure 2 shows the output after execution.

Algorithm encodes the data by public key (7, 33) and stored in cloud. Cloud has calculated area of the land using encrypted data and result is returned to user. User will now receive encrypted (area) and will decrypt that

by using private key (3, 33) thereby receive the original area of land. The Figure 2 below is the output screen of the algorithm implemented in C language.

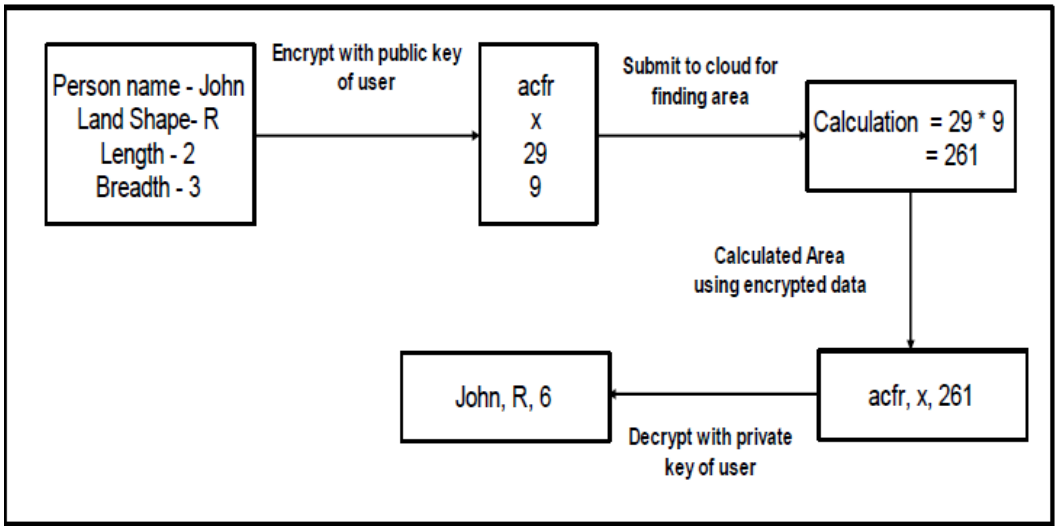

Figure 5: Area Calculation on cloud

```
Enter two relatively prime numbers                    : 3 11
        F (n) phi value = 20
Enter e which is prime number and less than phi       : 7
        Public Key      : {7, 33}
        Private Key     : {3, 33}
Enter the length: 2
        Encrypted keyword: 29
Enter the breadth       : 3
        Encrypted keyword: 9
Enter the cipher text (encrypted area): 261
        Decrypted keyword: 6
Do you wish to continue: -
```

Figure 6: Calculation of area using RSA as PHE

V.Application of HME

Practical application of homomorphism is the main motivation of many research works, below are two of its most promising applications:

a. Patient medical condition: Software for disease classification is one of the most intriguing uses of homomorphic encryption. Since the majority of hospital prediction models are constructed using plaintext, patient data privacy and confidentiality are an issue. The homomorphic encryption strategy has been outlined as a useful tool for protecting the privacy and confidentiality of patient data during computation in the preceding sections. As seen in figure 3, if a doctor wishes to know the patient's medical status, he runs a predictive application programme on the encrypted data of the patient. The patient's medical condition is shown on the screen by the software programme.

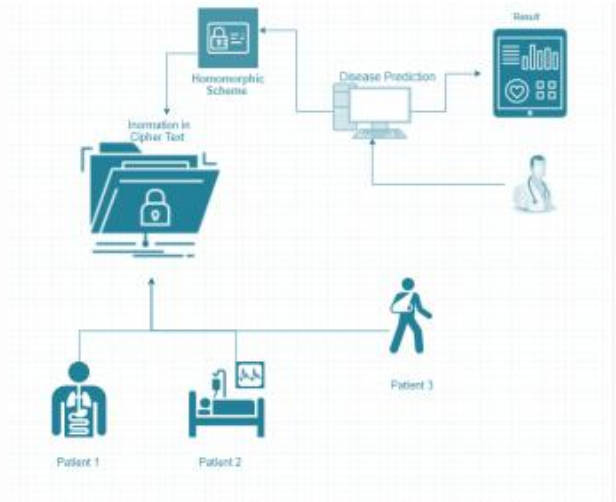

Figure 7:Homomorphic Encryption application in medicine

Without going through a decryption stage, a homomorphic scheme upgraded application software computes on the patient data stored in ciphertext. Using cyphertext as his datasets, the doctor ascertains (predicts) the medical status of his patients. Throughout computing, information privacy and confidentiality are maintained.

b. Finance and Banking: With the use of homomorphic encryption technology, it is possible to forecast client behaviour without having to decrypt their data. Most of the time, banks and other financial organisations need to conduct some sort of statistical analysis on their customers in order to provide financial advise, identify customers for identification purposes, or apply for loans. Because consumer information is private and sensitive, information analysis is exceedingly difficult. An analyst can successfully produce a detailed report using a homomorphic encryption strategy without jeopardising the privacy or confidentiality of the clients' information.

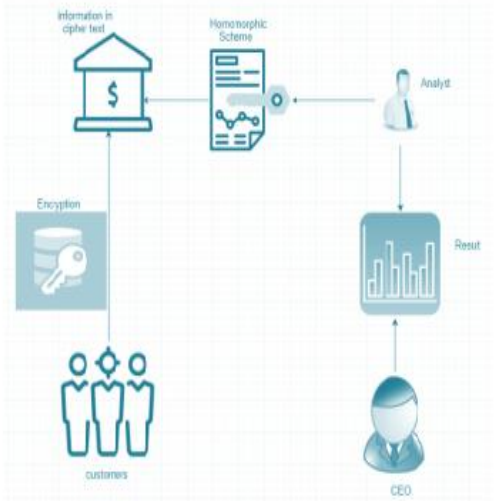

Figure 8: Homomorphic Encryption application in Finance and Banking.

A CEO requested information about some customers in the bank. The data analysis performs computation using homomorphic encryption scheme. Analytical results were made available to the CEO without compromising the customers' privacy and confidentiality.

VI.LIMITATIONS AND FUTURE WORK

1.Additional implementation problems can be found because technology is developing at a rapid pace. Undoubtedly, new problems are developing every day, and in the future, other solutions may be applied.

2. By adding more elements relating to user characteristics, culture, trust, etc. to the theoretical framework for cloud adoption and implementation that is now being used, it may be made better. To give the model greater legitimacy, it may also be empirically evaluated in the future.

3. By executing larger vs lighter apps on VMs Migration, the performance analysis of virtualization work can be further expanded. Moreover, many configuration options can be employed to test the studies for more universally applicable findings.
4. Future changes to the proposed CeIDMS could decentralize it and allow for the addition of a federation function. Moreover, any fully homomorphic algorithm may be used in place of RSA, which is a partial homomorphic algorithm.

VII. CONCLUSION

The ability to store information in the cloud has dramatically improved because to the use of the cloud. Technology is relevant to people and businesses in their daily lives. Nonetheless, everyone has been concerned about the security of information during data calculation. As a result, the development of homomorphic encryption has been a crucial element of a secure cloud computing system. The scheme is divided into full homomorphic encryptions and partial homomorphic encryptions. These algorithms all provide unique difficulties.

a. Partial encryption provides an encryption approach to protect data during computation, but it is limited to addition or multiplication operations. It is incapable of doing both.
b. Partial homomorphic is preferable than somewhat homomorphic. As a component of the cryptosystem, it included the evaluate function. The problem of arbitrary depth exists here.
c. The somewhat and full homomorphism both have the same potential. The approach slightly resolves the arbitrary depth problem, but memory consumption remains a drawback.

One of the most significant and applicable types of techniques to protect the privacy of data in the cloud is homomorphic encryption. The confidentiality of data is maintained by all forms of homomorphic encryption techniques, whether they enable processing of encrypted data fully, partially, or not at all. The fact that the entire process will be based on the plain text, which might potentially be contaminated or harmful at the end, means that the original result may not be compatible with the decrypted data, despite the fact that no security technique is without flaws.The purpose of implementing RSA as a multiplicative homomorphic encryption method is to create a novel IDM system with more secure features, which is described in the following chapter.

REFERENCES

[1]. ZvikaBrakerski, VinodVaikuntanathan“Efficient Fully Homomorphic Encryption “LWE, 2010
[2]. SigrunGoluch, “The development of homomorphic cryptography”Vienna University of Technology, 2009
[3]. Defense Signals Directorate "Cloud Computing Security Considerations" Cyber Security Operations Centre, vol. no. 2, Issue 5, 2011
[4]. Ponemon Institute “Encryption in the Cloud” Thales e-Security, 2009
[5]. Anthony T. Velte Toby J. Velte, Ph.D. Robert Elsenpeter, 2010 “Cloud Computing: A Practical Approach," 2011
[6]. FraunhoferVerlag “This security Of Cloud Storage Services” Fraunhofer Institute for Secure Information Technology, 2012
[7]. BhavnaMakhija, VinitKumar Gupta "Enhanced Data Security in Cloud Computing with Third-Party Auditor", International Journal of Advanced Research in Computer Science and Software Engineering, 2013
[8]. Dawn Song, Elaine Shi, “Cloud Data Protection for the Masses” IEEE Computer Society, 2012
[9]. Deyan Chen, Hong Zhao “ Data Security, and Privacy Protection Issues in Cloud Computing" International Conference on Computer Science and Electronics Engineering,2012 [10] DeepanchakaravarthiPurushothaman and Dr.SunithaAbburu “An Approach for Data Storage Security in Cloud Computing” IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 1, 2012
[10]. SimarjeetKaur “Cryptography and Encryption In Cloud Computing” VSRD-IJCSIT, Vol. 2 (3), 2012, 242-249, 2012
[11]. SanjoliSingla, Jasmeet Singh “Cloud Data Security using Authentication and Encryption Technique” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 7, July 2013
[12]. Mark D. Ryan, “Cloud Computing for Enterprise Architectures: Concepts, Principles, and Approaches," 2013, edition 4th