Secure data on multi-cloud using Homomorphic Encryption

Miss.Vaishali H. Bhavsar[1], PROF. Harish K. Barapatre[2]

*[1]SES's YADAVRAO TASGAONKAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,
DEPARTMENT OF COMPUTER ENGINEERING.
[2]PROFESSOR, VILLAGE –CHANDHAI, TALUKA–KARJAT, DISTRICT-RAIGAD (MAHARASHTRA), INDIA.*

---***---

***ABSTRACT**- The purpose of homomorphic encryption is to allow computation on encrypted data. Thus data can remain confidential while it is processed, enabling useful tasks to be accomplished with data residing in entrusted environments. In a world of distributed computation and heterogeneous networking this is a hugely valuable capability. Many conventional encryption schemes possess either multiplicative or additive homomorphic property and are currently in use for respective applications. Yet, a Fully Homomorphic Encryption (FHE) scheme which could perform any arbitrary computation over encrypted data appeared in 2009 as Gentry's work. In this paper, we propose a multi-cloud architecture of N distributed servers to repartition the data and to nearly allow achieving an FHE.*

Keywords-: multi- cloud; partial and fully homomorphic encryption; public key; asymmetric; secret key.

1. INTRODUCTION

Encryption is the process of using an algorithm to transform information to make it unreadable for unauthorized users. This cryptographic method protects sensitive data such as credit card numbers by encoding and transforming information into unreadable cipher text. This encoded data may only be decrypted or made readable with a key. Symmetric-key and asymmetric-key are the two primary types of encryption. Encryption is essential for ensured and trusted delivery of sensitive information.

Cloud computing is a comprehensive solution that delivers IT as a service. It is an Internet-based computing solution where shared resources are provided like electricity distributed on the electrical grid. Computers in the cloud are configured to work together and the various applications use the collective computing power as if they are running on a single system.

The flexibility of cloud computing is a function of the allocation of resources on demand. This facilitates the use of the system's cumulative resources, negating the need to assign specific hardware to a task. Before cloud computing, websites and server-based applications were executed on a specific system. With the advent of cloud computing, resources are used as an aggregated virtual computer. This amalgamated configuration provides an environment where applications execute independently without regard for any particular configuration.

RSA is a multiplicatively homomorphic encryption algorithm where the decryption of the product of two encrypted data will be the product of the two plain data. However, RSA doesn't allow addition operation nor the combination of multiplications and additions. A fully homomorphic encryption scheme enables computation of arbitrary functions on encrypted data.. This enables a customer to generate a program that can be executed by a third party, without revealing the underlying algorithm or the processed data. Unfortunately, all implementations of FHE schemes showed that this technique is still much too slow for practical applications.

Following are the sections organised as follows. Homomorphic encryption and related types are introduced in section II. In section III, we discuss some homomorphic encryption schemes and finally in section IV, we propose multi-cloud architecture and then V will be our conclusion….

2 .HOMOMORPHIC ENCRYPTION TYPES

Homomorphic encryption is the encryption scheme which means the operations on the encrypted data. Homomorphic encryption can be applied in any system by using various public key algorithms. When the data is transferred to the public area, there are many encryption algorithms to secure the operations and the storage of the data. But to process data located on remote server and to preserve privacy, homomorphic encryption is useful that allows the operations on the cipher text, which can provide the same results after calculations as the working directly on the raw data. In this paper, the main focus is on public key cryptographic algorithms based on homomorphic encryption scheme for preserving security.

There are two types of homomorphic encryptions:-

1] Partial Homomorphic Encryption.

2] Fully Homomorphic Encryption.

1] Partial Homomorphic Encryption:-

Cryptosystem is considered partially homomorphic if it exhibits either additive or multiplicative homomorphism, but not both. The efficiency of some partial Homomorphic encryption schemes is high enough for practical applications.

Most partial Homomorphic encryption schemes only support one type of operation, e.g. additions for Paillier and multiplications for RSA.

A) Paillier Additively homomorphic scheme:-

The Paillier cryptosystem invented by and named after Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing *n*th residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based. The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m1 and m2, one can compute the encryption of m1+m2.

The three steps (key generation, encryption and decryption) can be found in the following table:

Select two large primes, p and q.

Calculate the product n=p x q, such that

GCD (n, Φ (n)) = 1, where Φ (n) is Euler Function

Choose a random number g, where g has order multiple of n or gcd (L (gλ mod n2) n) = 1

Where L (t) = (t-1) / n and λ (n) = LCM (p-1 q-1)

The public key is composed of (g, n), while the

Private Key is composed of (p, q, λ).

The Encryption of a message m < n is given by:

- c=gmr n mod n2

The Decryption of cipher text c is given by:

- m= (L (gλ mod n2)/L (gλ mod n2)) mod n

Paillier Algorithm

B) RSA Multiplicatively homomorphic scheme:-

RSA (Rivest–Shamir–Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private). In RSA, this asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers. The "factoring problem "A user of RSA creates and then publishes a public key based on two large prime numbers , along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, and if the public key is large enough, only someone with knowledge of the prime numbers can decode the message feasibly breaking. RSA encryption is known as the RSA problem. Whether it is as difficult as the factoring problem remains an open question.

Key Generation:

1. Generate two large prime numbers, p and q.

2. Let n=p q

3. Let m= Ø (n) = (p-1) (q-1)

4 Choose a small number e, co prime to m

With GCD (Ø (n), e) =1; 1<e<= Ø (n)

5. Find d, such that de mod Ø (n) =1

Publish e and n as a public key.

Keep d and m as a secret key.

Encryption:

Cipher= (message) ^{e} mod n

Decryption:

Message= (cipher) ^{d} mod n

RSA algorithm

2. Fully Homomorphic Encryption :

A cryptosystem that supports arbitrary computation on cipher texts is known as fully homomorphic encryption (FHE) and is far more powerful .Such a scheme enables the construction of programs for any desirable functionality, which can be run on encrypted inputs to produce an encryption of the result .The existence of an efficient and fully homomorphic cryptosystem would have great practical implications in the outsourcing of private computations.

3. HOMOMORPHIC ENCRYPTION SCHEMES

Partial Homomorphic Encryption Schemes:-

1. Goldwasser-Micali Cryprosystem:-

The Goldwasser-Micali (GM) Cryptosystem is a public-key encryption algorithm developed in 1982. It is the first probabilistic public-key encryption scheme which is provably secure under standard cryptographic assumptions. It is based on the intractability of Quadratic Residuosity Assumption modulo a composite N. Very roughly, we select uniformly at random Quadratic Residues from Z ∗ N to encrypt 0 bit and to encrypt 1 bit we select quadratic non-residue from Z ∗ N. However, since the distribution of quadratic residues and quadratic non-residues are not same in Z ∗ N we confine ourselves to a subset of Z ∗ N where the number of quadratic residues is equal to the number of quadratic non-residues.

2. Paillier scheme:-

The Paillier Cryptosystem named after and invented by French researcher Pascal Paillier in 1999 is an algorithm for public key cryptography. The distinguishing technique used in public key cryptography is the use of asymmetric key algorithms, where the key used to encrypt a message is not the same as the key used to decrypt it. Each user has a pair of cryptographic keys — a public key and a private key. The private key is kept secret, whilst the public key may be widely distributed. Messages are encrypted with the recipient's public key and can only be decrypted with the corresponding

Private Key. The keys are related mathematically, but the private key cannot be feasibly (in actual or projected practice) derived from the public key.

3. Benaloh's scheme:-

The Benaloh Cryptosystem is an extension of the Goldwasser- Micali Cryptosystem (GM) created in 1994 by Josh (Cohen) Benaloh. The main improvement of the Benaloh Cryptosystem over GM is that longer blocks of data can be encrypted at once, whereas in GM each bit is encrypted individually.[

Fully Homomorphic Encryption Schemes:-

1. Gentry's scheme:-

He proposed fully homomorphic encryption consists of several steps: It constructs a somewhat homomorphic scheme that supports evaluating low-degree polynomials on the encrypted data. It squashes the decryption procedure so that it can be expressed as a low-degree polynomial which is supported by the scheme It applies a bootstrapping transformation to obtain a fully homomorphic scheme.

2. Smart and Vercauteren scheme:-

They present a fully homomorphic encryption scheme has smaller key and cipher text sizes .The construction proposed by the authors follows the fully homomorphic construction based on ideal lattices proposed by Gentry. It produces a fully homomorphic scheme form a somewhat homomorphic scheme. For somewhat homomorphic scheme, the public and the private keys consist of two large integers (one of which shared by both the public and the private key), and the cipher text consists of one large integer.

3. Gentry and Halev scheme:-

They presented a novel implementation approach for the variant of Smart and Vercauteren proposition which had a greatly improved key generation phase. In

Particular, the authors have noted that key generation (for cyclotomic fields) is essentially an application of a Discrete Fourier Transform (DFT), followed by a small quantum of computation and then application of the inverse transform. The key generation method of Gentry and Halevi is fast

4. MULTI CLOUD ARCHITECTURE

The fully homomorphic encryption schemes [1] are very time consuming. Considering the evaluation of one gate demanding a refresh, the run-time will be significant as well as the processing of security parameters. A suggestion of a nearly FHE scheme based architecture for enabling the evaluation of any function and processing encrypted data is illustrated in Figure 6. In our proposed architecture, the service provider repartitions the processing among the servers to fasten the evaluation process of any function

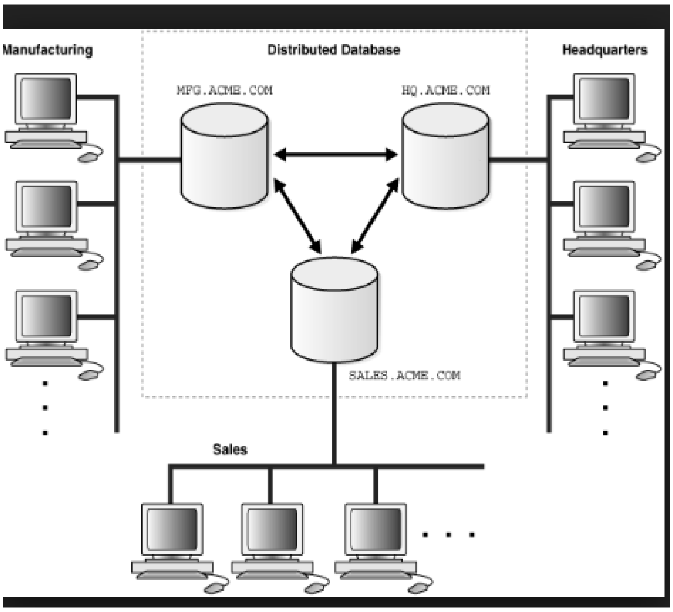

Fig A. architecture of distributed servers for processing encrypted data

In this system, we provide a high levelled architectural scheme through the usage of multiple Servers in the Computation. impersonating a legitimate user, there by infecting the entire cloud. This leads to affects many customers who are sharing the infected cloud. There are five types of issues raise while discussing security of a cloud Data Issues, Privacy issues, Infected Application, Security issues and Trust Issues. Homomorphic cryptosystem plays very important role with these issues.

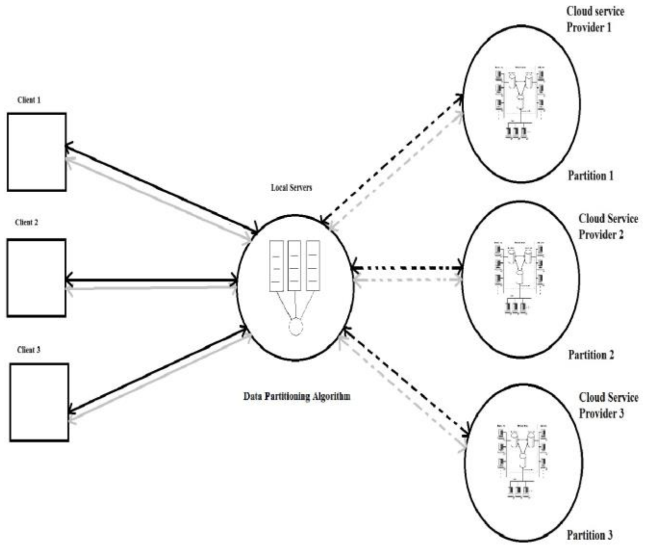

Fig B. Proposed architecture to secure data using Homomorphic encryption

This computation System will nearly allow achieving a FHE, and thus large number of operations including multiplications and additions can be performed. For instance, in Fig A, Client 1 requests the results of a given function, let's say f(x) =ax^2+bx+c. In this case the function elements are encrypted and divided into several chunks depending on the number of operations (Multiplication and addition), and will be processed separately on N different servers, equivalent to the number of addition operations Finally the result is sent back to a Central Server in order to be forwarded to Client 1 and then decrypted.

The benefit is that no longer chipertext after encryption unlike the classical method. The keys are easily handled and more security is maintained since is it impossible to read relevant information in distributed systems. In the cloud the N servers consists of hypervisors hosting multiple virtual machines which help improving the response time and augment the number of the involved computational entities in the distributed system. In this suggestion, we analyze the added value of the distributed systems in processing operations requested by clients. The scheme of homomorphic encryption is dispatched within the servers and this can be practical and help improving the security of the cloud in terms of confidentiality of data and performance.

5. HOMOMORPHIC ENCRYPTION APPLICATION

Security in Cloud Computing: The cloud service provider for cloud makes sure that the customer does not face any problem such as loss of data or data alteration. There is also a possibility where a malicious user can penetrate the cloud by.

Multiparty computation: In multiparty computation schemes, several parties are interested in computing a common, public function on their inputs while keeping their individual inputs private. This problem belongs to the area of computing with encrypted data. Usually in multiparty computation protocols, we have a set of n≥2 players whereas in computing with encrypted data scenarios n=2. Furthermore, in multi-party computation protocols, the function that should be computed is publicly known, whereas in the area of computing with encrypted data it is a private input of one party**.**

Secret sharing scheme: In secret sharing schemes, parties share a secret so that no individual party can reconstruct the secret form the information available to it. However, if some parties cooperate with each other, they may be able to reconstruct the secret. In this scenario, the homomorphic property implies that the composition of the shares of the secret is equivalent to the shares of the composition of the secrets.

6. CONCLUSION

In this paper we discuss the need of Homomorphic cryptosystem in today's digital era where security, confidentiality and eavesdrop issues are concerned. We try to list out different types of Homomorphic cryptosystem as present today, under the two broad categories namely partial and fully Homomorphic cryptosystem. Main advantage of Homomorphic cryptosystem is the ability to perform computation on encrypted data, with some limitation depending on a chosen algorithm. We discuss application of Homomorphic cryptosystem in various areas such as security in Cloud Computing, Untrusted Web Servers, Secret sharing scheme, Protection of mobile agents, Election schemes, Watermarking etc. Even there are many probable areas where this can be cleverly used to get desired benefits.

Distributed systems and multi-could architectures could bring lots of benefits to the application of homomorphic encryption and making it more practical in the case of the security of data and applications. In a future work, we will focus on the implementation of our proposal and conduct security and performance tests in order to show its practicability.

REFERENCES

[1] C. Gentry, "A fully homomorphic encryption scheme," Doctoral dissertation, Stanford University, 2009.

[2] Monjul Saikia,"A Brief overview of Homomorphic Cryptosystem and their Application", Nirjuli, 2015

[3]S. Sobitha Ahila et al Int. Journal of Engineering Research and Applications," State Of Art in Homomorphic Encryption Schemes", 2014

[4] Payal. V. Parmar,”Survey of Various Homomorphic Encryption algorithms and Schemes”, 2014.

[5] Mohd Rahul, Hesham A. Alhumyani, Mohd Muntjir,” An Improved Homomorphic Encryption for Secure Cloud Data Storage”, Taif University, 2017.

[6] J.Bringe and al., “An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication”, Springer-Verlag, 2007.

[7] C. Gentry, “Fully homomorphic encryption using ideal lattices,” InSTOC, Vol. 9, pp. 169-178, 2009.

[8]https://www.techopedia.com/definition/5507/encryption.

[9]] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” In 18th Annual Eurocrypt Conference (EUROCRYPT'99) Prague, Czech Republic , volume 1592, 1999.

[10] S. Goluch, “The development of homomorphic cryptography: From RSA to Gentry’s privacy homomorphism” Doctoral dissertation, Vienna university of Technology, 2010.