IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Comparative Study of various Partially Homomorphic Cryptography Techniques in Cloud

[1]Sourav Karmakar, [2]Ira Nath, [3]Rajesh Kumar Manjhi

[1,2,3]JIS College of Engineering

Abstract: Homomorphic encryption (HE) is an encryption technique where operations are performed on ciphertext. This encryption method can be used in varieties of applications by using public key algorithms. For transferring data from one place to another, there are various encryption algorithms for storage of data and securing the operations, but they do not preserve privacy. HE is useful in various applications in which HE performs the different operations on encrypted data and provides results after calculations performed directly on the plaintext. Nowadays, security of information and calculations to deal with the data of big business has expanded massively. In any case, a basic issue emerges when there is a necessity of registering on such encrypted information where protection is built up. This paper represents homomorphic cryptosystems for preserving security, properties, and categories of homomorphic encryption. In addition to this, privacy-preserving applications of homomorphic cryptosystems in the field of cloud computing, private information retrieval, and data aggregation in wireless sensor network are also presented.

Keywords: Cryptography,

1. Introduction

Cryptography [1] has been in practical use for a very long time. While it may well have been used previously, extant records indicate that the Spartans established the first system of military cryptography. As early as the fifth century B.C., they employed a device called the 'skytale' [which] consists of a staff of wood around which a strip of papyrus or leather or parchment is wrapped close-packed. The secret message is written on the parchment down the length of the staff; the parchment is then unwound and sent on its way. The existence of cryptography from this early a time, proves that there was a need for data security from a very long time. Cryptographic algorithms have been in existence from the time of Shakespeare (as mentioned in Julius Caesar). This is from where the Caesar algorithm gets its name from, wherein, a letter is encrypted by replacing it with a letter relative to its position in the alphabet. With an increase in the need for securing the data, the level of complexity involved to get through to encrypted data also increased. Algorithms with keys to encode and decode the data were developed. RSA encryption and decryption algorithm talks about secrecy of messages between two users. It involves a public key and a private key generation. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The private key is not publicly known, it is only known to receiver so that she/he can decrypt the encrypted message. These keys “public and private keys" for the RSA algorithm are generated using some mathematical operations. The principal goal of design of any encryption algorithm must be security against unauthorized attacks. within the last decade, there has been a vast increase in the accumulation and communication of digital computer data in both the private and public sectors. Much of this information has a significant value, either directly or indirectly, which requires protection. RSA is an Asymmetric key cryptography which provides security against unauthorized attacks. problem definition that is RSA Algorithm and mathematics behind it (RSA Algorithm) Some idea of RSA algorithm has been given in abstract above. we may refer that again for more information. The encryption and decryption in the RSA algorithm as follows) first, I had to generate the key pair and then those keys were to be used for encryption and decryption. Key generation RSA involves a public key and a private key.

1.1Basic about cryptography

Cryptography is the study and practice of techniques for secure communication in the presence of third parties called adversaries. It deals with developing and analyzing protocols which prevents malicious third parties from retrieving information being shared between two entities thereby following the various aspects of information security.
Secure Communication refers to the scenario where the message or data shared between two parties can't be accessed by an adversary. In Cryptography, an Adversary is a malicious entity, which aims to retrieve precious information or data thereby undermining the principles of information security.

Data Confidentiality, Data Integrity, Authentication and Non-repudiation are core principles of modern-day cryptography.

1.2Homomorphic cryptography

Homomorphic encryption in the cloud is still relatively young and is only being adopted at a slow rate. Even though FHE is currently not plausible to implement for real-world scenarios, there is no reason why PHE cannot offer cloud providers an extra level of security right now. Then in time migrate to FHE when schemes offer better performance. The cloud requires an increased level of security, and homomorphic encryption is a viable answer. Some cloud solutions have already realized this, like the samples in Section 6, but in the near future this will be more common across a far more diverse group of cloud applications and services.
As discussed throughout this chapter, there are still some problems with homomorphic encryption which will impact its future. For example, FHE can support more than a single operation. However, an open issue is that FHE also has limitations on supporting a wide range of operations/functions, even though this is the very definition of FHE. This is because two operations could be used to cancel each other out and make the security pointless. Supporting subtractions by an unencrypted value and comparing the encrypted value with zero would be one case. Because a malicious user can just subtract 1 until the encrypted value is zero, giving the answer. This leads to another issue, currently FHE is thought of as the perfect solution; however, it needs to be considered on an application-by-application basis. A one size fits all solution is not going to be as secure as a scheme which is designed for the application in mind. And finally, by having homomorphic encryption protect user information/data, it stops cloud services from learning information about them. This can stop targeted ads, selling anonymous user data and many other ways cloud services make money even though there is no cost to the end user. The issue is that even though users want to be more secure online, will they be willing to pay for the cloud service, or will they prefer the free, unsecured service instead? These are just some of the current issues that homomorphic encryption faces as it tries to become the future of security in the cloud.

1.3 Classification of homomorphic cryptography

There are three main types of homomorphic encryption. The primary difference between them boils down to the types and frequency of mathematical operations that can be performed on their ciphertext. The three types of homomorphic encryption include:

- Partially Homomorphic Encryption
- Somewhat Homomorphic Encryption
- Fully Homomorphic Encryption

2.Partially homomorphic encryption (PHE)[2]

It helps sensitive data remain confidential by only allowing select mathematical functions to be performed on encrypted values. This means that one operation can be performed an unlimited number of times on the ciphertext. Partially homomorphic encryption (with regard to multiplicative operations) is the foundation for RSA encryption, which is commonly used in establishing secure connections through SSL/TLS. Some examples of PHE include ElGamal encryption (a multiplication scheme) and Paillier encryption.

Pailler algorithm[1, 2]The Paillier cryptosystem, invented by Pascal Paillier in 1999, is a partial homomorphic encryption scheme which allows two types of computation:
addition of two cipher texts
multiplication of a cipher text by a plaintext number

Elgamal algorithm[3, 4]The Elgamal encryption system is an asymetric key encryption algorithm for public-key cryptography which is based on the Diffie Hellman key exchange.The system provides an additional layer of security key asymmetrically encrypting keys previously used for symmetric message encryption.

RSA algorithm[5] is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and Private key is kept private.

3.Description of various homomorphic encryption techniques

Homomorphicencryption is a cryptographicmethod that allows mathematical operations on data to be carried out on cipher text, instead of on the actual data itself. The cipher text is an encrypted version of the input data (also called plain text). It is operated on and then decrypted to obtain the desired output.Homomorphicencryption is a form of encryption allowing one to perform calculations on encrypted data without decrypting it first. The result of the computation is in an encrypted form, when decrypted the output is the same as if the operations had been performed on the unencrypted data.

3.1RSA cryptography[5]

RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and Private key is kept private.

An example of asymmetric cryptography:

1. A client (for example browser) sends its public key to the server and requests for some data.
2. The server encrypts the data using client’s public key and sends the encrypted data.
3. Client receives this data and decrypts it.

3.2Elgammal cryptography[3, 4]

The Elgamal encryption system is an asymetric key encryption algorithm for public key cryptography.Public key system proposed by Deffie and Hellman requires interacting of both parties to calculate a common private key.TheElgamal simplified the key exchange algorithm by introducing a exponent k.Today the Elgamal algorithm is used in many cryptographic products.

Mathematical Steps:

Key generator

- In Elgamal , only the receiver needs to create a key in advance and publish iy. As we discussed above , we will now follow through his procedure of key generation. Bob will take the following steps to generate his key pair:

1. Prime and group generation
2. Private key selection
3. Public key assembling
4. Public key publishing

3.3Pailler cryptography[1, 2]

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.

Key generation

Key generation works as follows:

- ✓ Pick two large prime numbers p and q, randomly and independently. Confirm that gcd(pq,(p−1)(q−1)) is 1. If not, start again.
- ✓ Compute n=pq.
- ✓ Define function L(x)=x−1n.
- ✓ Compute λ as lcm(p−1,q−1).
- ✓ Pick a random integer g in the set Z∗n2 (integers between 1 and n2).
- ✓ Calculate the modular multiplicative inverse μ=(L(gλmodn2)) −1modn. If μ does not exist, start again from step 1.
- ✓ The public key is (n,g). Use this for encryption.

The private key is λ. Use this for decryption.

Uses of Paillier Cryptography

Paillier's additive homomorphic encryption is increasingly used in recent research in the field of cloud secure outsourcing and privacy-preserving computation in addition to other cryptographic tools such as garbled circuits.

- ISBN Information: Electronic ISBN: 978-1-4673...
- INSPEC Accession Number: 15620417

4.Simulation Results

4.1 Comparative Study due to Encryption

Input size(bytes)	**Elgamal**(sec)	**Pailler**(sec)	**RSA**(sec)
32	0.12	0.24	0.01
64	0.24	0.36	0.007
105	0.69	0.146	0.085
124	0.75	0.225	0.105

From the above table we can see that RSA algorithm performs a faster encryption process than Pailler and Elgamal Algorithm. On the other hand it is also proven that Elgamal Algorithm performs a faster encryption process than Pailler algorithm..

4.1 Comparative Study due to Decryption

Input size(bytes)	**Elgamal**(sec)	**Pailler**(sec)	**RSA**(sec)
32	0.48	0.48	0.021
64	0.75	0.87	0.048
105	0.98	0.78	0.066
124	0.87	0.98	0.11

From the above table we can see that RSA algorithm performs a faster decryption process than Pailler and Elgamal Algorithm. On the other hand it is also proven that Elgamal Algorithm performs a faster decryption process than Pailler algorithm..

5.Conclusion

Generally, the Elgamal encryption system is an asymmetric key encryption algorithm for public key cryptography. The Elgamal simplified the Diffie-Hellman key exchange algorithm by introducing a random exponent K. We have surveyed the two algorithms that are commonly used in the previous paper. The paper comparestwo RSA and El-Gamal for secure file transmission.In this paper, the summary table reports the key length value, type of algorithm, security attacks, simulation speed, scalability, key used, power consumption, and hardware/ software implementation difference between RSA and EL-Gamal. In future we are implementing and performing comparative analysis of time taken for 2 Asymmetric key cryptography algorithms RSA and El-Gamal by changing some of the parameters. The size of encrypted message, key management method and algorithm efficiency is critical in respect of implementation in organizations that store large volumes of data.The execution time as a function of the encryption key and the file size will be examined and complexity and security will be checked.

REFERENCES

1. Nassar, Mohamed, Abdelkarim Erradi, and Qutaibah M. Malluhi. "Paillier's encryption: Implementation and cloud applications." In *2015 International Conference on Applied Research in Computer Science and Engineering (ICAR)*, pp. 1-5. IEEE, 2015.
2. Wu, Hao-Tian, Yiu-ming Cheung, and Jiwu Huang. "Reversible data hiding in Paillier cryptosystem." *Journal of Visual Communication and Image Representation* 40 (2016): 765-771.
3. Khoirom, Motilal Singh, Dolendro Singh Laiphrakpam, and ThemrichonTuithung. "Audio Encryption Using Ameliorated ElGamal Public Key Encryption Over Finite Field." *Wireless Personal Communications* 117, no. 2 (2021): 809-823.
4. Kiltz, Eike, and Krzysztof Pietrzak. "Leakage resilient elgamal encryption." In *International conference on the theory and application of cryptology and information security*, pp. 595-612. Springer, Berlin, Heidelberg, 2010.
5. Zhou, Xin, and Xiaofei Tang. "Research and implementation of RSA algorithm for encryption and decryption." In *Proceedings of 2011 6th international forum on strategic technology*, vol. 2, pp. 1118-1121. IEEE, 2011.