ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

An Enhanced Paillier's Algorithm Using Homomorphic Encryption Thresholding

Rahul Yogi[*1], Mrs. Pushpa.G[2]

[*1,2]Department of CSE, Siddaganga Institute of Technology, Tumkur – 572103 Karnataka, India

rahulyogi77@gmail.com

Abstract

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. When the data is transferred to the sink we use standard encryption methods to secure this data, however it is necessary that the sink has access to the raw data, and then it will decrypt them [01]. .Among many schemes the Paillier scheme fits the criteria but does not expose the complete required behavior of the system. However, if the Paillier Cryptosystem can be used to conceal information, with a few interesting advancements. These properties, when creatively applied, allow the Paillier Cryptosystem to be used in ways that other cryptographic systems simply can't be used. In many situations it is desirable to distribute the decryption process amongst a number of parties such that a message can only be decrypted if a certain qualified subset of these parties participate in the decryption process which is a Thresholding property and the other enhancement is the ciphering algorithm called blowfish algorithm This paper will explore how the Paillier Cryptosystem works, how these enhancements are used as an integration attempt to the Paillier cryptosystem [02].

***Key Words*—Homomorphic Cryptography, security, Paillier, Thresholding, Blow Fish, Additive Encryption, Multiplicative Encryption.**

I. INTRODUCTION

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. The main aim of homomorphic encryption is how to make it possible to combine two encryptions to get an encryption that encrypts some combination of the two original plaintexts.

The most useful variant of these is the additive homomorphisms, where the new encryption is the sum of the two original plaintexts. Systems with additive homomorphism's are useful for a lot of crypto logical protocols like electronic voting, multi-party computation, etc., providing that they also support verification protocols. There is also the question of threshold decryption. In multi-party computation models, all the actual computations are made inside encryptions using the homomorphic properties of the specific cryptosystem. However, this implies that to get the result someone has to decrypt at some point. This is where threshold decryption is used. It allows servers to decrypt if less than half the servers are trying to cheat or disrupt the protocol.

All these topics are interesting by themselves, but the real interesting things start to happen when they are combined. This leads to an additive homomorphic length-flexible threshold cryptosystem. This enables one to set up a system once and for all, and start doing computation on problems of various sizes. This is especially true in case of electronic voting where the same system might be used to hold a local election and a national election. These two forms of election have very different sizes and in normal cryptosystems it would be necessary to create two different cryptosystems, or one would risk that the local election could become very inefficient because it is using a cryptosystem designed for a national election. This is where length-flexibility comes in handy, in the sense that now the plaintext space can be adjusted to create both the elections in a setting that is essentially as efficient as possible for both problems [03]. In context to homomorphic encryption we know that lot of advantage to the computer science of today and tomorrow. But the adoption of sink passage applies only if the security is ensured. How to ensure better data security and how a client can keep their private information confidential? There are two major questions that present a challenge for providers of network to transfer the data from any source to any destination. Our basic concept was to encrypt the data before sending to the network provider. But there is a problem still faced by the client. Because the network provider needs to perform the calculations on data to respond the request from the client so he must provide the key to the server to decrypt the data before execute the calculations required, which might affect the confidentiality of data stored in the sink. A method enable to perform the operations on encrypted data without decrypted them is the Homomorphic Encryption [01].

There are huge amount of research has been done in cryptography but few of them are convincing, in those many encryption schemes which has been presented all are based on some basic ideas. In principle it is possible to distinguish two

ISSN: 2277-9655
Impact Factor: 1.852

main types of cryptosystems: the ones based on RSA (or related assumptions) and those based on discrete log (or related assumptions).

Of course some different solutions have been proposed, but either they suffer from inefficiency or security flaws. To this "class" belong almost all lattice based cryptosystems and the knapsack-type schemes. Another promising direction is the one of cryptography based on the theory of braid groups [12], but, again, this is a too insufficiently studied area to be completely reliable. At the very end all the "trusted" schemes are RSA or discrete log based schemes [04].

In many situations it is desirable to distribute the decryption process amongst a number of parties such that a message can only be decrypted if a certain qualified subset of these parties participates in the decryption process. For example, consider a cryptographic election protocol where the homomorphic properties of a cryptosystem are used to anonymously calculate the encrypted tally for each candidate. Granting knowledge of the decryption key to any single election official would allow that official to decrypt any ballot in the system, thus learning how an individual voter cast their vote. In order to combat this problem, the decryption key could distribute amongst several election officials, allowing a message to be decrypted if and only if some large enough subset of election officials agree to do so. A cryptosystem that supports such a process is called a threshold cryptosystem [05].

Hence in this paper our route takes a slightly modest approach of choosing the Paillier's algorithm and enhances its system by adding the Thresholding concept to suppress its malleability measure and amplify the performance parameter with the support of the blowfish algorithm which is used in the generation of the cipher.

II. HOMOMORPHIC ENCRYPTION

In 1978 Ronald Rivest, Leonard Adleman and Michael Dertouzos suggested for the first time the concept of Homomorphic encryption [13]. Since then, little progress has been made for 30 years. The encryption system of Shafi Goldwasser and Silvio Micali was proposed in 1982 was a provable security encryption scheme which reached a remarkable level of safety, it was an additive Homomorphic encryption, but it can encrypt only a single bit. In the same concept in 1999 Pascal Paillier was also proposed a provable security encryption system that was also an additive Homomorphic encryption. Few years later, in 2005, Dan Boneh, Eu-Jin Goh and Kobi Nissim [14] invented a system of provable security encryption, with which we can perform an unlimited number of additions but only one multiplication [01].

Homomorphic cryptography provides a third party with the ability to perform simple computations on encrypted data without revealing any information about the data itself. Typically, a third party can calculate one of the encrypted sum or the encrypted product of two encrypted messages. This is possible due to the fact that the encryption function is a group homomorphism, and thus preserves group operations. This makes homomorphic cryptosystems useful in a wide variety of privacy preserving protocols. The wide variety of protocols utilizing homomorphic cryptography makes it difficult to provide a comprehensive survey, and while an overview of applications is given, it is limited in scope and intended to provide an introduction to the various ways in which homomorphic cryptography is used beyond simple addition or multiplication of encrypted messages. In the case of strong conditional oblivious transfer, a new protocol implementing the greater than predicate is presented, utilizing some special properties of the Boneh-Goh-Nissim cryptosystem to achieve security against a malicious receiver [05].

Homomorphic encryption schemes are special cases of asymmetric cryptosystems. As in asymmetric systems there is a public key which is used for encryption and a private key for decryption. Additionally specific algebraic operations performed on a plaintext are equivalent to other (possibly different) algebraic operations performed on the cipher text. These encryption schemes can be additively and/or multiplicatively homomorphic [07].

A. Additive Homomorphic Encryption

Data aggregation is a popular approach employed in networks to minimize data transmission and storage. With aggregation techniques, the monitored data is expressed in a condensed form: Therefore, instead of storing all data sensed by several nodes, the network stores a condensed value only such as the sum of these values. However, data aggregation becomes problematic when the data to be aggregated is encrypted. As a solution, we apply an additive homomorphic encryption scheme, namely the elliptic curve cryptosystem, and present the performance results of our implementation for the prominent platform MicaZ mote. In synchronous networks the monitored data transmitted to a reader device is real-time responsive. In asynchronous networks, however, the monitored data is transmitted to a reader device only seldomly. Therefore, asynchronous networks need to store the data in the network in a distributed manner. However, the implementation of distributed data storage for open networks is very challenging. Thus, it is necessary to reduce the amount of the data being processed, e.g. stored or transmitted, in the network without losing relevant information. Secondly, the nodes have limited power capacity. Distributed data storage requires transmission between sink nodes. Since the transmission affects the power consumption, techniques for minimizing it are mandatory. Finally, the nodes are in general equipped with non-tamper-resistant hardware. Since open networks are usually employed in a public environment, data must be protected and concealed.

B. Multiplicative Homomorphic Encryption

All the currently existing homomorphic schemes are based on additive homomorphism. The other scheme based on multiplicative homomorphism is proposed. In the tallying phase, a decryption is performed to recover its product, instead of the sum of them (as in the additive homomorphic schemes). Then, the product is factorized. The new scheme is more efficient than the additive homomorphic schemes and more efficient than other schemes when the number of candidates is small. Strong privacy and public verifiability are obtained in the new scheme. Two main methods have been applied to design these schemes: mix network and

ISSN: 2277-9655
Impact Factor: 1.852

homomorphic tallying. Both methods can protect privacy when threshold trust is assumed. In regard to efficiency, mix network is more suitable for places with a large number of candidates or choices and homomorphic tallying is more suitable for places with a small number of candidates or choices as the latter's cost is linear in the number of candidates or choices. Current homomorphic schemes employ an additive homomorphic encryption algorithm (e.g. Paillier encryption) to encrypt and exploit additive homomorphism of the encryption algorithm to recover the sum of votes for any candidate or choice with a single decryption. As no single vote is decrypted, vote privacy is protected. It is surprising that multiplicative homomorphism has never been employed to design any scheme of data transferring, although it may lead to better performance. In a multiplicative homomorphic voting scheme, a multiplicative homomorphic encryption algorithm (e.g. ElGamal encryption) to encrypt the votes and a single decryption is performed to calculate the product of votes. Then the product is factorized and the votes are recovered. Like in additive homomorphic voting, no single vote is decrypted in multiplicative homomorphic voting, so vote privacy is protected too. The most important advantage of multiplicative homomorphic is that it is always more efficient than additive homomorphic voting and more efficient than other schemes when the number of candidates is small. In brief, multiplicative homomorphic scheme improves efficiency without compromising privacy or public verifiability.

One of two possible additive homomorphic encryption algorithm are usually employed: Paillier encryption or modified ElGamal encryption. Paillier encryption is inherently additive homomorphic and more frequently applied. The original El- Gamal encryption scheme can be simply modified to be additive homomorphic: a message is used as an exponent in an exponentiation computation, then the exponentiation is encrypted using the original ElGamal encryption. A passive result of this modification is that a search for logarithm must be performed in the decryption function, which becomes inefficient when the searching space is not too small. The modified ElGamal encryption is employed in homomorphic voting schemes [15], [16], [17], [18], where the details of the modification and the consequent search can are described in detail [11].

III. EXISTING SYSTEM

In 1978 Ronald Rivest, Leonard Adleman and Michael Dertouzos suggested for the first time the concept of Homomorphic encryption [13]. Since then, little progress has been made for 30 years. The encryption system of Shafi Goldwasser and Silvio Micali was proposed in 1982 was a provable security encryption scheme which reached a remarkable level of safety, it was an additive Homomorphic encryption, but it can encrypt only a single bit. In the same concept in 1999 Pascal Paillier was also proposed a provable security encryption system that was also an additive Homomorphic encryption.

Few years later, in 2005, Dan Boneh, Eu-Jin Goh and Kobi Nissim [14] invented a system of provable security encryption, with which we can perform an unlimited number of additions but only one multiplication [01].

A. ElGamal

In cryptography, the ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie–Hellman key exchange. We have seen that the security of the RSA cryptosystem is related to the difficulty of factoring large numbers. It is possible to construct cryptosystems based on other difficult number-theoretic problems. We now consider the ElGamal cryptosystem, named after its inventor, Taher ElGamal, which is based on the difficulty of a problem called the "discrete logarithm"[08]. ElGamal encryption consists of three components: the key generator, the encryption algorithm, and the decryption algorithm. A disadvantage of the ElGamal system is that the encrypted message becomes very big, about twice the size of the original message m. For this reason it is only used for small messages such as secret keys.

B. Goldwasser-Micali

This deals with the security challenges in authentication protocols employing like volatile authentication features, where the authentication is indeed a comparison between a fresh authentication template and that enrolled during the enrollment phase. We propose a security model for authentication protocols by assuming that the authentication features to be public. Extra attention is paid to the privacy issues related to the sensitive relationship between the authentication feature and the relevant identity. Relying on the Goldwasser-Micali encryption scheme, we introduce a protocol for authentication and prove its security in our security model. Security protocols generally rely on exact knowledge of some data, such as a cryptographic key, however there are particular applications where environment and human participation generate variability. In authentication based cryptosystems, when a user identifies or authenticates himself using his biometrics, the authentication feature, which is captured by a sensor, will rarely be the same twice.

Thus, traditional cryptographic handling such as a hash value is not suitable in this case, since it is not error tolerant. As a result, the identification or authentication must be done in a special way, and moreover precaution is required to protect the sensitivity (or privacy) of authentication.

We here consider a practical environment where a human user wants to authenticate himself to a database using his authentication procedure. A typical scenario is that some reference authentic data is stored inside a database, through which the server authenticates the user by checking whether or not a "fresh" template sent by the sensor matches with the reference one. It can be extracted into binary strings. Therefore, an authentication leads to a comparison between two binary vectors. If the Hamming distance is adopted, then a comparison consists of computing the Hamming distance between the reference data and the fresh template and comparing this to a threshold.

ISSN: 2277-9655
Impact Factor: 1.852

The Goldwasser-Micali scheme suffers from two aesthetic drawbacks:

1) The signature scheme is not completely memory less". That is, the signature generated by the signer slightly depends on the previous signed messages.

2) The signing process in the suggested factoring-based implementation is too slow [10].

C. Paillier's cryptographic algorithm

The Paillier Cryptosystem is a modular, public key encryption scheme, created by Pascal Paillier, with several interesting properties with the underlying mathematical principles that make the system work clearly outlined. It is assumed that the reader is familiar, to some degree, with modular arithmetic, as well as the concept of converting an alphanumeric message into a purely numeric message, which can be broken into blocks, mi, such that, for each i, 0 < mi < n, for a predetermined value, n. Also, the term plaintext will be used to refer to a message that is numeric, but is not encrypted, while the term cipher text will be used to refer to plaintexts which have been encrypted, but not yet decrypted. However the Paillier system forms the strong bond for maintaining the security but still lags behind in providing the improved security for the data. Lacking in providing both malleability and performance.

IV. PROPOSED SYSTEM

The Paillier Cryptosystem is a modular, public key encryption scheme, created by Pascal Paillier, with several interesting properties. This paper will explore the enhanced Paillier's work by concatenating the concept of the cryptographic Thresholding, Which distributes the process amongst a number of parties such that a message can only be decrypted if a certain qualified subset of these parties participates in the decryption process. A cryptosystem that supports such a process is called a threshold cryptosystem. It is assumed that the reader is familiar, to some degree, with modular arithmetic, as well as the concept of converting an alphanumeric message into a purely numeric message. Also, the term plaintext will be used to refer to a message that is numeric, but is not encrypted, while the term cipher text will be used to refer to plaintexts which have been encrypted, but not yet decrypted. In order to illustrate the system's potential. We will initiate with the encryption process with the combination of the blowfish algorithm to output the encrypted stream called the cipher text. . It takes a variable-length key from 32 bits to 448 bits which makes it more feasible than other encrypting algorithms maintain its reliability and superiority parameters. Blowfish is one of the fastest block ciphers which has developed to date. No attack is known to be successful against it. Since in the security algorithms performance along with malleability forms a mandatory criterion. Since there doesn't exist yet known concept on the Paillier's algorithms that confronts the malleability and performance concept and makes the concept stupendous. Hence in supporting the idea we try to bring out a novel system that takes into measures both the two parameters.

A. Encryption

The traditional form of the algorithm which fits the malleable limitation. However to over this drawback we characterize an enhancement to the Paillier's algorithm by adding the additional combinations of features so that the adversary probability of getting through the data would be effectively controlled. The rest of the content would conceptualize our algorithm which goes as follows.

Initially, the dealer can creates an instance of the cryptosystem and distribute l secret shares amongst l parties such that any subset of t or more parties can cooperate to decrypt a message. Initially to create a public key, Let $p = p_1, p_{2,\ldots\ldots}p_n$ and $q = q_1, q_{2\ldots\ldots}q_n$ represent the two prime number values to calculate the product $n = n_1, n_{2\ldots\ldots}n_n$. let g , $x = x_1, x_{2\ldots\ldots\ldots}, x_n$ and $r = r_1, r_2, \ldots\ldots, r_n$ are randomly choosen. Next we obtain invididual products of n by using n=p*q. the next possible step is in the creation of the cipher text by using the blowfish algorithm which can be calculated by the formula as shown below

$$c_i = g^{x_i}.r_i^{n_i} \bmod n_i^2 \quad (1)$$

The encrypted message is then just c_i. The holder of the private key does not need to know the value of r in order to decrypt c_i.

B. Decryption

Given an encrypted message, c, and knowing the values p_i, q_i and g, one can decrypt c_i. Note that Carmichael's function $\lambda(n_i) = \text{lcm}[(p_i - 1)(q_i - 1)]$ is easily computable given the values of p_i and q_i. Also note that g chosen for this public key, Carmichael's Theorem guarantees that $g^{\lambda(n_i)} = l_i \bmod n_i$. Carmichael's Theorem states that if two integers, g and n, are relatively prime, then $g^{\lambda(n_i)} = l_i \bmod n_i$. Since g is a unit modulo n_i^2, it is relatively prime to n_i^2, which means it's relatively prime to n, thus Carmichael's Theorem applies. For any decryption with the public key n_i, g ,regardless of the value of c, the calculation of $g^{\lambda(n_i)} \bmod n_i^2$ is necessary. This resulting value, an element of the participant will, by Carmichael's Theorem [19], be congruent to 1 mod n. Thus, subtracting one from this resulting value will give a number that is divisible by n (congruent to zero mod n). So, compute $g^{\lambda(n_i)} \bmod n_i^2$, subtract one from this value, then divide that number by n_i.

The above procedure helps in decrypting the message individually however the generalized form of the equations that follow the decryption is depicted below

$$L(u) = \frac{(u-1)}{n} \quad (2)$$

$$L\left(g^{\lambda(n)} \bmod n^2\right) = k \quad (3)$$

Notice that since $g^{\lambda(n)}$ is being calculated mod n^2, it can be viewed as a number greater than or equal to zero, but strictly less than n^2, so dividing this number by n results in a value, k, greater than or equal to zero, but strictly less than. Since n = p * q, so long as k is not congruent to a multiple of p or q mod n, then k has an inverse, This is the previously undefined property that must be satisfied by g, which was mentioned in Encryption.

Values of g such that $L\left[g^{\lambda(n)} \bmod n^2\right]$ is congruent to a multiple of p or q mod n which are the few exceptions of

ISSN: 2277-9655
Impact Factor: 1.852

semi-random g values with orders divisible by n that must be excluded. To be precise the equations above try to conceptualize about the complete decrypted data after the integration of t participants form the dealer. That is the adversary cannot manipulate the data until he breaks the data of the t participants thus making the system more stupendous and secured from malleable attacks. Thus satisfying the malleability parameter. Since the performance parameter is also an important one we tackle it by using the blowfish algorithm in ciphering the data. Since the blowfish is the most efficient algorithm in encrypting than any other algorithm.

C. Algorithm Of The Proposed System

1. Encryption

Input: A plain text message
Output: An encrypted message for multiple participants
Step 1: initialize the variables $p_1, p_{2,\ldots..}p_n$ and $q_1, q_2, \ldots q_n$such that they are large primes.
Step 2: generate the public key for each participant $n_1, n_2, \ldots . n_k$ such that n=p*q
Step 3: choose the semi random variable 'g'
Step 4: calculate the encrypted value for each participant by the formula

$$c_i = g^{x_i}.r_i^{n_i} \bmod n_i^2$$

2. Decryption

Input: An encrypted message divides among many participantsc_i.

Output: the plain text message among the participants.

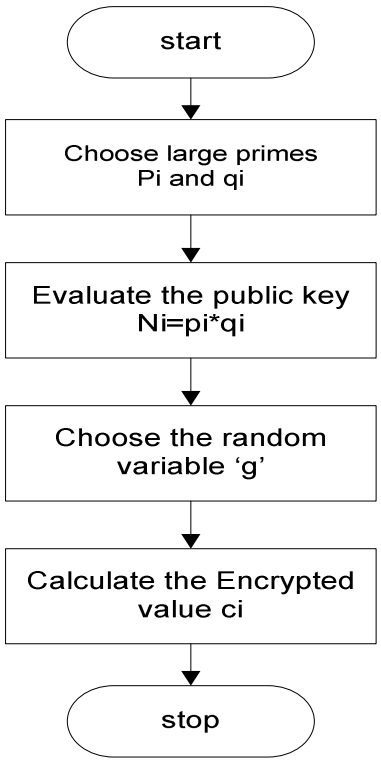
Step 1: Calculate the value of λ by the LCM of the p_i andq_i.

Step 2: evaluate the value of 'u' by applying the formula

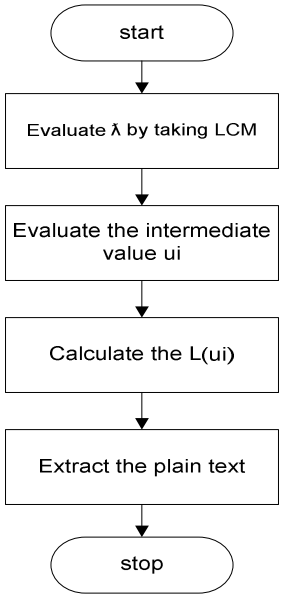
$$u_i = g^{\lambda(n_i)}.r_i^{n_i} \bmod n_i^2$$

Step 3: calculate the inverse of the $L(u_i)$ which will give you the plain text value

D. Flowchart Of The Proposed System



Flow Chart 1: Encryption of the Proposed System



Flow Chart 2: Decryption of the Proposed System

V. RESULTS AND DISCUSSIONS

The results and discussion implements the way of evaluating the expected parameters giving a way to define the probable results. Here, according to our proposed system we evaluate two parameters one is the malleability and other is the performance, which is unique parameter our paper takes into consideration. Since converting into the cipher text expectedly takes more time affecting the performance. Hence

ISSN: 2277-9655
Impact Factor: 1.852

we choose an alternative way of using blowfish algorithm for the encryption which yields the following results below in the figure 1. From the graph it is evident that using an additional algorithm during encryption enhances the performance of the proposed system. With the enhancement it is obvious we have decreased the consumption time 30seconds for 10Kb of data when compared to usual Paillier algorithm which can be easily seen in the figure 1.

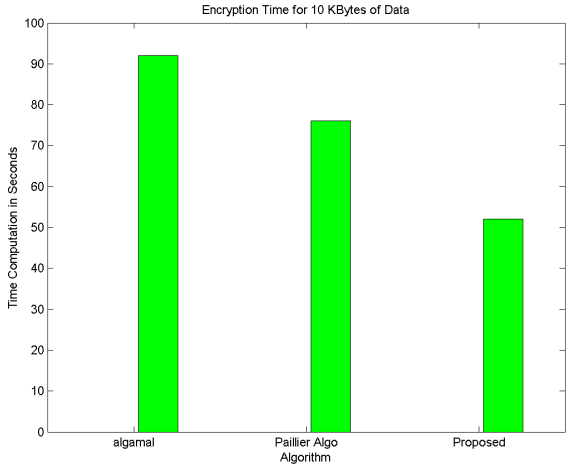


Figure 1: Performance Evaluation

The other parameter that is taken into consideration is malleability which is fundamental parameter for all the security algorithms. It defines the compactness of the algorithm that is how secure the algorithm is. Thus defining the randomness of the system. The graph is plotted between the number of trails and the randomness percentage. The randomness indicates how compact the algorithm is. In other words it indicates the possible percentage measurement to attacks. However it can be seen from the graph that our proposed system has more randomness which indicates the security boundary of our algorithm thus making it more feasible than other algorithms.

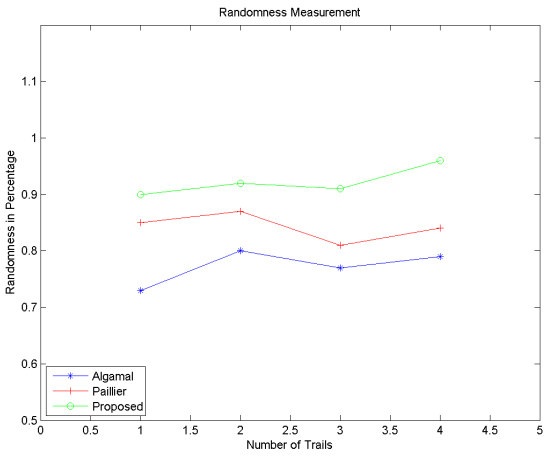


Figure 2: Malleability Evaluation

VI. Conclusion

Security on fully Homomorphic encryption is a traditional concept on security which enables us to provide the encrypted data without knowing the raw entries on which the calculation was carried out respecting the confidentiality of data. However, our work is based on the application of Homomorphic encryption which enables us to derive the security of the data and also its performance issue. As evaluated in the graphs of the malleability and performance it is evident that our algorithm provides a novel result which is found to be good making the security of the Paillier's algorithm much compact and flexible.

Acknowledgment

The authors are very grateful to the team of eMath Technology, India for the interesting discussions regarding this work

References

[1] Maha TEBAA, Said EL HAJJI, Abdellatif EL GHAZI," Homomorphic Encryption method applied to Cloud Computing" 978-1-4673-1053-6/12/$31.00 ©2012 IEEE.

[2] Michael O'Keeffe, "The Paillier Cryptosystem, A Look Into The Cryptosystem And Its Potential Application"

[3] Mads Johan Jurik, the Paillier Cryptosystem with Applications to Cryptological Protocols, Basic Research in Computer Science, ISSN 1396-7002 August 2003.

[4] "Dario Catalano",Paillier's Cryptosystem

[5] "Kevin henry"The Theory and Applications of Homomorphic Cryptography", Waterloo, Ontario, Canada, 2008

[6] Pratap Chnadra Mandal "Superiority of Blowfish Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 9, September 2012 ISSN: 2277 128X.

[7] Georg M. Penn, B.Eng,"Partially Homomorphic Encryption Schemes", Johannes Kepler University Linz, May 23, 2013.

[8] Dr. Michael Zwilling,"The El Gamal Cryptosystem", Mount Union College , March 12, 2008.

[9] "Julien Bringer, Herv´e Chabanne, Malika Izabach`ene, David Pointcheval2, Qiang Tang and S´ebastien Zimmer", "An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication, The 12th Australasian Conference on Information Security and Privacy".

[10] Oded Goldreich,"Two remarks concerning the Goldwasser-Micali-Rivest Signature Scheme", Department of computer science and applied Mathematics, Weizmann Institute of science, Israel.

[11] Kun Peng, Colin Boyd, Ed Dawson, Byoungcheon Lee, and Riza Aditya, "Multiplicative Homomorphic E-Voting", Information Security Research Centre.

[12] Maurice Chiodo, "An Introduction to Braid Theory "November 4, 2005

[13] Ronald L. Rivest, Leonard Adleman, and Michael L. Dertouzos. "On Data Banks and Privacy Homomorphisms", chapter On Data Banks and Privacy

ISSN: 2277-9655
Impact Factor: 1.852

Homomorphisms", pages 169- 180. Academic Press, 1978.

[14] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim."Evaluating 2-DNF formulas on cipher texts". In Theory of Cryptography Conference, TCC'2005, volume 3378 of Lecture Notes in Computer Science, pages 325-341. Springer, 2005.

[15] Alejandro Hevia and Marcos Kiwi., "Electronic jury voting protocols". 2000. http://eprint.iacr.org/2000/035/.

[16] Aggelos Kiayias and Moti Yung. "Self-tallying elections and perfect ballot secrecy. In Public Key Cryptography", 5th International Workshop—PKC 02, pages 141–158, 2002.

[17] Byoungcheon Lee and Kwangjo Kim. "Receipt-free electronic voting through collaboration of voter and honest verifier". In JW-ISC 2000, pages 101–108, 2000.

[18] Byoungcheon Lee and Kwangjo Kim. "Receipt-free electronic voting scheme with a tamper-resistant randomizer", In Information Security and Cryptology, ICISC 2002, pages 389–406, 2002.

[19] DeVries, Andreas. "Carmichael Function." Math IT. 2002. 14 Apr. 2008 <http://www.math-it.org/Mathematik/Zahlentheorie/Carmichael.html>.