Quest Journals
Journal of Software Engineering and Simulation
Volume 8 ~ Issue 7 (2022) pp: 54-61
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

An Enhanced Homomorphic Encryption for an Electronic Voting System using Hybrid Cryptographic System

Damaris Ferdinand
Daniel Matthias

ABSTRACT
The aim of this research is to enhance homomorphic encryption for an electronic voting system using hybrid cryptographic system. Two cryptographic algorithms such as Rivest Shamir Adleman (RSA) and Advanced Encryption Standard (AES) were combined in developing the hybrid system which was applied to an electronic voting system. Due to the security threats over the internet, there is a need to protect confidential information against malicious attacks during the process of data transmission and cloud storage. However, in order to address these security challenges, different cryptographic schemes have been proposed over the years which has in a great deal alleviated these security threats but, have also proven to be unreliable with an untrusted communicating party. The idea of homomorphic encryption has therefore been adopted to proffer immense solution the issues of data confidentiality, privacy and authenticity during the process of transmission and storage. To achieve this, a hybrid system was developed to enhance homomorphic encryption for an electronic voting system, where the combination of two standard cryptographic schemes (RSA and AES) was used to encrypt and compute on data homomorphically thus, ensuring data confidentiality, privacy and authenticity in the cloud. In the electronic voting system, votes cast were strongly encrypted, tallied and computed upon using homomorphic encryption and the result was stored in the cloud, in which the encrypted votes and the computed result were only decrypted by the authorized users (electoral officers) after retransmission, where the deciphered result tallied with the product of the total votes cast for each of the candidates as displayed on the interface, which represents the plaintext input as also shown in the hybrid system therefore, addressing the major problems of votes confidentiality, privacy and result authenticity in an electronic voting system. A Scripting Language, PHP and an Object-Oriented Programming Language Python, were used in the development of this system to make it more robust and to effectively optimize security.
KEYWORD: *Cryptographic,Algorithms, Homomorphic, Encryption, RSA, AES*

***R**eceived 02 July, 2022; **R**evised 13 July, 2022; **A**ccepted 15 July, 2022 © The author(s) 2022.*
P**ublished with open access at **www.questjournals.org

I. Background to the Study

The transmission of confidential data over the network presently requires high level of security for wide range of applications. This is because, the confidential information is highly vulnerable to malicious attacks. As a result of this, the need to protect confidential information has posed some security threats over the internet. Hence, cryptographic algorithms are considered to provide best solutions to these security threats, when various techniques are applied to ensure information security. Conventionally, Cryptography refers to the procedure of converting a plain or raw message into a ciphertext in order to protect information from malicious attacks where the encrypted message will have to be decrypted back to plaintext by the authorized user. This can be classified as Symmetric and Asymmetric Key Cryptography. Symmetric Key Cryptography is an encryption scheme or approach, where the sender and receiver of the message share and use a single key for encrypting and decrypting messages, this key is called "secret key". It is a cryptosystem characterized by two algorithms (Gupta & Iti, 2013). During the communication process, the encryption algorithm Enc (m, s_k), is used by the sender where "m" denotes the message to be encrypted, also known as the plaintext and "s_k" denotes the secret key used for encrypting the plaintext into a ciphertext denoted as 'c' which corresponds to the plaintext "m", when decrypted. The encrypted message is transmitted to the receiver who decrypts the ciphertext using the decryption algorithm Dec (c, s_k). A popular example is Advanced Encryption Standard (AES).

Statement of the Problem

With the increasing awareness and concerns regarding data communication and information security with privacy, adequate use of cryptographic algorithms in data communication and processing is highly required. A major problem in data communication process has been the preservation of confidentiality between sender and receiver, which demands efficient data protection against intruders. By convention, encrypting information for transmission and processing has alleviated the dangers of security threats but, over time has as well proven undependable. Homomorphic Encryption have been confirmed to proffer tremendous solutions to the problem of lack of data confidentiality, privacy and authenticity by allowing computations to be performed on encrypted data thus, yielding same results when decrypted as though computing on raw data.

Aim and Objectives of the Study

The aim of this research is to enhance Homomorphic Encryption for an Electronic Voting System using a Hybrid Cryptographic system, which combines Rivest Shamir Adleman (RSA) and Advanced Encryption Standard (AES) cryptographic schemes.

The following objectives are required to achieve the above aim.

1.To generate security parameters for RSA and AES using RSA modulus, Euler totient function and Extended Euclidean algorithm.

2.To encrypt the data to be outsourced for computation using RSA public key and AES symmetric key, then perform homomorphic computation on the resulting ciphertexts which is the encrypted votes.

3.To decrypt the computed ciphertext following the reverse order of the encryption steps thus, generating a result tallying with the total votes cast for each of the candidates.

4.To evaluate the system to ensure it meets the specified requirements.

II. LITERATURE REVIEW

Bogos *et. al.* (2016) however, challenged this claim by publishing the details of the case of an attack on a homomorphic encryption that was claimed to be semantically secure, which was later discovered not to be semantically secure after being a victim of several attacks.

Gentry & Halevi (2011) explained that the construction of the Fully Homomorphic Encryption begins with the construction of a Somewhat Homomorphic Encryption scheme that supports the evaluation of degree polynomials on the ciphertext. Afterwards, the decryption process has to be modified in such a way that it can be expressed as a low polynomial. Once the degree of polynomials that can be evaluated by the scheme exceeds twice the degree of the decryption polynomial then, the scheme is bootstrappable and can be transformed into a Fully Homomorphic Encryption. Modifying the decryption process is achieved by adding to the public-key an additional hint about the secret-key, this approach is knowns as: Sparse Subset Sum Problem (SSSP). Here the public-key is enhanced with a large set of vectors, and a sparse subset of the public keys adds up to the secret key. A ciphertext of the underlying scheme is post-processed using the additional hint, which allows the ciphertext to be decrypted with a low degree polynomial.

Lopez *et. al.* (2012) introduced a Somewhat Homomorphic Encryption that handles inputs from more than one public-key.

Joppe *et. al.* (2013), proposed another Fully Homomorphic Encryption scheme, where the security of the system on the standard assumptions of lattice. In the same context, three researchers led by Gentry described a Learning with Error (LWE) -based FHE scheme, which is highly customized to evaluate the Advanced Encryption Standard (AES) circuit efficiently. The results reported in their paper showed their proposal was capable of successfully evaluating a single AES encryption operation within five minutes.

III. MATERIAS AND METHODS

Computational Research Methodology

The research methodology adopted for this research work is the Computational Methodology. This methodology allows the utilization of new advances in the field of computing which include, algorithms, models, simulations, and systems which can be used to numerically study the behavior of systems. This methodology allows the generation of random numbers used as keys to perform mathematical operations, specifically in the study of cryptography. In this research work, we intend to use Hybrid Cryptographic Algorithm (HCA) to enhance Homomorphic Encryption (HE), as it permits the use of mathematical operations to encapsulate data thus, allowing the performance of mathematical computations on encrypted data without prior decryption to ensure data privacy and security during the process of data storage and communication.

Analysis of the New System

The new system is one which will be able to perform mathematical computations on encrypted data without prior decryption thus, yielding a result corresponding to the result of a plaintext computation, using the generated keys. The public key is selected based on the principles governing the selection, which will be used for encryption, and will be released to the public (cloud server) to perform required computation and further encryption. The Hybrid Cryptographic Algorithm (HCA) is adopted which is used in designing the proposed system. HCA is a combination of Rivest Shamir Adleman (RSA) and Advanced Encryption Standard (AES) cryptographic algorithms. This hybrid system was adopted in order to achieve not only a more efficient security and privacy of data during storage and communication process involving an untrusted party but, also has the advantage of algorithm performance.

HCA combines the convenience of a public-key cryptosystem with the efficiency of a symmetric-key cryptosystem. The public-key based system is considered convenient because it does not require the sender and receiver to share a common secret key for the purpose of ensuring proper security. This technique deals with complicated mathematical computations and thus, requires a more efficient scheme to be combined with it. Since encrypting long messages in most applications using public-key cryptosystem incurs high cost of computation however, this is addressed by proposing a hybrid system, which is a combination of both public-key cryptosystem and symmetric-key cryptosystem. A hybrid cryptosystem can be categorized in two distinct cryptosystems such as:

i Key Encapsulation Scheme: this is used in public-key cryptosystem.

ii Data Encapsulation Scheme: this applies to a symmetric-key cryptosystem.

The hybrid cryptosystem is considered a public-key cryptosystem, having public and private keys corresponding to that of the key encapsulation scheme.

In hybrid system, two large primes are selected, used to compute for public key parameters. This is achieved using the Euler's totient function denoted as $\varphi(n)$ which is a multiplicative function for multiplying two relatively prime numbers. This function gives the order of the multiplicative group of integers modulo n, which is used in defining the RSA encryption system. The public-key which is used for encryption key is generated by selecting a prime number which is not a factor of $\varphi(n)$. The private key which is used for the decryption key is generated using Extended Euclidean algorithm. This algorithm is used for computing the Greatest Common Divisor (GCD) of two integers where the largest number divides them both without a remainder. This is also known as the modular inversion. After computing for RSA public-key and private-key, the symmetric-key of AES is generated from the encrypted data to be outsourced for computation, where same process will be repeated during decryption.

AES algorithm is based on Substitution box commonly known as S-box, which is used in the Rijndael cipher. Substitution box is one of the most essential aspect of AES. During subByte transformation, the 8bit input, represented in hexadecimal form is substituted by 8bit output (or byte) using the S-box. S-box is constructed by composing two transformations such as the multiplicative inverse in Galois Field GF (2^8), followed by an affine transformation.

The Substitution box table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 1: Substitution Box Table

Inverse Substitution box table

The Inverse Substitution box table is depicted in figure 2

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure 2: Inverse S-Box Table

Use-case Diagram

The system will have four main actors; the admin, the voters, server 1 and server 2. The admin registers voters, add candidates and electoral position to the database. The voters cast votes for their choice of candidate. The first server encrypts the captured votes and transmits it as ciphertexts to the second server which tallies, computes and stores the total votes cast by voters for each candidate, afterwards the encrypted and computed votes are being decrypted and results verified and announced by electoral officials.

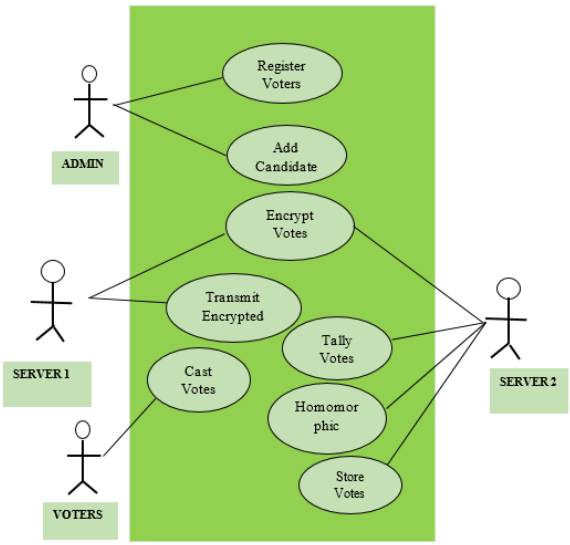

Figure 3. Use case Diagram

Table 1. Use-case Description

Actor	Use case	Description
Admin	Register voters	The admin registers voters in the database.
	Add candidates	The admin adds candidates to the database.
	Encrypt votes	Votes are encrypted and tallied via the Application Program Interface before it is sent to the cloud for further encryption and computation
Voters	Cast votes	Voters vote for their choice of candidates and the votes are encrypted.
Server 1	Encrypt, Transmit the encrypted votes	The first server performs encryption on the captured votes and transmits the encrypted votes to the second server.
Server 2	Tally, Homomorphic Encryption and Storage of Encrypted votes	The second server receives the encrypted votes from the first server and tallies the votes, performs homomorphic encryption on the votes then stores the results of the computation.

System Architecture

This section gives the general understanding on how the proposed system is designed. The diagram below shows the pictorial layout of the proposed system.

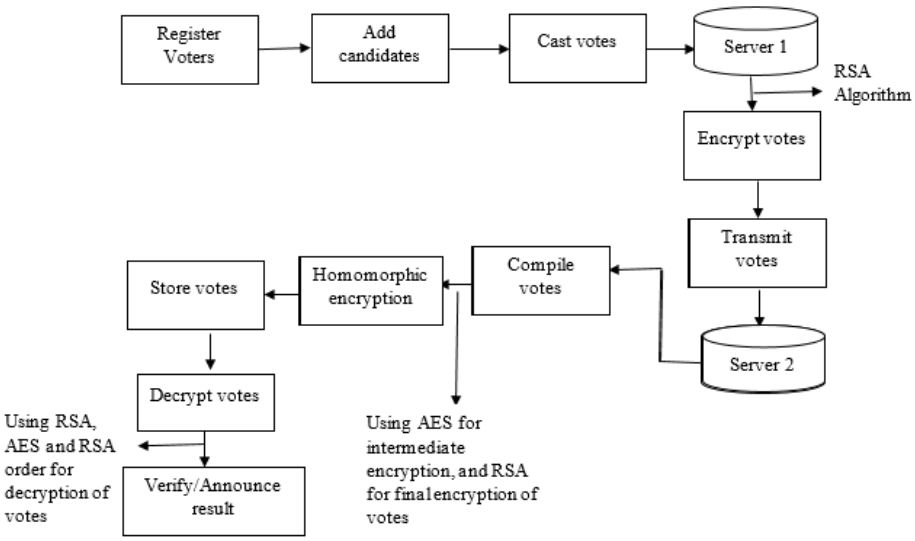

Figure 4. System Architecture

Sequence Diagram

sequence diagram is an interaction diagram that is used to represent the interaction between objects and how messages are exchanged between the objects required to perform the functionality of the system. The interaction between the objects in the proposed system is carried out in a sequential order as depicted in Figure5.

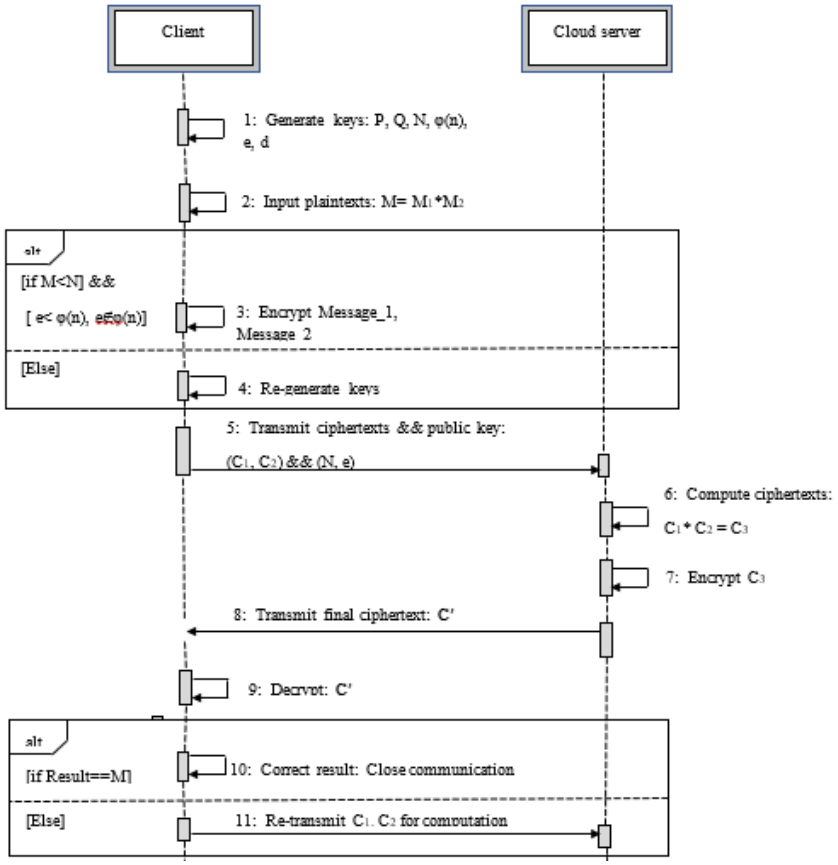

Figure 5: Sequence Diagram

IV. IMPLEMENTATION

System Requirement

Hardware Requirement	Software Requirement
Intel Core i3 processor	Windows 7 and above.
Hard disk file system: NTFS	XAMPP
Operating System of 64-bit	Antivirus of your choice.
Processor speed of 2.63 GHz and above	Python programming language
Hard disk Size of 136 GB and above	Python Anaconda IDE (Spyder)
4GB RAM capacity and above	Web browser

Programming Language

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected

How to Install

1.Open Anaconda and click on Spyder IDE
2.Click on the load folder icon
3.Load the file UI.py
4.Open Xampp application and click on start for both Mysql and Appache to get the local host
5.Click on Admin to go to the login page of the e-voting system
6.After all votes have been cast, go over to the anaconda IDE and click on run

V.RESULTS

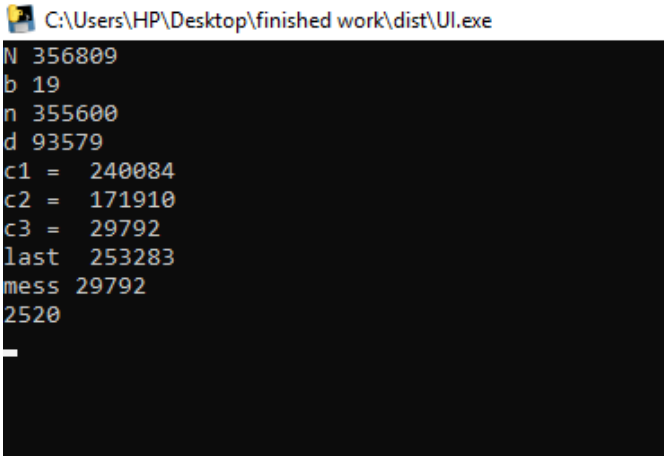

Figure 6: Command

Graphical User Interface

The Figure 7 depicts the interface created for the user where the user can view the encrypted data, after it has gone through homomorphic encryption, which also decrypts ciphertexts when required.

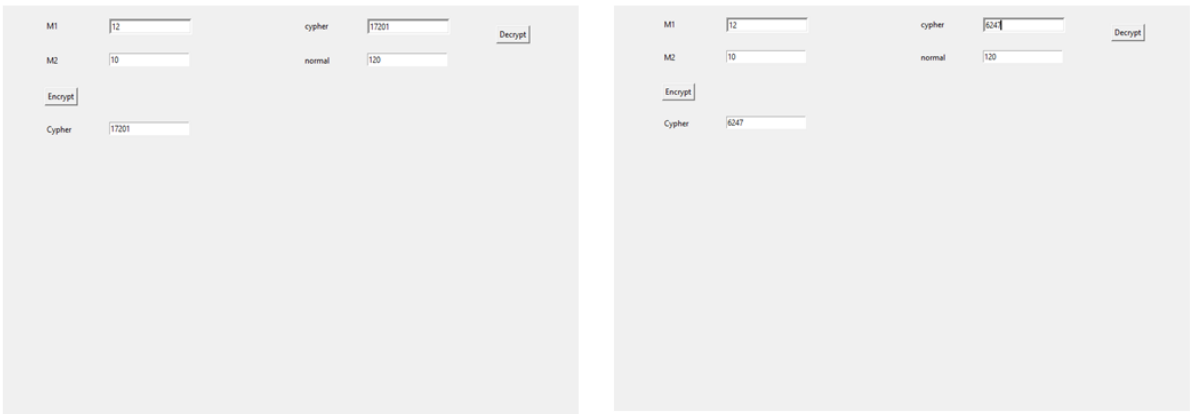

Figure 7: Graph ical User Interfaces

E-voting Admin Home Page

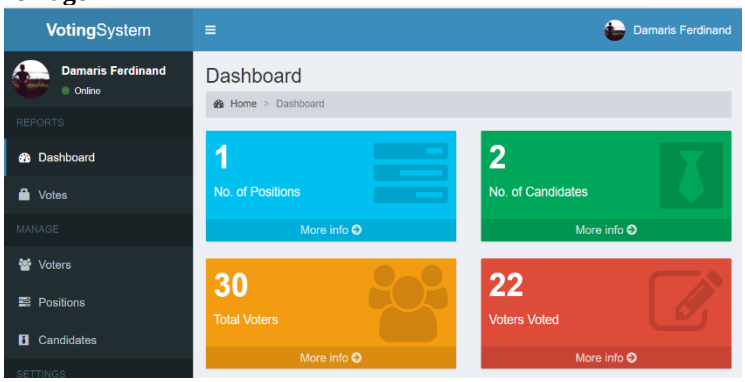

Figure 8: Admin Page

Figure 9: E-voting page

VII. CONCLUSION

Homomorphic approach of encryption allows computations to be performed on cyphertexts. Since sharing data with an untrusted party during the process of communication makes the data vulnerable to attacks. Over the years, several researchers have proposed various schemes and methods of achieving homomorphic encryption. However, these schemes have shown their individual potencies and limitations in developing a well-secured system which is as a result of total reliance on the encryption strength of just the conventional cryptographic schemes although, these schemes vary and outweigh one another in their security strengths. In view of some of these schemes, the conventional cryptography like RSA and others have shown their security strengths homomorphically but on the other hand have proven to be somewhat unreliable due to their vulnerability to attacks such as brute force and lots more. However, this work is developed in order to improve upon the homomorphic strength of encryption by combining both the public key and the symmetric key conveniences of RSA and AES cryptographic schemes. This work is developed in such a way that different layers of ciphertexts are generated by using both schemes in turns as shown in the algorithm in figure 3.8, which is implemented in an E-voting system. However, with the tabular results obtained in comparing the existing and the proposed system in terms of efficiency, satisfaction, validation, security, authenticity, voters' anonymity, and accuracy. Hence, considering these factors, this scheme has shown to provide a more powerful encryption in computing homomorphically on data as shown in the 'e-voting system' therefore, making it more reliable than the already existing single cryptographic schemes.

VII. RECOMMENDATION

Considering the knowledge acquired from this research work which has contributed immensely in cryptography research areas, e-voting system, and also to individual information security. The researcher is recommending that the proposed system can be deployed in the field of cryptography, and can also be used in the following areas:

1.Carrying out an online voting exercise with guaranteed high level of security.

2.Transmitting confidential information or data, even with the involvement of an untrusted communicating party.

3.Researchers who are interested in this field can also study to improve upon homomorphic encryption by combining more than two cryptographic schemes.

4.Research should also be carried out to implement a fully homomorphic encryption, whichexhibits both theproperties of multiplication and addition, since the proposed system is limited to only multiplicative properties.

REFERENCES

[1]. Alkady, Y., Fifi, F. & Rawya, R. (2015). Fully Homomorphic Encryption with AES in Cloud Computing Security. Electrical Engineering Department, Port Said University, Port Said, Egypt, 371 – 382.

[2]. Bogos, S., Gaspoz, J. &Vaudenay, S. (2016). Cryptanalysis of a Homomorphic Scheme. Retrieved from https: //infoscience.epfl.ch/record/220692.

[3]. Chandravath, D. & Lakshmi, P. V. (2019). RSA Homomorphic Encryption Technique using multiple keys.International Journal of Advanced Trends in Computer Science and Engineering, 8(4), 1476 – 1480.

[4]. Craig, G. (2009). A Fully Homomorphic Encryption Scheme. Department of Computer Science and the Committee of Graduate Studies. Stanford University. 51-56.

[5]. Craig, G. (2009). Fully Homomorphic Encryption using ideal lattices. Michael Mitzenmacher, editor. Proceedings of 41st annual ACM Symposium on Theory of Computing, 169 – 178.

[6]. Dan, B., Gil, S. & Brent, W. (2011). Targeted Malleability. Homomorphic Encryption for Restricted Computations. Stanford University, 350 – 366.

[7]. Emelie, W. (2018). Fully Homomorphic Encryption. A Case Study of Masters Thesis in Computer Systems and Networks. Department of Computer Science and Engineering, Chalmers University of Technology, University of Gothenburg,1 - 59.

[8]. [8] Frederik, A., Colin, B., Christopher, C., Kristian, G., Angela, J., Christian, A. R. & Martin, S. (2015). A Guide to Fully Homomorphic EncryptionUniversity of Mannheim {armknecht, jaeschke, reuter}@uni-mannheim.deDepartment of Telematics, NTNU {colin.boyd, ccarr}@item.ntnu.noDepartment of Mathematical Sciences, NTNU {kristian.gjosteen, martin.strand}@math.ntnu.no.

[9]. Gentry, C. & Halevi, S. (2011). Implementing Gentry's Fully Homomorphic Encryption Scheme. Annual International Conference on the Theory and Applicationsof Cryptographic Techniques, 129 – 148.

[10]. Gupta, C. P.&Iti, S. (2013) Fully Homomorphic Encryption Scheme with Symmetric Keys.A Dissertation submitted in partial fulfillment for the award of the Degree of Master of Technology. Department of Computer Science & Engineering (with specialization in Computer Engineering). Rajasthan Technical University, Kota, 15 – 25.

[11]. Hans, D. & Helmut, Knebl. (2007). Introduction to Cryptography. Principles and Applications. Springer Science & Business Media, 1 – 10.

[12]. Joppe, W., Bos, N. X. P., Semiconductors, K. L. M., Jake, L. & Michael, N. (2013). Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme. Conference: IMA International Conference on Cryptography and CodingMicrosoft, 8(4).

[13]. Junfeng, F. & Fredrick, V. (2012). Classes of Homomorphic Encryption. Cryptology ePrint Archive: Report, 144.

[14]. Kai-Min, C., Yael, T. K. & Salil, P. V. (2015). Improved Delegation of Computation using Fully Homomorphic Encryption. Rabin, 483 – 501.

[15]. Liangliang, X., Osbert, B. & Ling, I. Y. (2012). An Efficient Homomorphic Encryption Protocol for Multi-User Systems.Computer Science, Mathematics IACR Cryptol. ePrint Arch.

[16]. Lopez-Alt, A., Tromer, E., Vaikuntanathan, V. (2012) On-the-fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. STOC, 1219 – 1234.

[17]. Monique, O., Claude, T. & Pushkar, D. (2013). Topic: Homomorphic Encryption. Institution: Procedia Computer Science, Bowie State University, Department of Computer Science, 502 – 509.

[18]. Nigel, P. S. & Frederick, V. (2010). Fully Homomorphic Encryption. Masayuki Abe, editor, Advances in Cryptology. Lecture Notes in Computer Science, 377 – 394.

[19]. Nigel, S. (2015). Information Security and Cryptography. Cryptography Made Easy. University of Bristol, Bristol UK, 1 - 478.

[20]. Nisarg, M. (2019). Introduction to Homomorphic Encryption (HE). Principal Scientist | Galois, Inc,https://www.cs.virginia.edu/dwu4/papers/XRDSFHE.pdf

[21]. Pascal, P. (1999). Public-Key Cryptosystems Based of Composite Degree Residuosity Classes. International Conference on the Theory and Application of Cryptographic Techniques. 1592, 223 – 238.

[22]. Rivest, R. L., Adleman, L. &Dertouzos, M. L. (1978). On Data Banks and Privacy Homomorphism. Foundations of Secure Computation, Academia Press, 169 – 179.

[23]. Ron, R. & Jaydip, S. (2013). Homomorphic Encryption. Theory and Applications, Theory and Practice of Cryptography and Network Security Protocols and Technologies, Sen, J. (Ed.), INTECH Publishers, Croatia, 1- 31.

[24]. Rosario, G. & Daniel, W. (2013). Fully Homomorphic Message Authenticators. International Conference on the Theory and Application of Cryptology and Information Security, 301 – 320.

[25]. Rudragoud, P &Goudar, R. H. (2020). Fully Homomorphic Encryption (FHE): A Framework for Enhancing Cloud Storage Security with AES. International Journal of Scientific & Technology Research, 9.

[26]. Smart, N., & Vercauteren, F. (2009). Fully homomorphic encryption scheme having a smaller key and ciphertext sizes. Department of Computer Science, Cryptography and Information Security. Lecture Notes in Computer Science, 7 – 25.

[27]. Stanley, N. B. &Sankappanavar, H. P. (2012). A Course in Universal Algebra. Institution: State University of New York at New Paltz | SUNY New Paltz, Department Mathematics, 288.

[28]. Tilborg, H. C., Van, A. &Jajodia, S. (2011). Encyclopedia of Cryptography and Security. Springer Science & Business Media.

[29]. Zhou, J. & Yung, M. (2010). Applied Cryptography and Network Security: 8th International Conference. Springer Science and Business Media, volume 6123.