Advanced E-Voting System Using Paillier Homomorphic Encryption Algorithm

[1]Shifa Manaruliesya Anggriane, [3]Surya Michrandi Nasution, [3]Fairuz Azmi
[123]Electrical Engineering Faculty
Telkom University
Bandung, Indonesia
[1]shifamaa@gmail.com, [2]michrandi@telkomuniversity.ac.id, [3]worldliner@telkomuniversity.ac.id

***Abstract*—The increase of technology usage, especially in online storage data with cloud system, makes data security become one of the most important requirements for the user. One of the ways to protect the data is with encryption process, it as a process to change a data form into a new one that can only be read by the chosen recipient. This technology has been carried out in the election process with e-voting system. The high risk of security problem makes data encryption applied in the e-voting system. Thus, the ballot won't be interrupted by the insider or the outsider.**

The purpose of this research is to prove the effectiveness of the Paillier algorithm and its homomorphic property that implemented in an e-voting system. With homomorphic property, the system can calculate the sum of votes in ciphertext form without revealing the choice of the voters. The resulting ciphertext values will be different each other even though the same plaintext is encrypted, with a size of 4 times larger than the plaintext size. The success ratio for the system is 100% with the maximum number of messages that can be processed is 3.287.937.778 messages.

Keywords—paillier algorithm; cryptography; homomorphic; encryption; e-voting;

I. Introduction

Technology has been used in various aspects, both offline and online. The increase of its usage, especially in online storing data makes security become one of the most important requirements. One of the ways to protect data is with encryption. Encryption is a process to convert message or information into a form that can be read only by the recipient. The intended recipient must decrypt the encrypted data before it can be read. The recipient should have a key to decrypt it.

There are two cryptosystem categories: symmetric and asymmetric. The symmetric cryptosystem use the same key to perform the process of encrypt and decrypt a message. In an asymmetric cryptosystem or otherwise called as public key cryptosystem, the public key used for encrypt messages can accessed by anyone. The message can only be read by the specific recipient who has the paired key called private key. IDEA and DES are the symmetric cryptosystem example and RSA and Paillier are various asymmetric cryptosystem [1].

As pointed by Rivest, Adleman and Dertouzos [2] in encryption system an information system can just store and recover encrypted data for users. Decryption is required for further operation. The data is not secure any longer after it is decrypted. The new thought that permits direct computation on encrypted data without decryption was proposed, called "privacy homomorphism". With homomorphic encryption, the operations on the encrypted message such as additions and multiplications can be performed by using the public key algorithm.

The election process in Indonesia nowadays still uses paper as the media to elect and manual calculation to obtain the final result. This way has many obstacles such as slow counting process, and it takes more time and money in the manufacture and distribution process of the ballot. By utilizing the electronic voting system (e-voting system), the voting process can save cost, faster, and more accurate in the calculation process. It also more practical and safer [3].

However, it is still possible for the outsider or the administrator itself to manipulate the voting result. The high risk of security problem makes data encryption applied in the e-voting system. Thus, the ballot won't be interrupted by the insider or the outsider.

In this research, the focus is on the effectiveness of the Paillier algorithm to perform encryption and decryption process when implemented in the e-voting system. With its homomorphic property, the system can calculate the sum of votes without revealing which vote is voting for which candidate to the system.

II. Theory

A. Cryptography

In general, cryptology is the practice and study of mathematical techniques for keeping messages secure. In order to protect the information from disclosure to unauthorized parties, cryptosystem can provide one or more of the following four services [4]: Confidentiality, Authentication, Integrity, and Non-Repudiation.

Cryptosystem can be classified into two categories as described in the first section; symmetric key cryptosystem or secret key cryptosystem and asymmetric key cryptosystem or public key cryptosystem. In a symmetric key cryptosystem, both sender and receiver used the same key to perform encryption and decryption on a message. Popular symmetric cryptosystems are AES (Rijndael), DES, and IDEA. In an asymmetric key cryptosystem, there are two different keys paired, private or secret key and public key. The encryption

978-1-5090-1648-8/16/$31.00 ©2016 IEEE

process needs public key, while the other key used for decryption process. Both keys are produced as a relevant to each other [5].

B. Paillier Cryptosystem

Paillier cryptosystem is a probabilistic encryption and has an additive homomorphic property, invented in 1999 by Pascal Paillier [6]. Paillier cryptosystem has public key n and g, which is the RSA modulus. This cryptosystem encrypts a 'm' message with $c = g^m r^n \bmod n^2$, where r is a random integer. To obtain the n value, prime numbers p and q are needed. The prime numbers have to be different each other. After determined, compute the Carmichael's function with $\lambda = lcm\ (p-1),(q-1)$.

The applications of paillier cryptosystem are electronic voting and electronic cash [3]. In the implementation of the e-voting, the algorithm utilizes its additive homomorphic property.

C. Homomorphic Properties

Paillier cryptosytem has a homomorphic property. Without decrypting the encrypted message, the user can calculate the data of the message. The well-known implementation of homomorphic paillier cryptosystem is e-voting application. Additive homomorphic encryption applied here instead of ElGamal encryption, which is also an additive homomorphic encryption. Paillier cryptosystem is generally preferred due to the huge number of the voters in most e-voting system [7].

Assumed there are two messages (m_1 and m_2) encrypted to $E(m_1)$ and $E(m_2)$. To calculate the data of the messages the system will calculate it with $T = \prod_{i=1}^{N_p} c_i \bmod n^2$.. To get the final result of the calculated data, the system just needs to decrypt T [6].

D. E-Voting

Election allows people to choose their representatives for the future government. But the election sometimes littered with manipulation. The development of technology creates a new system of voting, electronic voting [6]. The e-voting system permits individuals to vote anywhere through computer which is connected to the internet.

The main problem in voting system is data manipulation. The electronic voting system also needs the security of the software. There has been cryptographic study on the e-voting system to secure ballots and the system from interferences.

III. System Design and Implementation

A. System Design

In this system, the user chose the candidate through computer which is connected to the internet. The ballot will be automatically encrypted and stored to the server in ciphertext form. To obtain the voting result, the administrator has to do the decryption process. This figure 1 below shows the system design for the e-voting system proposed in this research.

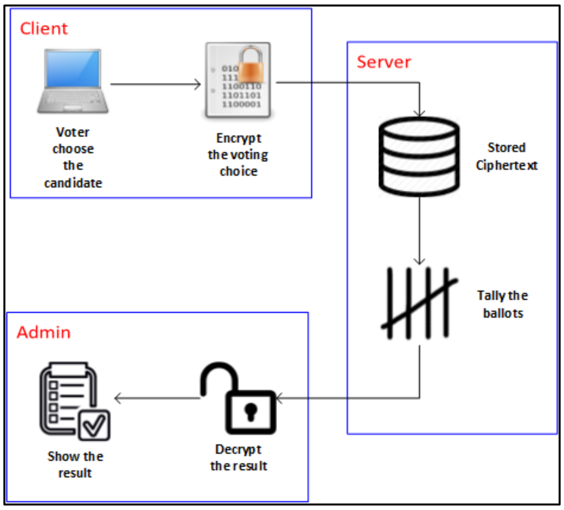

Figure 1. System Design

B. Key Generation

Paillier cryptosystem has two keys: the public key for encryption process and private key for decryption process as described in section 2. Key generation steps are as follows [7]:

1) Choose randomly different p and q as two large prime numbers with:

$$gcd(pq,(p-1)(q-1)) = 1 \quad [7]$$

2) Compute RSA modulus $n = pq$ and calculate Carmichael's function that can be computed with:

$$\lambda = \frac{(p-1)(q-1)}{\gcd(p-1,q-1)} \quad [7]$$

3) Select random g as generator where $g \in \mathbb{Z}^*_{n^2}$ with:

$$\gcd\left(\frac{g^\lambda mod\ n^2 - 1}{n}, n\right) = 1 \quad [7]$$

4) Find modular multiplicative inverse with:

$$\mu = (L(g^\lambda\ mod\ n^2))^{-1} mod\ n \qquad [7]$$

The function L is defined as $L(u) = \frac{u-1}{n}$

After following those four steps above, we could get the public key for encryption (n, g) and the private key for decryption (λ, μ).

C. Encryption Process

1. The message to be encrypted is m, where $m \in \mathbb{Z}_n$.
2. Choose random r where $r \in \mathbb{Z}^*_n$
3. To obtain the ciphertext do the computation: 7

$$c = g^m r^n \bmod n^2 \qquad [7]$$

D. Decryption Process

1. The ciphertext to be decrypted is c, where $c \in \mathbb{Z}^*_{n^2}$.
2. To obtain the plaintext message do the computation:

$$m = L(c^{\lambda} mod\ n^2).\mu \bmod n \quad [7]$$

E. Homomorphic Process

In the homomorphic process, the Paillier cryptosystem supports the additive homomorphic property. With publickey (n, g) and private key (λ, μ), The ciphertext will decrypt to plaintext with:

$$D(E(m_1, r_1).E(m_2, r_2)\ mod\ n^2) = m_1 + m_2 mod\ n \quad [7]$$

The ciphertext with a plaintext raising g will decrypt with:

$$D(E(m_1, r_1).\ g^{m_2}\ mod\ n^2) = m_1 + m_2 mod\ n \quad [7]$$

F. Implementation

The e-voting application is designed to be the simulation of e-voting system in general. This system has a graphical user interface (GUI). Whole system design mentioned above will be implemented in this application using Java programming language and uses MySQL for its database.

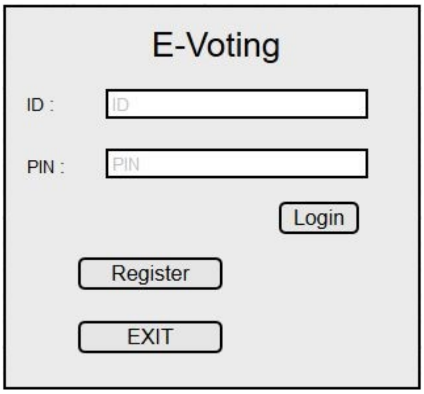

Figure 2. The application's main menu

Registered users must be validated to access the voting menu. Users who have been validated need to login first and they choose one of the candidates in the list to perform an election. After the user selects a candidate, the vote data will be encrypted by the application.

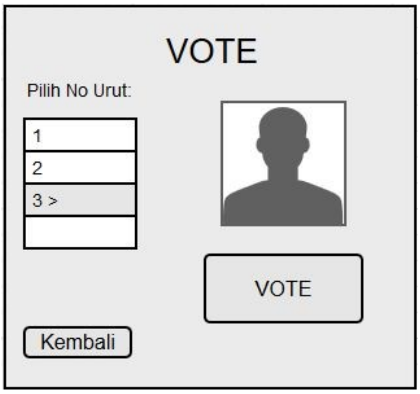

Figure 3. User voting interface

When the data successfully encrypted, the encrypted data or ciphertext will be stored in databases. Each voter will have ciphertext that is different each other, although they chose the same candidate.

The tally program will calculate each ciphertext. As explained before, Paillier algorithm can obtain $E(m_1 + m_2)$ with homomorphic property by multiplying m_1 and m_2 without decrypt it before.

To obtain the final result after the election ends, the administrator can decrypt the calculated results using the private key. The results of calculated ciphertexts will be same with the original calculated messages after being decrypted. The administrator will not know the choice of the voters. It is the main purpose of homomorphic tallying.

IV. Results and Analysis

There are several tests to prove the effectiveness of Paillier algorithm and the homomorphic property. The tests are ciphertext uniqueness test, decryption test, homomorphic test, message expansion factor test, runtime test, and the implementation to the application itself. The tests will be used 16 bit length of p and q, where $p = 50543$ and $q = 65053$.

A. Ciphertext Uniqueness Test

This test shows the chiphertext uniqueness produced by the algorithm.

Table 1. Ciphertext Uniqueness Test Result

No.	*Plaintext* (m)	*Random* r	*Ciphertext* (C)
1	7	49819	5091673028311841742
2	7	40942	5014329154833019458
3	7	40942	5014329154833019458
4	30	16765	939972971689839058
5	30	27529	8423042122639404412

Based on the test above, random value r will determine the uniqueness of the ciphertext. Different r produces different ciphertext aven though same plaintext is encrypted. But when same plaintext encrypted with same r value, the ciphertext will be same too.

B. Decryption Test

In this test, the ciphertext mentioned in table 1 will be decrypted with same p and q value. The decryption process use $m = L(c^{\lambda} mod\ n^2).\mu \bmod n$ function. With multiplying the prime numbers, can be obtained that $n = 3287973779$.

Table 2.Decryption test result with same plaintext

No.	*Ciphertext (C)*	*Decryption result*	*Conclusion*
1	5091673028311841742	7	True
2	5014329154833019458	7	True
3	5014329154833019458	7	True
4	939972971689839058	30	True
5	8423042122639404412	30	True

Table above shows the encryption process before was successfully performed.

Table 3.Decryption test tesult with different plaintext

No.	***Plaintext (m)***	***Ciphertext (C)***	***Decryption result***	***Conclusion***
1	900	2782712007130197049	900	True
2	1799238234	4431658342509263100	1799238234	True
3	3287973779	1391923990969058026	0	False
4	3287973787	10537662793546147947	12	False

From the table above can be concluded that when the plaintext m value is less than n, the decryption result will be true. But when greater than n value or $m \geq n$, the result will be the difference between both values.

C. Homomorphic Tests

In this test, homomorphic calculation is performed. The test using this homomorphic function $D(E(m_1, r_1).E(m_2, r_2) \; mod \; n^2) = m_1 + m_2 mod \; n.$

Table 4. Homomorphic test result

No	**$m_1 + m_2$**	**Ciphertext m_1 $m_2$mod n^2**	**Decryption Result**	**Conclusion**
1	20 + 40	8207684350209635468	60	True
2	25600000 + 65340000	7364355918022896602	90940000	True
3	3287973779 + 121	2478900265231643136	121	False

Table above shows that the homomorphic tallying with message value lesser than n, will produce the correct decryption result. However, the value of the decryption does not correspond to the original message when the homomorphic tallying of the message value greater than or equal to the value of n.

D. Message Expansion Factor Tests

In the following test performed encryption process to a message with different prime number size. The purpose is to determine the influence of large prime size to the resulting ciphertext size.

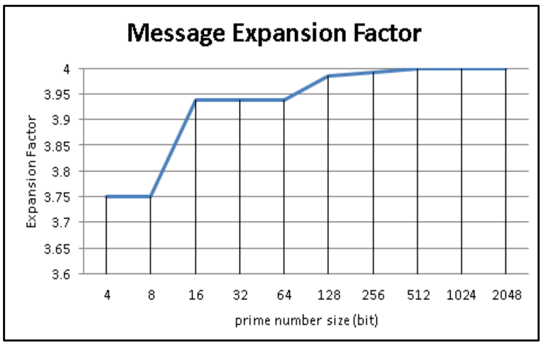

Figure 4. Comparison of message expansion factor

The resulting expansion factor value is increased and has a value range of 3.75 to 4, which means the size of the resulting ciphertext has a size of four times of the primes used.

E. Runtime Testing

Like the previous test, this test performed with different value and size of p and q. Plaintext used is a 4 bit length, which is 15. The accuracy of time tested in a nanosecond where 1 nanoseconds equal to 0.000000001 seconds.

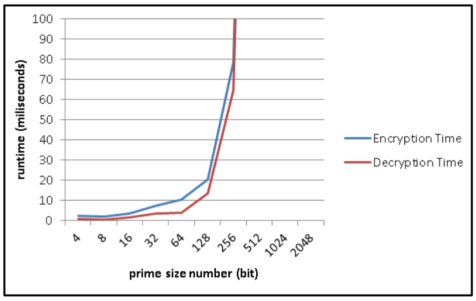

Figure 5. Runtime test result

Based on the test results, in general, the time required to perform encryption and decryption process is getting slower as the increasing size of primes that were tested.

F. Algorithm Implementation Test

In this tests the election by 30 voters with e-voting application uses homomorphic Paillier algorithm is performed. The prime numbers used are p = 50543 and q = 65053 with 16-bit length. The purpose is to prove whether the algorithm with its homomorphic properties quite well implemented in e-voting system.

Table 5.Ciphertext votes

No	ID	Candidate No.	*Ciphertext*
1	120	1	7048685014260710000
2	121	2	6199998058365330000
3	122	2	5721931450728270000
4	123	2	3271762689678700000
5	124	1	3697195553197800000
6	125	3	8132288716639470000
7	126	3	2791203262700100000
8	127	1	10071468847212300000
9	128	3	7661379461225350000
10	129	3	562902824791718000
11	130	2	9720589768897740000
12	131	2	4969482789316270000
13	132	3	7742302007199690000
14	133	3	6941721935240120000
15	134	3	1996417418952830000
16	135	3	10616339871119700000
17	136	1	3502074632894140000
18	137	2	3388899584094530000
19	138	1	2962120345059070000
20	139	2	259822751610120000
21	140	2	4493431325501730000
22	141	3	6689860453494120000
23	142	2	4603084197316520000
24	143	1	7286753142921830000
25	144	2	9469277846877200000
26	145	3	7986874716348130000
27	146	3	6653784498116810000
28	147	1	811286770593761000
29	148	1	2840117295313010000
30	149	3	5082519652939246972

The vote that has been encrypted into ciphertext stored in the database. The tallying process is done by the system using homomorphic algorithms, by multiplying the ciphertext. The result of the homomorphic tallying showed by the table below.

Table 6. Final result of the voting

Candidate No.	Candidate Name	Ciphertext Result	Decryption Result
1	Endang Sudimarno	8795881879271 410000	8
2	Siti Ani	1821048005701 860	10
3	Rudi Martaman	1337984066778 440000	12

According to the table above can be seen that the decryption result with homomorphic tallying is same with manual calculation. Based on the statement can be concluded that the homomorphic on e-voting application runs well.

V. CONCLUSION AND FUTURE WORK

This research presents the electronic voting system using Paillier algorithm with its homomorphic property. This system guarantees data confidentiality and utilizes homomorphic properties of the algorithm to calculate the votes that processed by the system. The success ratio of the system to perform calculation is 100%. Systems with Paillier homomorphic algorithms can accommodate input plaintext up to the values of n. The resulting ciphertext value is different each other though the plaintext to be encrypted has the same value.

Future work for the further research can be done are improving the security and confidentiality of the system.

REFERENCES

[1] Payal V. Parmar, Shraddha B. Padhar., Shafika N. Patel, Niyatee I. Bhatt, Rutvij H. Jhaveri, "Survey of various homomorphic encryption algorithms and schemes," International Journal of Computer Applications. Vol. 91(8), April 2014.

[2] Hyungjick Lee, Jim Alves-Foss, and Scott Harrison, "The use of encrypted functions for mobile agent security," Proceedings of the 37th Annual Hawaii International Conference on, pp. 10-pp. IEEE, 2004.

[3] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach, "Analysis of an electronic voting system," In: IEEE Symposium on Security and Privacy, 2004.

[4] Kouichi Sakurai and Tsuyoshi Takagi, "On the security of a modified Paillier public-key primitive," Information Security and Privacy. Springer Berlin Heidelberg, 2002.

[5] Harsh Marhur and Prof. Zahid Alam, "Analysis in symmetric and asymmetric cryptology algorithm," International Jurnal of Emerging Trends of Technology in Computer Science, Vol. 4(1), Jan-Feb 2015.

[6] Pascal Pascal, "Public-key cryptosystems based on composite degree residuosity classes," In Advances in cryptology-EUROCRYPT'99, pp. 223-238, Springer Berlin Heidelberg, 1999.

[7] Sansar Choinyambuu, "Homomorphic tallying with paillier cryoptsystem," HSR Hoschule für Technik Rapperswil, 2009.