Okamoto-Uchiyama Homomorphic Encryption Algorithm Implementation in E-Voting System

[1]Rifki Suwandi, [2]Surya Michrandi Nasution, [3]Fairuz Azmi,
[123]Electrical Engineering Faculty
Telkom University
Bandung, Indonesia
[1]rifki.s@hotmail.com , [2]michrandi@telkomuniversity.ac.id, [3]worldliner@telkomuniversity.ac.id

Abstract— **Nowadays, the use of electronic voting in an election has been widely adopted as a replacement for conventional voting system. Apart from the benefits of using this system, in fact, there are still many problems that can occur, such as system errors, network security, data confidentiality, and others. One of the main problems is it does not guarantee the confidentiality and accuracy of the data that to be stored and used. Cryptography is the science of mathematical techniques to secure the message offer solutions for the related problems. This research implements Okamoto-Uchiyama algorithm as the offered solutions that utilize cryptography on the security of e-voting system to ensure the security and confidentiality of the voting data, as well as utilizing the homomorphic properties of this algorithm to do the counting. This algorithm will hide the data by encrypting the votes that chosen by the voters and the system can perform calculations using the data that still encrypted without having to be decrypted first.**

Keywords—e-voting, encryption, okamoto-uchiyama, homomorphic.

I. INTRODUCTION

There are many problems in the electronic voting system, such as system errors, network security, data security, etc. One of the main problems is cheating committed by outsiders or even the administrator himself in manipulating the data that will be used or stored. To overcome these problems, cryptography needs to be applied. Before the voting data sent to the server, cryptography method is used to encrypt the data. Cryptography can guarantee security, completeness, and authenticity of voter data [1]. Various cryptographic algorithms can be applied in such systems, one of which is Okamoto-Uchiyama algorithm.

Okamoto-Uchiyama algorithm is one of the asymmetric cryptographic algorithms that used the public key to encrypt messages, while the other key is called private key, which is used to decrypt the messages [2]. This algorithm also has homomorphic property, which can perform mathematical calculations using messages that are still encrypted (ciphertext).

This research focused on the effectiveness of cryptographic using Okamoto-Uchiyama algorithm in terms of encryption and decryption of messages when implemented in e-voting system. As well as the accuracy of the calculation result of the ballots using the homomorphic property of the algorithm, which is a special property that no other algorithms in general.

II. THEORY

A. Asymmetric Encryption Algorithm

Asymmetric cryptography or also can be called public key cryptography is a cryptographic key that has a pair of related keys, which is public key to encrypt the messages, and private key to decrypt the messages. As the name implies, the public key can be put in a public place where everyone can access it, while the private key can only be accessed by its owner only. This algorithm usually has numbers based on the factorization and discrete logarithms [3].

The purpose of the existence of asymmetric cryptography is to minimize the number of locks required to perform cryptographic processes. Imagine if there will be a thousand people will communicate, then if not using an asymmetric key algorithm, it would take a thousand different keys, course it is not practical. If using asymmetric cryptography, then simply store the private key that is owned by the owner alone [3].

But there are also disadvantages of this cryptographic model, which require long processing times given enough computing needs complicated to perform factoring or discrete logarithm that characterizes this cryptography. In addition, the resulting ciphertext also usually larger in size compared ith the original plaintext [2].

B. Okamoto-Uchiyama Algorithm

This algorithm first developed by Tatsuaki Okamoto and Shigenori Uchiyama in 1998. As well as asymmetric key algorithms in general, this algorithm has the strength on the difficulty of factoring and discrete logarithm operation on primes with large digits. In this algorithm, the private key is expressed in two primes are p and q. While the public key is expressed by three numbers ie, n, g, and h, each of which has its own requirements and usability. The strength of the algorithm is slightly different from other asymmetric key algorithms that lie in the selection of very large values of n ie$n = p^2q$, compared with other algorithms, which generally has a value of $n = pq$ [4].

C. Homomorphic Property

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

Okamoto-Uchiyama algorithms have homomorphic properties. This means that the algorithm is able to compute ciphertext, the addition, and subtraction of the ciphertext, even multiplication with a certain constant, which will produce output that corresponds to the original plaintext [5].

Suppose that there are m1 and m2 which is the plaintext, while E(m1) and E(m2) is ciphertext or encrypted messages of m1 and m2. If desired an output T = m1 +/x m2, which must be done is to compute ciphertext, ie Te = E(m1) +/x E(m2) without first decrypting each ciphertext. After that just decrypt Te to get the same results with T. It can be utilized to maintain the security of the data processing that is true should not be known by anyone [3].

D. E-voting

Electronic voting is a voting system that uses electronic device as the media, it means people can vote for candidates using computers or smartphones instead of paper ballots. Generally, there are two main types of e-voting, first is remote e-voting that used internet connection. Using this method, voters do not need to be required to go to a certain place to do the voting. Second is electronic voting machines that located at the polling station. Both of them do not need to use paper as a ballot, so the costs that are typically used to print the ballot papers and the time used to distribute it will be much reduced [3].

Apart from the huge benefits that e-voting provides, the main problem of the e-voting system that needs to be concerned is the security. This system needs to transfer the data using the certain network. There are many possible ways for people that can attack the system to manipulate the data.

III. SYSTEM DESIGN AND IMPLEMENTATION

A. Key Generation

As already described in the basic theory, there are two keys in the Okamoto-Uchiyama algorithm, the private key and public key. Key generation steps are as follows :

1) Select two large prime numbers $\boldsymbol{p}$ and $\boldsymbol{q}$ randomly, then calculate $\boldsymbol{n = p^2 q}$.
2) Choose generator $\boldsymbol{g}$, when a randomly selects $\boldsymbol{g} \in (\mathbb{Z}/\boldsymbol{n}\mathbb{Z})^*$, where $\boldsymbol{g^p \neq 1 \bmod p^2}$.
3) Calculate $\boldsymbol{h = g^n \bmod n}$. [6].

After following the three steps above, then obtained as a public key tuple $< n, g, h >$ and the private key is two primes $< p, q >$.

B. Encryption Process

1) The message $\boldsymbol{m}$ that needs to be encrypted must be integer where $\boldsymbol{m} \in \mathbb{Z}/\boldsymbol{n}\mathbb{Z}$.
2) Choose a random number $\boldsymbol{r}$ with $\boldsymbol{r} \in \mathbb{Z}_{\boldsymbol{n}}^*$.
3) Compute ciphertext $\boldsymbol{C = g^m h^r \bmod n}$. [6].

C. Decryption Process

As for doing the decryption process, we must first define an auxiliary function, expressed as $\boldsymbol{L(x) = \frac{x-1}{p}}$. With the help of this function, the decryption process is expressed as follows :

$$\boldsymbol{m = \frac{L(C^{p-1} \bmod p^2)}{L(g^{p-1} \bmod p^2)} \bmod p} \qquad [6].$$

D. Homomorphic Process

The Okamoto-Uchiyama cryptosystem supports the homomorphic addition that will be used in this research. Under the same public key $< n, g, h >$ and private key $< p, q >$, given $\boldsymbol{C_0 = g^{m_0} h^{r_0}}$ **and** $\boldsymbol{C_1 = g^{m_1} h^{r_1}}$ as valid encryptions of $\boldsymbol{m_0}$ and $\boldsymbol{m_1}$ respectively,

$$\boldsymbol{C_0 C_1 = g^{m_0+m_1} h^{r_0+r_1} \bmod n} \qquad [5].$$

E. Implementation

Figure 1 shows about the system design that proposed in this research.

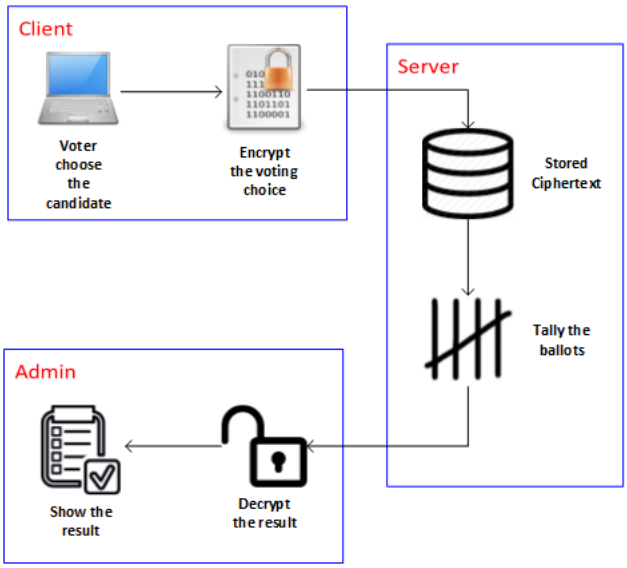

Figure 1. *System Design*

Basically, the e-voting system should have a graphical user interface (GUI), which voters can use to communicate with the server system. Whole design mentioned above will be implemented in e-voting application using Java programming language and uses MySQL for its database.

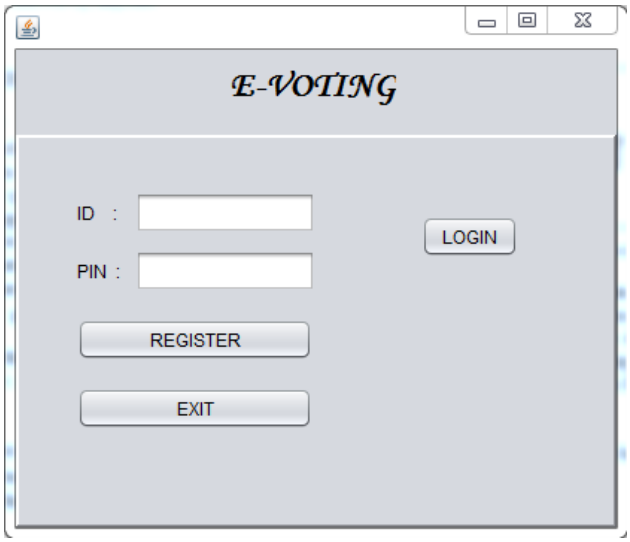

Figure 2. *The application's main menu*

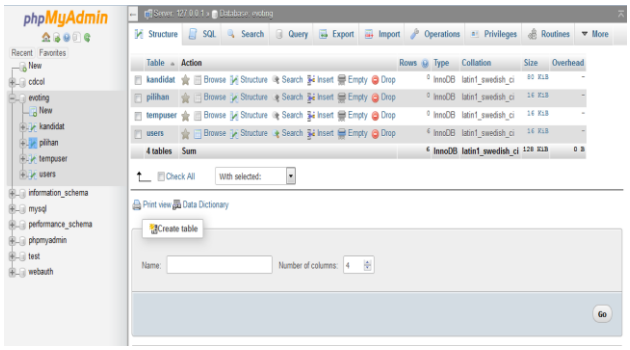

Figure 3. *Preview of databases*

Initially, users must register to perform validation. Users who have been validated need to login first and perform an election by choosing one of the candidates in the list. After the user selects a candidate, the choice's data will be encrypted by the application.

Figure 4. *Ballot's preview*

After the data encrypted, the encrypted data (ciphertext) will be stored in databases. Each voter will have a different ciphertext although they chose the same candidates.

The tally program will calculate each ciphertext. In Okamoto-Uchiyama algorithm, as explained before, can obtain $E(m1 + m2)$ using homomorphic property by multiplying $m1$ and $m2$ without decrypting beforehand.

After the election ends, the admin can decrypt the calculated results using the private key. The results of calculated ciphertexts will be same with the originally calculated messages after being decrypted.

IV. RESULTS AND ANALYSIS

The test will involve 10 voters will choose one of three different candidates. After each voter finish selecting the candidates, their choice's data will be encrypted by the encryption program. Table 1 shows that although the voters choose same candidates, each ciphertext will be different.

Table 1. *Vote messages to be encrypted*

Voter Name	**Candidates**			**Vote Messages (m)**
	C1 (10^0)	**C2** (10^1)	**C3** (10^2)	
V1		v		m = 10
V2		v		m = 10
V3			v	m = 100
V4	v			m = 1
V5			v	m = 100
V6			v	m = 100
V7			v	m = 100
V8	v			m = 1
V9			v	m = 100
V10		v		m = 10
Total	**2**	**3**	**5**	

The maximum vote message can be : $m_{max} = 10^2 = 100$. So the maximum possible tally can be : $T_{max} = N_V$ x m_{max} = 10 x 100 = 1000.

Step 1 : Key generation.

- Choose two primes randomly p and q, which the value of p must be greater than T_{max}. So the value of p and q at least must be of 16 bits length. $p = 60101$, $q = 49081$.
- Calculate $n = p^2 \; x \; q = \; 60101^2 \; x \; 49801 = 177286962395281$.
- Select generator g randomly. $g = 13$.
- Calculate $h = \; g^n \; mod \; n = 167855765653197$.

Step 2 : Encrypt each message m as shown in the table above.

- $C_i = \; g^{m_i} \; h^{r_i} \; mod \; n$

***Table 2**. Encrypted vote*

Voter name	Vote message to be encrypted	Random r_i	Encrypted vote C_i
V1	10	5571	29506521184002
V2	10	5618	53358847146148
V3	100	39152	172935136473931
V4	1	4253	126400426329454
V5	100	49910	36221372195014
V6	100	64626	53358010542153
V7	100	48589	118346020440719
V8	1	50124	25872362462795
V9	100	44966	157128349866557
V10	10	63484	80675382575000

After the election process is completed, the admin can tally the encrypted data by using the homomorphic property of Okamoto-Uchiyama algorithm. The server will multiply all encrypted messages. $\boldsymbol{Tally\ (T_C) = \prod_{i=1}^{N_V} C_i =}$ **25817018325636756507859004420889084523864826798005850029582615900971649181925451695071278840345483720225293024849738002772353872548360000000.**

Then the admin can decrypt T_C to get tally message m. Using decryption formula :

$$\boldsymbol{m = \frac{L(T_C{}^{p-1} \; mod \; p^2)}{L(g^{p-1} \; mod \; p^2)} \; mod \; p = 532}$$

Finally, the result of this voting is decided. As calculated before, $m = 532$, which means candidate C1 has 2 votes, candidate C2 has 3 votes, and candidate C3 has 5 votes. So the candidate C3 is the winner of this election.

If we count manually the choices in figure 6, two voters choose candidate number 1, three voters choose candidate number 2, and five voters choose candidate number 3. The result will be same as the final score as calculated before.

As we can analysis, the size of the ciphertext is much larger than the original message, because this algorithm based on large prime numbers of p and q with quite complicated to perform factoring or discrete logarithm, such as calculating values of $n = p^2q$, which is very large compared with other algorithms, which generally has a value of $n = pq$.

The possibility of ciphertext turn out has similar value based on the size of prime number p and q. Even though the message that will be encrypted similar, the random number r that different each encrypting process will make the ciphertext different each other. that means this algorithm proved that has uniqueness in the encrypted result.

The homomorphic properties that we use in this study proved to produce accurate results between calculated result using ciphertext that similar with desirable result that calculated manually after decrypted. So we can use that property to anticipate cheating by the outsider to manipulate the stored data or trying to interrupt the tally process.

V. CONCLUSION AND FUTURE WORK

In this research, we implement Okamoto-Uchiyama homomorphic encryption algorithm in e-voting system. With this algorithm, we can ensure data confidentiality and utilizes homomorphic properties of the algorithm to calculate the votes that processed by the system. The result of testing and analysis shows that the voting data is well encrypted and has a unique value for each ciphertext. The final result that calculated using homomorphic property is similar to the desirable result after being decrypted.

Future work that can be done in further research is to complete the full security objectives of e-voting system such as integrity, authenticity, non-replay, etc.

REFERENCES

[1] Agustina Esti Rahmawati, and Agus Kurniati. 2009. *Pemanfaatan Kriptografi dalam Mewujudkan Keamanan Informasi pada E-voting di Indonesia*. Informatics National Seminar 2009 (semnasIF 2009), UPN Veteran, Yogyakarta.

[2] Nathanael Ezra Hizkia. 2013. *Implementasi Algoritma Kriptografi Kunci Publik Okamoto-Uchiyama*. Informatics Engineering, Bandung Institue of Technology.

[3] Ngo Cuong. 2014. *Secure Voting System Using Paillier Homomorphic Encryption*. Faculty of the Department of Computing Sciences Texas A&M University – Corpus Christi, Texas.

[4] Coron Jean-Sebastian, David Naccache, Pascal Paillier. 1999. *Accelerating Okamoto-Uchiyama's Public-Key Cryptosyste*. Electronic Letters. 35(4):291-291.

[5] Henry Kevin, 2008. *The Theory and Applications of Homomorphic Cryptography*. Computer Science, University of Waterloo, Canada.

[6] Okamoto Tatsuaki, and Shigenori Uchiyama. *A New Public-Key Cryptosystem as Secure as Factoring*. NTT Laboratories, 1-1 Hikarinooka, Yokosuka-shi, 239-0847 Japan.

[7] Peng Kun, Colin Boyd, Ed Dawson, Byoungcheon Lee, and Riza Aditya. 2004. *Multiplicative Homomorphic E-Voting*. In Canteaut, A & Viswanathan, K (Eds.) Progress in Cryptology – INDOCRYPT 2004. 5th International Conference on Cryptology in India.

[8] Rahmmadian Tika, Aswin Suharsono, Ahmad Afif Supianto. *Desain dan Implementasi Sistem Keamanan E-Voting dengan Jaminan Confidentiality Data*. Computer and Informatics Engineering, University of Brawijaya.

[9] Vaghasia Chandni, Kirti Bathwar. 2013. *Public Key Encryption Algorithms for Wireless Sensor Networks in TinyOS*. International Journal of Innovative Technology and Exploring Engineering (IJITEE). ISSN: 2278-3075, Volume-2, Issue-4.

[10] Zhao Yingming, Yue Pan, Sanchao Wang, and Junxing Zhang. 2014. *An Anonymous Voting System Based on Homomorphic Encryption*. Inner Mongolia University, School of Computer Science, Huhhot, China.