Design and Implementation of Secure Remote e-Voting System Using Homomorphic Encryption

Ihsan Jabbar and Saad Najim Alsaad
(Corresponding author: Ihsan Jabbar)

Department of Computer Science, College of Science, University of Mustansiriyah
Safi Al Din Al Hilli Street, Baghdad, Iraq
(Email: ihsan.jabbar90@gmail.com)
(Received Jan. 28, 2016; revised and accepted Apr. 23 & Oct. 25, 2016)

Abstract

Internet polling also known as "e-voting" became popular in past few years, since it reduces the tallying cost and time, increases the number of voter participation, also reduces the human resources and the traditional work that means less fraud and corruption. In this paper, a remote e-Voting system is designed and implemented using homomorphic encryption. The homomorphic property in ElGamal cryptosystem are exploited to achieve two important voting requirements: first, the security of device used for electronic voting by voter. Second, the voter has the ability to choice willfully and uncoercionly. The general voting system requirements such as eligibility, privacy, accuracy, fairness, Receipt-freeness, coercion resistance, mobility, simplicity, individual verifiability, scalability and availability are also achieved in the system.

Keywords: Electronic Voting; ElGamal; Homomorphic Encryption

1 Introduction

Electronic voting is a process completely conducted by electronic devices such as computers and communication technologies. Its applications such as elections are so sensitive in terms of security. Current e-Voting schemes are based on either mix network, or blind signatures, or homomorphic encryption. Homomorphic encryption is used to make sure that votes holds its confidentiality by encrypting and calculating all votes without decrypting.

Voting schemes based on homomorphic encryption were first introduced by Benaloh [5]. Several improved schemes were developed after that. These schemes follow the similar election procedures, but they introduce new security properties, such as receipt-freeness. In 1997, Crammer et el. [7], introduced a new multi-authority secret-ballot election scheme based on the discrete log assumption. Their scheme achieves privacy, universal verifiability and robustness. In 2002, Rivest [18] discussed in his lecture notes a voting scheme based on homomorphic property of Paillier cryptosystem to achieve the privacy of voters by tallying the encrypted votes. This scheme used blind signature which allow for anonymous voting. In 2010, George and Sebastian [10] presented a voting scheme based on homomorphic encryption. The scheme achieves privacy, uncoercibility, and receipt-freeness. The scheme can be used for both yes/no and multi-candidates types of voting. In 2011, Huszti [13] proposed a homomorphic encryption-based voting scheme based on Crammer Scheme [7]. The scheme achieves eligibility, unreusability, privacy, verifiability, receipt-freeness, and uncoercibility. It only needs anonymous channels. In 2013, Hussien and Aboelnaga [12] proposed a new voting scheme based on additive homomorphic property of Paillier cryptosystem and blind signature based on RSA. The scheme achieves eligibility, secrecy, uniqueness, privacy and accuracy. In 2013, Yi and Okamoto [22] presented voting scheme which maintains the privacy of voter even if the voter's PC infected by malware or the voter is physically controlled by the adversary. The scheme can only tell if the candidate wins or loses without the number of yes or no votes. In 2014, Zhao et al. [23] presented a voting scheme based on homomorphic encryption to ensure anonymity, privacy and reliability. The scheme using RSA cryptosystem to encrypted the data. In 2015, Will et el. [21] described a partially homomorphic cloud-based mobile voting system. They implemented the system to show its practicality. The system achieves eligibility, unreusability, untraceability, verifiability, tally correctness, uncoerceability, auditability, accessibility, fairness, soundness and integrity.

In this paper, a secure e-Voting system based on homomorphic property of ElGamal cryptosystem is designed and implemented. Our system achieved the following e-Voting system requirements: eligibility, privacy, accuracy, fairness, receipt-freeness, coercion resistance, mobility, simplicity, individual verifiability, scalability and availability.

The rest of this paper is organized as follows: Section 2 provides the background of homomorphic encryp-

tion. Section 3 presents ElGamal cryptosystem. The design and implementation of the system are introduced in Section 4. In Section 5, a very simple testing example is given. The security analysis is discussed in Section 6. Finally, our conclusions are drawn in Section 7.

2 Homomorphic Encryption

Homomorphic encryption is a type of encryption that allows particular computations to be conducted on ciphertext and return an encrypted result, the decrypted of result is equal the result of conducting the operation on the plaintext. The property of homomorphic is useful to develop a secure e-voting system with high privacy data retrieving scheme, also it makes the use of cloud computing by ensuring the privacy of processed data. An example for its mathematical consistency, if there are two numbers 10 and 20 then both are encrypted to 56 and 69 respectively, the addition operator gives a number with value 125, the decrypted of this value is 30 [14].

The concept of homomorphic encryption was suggested in 1978 by Rivest and Adleman [19], But for 30 years the progress is very slow. In 1982, Goldwasser and Micali [11] proposed their encryption system that was able to encrypt one bit in additive homomorphic encryption. Paillier [16] in 1999 suggested another additive homomorphic encryption. Boneh, Goh and Kobi [2] in 2005 were invented a security system of encryption which conduct only single multiplication but large number of additions. In 2009, Gentry [9] construct a fully homomorphic encryption based system that able to conduct both of addition and multiplication, but the scheme is impractical. Several optimizations and refinements were proposed after that, but this schemes are still inefficient and impractical [20].

3 ElGamal Cryptosystem

Based on the Diffie-Hellman key exchange, Taher ElGamal [8] presents his public key cryptosystem in 1985. The security of ElGamal encryption scheme depends upon the difficulty of computing discrete logarithms over finite field. ElGamal scheme can be defined over any cyclic group **G** with a large prime order q and a generator g. The three components that configure the scheme are as shown in Figure 1.

ElGamal cryptosystem has a homomorphic property as follows [6]:

$$(C_{1,1}, C_{1,2}) = (g^{r_1}, p_1.y^{r_1})$$

and

$$(C_{2,1}, C_{2,2}) = (g^{r_2}, p_2.y^{r_2})$$

where r_1 and r_2 are randomly chosen from $\{1, 2, \ldots, q-1\}$

- **Key Generation**
  - Select a large prime as a q
  - Select x *to* be a member of the group **G** = < Zq^*, X >, x must be "$1 \le x \le q-1$"
  - Select g *to* be a primitive root (generator) in the group **G** = < Zq^*, X >
  - $y = g^x \bmod q$
  - Public key ← (g, y, q)
  - *Private key* ← x
- **Encryption**
  - Select a random integer r in the group **G** = < Zq^*, X >, r must be "$1 \le r \le q-1$"
  - $C_1 = g^r \bmod q$
  - $C_2 = (p \,.\, y^r) \bmod q$ // p is the plaintext
- **Decryption**
  - $P = [\, C_2 (C_1{}^x)^{-1} \,] \bmod q$

Figure 1: ElGamal cryptosystem pseudocode

and $m_1, m_2 \in \mathbf{G}$ someone can compute:

$$\begin{aligned} E(p_1).E(p_2) &= (C_{1,1}, C_{1,2}).(C_{2,1}, C_{2,2}) \\ &= [g^{r_1}.g^{r_2}, (p_1.y^{r_1}).(p_2.y^{r_2})] \\ &= [g^{r_1+r_2}, (p_1.p_2), y^{r_1+r_2}] \\ &= E(p_1.p_2), \end{aligned}$$

where E symbolizes to the encryption process.

4 Design and Implementation

Developing an e-Voting system requires the collaboration of many participants with different background. In our system, there are five participating actors: Administrator, Registrars, Tally Authorities, Candidates and Voters. Table 1, summarizes the participating actors and their responsibilities. The design of e-Voting System depicted in Figure 2, consists of four stages: election setup, registration, voting and tallying. These stages are consecutively, that's mean no feedback between one stage to another. We assume that there is a Bulletin Board **(BB)**, an insert only board readable by the public. This system supports multi-candidate elections which has n_C of candidates. Each voter V_i cast his vote for each candidate. This vote may be Yes or No, this is equivalent to 1 and -1, respectively.

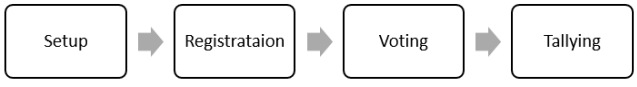

Figure 2: General structure of the system

1) **Election Setup:** The aim of this stage is to initialize election database and calculate necessary parameters to pass them to the next stages. The algorithm of this stage is depicted in Algorithm 1. It illustrates that two actors participate in this stage: Admin and Tally authority.

   The implementation is based on ElGamal encryption scheme over a group G with a large prime order q and a generator g. These parameters should be determined by admin. In addition, the election database

Table 1: Actors participating and responsibilities

Actor	Responsibilities	Notes
Administrator	responsible for the eligible voters list, election setup and controlling the entire system	N\A
Registrars	authorize the voters for the election during registration stage	Registrars are distributed on regions to facilitate the registration process
Tally Authorities	responsible for counting the votes and the announcement of the final results of the election	The list of tallying authorities $T = \{T_1, T_2, \cdots, T_{n_T}\}$
Candidates	asking for the vote and competing with each other to get highest number of votes	The list of candidates $C = \{C_1, C_2, \cdots, C_{n_C}\}$
Voters	qualified to vote and casting ballots	The list of voters $V = \{V_1, V_2, \cdots, V_{n_V}\}$

Algorithm 1 Setup stage

Input: list of all eligible citizens to vote
Output: Election DB, $g, q, PVTKEY$ for T, n_T, n_C
1: Admin creates database and chooses $q, g,$ and n_T
2: **for all** i in T **do**
3: Choose a prime t_i as private key $(PVTKEY_i)$
4: Calculate $PUBKEY = g^{t_i}$
5: **end for**

should be created by admin. It consists of five tables: administrator, registrars, tally authorities, candidates, and registered voters. Also, in this stage each tally authority T_i chooses a random prime t_i as a private key $PVTKEY_i$ from Z_q^* then calculates the public key using Equation (1)

$$PUBKEY_i = g^{t_i} \tag{1}$$

The sequence diagram of this stage is shown in Figure 3.

2) **Registration:** The aim of this stage is to enable eligible citizens to be registered for voting stage. The algorithm of this stage is depicted in Algorithm 2.

Algorithm 2 Registration stage

Input: Election DB, $g, q, PUBKEY$ for T, n_T, n_C
Output: $n_V, (A, B)$ for each v
1: **for all** v want to register **do**
2: Registrar checks the eligibility of v
3: Generate r for v
4: Generate $y \in Z_q^*$
5: $(A, B) = Enc(r)$
6: Delete r and save ciphertexts (A, B)
7: **end for**

In registration stage, remember that the number of candidates n_C is given. The voter V_i identify himself to the registration employee (registrar) by the identification card (citizen card). The registrar enters the required information into system to verify if the voter's information exists in the eligible voters list. If the voter is eligible, the system sends a password to his provided email address. The voter V_i will proceed the process of registration in private booth. Each V_i can login into system with the password which was sent by the system to the voter's email address. The system generates a reference r_i for each V_i. This reference is an integer, it is a string of bits with length equal to the total number of candidates n_C .The system generates the reference using Equation (2).

$$r_i = a_{i,1} + a_{i,2} + \cdots + a_{i,n_C}{}^{2^{n_C-1}}, \tag{2}$$

where $a_{i,j} \in \{0, 1\}$.

The reference r_i is encrypted using ElGamal cryptosystem into two ciphertexts using Equations (3) and (4).

$$A_{i,j} = g^{y_{i,j}} \tag{3}$$

$$B_{i,j} = \begin{cases} g(\prod_{t=1}^{n_T} PUBKEY_t)^{y_{i,j}} & \text{if } a_{i,j} = 0 \\ g^{-1}(\prod_{t=1}^{n_T} PUBKEY_t)^{y_{i,j}} & \text{if } a_{i,j} = 1, \end{cases} \tag{4}$$

where $y_{i,j}$ is randomly chosen by the system from Z_q^*.

Hence, the system permanently deletes the reference and keeps only its ciphertexts. Encrypted reference saved in a separately file until the elections day. At the end, the voter's information will be saved as new entries to the "registered voter" table in the database. The sequence diagram of this stage is shown in Figure 4.

3) **Voting:** The aim of this stage is to enable registered voters to cast their votes. In this stage, the system does not require a secret channel for the voters to cast their votes, also does not need to encrypt the votes, this allow the voters to verify that their votes are correctly included and not manipulated by malwares or viruses. The algorithm of this stage is depicted in Algorithm 3.

After the announcement of the candidates by Admin on Bulletin Board (**BB**), a voting stage is started. To cast his vote, the voter V_i should remember his

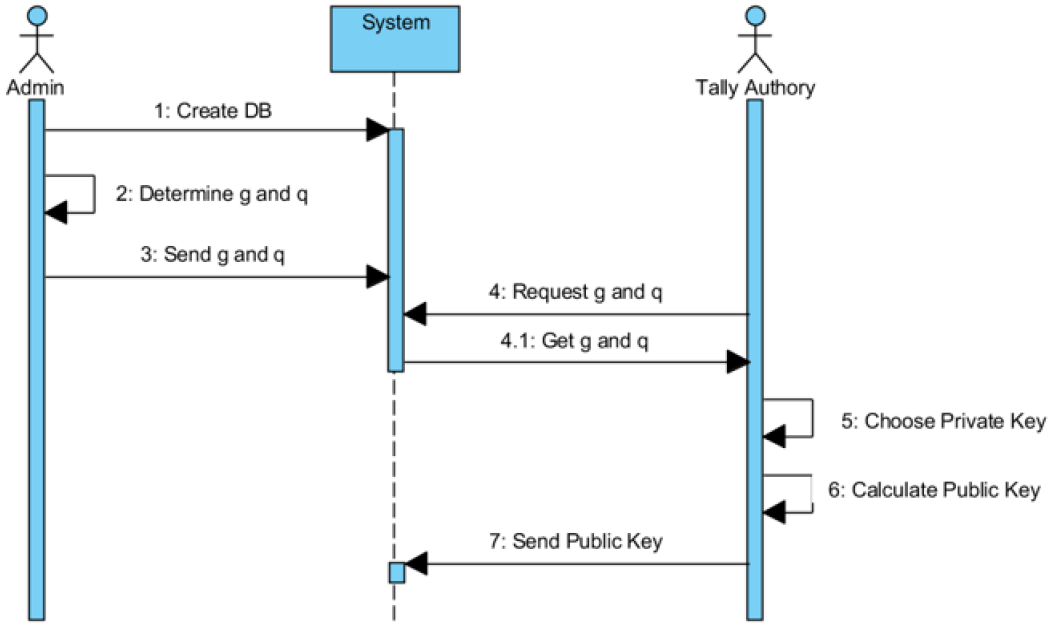

Figure 3: Election setup stage sequence diagram

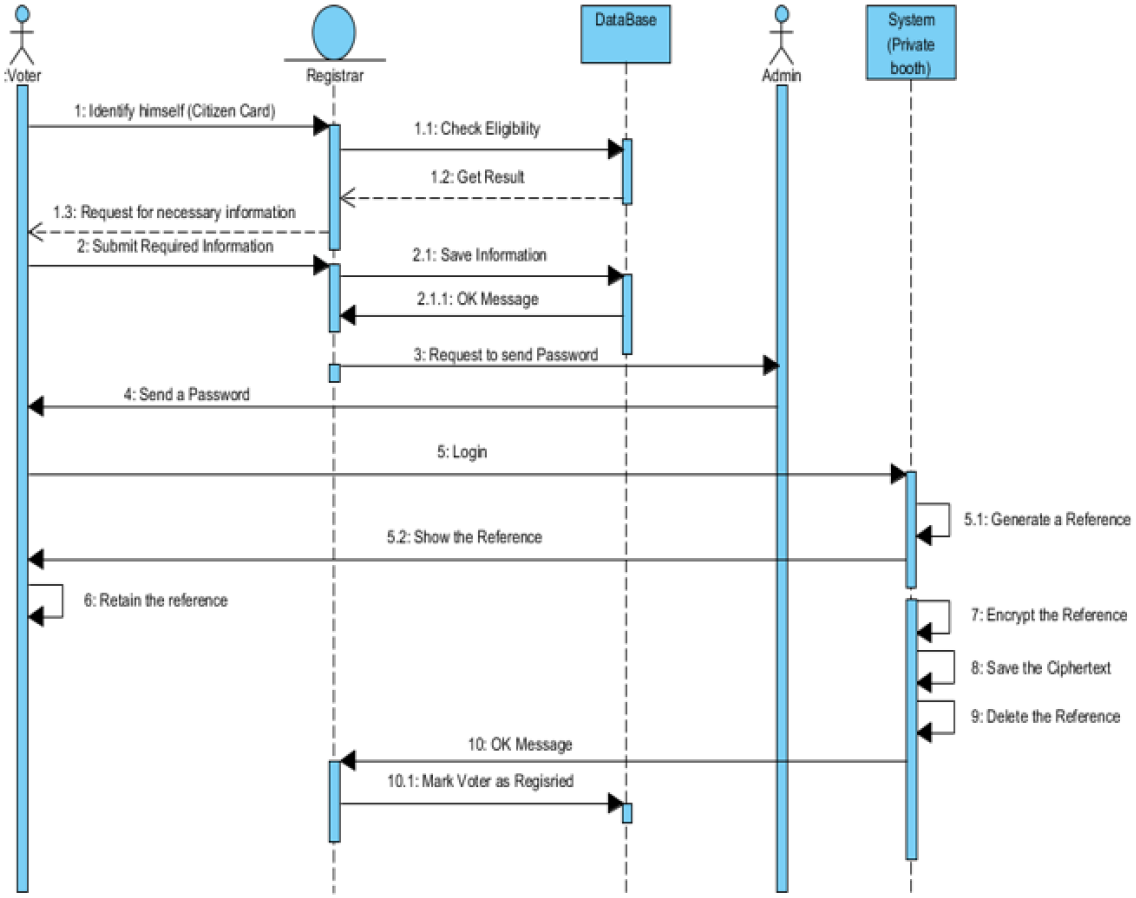

Figure 4: Registration stage sequence diagram

Algorithm 3 Voting stage

Input: Election DB, n_V, n_C
Output: β

Admin announces the list of candidates on **BB**
for all i in V **do**
  for all j in C **do**
    V_i determines his $\beta_{i,j}$ and send it to the system
    Admin publishes $\beta_{i,j}$ on **BB**
  end for
end for

reference r_i and login to the system with the required credentials. The voter V_i chooses a sequence of his $\beta_{i,j}$ with values of 1 or -1 according to the Table 2.

Table 2: Voter guide to determine β

Reference	Vote	β
0	Yes	1
1	Yes	-1
0	No	-1
1	No	1

For example, if the number of candidates n_c is 4 and the reference number r_i is $(0, 1, 1, 0)$. Suppose that, the voter V_i wants to select (Yes) for the third candidate and (No) for the other candidates, then the β_i should be $(-1, 1, -1, -1)$. This is the point, the string of β_i dose not refers to what the voter is selected.

After entering β for each candidate, β sent to the system saved in separately and stand by for tallying. At the same time, the β will be published publicly on the (**BB**). The voters can know if their votes have been changed or not, everyone can access to the , but only the voter himself knows what does it means. The sequence diagram of this stage is shown in Figure 5.

4) **Tallying:** The aim of this stage is to count the ballots and get the final result for each candidate. Each tally authority should apply the algorithm (A) of this stage. This algorithm is depicted in Algorithm 4.

In this stage, tally authorities combine all valid ballots (β) posted on (**BB**) using the homomorphic property of ElGamal scheme as shown in Equations (5) and (6).

$$X_{T,j} = \prod_{i=1}^{n_V} {A_{i,j}}^{B_{i,j}} \quad (5)$$

$$Y_{T,j} = \prod_{i=1}^{n_V} {B_{i,j}}^{B_{i,j}}. \quad (6)$$

By using ElGamal encryption scheme, each tally authority T_i, calculates $X_{i,j}$ using Equation (7).

$$X_{i,j} = {X_{T,j}}^{PVTKEY_i}. \quad (7)$$

Algorithm 4 Algorithm (A) of tallying stage

Input: $n_T, n_C, n_V, PVTKEY, (A, B), \beta$
Output: X_{n_T} and Y_{n_T}

for all j in C **do**
  $X_{n_T} = 1, Y_{n_T} = 1$
  for all i in V **do**
    $X_{T,j} = X_{T,j} * {A_{i,j}}^{\beta_{i,j}}$
    $Y_{T,j} = Y_{T,j} * {B_{i,j}}^{\beta_{i,j}}$
  end for
end for
for all i in T **do**
  for all j in C **do**
    $X_{i,j} = {X_{T,j}}^{PVTKEY_i}$
  end for
  T_i sends $X_{i,j}$ and $Y_{T,j}$ to Admin
end for

The algorithm (B) of tallying stage is applied by Admin as depicted in Algorithm 5.

Algorithm 5 Algorithm (B) of tallying stage

Input: Election DB, $g, n_T, n_C, n_V, X_{n_T}, Y_{n_T}$
Output: No. of (Yes/No) for each Candidate

for all j in C **do**
  $eq_j = 1$
  for all i in T **do**
    $eq_j = eq_j * {X_{i,j}}^{-1}$
  end for
  $g^{y_j - n_j} \leftarrow (eq_j * Y_{T,j})$
  $Z_j = \dfrac{ln(g^{y_j - n_j})}{ln(g)}$
  $y_j = \dfrac{Z_j + n_V}{2}$ // y_j : No. of (Yes) votes for C_j
  $n_j = n_V - y_j$ // n_j : No. of (No) votes for C_j
  Admin publishes y_j and n_j on **BB**
end for

All X_{n_T} and Y_{n_T} are sent to admin. In turn, he calculates eq_j using Equation (8).

$$eq_j = Y_{T,j}. \prod_{i=1}^{n_T} {X_{i,j}}^{-1} \quad (8)$$

eq_j is:

$$\begin{aligned} eq_j &= \prod_{i=1}^{n_T} g^{\beta_{i,j}(-1)^{a_{i,j}}} \\ &= \prod_{i=1}^{n_T} g^{v_{i,j}} \\ &= g^{y_j - n_j}, \end{aligned}$$

where y_j and n_j are represented the number of (Yes) and (No), respectively, for the candidate C_j. and:

$$y_j + n_j = n_V.$$

To determine the values of y_j and n_j using the fol-

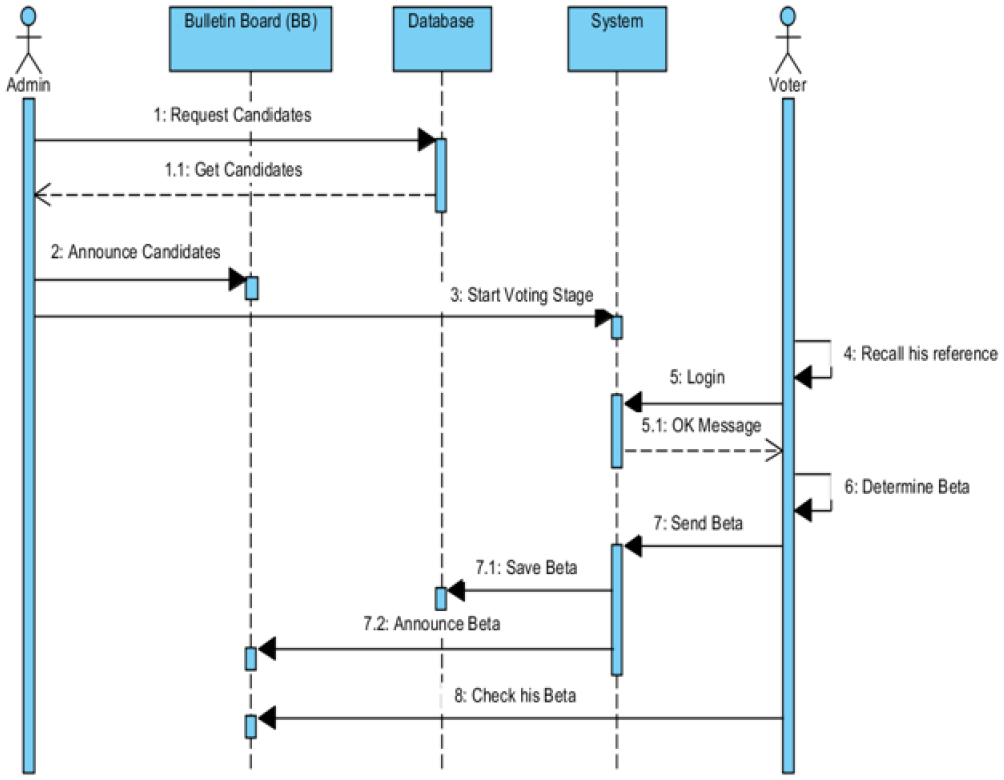

Figure 5: Voting stage sequence diagram

lowing equations:

$$g^{2y_j - n_V} = g^{y_j - n_j}$$

$$Z_j = \frac{ln(g^{y_j - n_j})}{ln(g)} \quad (9)$$

$$y_j = \frac{Z_j + n_V}{2} \quad (10)$$

$$n_j = n_V - y_j. \quad (11)$$

At the last, admin announce final results y_j and n_j for each candidate C_j on (**BB**). The sequence diagram of this stage is shown in Figure 6.

Figure 7, summarizes the four stages of the e-Voting system in more details focusing on the parameters of each stage.

5 Testing Example

Here is a simple example. We choose $q = 13$ and $g = 5$. The number of tally authorities $n_T = 3$. Table 3, illustrates the details of each tally authority.

The number of candidates $n_C = 3$, the number of voters $n_V = 10$. Suppose that, the references and votes of the voters as shown in Table 4. Table 4 also shows the corresponding β of these assumptions.

The voter V_7 in the table is taken to be an example of this testing. the reference of V_7 is: $r_7 = (0, 1, 0)$.

Table 3: A simple example (private key & public key for each tally authority)

i	T_i	$PVTKEY_i$	$PUBKEY_i$
1	T_1	7	78125
2	T_2	11	48828125
3	T_3	13	1220703125

The following results are the values of cipher texts A_7 and B_7 after encrypt r_i by applying Equations (3) and (4).

$$A_{7,1} = 125$$

$$B_{7,1} = 504870979341447555463506281780983186 99085211846977472305297851562 5$$

$$A_{7,2} = 25$$

$$B_{7,2} = 433680868994201773602981120347976684 5703125$$

$$A_{7,3} = 25$$

$$B_{7,3} = 108420217248550443400745280086994171 142578125.$$

When voter V_7 wants to vote in voting stage, he should login into system by his email and password. After that, the voter should determine his string of

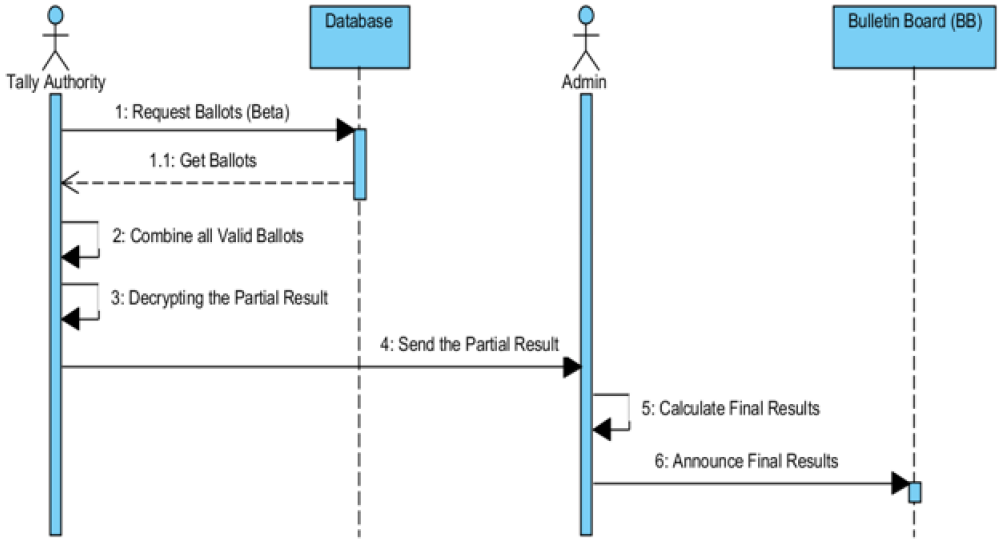

Figure 6: Tallying stage sequence diagram

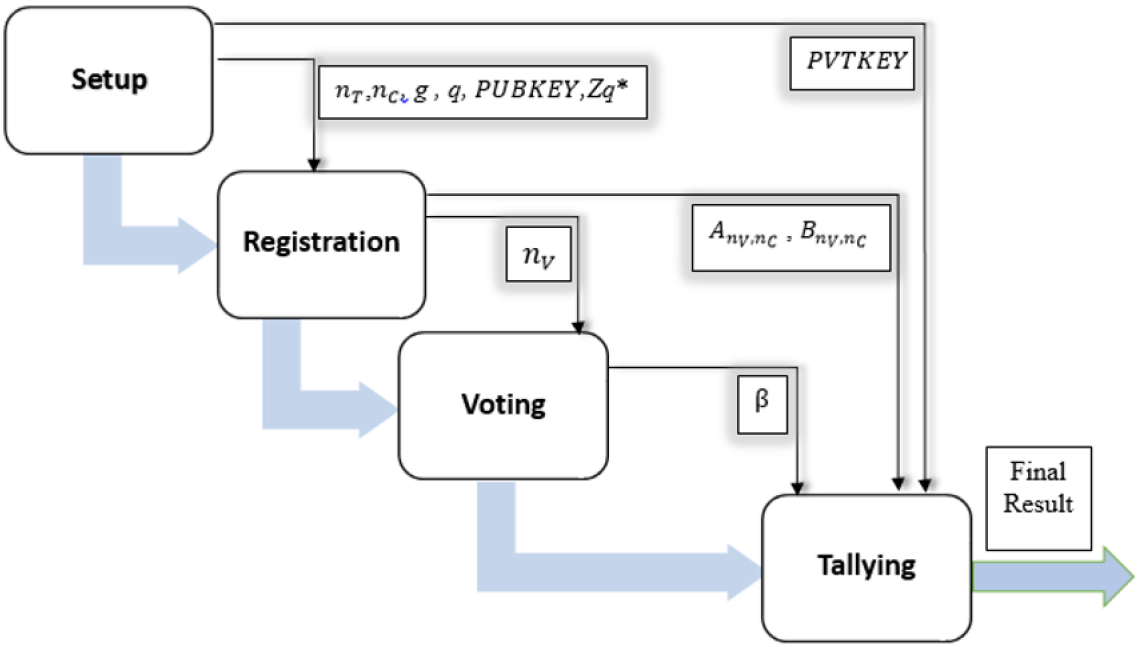

Figure 7: The structure of the e-Voting system

Table 4: A simple example (reference, vote and β for each voter)

Voter	r_i			Vote			β_i		
V_1	1	1	0	Yes	No	No	-1	1	-1
V_2	0	1	0	No	No	Yes	-1	1	1
V_3	1	1	0	No	No	Yes	1	1	1
V_4	1	1	0	Yes	No	No	-1	1	-1
V_5	0	1	1	No	No	Yes	-1	1	-1
V_6	1	0	0	No	No	Yes	1	-1	1
V_7	0	1	0	No	No	Yes	-1	1	1
V_8	1	1	1	No	No	Yes	1	1	-1
V_9	1	0	1	Yes	No	No	-1	-1	1
V_{10}	1	1	0	No	No	Yes	1	1	1

β corresponding to his reference and vote. As it has been imposed in Table 4, the voter V_7 intends to vote with select (Yes) for the third candidate and (No) for the others. Then, the string of his β should be $(-1, 1, 1)$.

In tallying stage, the results of applying Equation (5) by tally authorities are the following:

$$\begin{aligned} X_{T,1} &= 0.00032; \\ X_{T,2} &= 244140625; \\ X_{T,3} &= 625. \end{aligned}$$

The results of applying Equation (6) also by tally authorities are the following:

$$\begin{aligned} Y_{T,1} &= 7.3075081866545145910184250549280887399251968 \\ Y_{T,2} &= 1064489960002037679977513429061827212061649183577670400231262522961602207186388034071259220931839225487022711293188881345298645423989667325397992779882663154942462236970742476180620342216409767200823580685565725948565907188481105549726635217666625976562 5 \\ Y_{T,3} &= 2938735877055718769921841343055614194546663891930218803771879265696043148636817932128906 25. \end{aligned}$$

The following results of applying Equation (7) by each tally authority: $X_{i,j}$, where $i = 1$ to n_T and $j = 1$ to n_C.

$$\begin{aligned} X_{1,1} &= 3.4359738368 \\ X_{1,2} &= 516987882845642296794630432543726783478632569313049316406 25 \\ X_{1,3} &= 37252902984619140625 \\ X_{2,1} &= 3.6028797018963968 \\ X_{2,2} &= 183670992315982423120115083940975887159166493245638675235742454106002696789801120758056640625 \\ X_{2,3} &= 56843418860808014869689941406 25 \\ X_{3,1} &= 3.6893488147419103232 \\ X_{3,2} &= 109476442525376333665916373694524697756270464209102794668520959678889928334832859491143608465790748596191406 25 \\ X_{3,3} &= 22204460492503130808472633361816406 25 \end{aligned}$$

The following results of applying Equation (8) for each candidate, where $j = 1$ to n_C:

$$\begin{aligned} eq_1 &= 0.00160000000000000000000000000195162925774229967039972988285256860763183794915676116943359375 \\ eq_2 &= 1.02400000000000008533649470216805061997558804476292671752859762412286530608429290809578771490697461634273449669102906757236935667916355866785777021114639109778566063810973265372200026160300237081865470933905726269586011767387390136718 75 \\ eq_3 &= 625. \end{aligned}$$

By applying Equation (9), get the following results for each candidate:

$$Z_1 = -4, \quad Z_2 = -10, \quad Z_3 = 4.$$

Now get the number of (Yes) votes and number of (No) votes for each candidate by applying Equations (10) & (11) as shown in Figure 8. Finally, admin announces these results on (**BB**).

6 Analysis

The design of any e-Voting system should be satisfy a number of basic and extended requirements. However, it is impossible to satisfy all of these requirements at the same time [4]. Many researchers described these requirements such as [1, 3, 15, 17]. Our e-Voting system achieves the following properties and requirements:

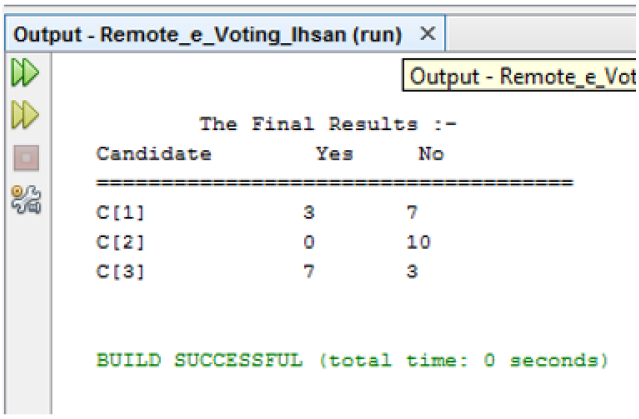

Figure 8: The final results of the example

1) **Eligibility:** Registrars Prove and confirm the eligibility of the person to vote. They denied from election any person who does not met the pre-defined requirements. In addition, there is a field called (isVoted) in class of voter. This field is marked when the voter casts his vote to prevent him from voting one more time.

2) **Privacy:** In the system, no one can link the identity of the voter and his vote, even the tally authorities. The privacy of voter is preserved by using homomorphic encryption protocol and reference technique.

3) **Accuracy:** The system can tally the valid votes with high accuracy by using homomorphic property of El-Gamal cryptosystem.

4) **Fairness:** No one can know the intermediate results or any partial result during election since the system is designed for multiple tally authorities. All tally authorities and admin should be jointly compute the final results.

5) **Receipt-freeness/coercion resistance:** The using of reference in the system make the system is receipt-freeness and that prevents the vote-buying. The voter cannot prove what is he voted in election to others.

6) **Mobility:** The system require the voter comes to specific locations only during registration days. The voter can cast his vote from anywhere he access to Internet during voting day.

7) **Simplicity/Convenience:** The system is relative simple. The user interface is user-friendly and not require high skills from the voters.

8) **Individual verifiability:** A voter must have the ability to verify that the vote he casted is accounted in the tally without any modification. The system achieved this property by publishing the β on (**BB**). Each voter can reach his β easily and check if it is changed or not.

9) **Scalability:** The practicality of a voting system depends on the factor of protocols complexity that used in the system. With aspect to storage, computation and communication, the system has to be scalable to any number of voters with more computations and hardware requirements.

10) **Availability:** In our system, the voters can be access all features during the election period.

7 Conclusions

In this paper, a practical secure e-Voting system is presented. By using homomorphic encryption, the system achieves the confidentiality of the voters. The system does not require a secure channel during a voting stage. The needless of encryption of votes allows the voter ensure that his vote is not changed. Unfortunately, e-Voting schemes based on homomorphic encryption require high computational space and time. The overhead for tallying is increasing depending on the increase in the number of voters, candidates and values of parameters. The system is practical with small and large scale elections with more computation as the election size increased. The system is implemented using JAVA, the language that deals with the huge numbers that consist of thousands of digits in both the integer and decimal form by using BigInteger and BigDecimal classes.

References

[1] M. A. Based and S. F. Mjølsnes, "Security requirements for internet voting systems," in *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering*, pp. 519–530, Springer, 2013.

[2] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography Conference*, pp. 325–341, 2005.

[3] Y. Chen, J. Jan, and C. Chen, "The design of a secure anonymous internet voting system," *Computers & Security*, vol. 23, no. 4, pp. 330–337, 2004.

[4] B. Chevallier-Mames, P. Fouque, D. Pointcheval, J. Stern, and J. Traoré, "On some incompatible properties of voting schemes," in *Towards Trustworthy Elections*, pp. 191–199, 2010.

[5] J. D. Cohen and M. J. Fischer, *A robust and verifiable cryptographically secure election scheme*, Department of Computer Science, Yale University, 1985.

[6] J. C. Corena and J. A. Posada, "Multiplexing schemes for homomorphic cryptosystems," *Elementos*, vol. 1, no. 1, 2013.

[7] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *European Transactions on Telecommunications*, vol. 8, no. 5, pp. 481–490, 1997.

[8] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Workshop on the Theory and Application of Cryptographic Techniques*, pp. 10–18, 1984.

[9] C. Gentry, *A Fully Homomorphic Encryption Scheme*, PhD thesis, Stanford University, 2009.

[10] V. George and M. Sebastian, "An adaptive indexed binary search tree for efficient hom om orphic coercion resistant voting scheme," *International Journal of Managing Information Technology*, vol. 2, no. 1, pp. 1–9, 2010.

[11] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pp. 365–377, 1982.

[12] H. Hussien and H. Aboelnaga, "Design of a secured e-voting system," in *International Conference on Computer Applications Technology (ICCAT'13)*, pp. 1–5, 2013.

[13] A. Huszti, *A Homomorphic Encryption-based Secure Electronic Voting Scheme*, Faculty of Informatics, University of Debrecen, Hungary, 2011.

[14] I. Jabbar and S. N. Alsaad, "Using fully homomorphic encryption to secure cloud computing," *Internet of Things and Cloud Computing*, vol. 4, no. 2, pp. 13–18, 2016.

[15] M. F. Mursi, G. M. Assassa, A. Abdelhafez, and K. M. Abo Samra, "On the development of electronic voting: A survey," *International Journal of Computer Applications*, vol. 61, no. 16, 2013.

[16] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, 1999.

[17] C. Porkodi, R. Arumuganathan, and K. Vidya, "Multi-authority electronic voting scheme based on elliptic curves.," *International Journal of Network Security*, vol. 12, no. 2, pp. 84–91, 2011.

[18] R. Rivest, S. Ledlie, et al., *Lecture Notes 15: Voting, Homomorphic Encryption*, 2002.

[19] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[20] E. Saleh, "Processing over encrypted data: Between theory and practice," in *Proceedings of the 8th Ph. D. Retreat of the HPI Research School on Service-oriented Systems Engineering*, vol. 95, p. 163, 2015.

[21] M. A. Will, B. Nicholson, M. Tiehuis, and R. K. Ko, "Secure voting in the cloud using homomorphic encryption and mobile agents," in *IEEE International Conference on Cloud Computing Research and Innovation (ICCCRI'15)*, pp. 173–184, 2015.

[22] X. Yi and E. Okamoto, "Practical internet voting system," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 378–387, 2013.

[23] Y. Zhao, Y. Pan, S. Wang, and J. Zhang, "An anonymous voting system based on homomorphic encryption," in *10th International Conference on Communications (COMM'14)*, pp. 1–4, 2014.

Biography

Ihsan Jabbar received his bachelor degree in Computer Science from the College of Science/University of Al-Mustansiriyah in 2013. And his Master of Science degree in Computer Science from the same college.

Saad Najim Alsaad is a professor in Computer Science department at the College of Science/ University of Al-Mustansiriyah / Iraq. He is working as teaching staff member more than 20 years. He is interested in research domains such as: speech processing, information security, and object oriented software engineering. He is now working as Editor in Chief in Al-Mustansiriyah Journal of Science.