A STUDY ON SOME VERIFIABLE ELECTRONIC VOTING SCHEMES: DYNEVOTA, END-2-END VERIFIABLE VOTING AND EFFICIENT RECEIPT FREE VOTING

OMAR AL-GBURI

JANUARY 2015

A STUDY ON SOME VERIFIABLE ELECTRONIC VOTING SCHEMES: DYNAVOTE, END-2-END VERIFIABLE VOTING AND EFFICIENT RECEIPT FREE VOTING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
OMAR AL-GBURI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
MATHEMATICS AND COMPUTER SCIENCE\INFORMATION
TECHNOLOGY PROGRAM

JANUARY 2015

Title of the Thesis: **A Study on Some Verifiable Electronic Voting Schemes: DynaVote, End-2-End Verifiable Voting and Efficient Receipt Free Voting**

Submitted by **Omar AL-GBURI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Billûr K

Prof. Dr. Billur KAYMAKÇALAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. A. Nurdan SARAN
Supervisor

Examination Date: 26.01.2015

Examining Committee Members:

Assist. Prof. Dr. Özgür Tolga PUSATLI	(Çankaya Univ.)
Assist. Prof. Dr. Elif SAYGI	(Hacettepe Univ.)
Assist. Prof. Dr. A.Nurdan SARAN	(Çankaya Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I wish to declare that it has been getting the information in this document is in accordance with the rules, academic and moral behavior. I would also like to announce that the conduct and rules that are not original have labeled to sites and sources.

Name, Last Name: Omar, AL-GBURI

Signature :

Date : 26.01.2015

ABSTRACT

A STUDY ON SOME VERIFIABLE ELECTRONIC VOTING SCHEMES: DYNAVOTE, END-2-END VERIFIABLE VOTING AND EFFICIENT RECEIPT FREE VOTING

AL-GBURI, Omar
M.Sc., Department of Mathematics and Computer Science\ Information Technology Program
Supervisor: Assist. Prof. Dr. A. Nurdan SARAN

January 2015, 53 pages

In the past decade the electronic voting has been developed of being a way of counting the votes. The process of electronic voting or voting remotely (voting via the Internet) has become one of the interesting research topics, has been built on the provision of many of the regulations that have been implemented to a vote-mail or remote voting. Remote voting systems are more vulnerable to attack by outsiders than electronic voting systems in controlled environments. The remote voting scheme should provide the accuracy and security of the system and also ease of use and coercion (and other things will be mentioned later). In this study, a theoretical study of the electronic vote is given and also countries that have used electronic voting are listed, also barriers and requirements facing the process of designing the electronic election systems have been provided. In addition, a brief background of some encryption algorithms and systems which are used for the purpose of the design of electronic voting systems are given. Some models of schemes (in terms of cryptographic algorithms) have been designed for the purpose of electronic voting are described and the success and failure of each of them are given. The three electronic voting systems, have been described in details and analyzed in terms of success and also shortcomings of these systems. In this study, these three schemes have been chosen since the authors of these three schemes claim to satisfy different aspects of electronic voting requirements . To satisfy one requirement, one may compromise on other requirement. By this way, requirements of electronic voting may be examined.

Keywords: E-voting, Anonymity, Mix-nets, Blind Signature, Homomorphic Encryption, Voter Verifiability.

ÖZ

DOĞRULANABİLİR ELEKTRONİK OYLAMA YÖNTEMLERİ ÜZERİNE BİR ARAŞTIMA: DYNAVOTE, İKİ YÖNLÜ DOĞRULANABİLEN OYLAMA VE VERİMLİ ALINDI ONAYLI OYLAMA

AL-GBURI, Omar
Yüksek Lisans, Matematik-Bilgisayar Anabilim Dalı\Bilgi Teknolojileri Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. A. Nurdan SARAN
Ocak 2015, 53 sayfa

Son on yılda elektronik oylama oy verme sürecinde kullanılan bir metot olma yolunda gelişme sağlamıştır. Elektronik oylama ya da uzaktan oylama(internet ile oylama) usulü ilginç araştırma başlıklarından biri haline gelmiş, posta yoluyla oylama ya da uzaktan oylama için uygulanmış olan düzenlemelerin çoğunun temin edilmesi üzerine inşa edilmiştir. Uzaktan oylama sistemleri dışarıdaki grupların yabancıların saldırısına uğramak açısından kontrollü çevrelerdeki elektronik oylama sistemlerinden daha kırılgan bir yapıya sahiptir. Uzaktan oylama düzeninin sistemin hatasız bir şekilde işleyişini ve güvenliğini sağlaması ve aynı zamanda kullanımını ve baskıyı (ve ileriki bölümlerde bahsedilecek diğer hususları) kolaylaştırıp rahatlatması gerekmektedir. Bu tezde, elektronik oy ile ilgili teorik bir çalışma verilmekte ve aynı zamanda elektronik oylama sistemini kullanan ülkeler sıralanmakta olup, bilahare elektronik oylama sistemlerinin karşılaşmakta olduğu gereksinimler şartlar ve güvenlik konuları da sunulmaktadır. İlaveten, elektronik oylama sisteminin amacı doğrultusunda kullanılan bazı şifreli algoritmalar ve sistemlerin geçmişleri de kısa bir şekilde anlatılmaktadır. Elektronik oylama amacı ile tasarlanmış olan bazı proje modelleri (şifreleme algoritmaları açısından) tarif edilmekte ve her bir modelin başarılı yönleri ya da uğradığı başarısızlıklar anlatılmaktadır. Üç elektronik oylama sistemi arasında bir karşılaştırma yapılmış olup, burada bahis konusu algoritmalar detaylı bir şekilde tarif edilmiş ve sistemlerin başarılı ve başarısız yönlerinin bir analizi yapılmıştır. Bu çalışmada, bu üç elektronik oylama sistemi seçilmiştir çünkü bu üç elektronik oylama sisteminin yazarları, yöntemlerinin elektronik oylamanın sağlamak zorunda olduğu farklı gereksinimleri karşıladıklarını ifade etmişlerdir. Gereksinimlerden birini karşılamak için diğerinden ödün vermek gerekebilir. Bu yolla, elektronik oylamanın gereksinimleri irdelenilebilecektir.

Anahtar Sözcükler: E-oylama, Anonimlik, Karıştırıcı-ağ, Kör İmza, Homomorfik Şifreleme, Seçmen İzlemesi.

ACKNOWLEDGEMENTS

Thanks to God the most compassionate and the most merciful. My Allah’s mercy and peace be upon our leader Mohammed, who invites us to science and wisdom, and members of his family and his followers.

I would to express my deep gratitude after God almighty in the completion of this research to my supervisor Dr. Nurdan Saran who suggested this project and gave me a lot of her time. I am indebted for her suggestions and valuable remarks.

Finally, my thanks go to the members of my family for their help and encouragement, and to everyone who helped in one way or another in bringing out this work.

My God bestow health and happiness to all of them .

TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM... iii

ABSTRACT... iv

ÖZ…………………………………………………………………………….. v

ACKNOWLEDGEMENTS……………………………………………………….. vi

TABLE OF CONTENTS…………………………………………………………. vii

LIST OF FIGURES……………………………………………………………... ix

LIST OF TABLES……………………………………………………………… x

LIST OF ABBREVIATIONS……………………………………………....... xi

CHAPTERS:

1. INTRODUCTION.. 1

1.1 E-voting Requirements... 2

1.2 E-voting in the World..……………………………………………… 4

1.3 Motivation………………... 7

2. E-VOTING SCHEMES……………………………………………………. 9

2.1 Overview of E-Voting Schemes ……….. 9

2.1.1 Blind Signature Scheme.. 9

2.1.2 Mix Network Scheme... 10

2.1.2.1 Decryption Mix Network Scheme………………… 11

2.1.3 Homomorphic Encryption…….. 14

2.2 CRYPTOGRAPHIC PRIMITIVIES………………………………….. 14

2.2.1 RSA Cryptosystem……………………..………………… 14

2.2.2 ElGamal Cryptosystem……………………..……………….. 18

2.2.3 Hash Function………………………….…….…………… 19

2.2.4 Zero-Knowledge Proof…………………………………….. 20

2.2.4.1 Properties of Zero-Knowledge Proof……………… 21

2.2.4.2 Advantage of Zero-Knowledge Proof…..………… 21

2.3 Literature……………………………………………………………….. 21

3. ON SOME UNIVERSAL VERIFIABLE SCHEMES.................................….. 26

3.1 Universal Verifiability…………………………………………………... 26

3.2 Brief Description of Algorithms……………………………………… 27

3.2.1 DynaVote ………………... 27

3.2.2 End-2-End Verifiable Voting……………………………….. 36

3.2.3 Efficient Receipt Free Voting.................……………...….. 41

3.3 Comparison of Algorithms.. 43

3.3.1 Authentication and Authorization Phase …....……………... 43

3.3.2 Voting and Submitting Phase ……………………………… 45

3.3.3 Counting Phase......................................………………….... 46

3.4 Robustness Against Attacks 49

3.5 Comparison According to Cryptographic Primitives Used.................. 50

3.6 Comparison According to Security Requirements that Satisfied.......... 50

4. CONCLUSION.. 52

REFERENCES... R1

APPENDICES... A1

A. CURRICULUM VITAE.. A1

LIST OF FIGURES

FIGURES

Figure 1 The General Requirements of any Voting system.................... 8

Figure 2 The Blind Signature Scheme……………………………….……. 10

Figure 3 The Anonymization Component in a Voting Process………… 11

Figure 4 Visualization of The structure of a Typical n-layered Onion… 12

Figure 5 Explanation of Onion Decryption……………………………. 13

Figure 6 Overview of the Voting Stage..……………………………….. 30

Figure 7 Overview of the Counting Stage.……………………………... 33

Figure 8 The Voting Schemes……………………………………...... 47

Figure 8 The Counting Schemes….……………………………………. 48

LIST OF TABLES

TABLES

Table 1 Countries that Have Used Internet Voting..…………….…….. 6

Table 2 RSA Cryptosystem Scheme …………………………………. 15

Table 3 ElGamal Cryptosystem Scheme ……………………………… 19

Table 4 Schemes and Their Primitives…………………………………. 24

Table 5 The Requirements Achieved by the Schemes..………………... 25

Table 6 The Authentication and Authorization Stages…………………. 44

Table 7 The Schemes and Their Primitives…………………………….. 51

Table 8 The requirements Achieved by the Schemes…………………. 51

LIST OF ABBREVIATIONS

ACM	Association for Computer Machinery
AES	Advanced Encryption Standard
HMAC	Hash-based Message Authentication Code
MAC	Message Authentication Code
NIST	National Institute of Standard and Technology
RPI	Return Path Information
SHA	Secret Hashing Algorithm

CHAPTER 1

1. INTRODUCTION

In recent years there are a growing preference on a large scale to carry out the election using of electronic media. Forms of this growing have been replaced by the process of voter registration using the punch card and scanning systems to process the use of direct electronic registration machines. At the same time popped increased use of mechanical and electronic media. Mechanical was served in the voting process has been used in the early 1980 with the invention of Herman Hollerith punch card machinery that used in the census in the United States and then to process of electronic voting [1]. Electronic voting (E-voting), it is a term that encompassing several different types of voting, includes electronic means of casting a votes and electronic means of counting votes. The area of electronic voting has grown used of a paper ballot market choice early in United States. A paper ballot marked choices, had evolved development the field of electronic voting. Electronic voting has evolved through years of use Punch Cards to the development of optical scanning systems and then to specialized compartments of the electronic voting process. The process of transferring votes has been developed to, from transferring votes through networked computers or phones and most recently process is voting through the internet. The technology of electronic voting in polling stations can accelerate the process of sorting the ballots, and provide improved access voters with disabilities. The voting system can provide remote voting via the web to improve process and availability of the voting process even more convenient. Also it may lead to increase voter turnout in the elections [2]. The basics primitives that motivated the electronic voting systems frontward is the accuracy and security. But the obstacles facing the proliferation of systems on a large scale is the fear of manipulated or rotten programs. The number of issues which represent by threats and attack lead to corrupt network computers and system components.

Also the field of cryptographic technology has evolved. Online voting systems and new generation in developing classified as cryptography systems to vote, not just for saving adequate secrecy, but to provide verification capabilities too. Josef Stalin said:" Those who cast the votes decide nothing, but those who count the votes decide everything". The perseverance of encrypted voting systems to identifying and preventing the registration process and counting incorrectly. Electronic voting systems should also include the possibility of re-counting the votes, when an error arose, as paper based elections [3].

1.1 E-voting Requirements

The electronic voting systems faced challenges raised by the functional requirements and constitutional ruled by the country in which they are operate. An electronic voting systems must respect the constitutional principles of the elections. For the technical solutions, this translates into the safety requirements that must be met by the operating environment where the voting process takes place. Some requirements are essential for every voting system such as privacy, reliability, uniqueness, verifiability, accuracy and authentication. Additional requirements are desirable like scalability, transparency, cost effectiveness and convenience. The security requirements of the voting systems has been described by many researchers [4-5], and display these needs through the following formal definitions:

- Eligibility: Eligible voters can participate in elections.
- Uniqueness: Voter cannot vote more than once. No one is able to change or duplicating a ballot from another voter.
- Privacy: There are no party or person can connect the ballot to the person who had made. The privacy of the voter must be kept during and after the election for a long period.
- Convenience: The voter must be able to cast his vote with less equipment and expertise. This include the elimination of all physical limitations, and reduce the necessity of learning too complex techniques.
- Transparency: Bulletin board may be used for the deployment of the election process. Security and reliability of the system should be assured.

- Practicality: The election scheme must not contain any assumptions difficult to implement in a wide range elections.
- Fairness: It is not allowed to declare preliminary results of the election prior to the end of the election process to make sure that all candidates are given a just decision , even the authorities of the vote should not have any idea about the results of the votes.
- Incoercibility: Voter must get freedom to vote. No one even the authorities have the right to extract the value of the vote or force the voter to vote in a particular way.
- Accuracy: Voting system must calculates all the votes correctly. Any vote cannot be deleted, nor invalidated, nor altered.
- Verifiability: The voter must be sure that his vote has been calculated correctly in the final tally (individual or universal verifiability).
- Receipt-freeness: The inability to know what the vote is. Voter cannot take a receipt to prove his vote to external party during or after the election. In order to prevent the sale and purchase of votes.
- Robustness: Any number of authorities or parties don’t disable or impact of the elections and the final results. Robustness should be assured to possess confidence in the election results.
- Integrity: We can divide the integrity into two parts software and data integrity. Software integrity is to make sure that only genuine programs is working. Data integrity is maintaining integrity of the voting information and verification of the records.

Few disputes in the demands made the definition. The disputes example is the Authentication vs. Privacy, for identifying and verify the credentials of voters and; at the same time protect the privacy of his/her ballot. Another example is the Verifiability vs. Receipt Freeness to enable the voters to verify that their ballots have calculated correctly and properly cast without providing a receipt from the authority right to cast a vote. In the view of the short history for the electronic voting systems in all parts of the world and the inherent limitations in the scope of the implementation, it is quite difficult for us to measure the success or failure of any of the above issues, or all of them. Furthermore, any voting process will be mentioned later involved committed by the rules and cultures that belong to different communities. Thus, the example of a single country may not fit directly with the example of another country [6].

1.2 E-voting in the World

Electronic voting have been put on electronic voting experiences (Table 1) maps operations, according to the results of a global Studies [7]. Standards machines that are used in electronic voting have been collected in thirty countries for use in political elections. The criteria of electronic voting have met in thirty countries to use electronic voting machines in the voting binding of a political nature.

The following states (Italy, Ireland, UK and Norway) have stopped using electronic voting. Germany and the Netherlands have stopped the use of electronic voting machines. Belgium and France are the two countries that are using the electronic voting machines till now [7].

Electronic voting in France has been used in a limited a number of towns. US presidential election of 2000 has been affected by the limited use of electronic voting processes, as well as the decision of some European countries, such as Germany, Ireland and the Netherlands will not continue with the electronic voting because the basis of the security and transparency concerns.

There are two examples about the use of electronic voting in the national elections in South Africa (Venezuela and Brazil) and one in Asia (India).

Brazil and India are using electronic voting for decade and they have important benefits in the quality of the election results. This does not mean that the use of electronic voting machines have not faced opposition in these countries but, the opposition failed to gather a lot of momentum.

Electronic voting processes were not used in the African continent as a result of the large number of problems, including the financial problems of these countries. Some states are struggling to get the money and efficient infrastructure for the purpose of the success of electronic voting processes. Not all of the African countries are weak for the investment, nevertheless, there are other reasons faced electronic voting to make their way to Africa [8]. The infrastructure of Electronic voting relies on some features such as electricity and telecommunications, which are not available in the national level in many African countries [9].

The following brief is describing and highlighting on the interesting aspects for each case.

- Belgium- electronic election started in 1991. Used widely term in the general and municipal elections as well. It is based on two systems-named Jites and Digivote. Both systems had characterized as “indirect recording electronic voting systems” because the machine does not record and tabulate the vote directly [10]. Jites and Digivote systems have used Cardboard for the purpose of recording the vote either magnetic tape has been used to schedule the vote. Differences are resolved by re-calculate the cards by the machine [11].
- Brazil electronic voting started in 1996. The first test had been in Santa Catarina state. All elections in Brazil had become electronically since 2000. In the elections that held in 2000 and 2002, over 400 thousands electronic voting machines has been used and the results calculated electronically in minutes after the polls closed [12].
- France used remote voting in elections for the first time in the year 2003, so the French citizens who live in the United States have selected their candidates to Association of French Citizens Overseas. More than 60% of the voters have voted online rather than the paper based ballot.

Country	region
Argentina	Currently used in some parts of the country
Australia	Piloted and Not Continued
Belgium	Currently used in parts of the country
Bangladesh	Pilots Ongoing
Bhutan	Pilots Ongoing
Brazil	Currently used Nationwide
Canada	Currently used in some parts of the country
Costa Rica	Piloted and Not Continued
Ecuador	Pilots Ongoing
France	Currently used in parts of the country

Germany	Discontinued
Guatemala	Piloted and Not Continued
India	Currently used Nationwide
Indonesia	Pilots Ongoing
Ireland	Piloted and Not Continued
Italy	Piloted and Not Continued
Japan	Currently used in parts of the country
Kazakhstan	Piloted and Not Continued
Mexico	Currently used in some parts of the country
Magnolia	Pilots Ongoing
Nepal	Pilots Ongoing
Netherlands	Discontinued
Norway	Piloted and Not Continued
Paraguay	Discontinued
Peru	Currently used in some parts of the country
Philippines	Piloted and Not Continued
Russia	Pilots Ongoing
United States	Currently used in parts of the country
Venezuela	Currently used Nationwide
United Kingdom	Piloted and Not Continued

Table 1 Countries that have used Internet Voting.[1]

An article was published on the Internet Rights Forum with regard to the election process that the French who live overseas must vote electronically for the benefit of Association of French citizenship [13]. In 2009 this recommendation became reality, 6000 French citizens have voted through the system [14].

- India- There is no country has used the electronic voting system in a wide range elections such as India. Since India currently the second largest country in terms of population in the world, so it is natural to become the largest number of voters in the democratic process. Indian Electronic voting used for the first time in 1982 in Parur assembly constituency in Kerala State.

[1] Data presented in table 1 has been collected from many different sources, including sources such as reports from election management bodies, election management body

The electronic voting mechanisms used during the national elections that held for the Indian parliament in 2004 and 2009. In accordance to statistics available for the Indian parliamentary elections in 2004 and 2009 it had participated in the elections about 60% of Indian eligible voters either the counting process it took only a few hours [12].

1.3 Motivation

In recent years the electronic voting became so popular search and hot topic. It has been found electronic means of counting the ballots, but the focus now is more on how to cast their ballots online. There are many benefits behind the electronic voting process including increase the speed up of counting process as well as improve turnout of voters with disabilities. The machines of direct recording and voting across the internet also reduce the use of ballot papers and manual work of preparation. Validation of such systems too exaggerated, also since the registration and counting are done electronically, there are no way for mistakes.

The electronic voting process especially across Internet can facilitate portability of counterfeiting. Figure 1 has shown the general requirements of any voting system. The online system more subject to threats; fraud, compromised computers and other attacks being waged across networks. The need to improve tools of integrity, confidentiality, anonymity, authentication and other requirements is extremely important. Recently we has seen the development of systems for electronic voting processes needed to develop encryption algorithms to protect against these attacks and verification purposes alike. The probability of verification of many steps of operation are also an essential way to discover and cope with many of the attacks.

NIST had a workshop on what is called end-to-end voting system verifications [15]. This voting system may provide the voter with possibility of the verification of the votes cast during the entire process. These systems have discussed widely and considered a great evaluation of the process of electronic voting, but these systems can lead to possible defects, such as the problem of coercion. Since the subject of protection of electronic election is increasing interest. This thesis includes the study of the security technologies used in the process of electronic voting. It will include the solution and technologies that are using to provide the security issues in the electronic election, the requirements of the electronic voting also will include experiences of electronic voting around the world. Finally, there will be a comparison between stages of electronic voting through different experiences [15]. As similar, but still significant the modern cryptographic techniques of homomorphic encryption (using El-Gamal), Blind Signature, Mixnet and other cryptographic techniques being developed for e-voting will be presented[16].

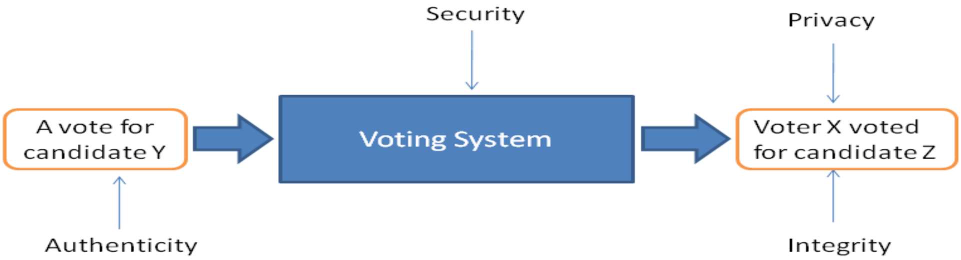

Figure 1 The General Requirements of Any Voting System [15]

To gain greater understanding and get some hands on experiences, also examine three algorithms whether they are satisfying the requirements of electronic voting or not.

CHAPTER 2

2. E-VOTING SCHEMES

2.1 OVERVIEW OF E-VOTING SCHEMES

Electronic voting system consists of conceptual design, which is called electronic voting system. So, the electronic voting systems built on electronic voting scheme which is the essence assurance system to meet the requirements. Most of them using the principles of cryptographic mechanisms. For more than two decades research on this topic (e-voting) have been done [17]. Chaum presented the first e-voting protocols in 1981. He started his huge amount of researches on many ways to achieve safe results to process of electronic voting. This approach has seen the efficiency of successful of practical application significantly over recent years. They had provided hereafter the most important classes and schemes of the encryption mechanisms which forms the protocols units of the election scheme [18].

2.1.1 Blind Signature Scheme

A blind signature had been introduced by Chaum [19]. It allows a person to get another person signature to sign a message without revealing any information about the content of the message. Figure 2 illustrate the blind signature scheme.

Blind signature is equivalent to signing a paper-lined envelopes carbon. Writing the signature on the outside of the envelope, such as leaving a carbon copy of the signature on a piece of paper inside the envelope. When the envelope is opened, the slide will show the signature of the carbon image [20].

This cryptography technique is the most popular in the e-voting scheme through the provision of a secret ballot voter [21]. The blind signature is used to authenticate the voters without revealing the contents of the ballot. Thus, the authority which task is to verify the eligibility of voters do not know whom the voters vote for. In order to achieve the confidentiality requirement in electronic voting the ballot must be blinded.

The voters ought to get the signature of the auditor when he votes. To ensure the voting secrecy voter cast a ballot. A ballot blinds using random number and sends it to the authentication [20]. We can restated the blind signature based on RSA (section 2.2.1) as following. (e, n) is the signer public key and (d, n) is the signer private key.

1) At the beginning the provider must generate a random number which is , then calculates: $x = r^e\ m\ mod\ n$. The provider sends x to the signer.
2) The signer receive x and sign it: $t = x^d\ mod\ n$. After that, sends t to the provider x
3) Then, the provider reads t. Since $t = x^d\ mod\ n$, since $t = (r^e\ m)^d\ mod\ n = r^{ed}\ m^d\ mod\ n = r\ m^d\ mod\ n$
4) The provider calculates $s = t\ r^{-1}\ mod\ n = md\ mod\ n$

This is the signing formula for m. Since m is blinded by using blinding factor of r, signer cannot know the contents of m [22].

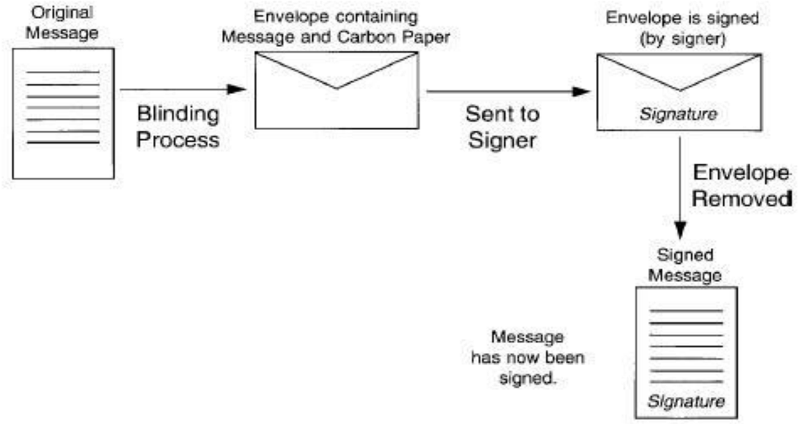
Original Message
Blinding Process
Envelope containing Message and Carbon Paper
Sent to Signer
Envelope is signed (by signer)
Signature
Envelope Removed
Signed Message
Message has now been signed.
Signature

Figure 2 The Blind Signature Scheme [22]

2.1.2 Mix Network Scheme

It is a method of encryption works on the basis of the public key, A mix net is a method based on public key encryption, which includes mixing of the components between the opposite sides using encryption process. The process of sending all data prior to delivery [23]. A mix net encrypted data takes as input quirks and reorganize data and decrypt them. One of the main purposes of using a mix net is to pelt the communications among the elements of input and production. The idea of a mix net invented by Chaum in the early 1980s which based on the bases of encryption overlapping. It is consist several types of mix net exist based on re-encryption. Figure 3 shows how it can be involved an element of mix net in the voting to guarantee anonymity and secrecy [23].

2.1.2.1 Decryption Mix Network Scheme

Mixnet decryption category required that the sender to use the keys of the stages in order to send the message that he intends to send it. Thus, to decrypt the message using the secret key for each stage all this is declared using public key algorithms such as RSA. It is also possible to use of symmetric key cryptosystem, when the sender subscribe key with each of the mixnet stage.

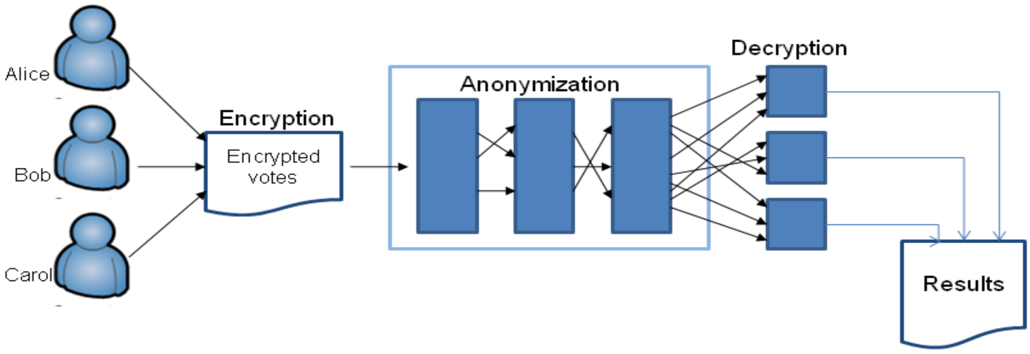

Figure 3 The Anonymization Component in a Voting Process [23]

Decryption mixnet just using the public keys of the stage in anonymous connection process as:

$Forwarding\ onion_n = E_{k_x}(m)||r_n,$

$Forwarding\ onion_{n-1} = E_{k_n}(A_x||forwarding\ onion_n||r_{n-1},$

:

$Forwarding\ onion_j = E_{K_{j+1}}(A_n||forwarding\ onion_{j+1})||r_j,$

:

$Forwarding\ onion_1 = E_{K_1}(A_2||forwarding\ onion_2)||r_1,$

$E_K(m, r) = A_1||forwarding\ onion_1$ (Eq. 1)

Where $K = K_1, K_2, K_3, \dots, K_n)$ represent the public key of each of the n stages, $A_1, A_2, A_3, \dots, A_n$ represent the address of the n stages, $r_1, r_2, r_3, \dots, r_n$ represent the random strings to randomize the encryption of each stage, A_x represent the receiver address and K_x represent the public key of the receiver. The amount of output in (Eq. 1) represents the n-layered of the onion [23], through onions shipping nested in all of the n-layer, which is transmitted by the transmitter is explained in Figure 4. The transmitter with onion formula in (Eq. 1) can be given as:

$$E_K(m,r)$$
$$= A_1||E_{K_1}\left(A_2\left|\left|E_{K_2}\left(A_3 \dots E_{K_{n-1}}\left(A_n\left|\left|E_{K_n}\left(A_x\left||E_{K_x}(m)|\right|r_n\right)\right)\right|\right|r_{n-1}\right)\dots\right||r_2)\right)\right|\right|r_1\right). \quad (Eq.2)$$

Figure 5 illustrate graphical clarification of the decryption that is made on the transmitter on onion by the mixnet. In every phase of the path shells of the layer of the onion, i.e., decryption operation by using the private key of each stage K_j^{-1} as:

$D_{K_j}(E_{K_j}(A_{j+1}||forwarding\ onion_{j+1}$ (Eq. 3)

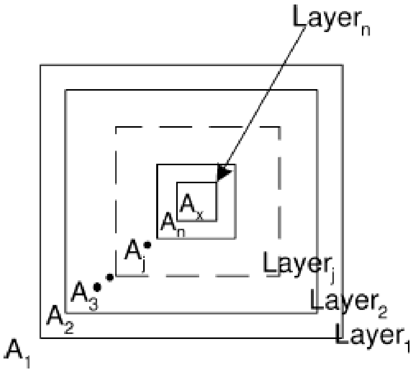

Figure 4 The Visualization of The Structure of a Typical n-layered of Onion (within every layer we discovery the title of the next stage, and also the orientation of the onion, and additional control information [23]).

After decrypting more onions that has received (from transmitters or from other phases), the phase j applies random permutation on them. $.i.e.\pi_j: l \rightarrow l$, 1 is the batch size). Decryption Mix net include mixing operation by stage j. Generated quantities represent the forwarding onions that have reduced in the size and form the mixed output batch of stage j. Then, this onion is forwarded at one time to the next stage of their respective. The next stage for the sender is represent the address A_{j+1}. The mixing operation is continuous process until access to the last stage n. The decrypting quantity from stage n $(EK_x(m))$ is sending to the receiver that has the address $\mathrm{A_x}$. The previous result represents the decrypting mix net in one way anonymous communication.

In decrypting mixnet two-way anonymous communication, when the receiver has to answer the unknown sender, it must include return path information (RPI) with a key $\mathrm{k_s}$, along with m. The RPI looks like the formula of the sender onion in (Eq. 2), but here only the address of the sender A_s will be encrypted through the onion as follows:

RPI= $A_n||E_{K_n}\ (k_n||A_{n-1}||$
$E_{K_{n-1}}(k_{n-1}||A_{n-2}||\ldots E_{K_2}(k_2||A_1||E_{K_1}(k_1||A_s||r'_1))||r'_2)\ldots||r'_2)\ldots||r'_{n-1}||r'_n$
(Eq. 4)

Where $r'_1, r'_2, r'_3, \ldots, r'_n$ are the random strings and $k_j, j = 1,2,3, \ldots, n$ are the symmetric keys, so, the receiver at A_x will obtain the following from mixnet: $E_{K_x} = (m||\mathrm{RPI}\ K_s)$ (Eq. 5)

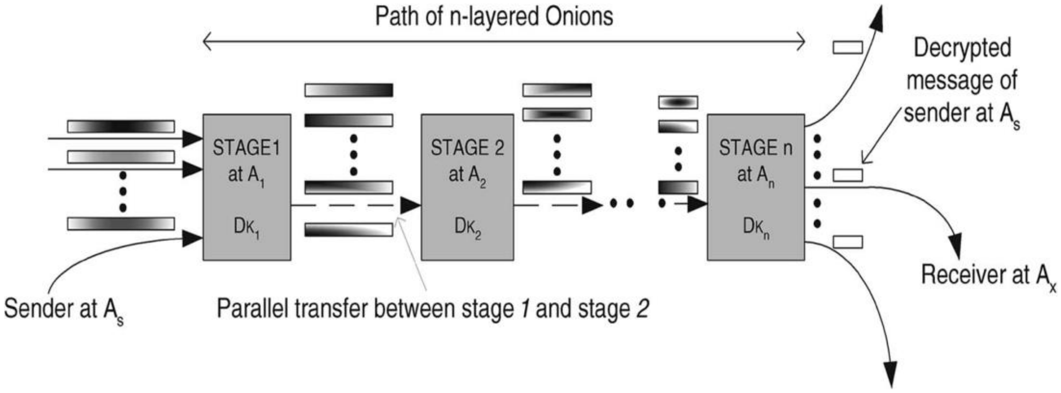

Figure 5 Explanation of Onion Decryption, the size of the onion indicate their size. The onion size and its decreases are traversed. The decrypted message that received by A_x can not be linked back to the sender at A_s [23].

After decryption the message the receiver may send a reply to the sender as follows: $E_{K_s}[m]||RPI$ (Eq. 6)

Thus, stage n will receive the encrypted response and peels the attached RPI onion from its layer to get k_n, and then, re-encrypts $E_{K_s}[m']$ with k_n to change its appearance. The rest of the mixing process happiness as it happened before in the path forward from the sender of the message to the receiver.

Lastly, stage 1 transmit the re-encrypted response to the transmitter at A_s. Therefore, we can achieve the two-way anonymous communication by using decryption mxinet.

Note it is unnecessary for the return and forwarding route to include the similar stage [22][23].

2.1.3 Homomorphic encryption

Values of homomorphic encryption can find a set of encoded values without the need to decrypt them. We can say that encryption function $E()$ is homomorphic, if it's possible after the given $E(X)$ and $E(Y)$ to get $E(X\perp Y)$ $E(X\perp Y)$ without making decryption of X or Y with some operation $\perp$.

$$E(X)\perp E(Y) = E(X\perp Y).$$

ElGamal is one of the functions that have the characteristics of multiplicative homomorphism. If you encrypt the message m_1, and encrypt another message m_2, and you multiply them together, the result is the encryption product of the two messages.

What is happening to ElGamal if we multiplying the two encrypted messages(m_1 and m_2), that is the randomization will get added up at the exponent and the messages will multiply.

$$E(m_1) * E(m_2) = E(m_1 * m_2)$$

RSA and ElGamal are the two examples of homomorphic cryptosystems [24]. With the RSA for example, after encrypt the plain text P to a cipher text C, you can double C with 2, and then decrypt 2C, and you will obtain on 2P. This is not possible by the normal symmetric cipher like DES and AES, since when multiplying on the cipher text by 2 with AES, and then decrypt it, you will get some random rubbish, not the original text P.

That is all homomorphism properties that are based on the multiplication process because it does not yet certain criteria to standardize homomorphic encryption which includes the addition and multiplication, so that in the voting scenario where the addition is desirable property we can use a variant the ElGamal cryptosystem [25].

2.2 Cryptographic Primitives Used in E-voting Schemes

2.2.1 RSA Cryptosystem

RSA is one of the most popular public key cryptosystem, it is named by the three following developers Ron Rivest (b.1947), Adi Shamir, and Leonard Adleman (b.1977). they have worked in laboratory belonging to computer science named MIT. They have announced their algorithm for the first time in Martin Gardiner "Mathematical Games" in column Scientific American in August, 1977.

They published their paper in 1978 which is named "A method for obtaining digital signatures and public-key cryptosystems" in the Communication of the Association for Computing Machinery. RSA received a patient. Clifford Cocks (British Mathematician) has worked for the CESG and also developed a similar algorithm in 1973 [26]. Nevertheless, there is nothing official indicating that it has been executed. RSA and ElGamal, both of them are public key cryptosystem which we shall study them in details, both depend on the number of theoretical problems. RSA works on the principle that it is not difficult to make the multiplication process on two large prime numbers, but the great difficulty lies on how to analysis the large integer number. ElGamal relies based on the fact that it is not difficult to raise n to the exponent of p, but it is difficult to conclude p base on b. RSA may be decrypted one day if a mathematician found a solution to the factoring problem. Likewise, if they wakes up and find a solution to the discrete logarithm problem, ElGamal encoded

Key Generation: KeyGen (p, q)
Input: Two large primes p, q
Compute $n = p.q$
$\varphi(n) = (p-1)(q-1)$
Choose e such thatgcd$\big(e, \varphi(n)\big) = 1$
Determine d such that $e.d \equiv 1 \; mod \; \varphi(n)$
Key:
public key = (e, n)
secret key= (d, n)
Encryption:
$c = m^e \; mod \; n$
where c is the cipher text and m is the plain text

Table 2: RSA Cryptosystem Scheme[2]

[2] From the International Journal of Advanced Research in Computer Science and Software Engineering. It also available online through this link: www.ijarcsse.com.

message may be deciphered. Search in the theory of numbers can have a significant on the public key cryptosystem in order to avoid "putting all of one's eggs in one basket", cryptosystem in the basis of other than of the numbers of theoretical problems were developed, but are not popular as RSA and ElGamal. RSA based on the number of theoretical result due to the Swiss mathematician Leonard Euler (1707-1783) (pronounced "oiler") [29]. Table 2 is RSA cryptosystem.

In order to construct the key we will need to two large primes p and $q. n = p. q$ is the modulus. Whereas, practically, the modulus is typically 1024 or more, for example e will use too small primes, let's assume $p = 53$ and $q = 61$ (both of them are primes). So, $n = p. q = (53 * 61) = 3233, p$ and q is still secret but n is the public. Security of RSA algorithm relies on the attacker being unable to factor n to the head of the factor n in two of the head of the factors p and q. The encryption exponent which is pointed by e may be any integer having multiplicative inverse module.

$(p - 1)(q - 1) = (53 - 1)(61 - 1) = 52 * 60 = 3120$, by using Euclidean algorithm, the inverse multiplicative of e modulo $(p - 1)(q - 1)$ has been calculated; which represent the decryption exponent d, for our example, $d = 253, d$ is the private key and e is the public key. For our example, we have Alex will send a message to Nick. Alex will receive the public key of Nick, n and e available to anyone who want to send any encrypted message to Nick (i.e. N and e). Nick will keep the secret exponent, which represent Nick's public key. A secret key which is used by Nick decrypt any message have been sent to him, which may encrypted using his public key. If Alex want to send a message to Nick. First, convert the message to a string of numbers formula. Practically ASCII (American Standard Code of Information Interchange) of numbers used usually; our usual area is $a = 00, b = 01, c = 02, \dots, z = 25$. So that, will be converted to a series of math 12001907. The string is divided into blocks, each smaller than the modulus---3233 in our case. Blocks of four digits will be work: 1200 1907.

We will obtain C which represent the cipher text message by raising each block from the plain text m to exponent modulo n.

$m^e \; mod \; n = c$

$1200^{37} \; mod \; 3233 = 2223$

$1907^{37} \; mod \; 3233 = 3126$

Alex will transmit to Nick the message 2223 3126

How Nick will decrypt the message from Nick?

The decryption will depend on Euler's result that$a^{(p-1)(q-1)} = 1\, mod\, pq,\ n = pq$. So that $a^{(p-1)(q-1)} = 1\, mod\, n$. Now$e$ and d are inverse modulo $(p-1)(q-1)$; $ed =\ 1\, mod\ (p-1)(q-1);\ i.e., ed = 1 + k(p-1)(q-1)$, for some integer k. The cipher text is $C = m^e\, mod\, n$. When Nick receives block of cipher text, he raises them all to the power of d, his decryption exponent.

$C^d = (m^e)^d = p^{ed}$

$m^{1+k(p-1)(q-1)} = m\, m^{k(p-1)(q-1)}$

$m(m^{(p-1)(q-1)})^k = m(1)^k = m\, mod\, n$

It is represent the plain text message. Here is a decryption for our example:

$C^d\, mod\, n = m$

$2223^{253}\, mod\, 3233 = 1200$

$3126^{253}\, mod\, 3233 = 1907$

So, it was decrypted to $1200\ 1907$ which was converted to the math formula.

For Beth to cryptanalyze Alex's message to Nick, she should be able to build Nick's private key (d), which is representing the inverse of modulo (p-1)(q-1). Using the extended Euclidean algorithm and she can easily build inverse e if she know p and q, but in order to get p and q, Beth will need to be capable of factor n, that is part of the public key of Nick. This has problem because there is no effective way to factoring a large integers.

Nevertheless, for the future, the factoring of large integers in a polynomial time might be possible then can broke RSA encrypted message. In 1994, Peter Shor (b.1959), then he worked at Bell Labs, found the quantum computer algorithm which would factoring a large integers in the polynomial time. Thankfully, for cryptographers in order to building a quantum computer that would benefit from break the RSA algorithm and an encrypted message not on the horizon; the best effort to date: in 2001, a7-qubit quantum computer was able to using Shor's algorithm to factoring 15. RSA used because it is surviving. RSA been attacked from its development time 1977. The mathematician had improving the factoring algorithms, but cryptographers responded by increasing the sizes of keys. Quantum computer poses a threat, but not directly. The algorithm was tested so thoroughly that there is a sense that the well-known weaknesses-there. For instance, good choices and for p, q and e. But the problems are known to be dealt in the implementation. In particular, the encryption is now out in the open in the universities and not only secretly in the black rooms. The algorithms can be tested publicly by experts.

Coding systems need to gain confidence, and RSA has gain it. But can penetrate the unexpected in factoring algorithms leads to the elimination immediately.

New algorithms should be seen with suspicion. An encryption, which has known vulnerabilities can be better choice of encryption that are not strictly tested. For many years, DES encryption system became closer. Has been replaced by a choice of AES very open manner, which give it a lot of confidence in the primary, but now it has been determined it will be even more carefully examined for weaknesses[27].

2.2.2 ElGamal cryptosystem

Such as RSA, ElGamal is a public key encryption system. You will have two keys public key which is deployed and the private key that kept hidden. The public key is using the encryption and the private key is using for the decryption process. This allows the entity (human or computer) to receiving the encrypted messages from a variety of senders with a reasonable degree of confidence that cannot decrypt the messages sent by any person other than the recipient.

Like RSA, there should be a relationship between the private and the public key when encryption and decryption because the encryption and decryption is an inverse relationship. Security in public key cryptography depends on this relationship, as one which cannot easily be exploited to conclusion the private key, which represent the decryption key of knowledge the public key, which represent the encryption key.

The basic mathematical relationship between encryption and decryption keys in ElGamal (as illustrated in Table 3) is rely on the so-called discrete log problem, which will be clarified later. For sending encrypted message, the plain text must be converted to digital format, like what happiness in RSA, there are numerous concerns about doing this. For our purpose, it is assumed that you have already done this conversion, result will be potentially very large in a positive integer m. We will assume also that $m < p$. Usually, this is the case, if not, you will need to be away agreed to modify the encryption process (perhaps splitting m in some way and then encrypt the message in “packets”) there.

For practice we will encrypt $m = 2$, using ElGamal cryptosystem with prime number $(p) = 139$, generator $(g) = 3$ and private key $(X_A) = 12$. So, the public key $Y_A = g^{X_A}\, mod\, p$, so $Y_A = 3^{12}\, mod\, 139 = 44$. Then, we select random exponent $X_B = 52$. So, $c = g^{X_A}\, mod\, p$. That, $c = 3^{52}\, mod\, 139 = 112$. Thus, $ciphertext = m * s, CipherText = (2 * 112\, mod\, 139) = 85$. If we want to decrypt the cipher text $(CipherText)$, we will compute R. $R = (CipherText)^{p-1-X_A} mod\, p = 36. Decryption = (36 * 85) mod\, 139 = 2$, which, is equal to original m. ElGaml encryption scheme is invented in 1985 by Taher ElGamal [28].

Generate a pair of key:

- Select a large prime p and the generator g of a multiplicative group Zp $*$ of the integers modulo p .
- Select an integer X_A from the group Z by random and with the constraint $1 \leq X_A \leq p - 2$.
- Private key: X_A
- Public key: $Y_A = g^{X_A} \bmod p$

Encryption procedure:

- Select a random exponentX_B
- Compute $c1 = g^{X_B} \bmod p$ and combine it with the cipher text that will be sent to the receiver.

Table 3: ElGamal Cryptosystem Scheme[3]

2.2.3 Hash function

Has function is fragmentation the message to different sizes of data to the output of a fixed size via one-way mathematical function [28]. The result of this fragmentation is the retail value of the tag data and called hash code or value . Hash function, works by taking the message as input then produce a code that represents the function of each bit of the message. It can be visualized as a certification of the

[3] From International Journal of Advanced Research in Computer Science and Software Engineering. It also available online through this link: www.ijarcsse.com.

message , since, if the value of any bit in the message changed, the value of the hash director going to be changed [29].

Hash code is one value extract by several parties from one data element, but it is difficult to extract multiple hash values from one type of data. So any party may reproduced the hash value or the message digest with the same stream of data. Therefore, we can get the integrity through the use of hash value. If a message has been sent from Alice to Bob and Alice wanted to be sure that his message has been received correctly, Alice must send the message with its digest. Bob to makes sure that the message that has been received coming from the right place he should be used the message and secure hashing algorithm an calculates the digest of the message again and compares it with that received. All of these provide protection against accidental changes in the data, but does not provide protection in the event that one of them intercepted Alice's message and changed the original text with digestion [28]. In the case of protection against interception hash is used in addition to the common secret key to create a hash value based on the authentication code which is named HMAC be similar to MAC. Thereafter, Alice sends the message to Bob with HMAC, Bob then calculates the HMAC to protect against changes that can get the message and no one can intercept or tempered with it because no one knows the secret key. The standards that used are the Secure Hash Algorithms (SHA256, SHA1, SHA512, SHA384) [30].

2.2.4 Zero-Knowledge Proof

Rackoff, Micali and Goldwasser the first to put the idea of Zero-Knowledge Proof in the year 1985. Interactive Guide Protocol is ratified prover to verifier busing mechanical challenge and response. In this type at the end of the contacts the verifier can accept or discard the prover.

Zero-Knowledge overcomes the major fears that ratify on passwords dramatically. The authentication that relying on a password, verifier approves prover based on password only. Verifier has a simple knowing, if they were not full-on prover passwords. So that, the can participate prover password to a third party. The main goal of a zero-knowledge protocol is to convince the prover that he will not participate any information about the secret that knows itself. ZK protocol gives great possibilities for success and not necessarily to be an absolute success. So, after exchanging several messages (each one of them containing challenges and responses) verifier can accept or reject the proof. Increasing the number of challenges and responses can reduce the likelihood of errors to a reasonable level [31].

2.2.4.1 Properties of Zero-Knowledge Proof

Zero-Knowledge Proof specifications derived from cooperative resistant procedures:

1) Completeness: The probability of success and failure of the protocol is determined by the amount of confidence by the prover and the verifier. Probability smooth of acceptance varies from one application to another.
2) Accuracy: The protocol is accurate, if it is failed on all other false claim, agreed a unfair prover and a truthful verifier [32].

2.2.4.2 Advantage of Zero-Knowledge Proof

1) Zero Knowledge Proof Transfer: As long as the prover does not know any information or secrets about the prover, so he cannot divulge the secrets of the prover to any other party.
2) Efficiency: As a result of the presence of natural interactive proof, it must be calculate the efficiency of Zero-Knowledge Proof. The cost of the calculations associated to encryption are evaded.
3) Degradation: The safety of Zero-Knowledge Proof protocol does not deform with the continues using where not revealed any information about the secret.
4) Unsolved mathematical assumptions: Zero-Knowledge Proof protocol depends on many computational issues like discrete logarithms and integer factorization [33].

2.3 Literature

In electronic voting, generally there are four stages: initialization, registration, voting and counting.

Michail J. RADWIN (1995) schemes. This scheme is divided into four stages: the initialization stage, the authority setup RSA during this phase. The registration stage, during this phase the authority guides the voter in order to construct a token of his vote. The voting stage, the voter at this stage constructs and cast his vote and the vote will be decrypted. The counting stage, this is the last stage in which the results been issued. This scheme has used for Yes/No elections process [34]. This scheme was used in small elections, which require one authority and it is applied in some cases, in the elections that require more than one authority. The term token (is an entity that helps somebody to vote similar to a ballot paper) here is a pseudonym been built to the voter with the help of the authorities[34].

Atsushi FUJIOKA, Tatsuaki OKAMOTO, and Kazuo OHTA (1992) scheme is divided to four stages: an initialization stage, registration stage, voting stage and counting stage. During the first two stages a voter will receive a token from the authority using the scheme of a blind signature. Later, the voting stage in which the voter sends the encrypted vote beside his token and the last phase is a counting phase, where the votes are decrypted in order to count them [35]. The scheme needs only two authorities, the administrator in charge of the token issues, and the collector, which combines tokens and distributes the votes outcome. The administrator will sign the blindly signed token [35].

Miyako OHKUBO et al. scheme is composed into three stages, the registration stage during this stage the voter preparing the ballot and getting authorization by obtaining blind signature from the administrator. The voting stage, voter sends his vote by anonymous way using mix-net. Counting stage, at this stage the ballot papers are opened for the purpose of counted [36].

KIM, Jinho KIM et al. scheme for online voting. It is composed of three phases: the registration phase, voter will get pair of keys along with a certificate from the administrator. The voting phase, at this phase the voter will cast his vote and it will be signed blindly by the administrator to the tallier. Finally, the counting stage, at this stage all the ballots are decrypted and counted [37].

Juang et al. is proposed in 1997. It is composed of six stages. The initialization stage, during this stage the system parameters are setup. Key generating stage, during this stage all auditors work together without taking help from anyone else in order to generate the public key and share it with each other [38].

At this scheme administrator and some auditors are being used, in order to observer the election process. The original paper states that the voter could not refrain after the registration stage as well as at least $(m - k + 1)$ of auditors cannot detect their shadow keys until the screening stage and number of k auditors are truthful and transmit their shadow of the keys [38].

RYAN et al. proposed their scheme in 2005 into five stages. Roughly speaking, “Prêt-à-Voter” is the protocol that replaces the list of candidates randomly. The public key of the mix-servers which is named tellers will encode the permutation function sequentially to construct an “onion” that represents the final encoded value [39].

After that, all onions are printed on the ballots randomly because of the possibility of the encryption used in each layer of the onion. The voters can put a mark in the selection of candidates and cast ticks and onions without the list. When the voters cast the ballots, taller stallers decrypts the layer of the cover because they have the private key. Finally, the votes are decrypted by the tallied to get the results [39].

WEBER proposed his scheme in 2006 and it is composed of 4 stages. The initialization stage, at this phase ElGamal threshold cryptosystem are set up and also the public parameters are published. The registration stage, at this stage the voters get their credentials. The voting stage, at this stage the voters will cast their encoded votes. Finally, the ballots that has casted are verified, mixed, decrypted and counted by the authorities [40].

The authors Martin et al proposed a generic scheme in 2000. Their scheme proposed for converting voting protocol based on homomorphic encryption [41].

Relies on many of the authorities, every one of them causes a randomly order list by all encrypted votes, similar in [42]. Exhausting procedure called designated-verifier proofs [43], allows voters to trace their votes by pointing the encryption of their choice. The tallying of the votes are performed by using homomorphic encryption [41].

RJASKOVA proposed her scheme in 2002, and it is composed into three stages. During the first stage, authorities setup a robust threshold ElGamal cryptosystem. Later, the voter shares his vote with the authorities, which re-encrypted before rebuilt. The latest stage is decrypting the votes and tallying them by using homomorphic encryption [44].

Benaloh et al. proposed their paper in 1994 presents two protocol. Reduces protocol in which voters react with single authority and multiple tallying authorities protocol that improving privacy. All these protocols are based on homomorphic encryption properties. Verifiability are achieved at this scheme by using zero knowledge proofs. This paper has introduced the concept of receipt-freeness for the first time [45].

CRAMER, et al. in 1997, and composed into three stages. The initialization stage, a vote casting stage (voter cast ballots in public with proof that he would be calculated well on the bulletin board).

Protocols	**Cryptographic primitives used**			
	ZKP	HE	BS	MN
[34]	X		X	
[35]	X		X	
[36]	X		X	X
[37]	X		X	
[38]			X	
[39]	X			X
[40]	X			X
[41]	X	X		X
[44]	X	X		
[45]	X	X		
[46]	X	X		

Table 4: The Schemes and Their Primitives[4]**.**

The abbreviations that used: ZKP: Zero-Knowledge Proof, HE: Homomorphic Encryption, BS: Blind Signatures, MN: Mix-Nets.

[4] This had been taken from Survey on Electronic Voting Schemes, the authors are Laure Fouard, Mathilde Duclos, and Pascal Lafourcade. Page 60.

Finally, the tally stage by using homomorphic encryption properties all the votes are decrypted as one block thanks to the technique of threshold of cryptosystem. The main protocols offer to the voters the choices among double selections. There is in the same paper extension to allow the elections to provide many options [46]. The above schemes did not achieve the same characteristics, because of the properties of various encryption technique and how they are combined. The user will use one protocol more than the others, according to the specifications that want to achieve. Nevertheless, some of these schemes are not easy to achieve because of some theoretical assumptions are not constructed practically. Table 4 show you the above mentioned cryptographic primitives that are described previously. In Table 5 we listed the different kinds of requirements that accomplished by the above structures.

Protocols	**Requirements**							
	Privacy	**Receipt -Freeness**	**robustness**	**Verifiability**		**Democracy**		**fairness**
				U	I	**E**	**PMV**	
[34]	C	A	A	A	C		C	C
[35]	P	A	A	A	C	P	P	P
[36]	C	A	A	A	C	C	C	C
[37]	A	C	A	A	C	C	C	C
[38]	C	A	A	A	C	P	P	P
[39]	C	C	C	C	C			C
[40]	C	C	C	C	A	C	C	C
[41]	S	S	S	S	S	S	S	C
[44]	C	C	C	C	C	A		C
[45]	P			C				C
[46]	C		C	C		C	C	

Table 5: The Requirements Achieved by The Schemes[5]

The abbreviations that used: ZKP: Zero-Knowledge Proof, HE: Homomorphic Encryption, BS: Blind Signatures, MN: Mix-Nets.

[5] This had been taken from Survey on Electronic Voting Schemes, the authors are Laure Fouard, Mathilde Duclos, and Pascal Lafourcade. Page 61.

CHAPTER 3

3. ON SOME UNIVERSAL VERIFIABLE SCHEMES

3.1 Universal Verifiability

According to the literature, there are not unique or comprehensive definition for the Universal Verifiability. But nevertheless, if they are examined in detail we will find it holds almost the same meaning. Therefore, in order to understand the Universal Verifiability well we will examine the literature and see how the definition of the Universal Verifiability from point of view of some modern scientists.

Sako et al.(1995) presented the principles of the Universal Verifiability to focus on the importance of the audit for the entire election to classify the verifiability to individual and universal verifiability. This classification was accepted by other e-voting studies. Sako et al. has defined the individual and universal verifiability respectively as "A sender can verify whether or not his message has reached its destination, but cannot determine if this is true for the other voters" and "In the course of the protocol the participants broadcast information that allows any voter or interested third party to at a later time verify that the election was performed properly" [47].

Cranor et al.(1997) tightening the definition of the universal verifiability on it is just count the votes and defined it as "Anyone can independently verify that all votes have been counted correctly". Greatest of the studies that have followed has been used this definition because of its clarity and its ability to measure. [48].

Karlof et al.(2005) merge the definitions of the verifiability without differentiates between the individual and the universal one as follows: "Verifiably cast-as-intended means each voter should be able to verify his ballot accurately represents the vote he cast. Verifiably counted-as-cast means everyone should be able to verify that the final tally is an accurate count of the ballots" [49].

We can sum up the individual and universal verifiability that have used through the literature as follows: “every voter can check if his vote has been properly counted” and “anyone can check that the calculated result is correct and election is performed correctly” (Cranor 1997), (Sako 1995), (Fujioka 1992) and (Karlof 2005).

3.2 Brief description on some electronic voting scheme achieved the universal Verifiability

The goal of this stage is to give general overview of uses of the primitive encryption methods already mentioned in various electronic voting schemes. We note that there are some hybrid protocols that combines three schemes (Blind Signature, Homomorphic Encryption and Mixnet). Such merges must be built with great care. In fact, two or more protocols may verify property which is not achieved by the combination of them.

Thus, for each of them, we will make a brief description of the protocols that are satisfied universal verifiability. Some properties that have been achieved and supposed to achieve will be discussed. Also, we will refer to existing or new attacks.

3.2.1 DynaVote scheme

Authors: Orhan Cetinkaya and Ali Doganaksoy (2007).

Primitives Used: Dynamic ballot, blind signature scheme.

Requirements: The scheme has used public channels.

Summary: The scheme proposed in [51] and called DynaVote. DynaVote is composed into three stages: Authentication & Authorization (is completed before the election day), Voting and Counting. The proposed protocol did not use complex encryption algorithms nor hidden communication channels because of having to $PVID$ entity in addition to the following entities: Voters, Ballot Generator, Key Generator, Counter. The $PVID$ $(Pseudo\,Voter\,Identity)$ is the most important feature in this scheme which depends on blind signature. Voter may use $PVID$ and hide his real identity.

During the Authentication & Authorization stage the voter receives a $PVID-list$. Later, he may use the $PVID$ at any time and place during the election time.

The voting phase is divided into two phases: Ballot Obtaining Phase and Vote Casting Phase. Through the Ballot Obtaining Phase, voter obtains on the dynamic ballot. The main building block of DynaVote protocol is the dynamic ballot. In dynamic ballot the order of candidates are changed for each ballot. Through the Vote Casting Phase, voter selects his candidate by using the dynamic ballot mechanism [50].

Lastly, in counting stage votes are decrypted and counted.

Protocol description: We shall first describe set of symbols and abbreviations used in this protocol before going into explanation in detail.:

(e_v, d_v): Is the session of Public and Private keys respectively that voter uses them to communicate with Key Generator and with the Counter.

(e_s, d_s): Is the session of Public and Private keys respectively that voter uses them to communicate with Ballot Generator.

(e_p, d_p): Is the session of Public and Private keys respectively that voter uses them to communicate with $PVID$ Authority.

(e_a, d_a): PVID Authority's public-private key pair.

(e_b, d_b): Ballot Generator's public-private key pair.

(e_k, d_k): Key Generator's public-private key pair.

(e_c, d_c): Counter's public-private key pair.

(e_z, d_z): Voting public-private key pair generated for voter to cast his candidate selection.

$\check{E}_x(m)$: Encrypt the message m by using the public key e_x.

$\check{D}_x(m)$: Decrypt (sign) the message m by using the private key d_x.

$H(m)$: It is a hash function kind of one way that used by the voter and the authorities applied on the message m.

B: Dynamic ballot.

V': It is the voter candidate choice according to the dynamic ballot.

V: Voter's actual vote.

$PVID - list: \{PVID_1, PVID_2\}$: Is a list of pseudo hidden identities that are not connected to the real identity of the voter.

Authentication & Authorization stage:

- Voter creates on $ID - list\ \{ID_1, ID_2\}, ID$ containing random number and some useful information as $ID = \{Election\ Date, Authority\ Date,$ $Random\ Number\}$.
- Voter blinds each ID separately with different random blinding factor and gets m_b.
- m_b is the combination of blinded IDs.
- The voter sends $\check{E}_a(Reisteration\ ID, D_p(m_b))$ to $PVID$ authority.
- $PVID\ authority \begin{cases} if\big((voter\ is\ eligible)\&\&(has\ not\ made\ any\ request)\big) \rightarrow signs\ the \\ else \rightarrow discard\ the\ request \end{cases}$
- The authority of a $PVID$ signs the blinded IDs on a message m_b and gets m_{bs}. m_{bs} represents the mixture of IDs that are blindly signed.
- The authority of $PVID$ will send $\check{E}_p(\check{D}_a(m_{bs}))$ again to the voter, then the voter unblinds each blindly signed IDs.
- Voter checks the $PVID$ authority signature on m_{bs} and gets $PVID - list =$ $\{PVID_1, PVID_2\}$.
- The above results represents the Authentication & Authorization stage which held prior to the election time.

Voting stage:

- The voter casts the candidate who has chosen through the dynamic ballot that gets it.
- In this protocol a dynamic ballot represents the most important part.
- Dynamic ballot show only physical ballot in the ballot with the corresponding dynamic only.
- Figure 6 shows overview of the voting stage.
- In this protocol the voting process is divided into two parts: The Ballot Obtaining Phase and the Vote Casting Phase.

Ballot Obtaining Phase

- Voter encrypts e_v with Election Date by using the public key that belong to Key Generator's $\check{E}_k(e_v, Election\ Date)$, before sends it to the Ballot Generator, voter will produce m_1. $m_1 = \check{E}_b(PVID_1, \check{E}_k\ (e_v, Election\ Date),$

e_s). Now voter sends m_1 to Ballot Generator.

- Ballot Generator decrypts m_1, when it receives it.
- If the message has come from the right place Ballot Generator signs it and produce the message $m_2 = \check{E}_k(\check{D}_p(\check{E}_k(e_v, Election\ Date), e_b))$ and sends it to Key Generator.
- Key Generator decrypts it to check the signature of Ballot Generator. If it comes really from Ballot Generator, Key Generator process to the next steps otherwise discards it.
- Key Generator generates pair of keys and saves them in the Votingkeylist(e_z, d_z). These keys are used by the voter to cast their vote in order to choose their candidates to the Counter authority.
- The hash of the voter will be published in addition to the voter keys public and private separately such $H(e_v, e_z)$ and $H(e_v, d_z)$ in separate screen called Key Generator Bulletin Board (KGBB).
- There are two important use of $H(e_v, e_z)$. First, the voter will use it in order to confirm the accuracy of the voting key. In addition, it can be used by the Counter authority to inhibit the Key Generator handling of the voting keys that been generated.

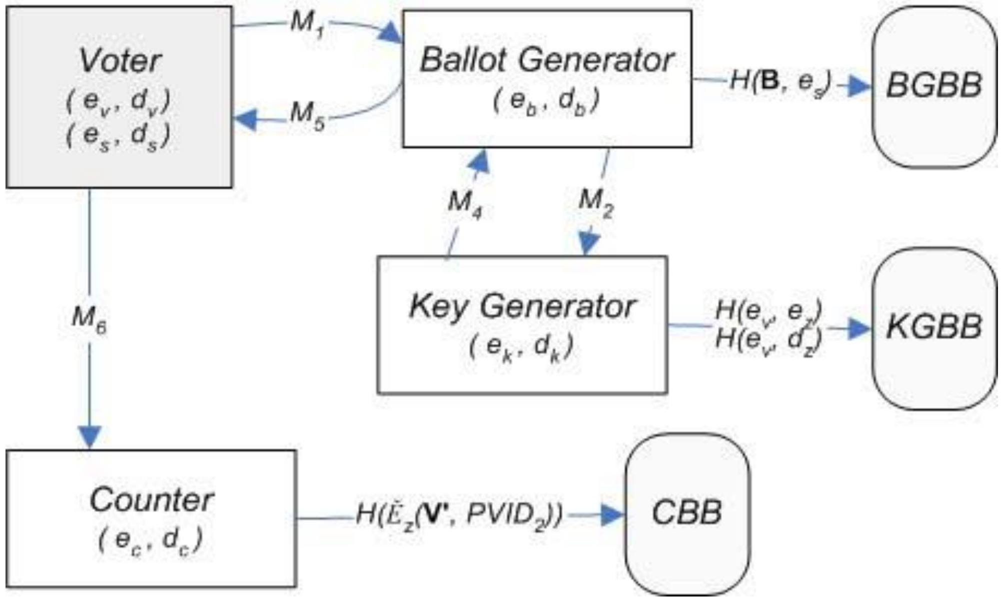

Figure 6 Overview of The Voting Stage[6]

[6] From the paper that belong to Orhan Cetenkaya & Ali Doganaksoy published in IEEE library.

- The authority of Key Generator will save (e_v, e_z, d_z) in VotingKeyList and generate m_3 and m_4.

  $m_3 = \check{E}_v(\check{D}_k(e_z, Election\ Date), e_v)$

  $m_4 = \check{E}_b(\check{D}_k(m_3, e_k))$
- The authority of Key Generator will send m_4 to the authority of Ballot Generator. Ballot Generator will decrypt the message and checks the sign of Key Generator's authority.
- The authority of a Ballot Generator's will create the dynamic ballot **B** by using arbitrary creator function.
- The authority of a Ballot Generator will publish the public keys of the voter such $H(\mathbf{B}, e_s)$ and also a hash of the dynamic ballot **B** in a separate screen named Ballot Generator's Bulletin Board(BGBB).
- $H(\mathbf{B}, e_s)$ is published to give an opportunity to the voter to verify the correctness of dynamic ballot.
- The authority of the Ballot Generator will save $(PVID, m_3, \mathbf{B}, e_b)$ in the BallotList that is an internal list belong to the dynamic ballot **B**. Later, it will produce the messag**e** m_5.

  $m_5 = \check{E}_s(\check{D}_b(m_3, \mathbf{B}, e_b))$
- The voter will receive m_5 that sent to him by the authority of Ballot Generator. Later, he will decrypt m_5 using the public key of the Ballot Generator and obtains the dynamic ballot B and extracts m_3.
- The voter will calculate $H(\mathbf{B}, e_s)$ in order to confirm the dynamic ballot that he obtained it and checks against the one that published on the Ballot Generator Bulletin Board (BGBB).
- In order to obtain the voting key e_z the voter will decrypt the message m_3 by applying the Key Generator public key on it.
- The voter confirm the result on the Key Generator Bulletin Board (KGBB) by creating the hash value $H(e_v, e_z)$.
- At this moment the voter has the key of the voting e_z in addition, to the dynamic ballot and been ready to cast his vote.

Vote Casting Phase

- Through the dynamic ballot **B** and the voting key e_z that voter has received them, he selects his candidate **V'**, later he produce the message $M_6 = \check{\mathrm{E}}_c(PVID_1, \check{\mathrm{E}}_z(\mathbf{V}', \mathrm{PVID}_2), e_v)$.
- He cast his vote without anonymous channels because the presence of $PVID$ scheme by sending M_6 to the Counter.
- No one can connect any relation between PVID scheme and voter real identity.
- The message m_6 decrypts by the Counter for further steps.
- Counter checks the validity of $PVID_1$ by using authorities public key, if it is valid Counter process the request, otherwise the request is discarded.
- The Counter publishes the hash of encrypted **V'** $H(\check{\mathrm{E}}_z(\mathbf{V}', PVID_2))$ on Counter Bulletin Board (CBB).
- The authority of Counter will append the date and time to the encrypted vote **V'** and add it the VoteList such as $(PVID_1, \check{\mathrm{E}}_z(\mathbf{V}', PVID_{\mathbf{2}}), e_v, DateTime)$.
- The VoteList is an internal list of the candidate that been selected by the voter that associated with the $PVIDs$ that he has received it.
- The Counter authority will inform the voter from the correctness of his vote by sending an acknowledgement (**Ack**) to him $\check{\mathrm{E}}_v(\check{\mathrm{D}}_c(\boldsymbol{Ack}))$.
- Through the **Ack** that the voter receives it the voter can verify individually from his vote by looking on CBB, voter finds the sequence of numbers as $H(\check{\mathrm{E}}_z(\mathbf{V}', PVID_2))$ and this sequence of numbers represent a receipt.
- All of that represents the voting stage which held at the election time.

Counting Phase

- When the time of the election ends, the counting stage starts.
- The authority of Ballot Generator, the Key Generator and the Counter will declare the following sub list as SubBallotList ($|PVID_1, \boldsymbol{B}|$), SubVotingKeyList ($|e_v, d_z|$) and SubVoteList ($|\check{\mathrm{E}}_z(\mathbf{V}', PVID_{\mathbf{2}}|$) one-to-one.
- The consistency of the election will be announced on bulletin boards and can be checked from any organization.
- The counting will start by the Counter authority.

- Counter contests every element in the VoteList $|e_v, d_z|$ over voter's session key e_v. Then, Counter gets the list $\left|PVID_1, \check{\mathrm{E}}_z(\mathbf{V}', PVID_{\mathbf{2}}), e_v, DateTime, d_z\right|$.
- The list $\check{\mathrm{E}}_z(\mathbf{V}', PVID_{\mathbf{2}}))$ will be simplified by the Counter by decrypting it by using the private key (d_z) and produces the list $|PVId_1, PVID_2, \mathbf{V}'|$ which represent the voter candidate selection.
- The Counter authority will confirm the correctness of the vote by applying the public key of the $PVID$ authority on $(PVID_2)$. If the checking process failed the vote will discard.
- Figure 7 showing an overview of the counting phase.
- This scheme allows the voter to vote more than one time because the $PVID$ is applied and only the last vote will calculate and previous votes are discarded and the Counter authority will keep the time and the date for each single vote.
- Then, the Counter authority will match the selection of the candidate $(\mathbf{V}')$ in the list $|PVID_1, PVID_2, \mathbf{V}'|$ with the dynamic ballot (**B**) that corresponding to SubBallotList $|PVID_1, \boldsymbol{B}|$ over $PVID_1$.
- Later, the Counter authority will obtain a list $|PVID_1, PVID_2, \mathbf{V}', \boldsymbol{B}|$ that represents the real vote of the voter.
- The real vote of the voter $\mathbf{V}$ can be defined as $\mathbf{V} = C_i \in \boldsymbol{B} \qquad where\ \ i = \mathbf{V}', \boldsymbol{B} = \{C_1, C_2, \dots, C_n\}$.

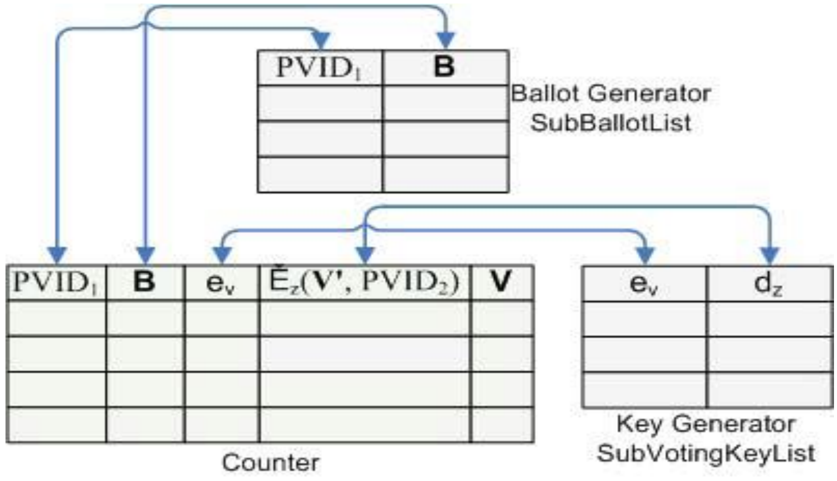

Figure 7 overview of the counting stage[7]

[7] From the paper that belong to Orhan Cetenkaya & Ali Doganaksoy published in IEEE library.

- The Counter authority at the end of the counting stage will announce the valid votes list $|PVID_1, H\left(\check{E}_z(\mathbf{V}', PVID_2)\right), \mathbf{V}|$ that consist to the votes at CBB and the invalid votes are discarded.
- After the counting stage the votes are easily to be tallied and announced.
- The voter can confirm that his vote been counted correctly after the counting stage by using his $PVID_1$, in addition, to the sequence number of $H(\check{E}_z(\mathbf{V}', PVID_2))$ through using the issued list.

Security analysis

We will show the electronic voting requirements, which DynaVote is able to meet:

1- Privacy: Voter does not use his real identity because the scheme uses a blind signature, so the PVID that belong to each single voter does not lead to his real identity for that privacy had kept.

2- Eligibility: Since DynaVote employs the PVID and through the PVID only the IDs of the eligible voters are signed. The PVID works by the concept of blind signature that signs each ID blindly, the ineligible voters IDs are not signed because n authorities are participate in signing process, so any request does not carry the eligibility will not sign.

3- Uniqueness: The counter authority obtains the list $|PVID_1, PVID_2, \boldsymbol{V}', \boldsymbol{B}|$. Since we stated before that the $PVID_1$ is unique and the $PVID_2$ contains meaningful information as vote casting date, so, the counter authority will take the result of the last voting process, for that, the uniqueness is achieved.

4- Uncoercibility: The protocol relies on concepts of multiple voting through the election time (which may continue several days) and the final vote will be take into consideration as we mentioned earlier. So, the process of vote-buying will not be easy, since voter can re-casting his vote again. For that, coercer will not find any way to coerce a voter to chooses the candidate that he like.

5- Fairness: Counting phase begins after the voting phase is completed. Since the dynamic ballot $\mathbf{B}$ is employed, Counter can only know that the voter has casted his vote, he cannot reveal any information more than that, he will need to the ballot generator to inform him the corresponding dynamic ballot. He will also need to the voting key which is controlled by key generator. So, Counter authority cannot calculate any partial results before the end of the election time, for that fairness is kept.

6- Accuracy: Voter can observes the processes taking place step by step, so when he gets the dynamic ballot $\mathbf{B}$ and voting key e_z, he can ensures it is correct by looking to BGBB and KGBB, if there is any mistake or corruption, he can provides objection to ballot generator. Also, he can looks to the CBB and ensures that his vote is published on it. Any authority cannot process any step without depends on the others. Even if all authorities (Key Generator, Ballot Generator and Counter) want to make any vote cannot because they need to PVID authority. PVIDs authority cannot make any fake PVID because it apply threshold cryptography.

7- Robustness: No one can distortion on the electoral system, not the voters nor authorities. Since if voter decided to send more than one vote the last vote is calculated only because the $PVID_1$ is employed. As for authority, the bulletin board and hash value from each single vote is employed publicly. Even if the authorities plotted with some voters they can influence the votes of those voters only.

8- Individual verifiability: Each voter can verify the verifiability of each stage before moving to the next stage. For instance, Key Generator and Ballot Generator will publish the two values $H(e_v, e_z)$ and $H(\boldsymbol{B}, e_s)$ respectively on their bulletin board KGBB and BGBB respectively. The voter can make sure that the keys (e_v, e_z and e_s) that he have them is correct by applying the same hash function that the Key Generator and Ballot Generator applied them. If the two values match he will send his candidate to the Counter, otherwise he make object to the Key Generator or to the Ballot Generator. In the voting stage, after the voter receiving the acknowledgement from the Counter, he will create the hash value for $H(\check{E}_z(\boldsymbol{V'}, PVID_2)$ and compares it with the hash value on CBB. If were not matching, the voter submits an objection to the Counter authority by clarifying $\boldsymbol{V'}$ and e_z.

9- Universal verifiability: Key Generator, Ballot Generator and Counter publish their sublists, at the end of the election before the tallying process. As soon as, the Counter announces the election results. The authorities can verify the validity of the election by monitoring the results on the bulletin board. Therefore, any authority and any observer can see the results. The Counter authority mission is to verify all results [50].

3.2.2 End-2-End Verifiable Voting System Scheme

Authors: Ahmet Sınak, Seçil Özcan¸ Hakan Yıldırım and Mehmet Sabir Kiraz.

Primitives Used: Homomorphic encryption, ElGamal encryption process.

Requirements: The scheme uses anonymous channels.

Summary: The End-2-End Verifiable Scheme is divided into four stages: the Vote Submission, the Transporting Process, Counting and Announcement Processes. Also, it contains eight players: The Voter's Computer, Authentication Box, Control Box, Authority, Bulletin Board are online players. The Voter, Counter and Decryption Service are offline players in our protocol. The other components of the system are the vote, Thin Client, Terminal Server, Trusted Parties and Certification Authority.

The protocol works on principle of generation private key in the form of threshold. Thus, it generated by the participation of t entities in independent of each other together to generate the private key, and any number of these entities less than t is not able to generate the private key. The private and public key are generated pre-election period, where each party receives a share of private key but the public key is known to everyone [51].

During the Vote Submission the following two parts are active: Voter and his Computer. The voter V will cast his vote after using his candidate selection and the PC will encrypts the voter's vote and creates hash of it. During the Transporting Process, the AB will authenticate the voters, CB will generate a receipt code an sends it to PC and V, the A will multiply pair of vote coming from AB and CB and sends them to C at the end of the election time. The Counting and Announcement Processes start when the election period ends, at this time C and DS are active players, if there is not any problem the votes will be counted and announced [52].

Protocol description: Before going to describe the protocol we must explain some abbreviation and keywords that will be used in the protocol.

V: The voter.

PC: The voter's computer.

sn: The voter's secret number.

v: The voter's vote.

r: Random Number.

x: The private key of the election.

g: Generator.

y: The election public key.

c: The encrypted ballot.

S: The private key of voter.

h: The receipt code value.

$Sign$: The PC signature on the encrypted hash vote using voter's secret key.

ZK_{PC}: The PC zero knowledge proof.

AB: Authentication Box.

C: Counter.

A: Authority.

u: Unique Number.

ZK_{AB}: The AB zero knowledge proof.

s: Sequence number.

CB: Control Box.

DS: Decryption Service.

TP: Trusted Parties.

t: Number of TP.

Vote Submission:

Online Process:

- V by using his ID card authenticates himself to the voting system and enters his sn after selects his candidate.
- The PC will encrypt the vote with r and produce the c.
- $E(v,r) = (g^r \ mod\ p, g^v\ y^r\ mod\ p) = c$. Later, PC will create hash of c and signs it using S.

  $h = H(c)$.

  $Sign = S(h)$.

- Later, PC will give h to V.

Transporting Process:

- PC will send ZK_{PC} and sn to AB.

$$\text{PC} \xrightarrow{c, sign, ID, ZK_{pc}, sn} \text{AB}$$

- By using the voter public key AB verifies $sign \rightarrow h$ and checks ZK_{PC}.
- AB gives the vote u so that, A matches the vote coming from AB with the vote coming from CB. AB also, give the vote s to ensures how many votes V has voted and sends to CB ZK_{AB}, u, s and all data received from PC except $sign$.

$$\text{AB} \xrightarrow{c, ID, ZK_{pc}, ZK_{AB}, sn, s, u} \text{CB}$$

- CB will check the ZK_{pc} and ZK_{AB}. Also, CB will calculate the hash of c and sends to PC in addition to V via safe channel as SMS. Through that, V will check his receipt and the receipt that received from CB to ensures that his vote is transmitted by AB in correct way with no alternation. Otherwise he knows that his computer is compromised or there is some problem so the vote is canceled.

$$\text{CB} \xrightarrow{h} \text{V and PC}$$

- If the vote is valid, AB and CB will mask c^{a1} and c^{a2} respectively. AB will send c^{a1}, u, s, ZK_{AB} and all the received data from PC except $sign$ and c to A. also CB will send c^{a2}, ZK_{CB} and all the received data from AB except c to A.

$$\text{AB} \xrightarrow{c^{a1}, ID, ZK_{PC}, ZK_{AB}, sn, s, u} \text{A}$$

$$\text{CB} \xrightarrow{c^{a2}, ID, ZK_{PC}, ZK_{AB}, ZK_{CB}, sn, s, u} \text{A}$$

- A verifies the zero knowledge proofs ZK_{PC}, ZK_{AB} and ZK_{CB}. After that, A will match the pairs by using u and multiplies the pair of covered encoded votes came from AB and CB. Then, A will store all the data that came from AB and CB in a database.

$$\begin{aligned} c^{a1}.c^{a2} &= E(v^{a1}, ra1).E(v^{a2}, ra2) \\ &= E\left(v^{a1+a2}, r(a1+a2)\right) \end{aligned}$$

$$= (g^r, g^v\, y^r)^{a1+a2} = c^{a1+a2}$$

Offline Process:

- A periodically sends ZK_A,c^{a1+a2} and all the received data except c^{a1}, c^{a2} and u to C (offline).

  $$\text{A} \xrightarrow{c^{a1+a2}, ID, ZK_{PC}, ZK_{AB}, ZK_{CB} ZK_A, sn, s} \text{C}$$

- C using a_3 to decrypts c^{a1+a2}, after verifies the proofs for ZK_{PC}, ZK_{AB}, ZK_{CB}, ZK_A.

  $(c^{a1+a2})^{a3}\ mod\ p \rightarrow (c^{a3^{-1}})^{a3} = c\ mod\ p$. Later C checks the sn, if the sn is correct the vote will be valid. If it is not the vote will not valid.

- C will calculate the h for each vote and send the V ID and the sn to BB offline using external disk.

  $$\text{C} \xrightarrow{h, ID, sn} \text{BB}$$

Online Process:

- BB will send h and sn to the voter, so the voter can verify that his vote has come to C and also, to ensures that sn and h are correct.

  $$\text{BB} \xrightarrow{h, sn} \text{V}$$

- BB will publish the sn and h for all the votes even if sn is not correct, so every voter does not see his sn and h he should inform the election authority.

Counting and Announcement Processes:

- When the election time is over C permutes the valid list which has the greatest s and valid sn, using homomorphic of ElGamal encryption system. C will get l of them randomly by using the discrete logarithm problem, then multiplies them.

  $$E(v1, r1).E(v2, r2) \dots E(v1, r1) = E(v1 + v2 + \cdots + v1, r1 + r2 + \cdots + r1)$$

- Through one way filter all votes are sent from C to DS and there is no any traffic from C to DS. At DS votes are decrypted using x.

  $$\text{C} \xrightarrow{E(v1, v2 + \cdots + v1, r1 + r2 + \cdots + r1)} \text{DS}$$

- In DS the private key of the election is built by using t TP to decrypt the encrypted votes.

- $D_x\big(E(v1, v2 + \cdots + v1, r1 + r2 + \cdots + r1)\big)$
  $= g^{v1+v2+\cdots vl}$.
- Finally, the election authority will announce the results.

Security Analysis:

The above protocol achieves the following security requirements:

1- Coercion-resistance: The protocol allows voters to vote more than once and the last vote is calculated as a result. As well as, the existence of a secret number sn that the voter can be enters it through the voting process to impress the coercer without sending a warning message to a voter that the secret number sn which enter it incorrectly, and the voter can votes again using correct secret number . So, the protocol guaranty that it will resistance against coercion.

2- Integrity: Since the protocol employs a zero-knowledge proof at every stage of its stages, so integrity is guaranteed. Through the SMSs, that the voter receives them from BB and CB he can confirm the validity of his vote during both transporting and counting stages. Therefore, if the values received by SMSs messages represent the same value that he submitted it during the voting, then the integrity is guaranteed. Otherwise, he can submit an objection to the election authority.

3- Vote Privacy: The protocol uses the principle of the homomorphic encryption. In addition, the vote of the voter encrypts in the PC, at each stage the vote arrives cannot be read since the stages through which do not have the private key. After counted, the votes are decrypted using homomorphic encryption as one block, so the privacy of the voter remain reserved.

4- Receipt freeness: The coercer cannot be sure that the voter has voted for his candidate or not because voter can use the secret number, that the system gave it to him instead of using the number in the receipt code paper. So, in this situation coercer (the person who buy the vote) cannot make sure that the voter has used the receipt code paper and voted for his preferred candidate or not. For that, receipt freeness has achieved.

5- Individual and universal verifiability: Each voter can make sure that his encrypted vote may transferred, counted and tabulated correctly in the final tally. Furthermore, each voter can make sure that his vote on the subject of the BB through the ID of each voter with his hash value [52].

3.2.3 Efficient receipt- free voting based on homomorphic encryption

Authors: Martin HIRT and Kazue SAKO (2000).

Primitives Used: Homomorphism encryption.

Requirements: Untappable one-way channels between authorities and voters, Properties of homomorphic re-encryption.

Summary: The authors Martin HIRT and Kazue SAKO proposed a generic scheme in 2000. Their scheme proposed for converting voting protocol based on homomorphic encryption [53].

Relies on many of the authorities, every one of them produces a arbitrarily order list through all votes that been encrypted, similar in [54]. Exhausting a method named designated-verifier proofs [55], allows voters to trace their votes by pointing the encryption of their choice. The tallying of the votes are performed by using homomorphic encryption [53].

Vote Generation:

- For each valid vote v_i, there exist standard encryption e_i^0, in turn, for each $A_k(k = 1, \dots, N)$.
- A_k picks the encrypted valid votes list $e_1^{(k-1)}, \dots, e_L^{(k-1)}$, shuffles it randomly and hand it to the next authority.
- Shuffles means to re-encrypted valid votes list $e_i^{(k-1)}$ and permute randomly the order of the list by choosing a permutation $\pi_k: \{1, \dots, L\} \to \{1, \dots, L\}$, ($1, .., L$ is equal to the number of the valid votes). Then computes $e_i^k \leftarrow e_{\pi(i)}^{(k-1)} \oplus \mathrm{E}\left(0, r_i^{(k-1)}\right), r_i^{(k-1)} \overset{\$}{\leftarrow}$.
- A_k proves publicly it shuffled honestly by proving for each i there exist a re-encryption of $e_i^{(k-1)}$ in the list $e_1^{(k)}, \dots, e_L^{(k)}$ without revealing which.
- A_k conveys to the voters the permutation π_k that it used and privately prove that it is correct.

- The voter can complains against the authority if he does not accept the prove. If he does, we set $e_i^{(k)} \leftarrow e_i^{(k-1)}, i = 1, \ldots, L$. So, we ignore the shuffling of this authority. Voter may complain against $n - t$ authorities.

Vote Casting:

- The voter publicly announce his vote by driving the position i of the encrypted vote $e_i^{(N)}$.

Tallying:

- After the votes are summed and encrypted we obtain $E(T)$. After that, the authorities decrypt $E(T)$ to obtain T, with a proof of correctness [56].

Component

Additive Homomorphic ElGamal Encryption

- The scheme is stand on ElGamal encryption scheme.
- $E(v) = (g^{\alpha}, y^{v}\, h^{\alpha})$, where $\alpha \in R\, Z_q$ is a random number and y^{v} is the message.
- $e_v^{(0)} = (1, y^{v})$ is the standard encryption of v, and the secret key z is chosen uniformly from Z_q, and the public key is $h = g^{z}$. The key pair (z, h) is constructed in a way that each authority receives a share z_i of z in a (t, N)-threshold secret sharing scheme and is publicly committed to this share by $h_i = g^{z_i}$.
- y is another self-governing generator of G. The set V of valid votes contains L values in Z_q. Also, an encryption of a vote $v \in V$.

Verifiable Decryption:

- The authorities first jointly compute, reveal and prove $\hat{x} = x^{z}$ in order to decrypt T from $e = (x, y)$. Since z is the secret key and is shared by all the authorities i.e. every authority has share of z_i. So, each authority A_i compute $\hat{x} = x^{z_i}$. This is achieved if t of the authorities reveal and prove $\hat{x}_i$.
- If $\hat{x}$ became known, one can compute $\frac{y}{\hat{x}} = \frac{\gamma^{T}.\, h^{\alpha}}{(g^{\alpha})^{z}} = \gamma^{T}$.

Re-encryptability:

- Let’s assume that $e(x, y)$. So, it’s re-encryption is $e' = (x', y') = (g^{\xi}x, h^{\xi}y)$. Where ξ is a random integer $\in R$ in Z_p and ξ is chosen uniformly in Z_p, so (x', y') is uniformly distributed.

Security analysis:

We will show the security requirements that the previous protocol has achieved:

1- Privacy, robustness: Since $t - 1$ from the authorities can decrypt the final results, so privacy is achieved.

2- Universal and individual verifiability: Any authority, observer or honest verifier can check that the result has been decrypted and counted correctly. So, the universal verifiability is achieved.

3- Fairness: Since the counting start when the voting time over so fairness is achieved.

3.3 Comparison between the three electronic voting systems

As a result of the differing priorities of encryption that works out, so schemes that are described previously did not achieve the same characteristics. Thus, the following comparative data help the user to choose one over the other schemes to achieve the requirements needed.

Compared with previous schemes and understand how each of them works, we must understand what is happening in all of its stages, the start of the period prior to the voting process until the announcement of the results, so we will start by comparing the period leading up to the voting process first, and then what happens after.

3.3.1 The Authentication and Authorization phase

Table 6 shows the authentication and the authorization stage for the voting systems that held prior or during the voting period, so the voter may have some IDs and secret number,...etc. Through different proposals each scheme will have the rudiments encryption is different from the other. The Authentication phase in DynaVote the voter will prepare his own ID. In this ID, there will be a necessary and important information for the completion of the election process, including

$\{election\ date, authority\ date, random\ number\}$. Election date here means the day and time scheduled to open the system for the purpose of voting process where (as we mentioned earlier) the authorization process will precede the election period. The authority date, means of authenticating the voter that he has been registered his information correctly and since this moment he qualified for the election process. Finally, the random number, which is used by the voter with the blind signature in order to conceal his name, identity and all other information by relying on the advantages of the blind signature. These information that been concealed will encrypt inside a message called mb by the public key of an authority called $PVID$.

In Authorization part, the voter will send the message mb to PVID authority. $PVID$ will receive the message sent by the voter and decrypt the code of this message by using its private key. Thereafter, the eligibility of the voter will be checked, if the voter is eligible and has not verified beforehand, then it will sign this message and send IDs called $PVID - list = \{PVID_1, PVID_2\}$ used later for the completion of the election process in the voting stage.

It is clear in Table 6 that End-2-End does not contain any authentication phase because in each stage there is no authority that asks the voter on his eligibility in whole the election process.

The authorization phase for End-2-End is done when the voter uses his ID card and enter his sn after choosing his candidate. Through the sn number that the voter enters it when he casts his vote, the election authority will ensure that the voter is eligible and practiced his right by voting. The sn number will encrypt with the voter information, so the voting authority will be unable to uncover the identity of the

	Authentication	**Authorization**
DynaVote	$Voter \xrightarrow{\check{E}_a(\ Registeration\ ID, D_p(m_b))} PVID_{authority}$	$PVID_{authority} \xrightarrow{\check{E}_p(\check{D}_a(m_{bs}))} Voter$
End-2-End	-	Voter uses his ID and sn to authenticate himself
Receipt-Free	-	-

Table 6: The Authentication and Authorization Stages

voter. The key players at this stage will be the voter and his PC. There are other algorithms (have been explained) to deal with the voter PC in the case of breakout. So, if the sn is true the election authority will address the request to other phases but if it was not true the request will be rejected.

Receipt-Free will not has any authentication and authorization stage through the whole election process. Thus, the protocol is intended to work in controlled environment where is not possible for any ineligible voter to reach the election process. So that, there is no need for any authentication or authorization stages because the predisposing environment for the election process is closed for a certain people.

3.3.2 Voting and Submitting phase

Let’s start with DynaVote, the voting here are made by a voter to choose its candidate through the dynamic ballot that get it during this phase. Figure 8 (a) shows the voting process. This stage is divided into two main stages are the Ballot Obtaining Phase and Vote Casting Phase (as mentioned earlier). At this stage the important thing is the Dynamic Ballot , which the voter gets it from an authority called the Ballot Generator. The Dynamic Ballot will contain a sequence of candidates that will change in each voting process (because the protocol will allow elected more than once and only the last voting process will be calculated). Later the voter will choose his preferred candidate from the Dynamic Ballot . The Ballot Obtaining Phase will finish when the voter choses his candidate. The Vote Casting Stage will start before the voter sends his Dynamic Ballot which has received it and choose his candidate through it. The voter choice will be encrypted and be send to another authority called Counter. At this point the voting phase will complete and then another stage will start.

For End-2-End the voting process is done when the voter enters his sn (as we stated previously) and selects his candidate selection (Figure 8 (b)). The PC will send the voter votes after encrypt it in addition to other information belong to the voter to an authority called AB. AB will decrypt the voter and make sure that it is valid. Later, AB will send this vote and all the information that has received from PC to another authority called CB. CB will send a hash of the encrypted vote to the voter via secure channels like sms, so the voter can ensure that his vote has been transferred properly, otherwise, he will be known that his vote has been manipulated by AB authority or that his PC has been compromised. Thereafter, CB will send all the information to A authority and then the voting process will be completed and the counting process will start that will be offline.

Lastly, for Receipt-free the voting stage will start when the voter drives the position of his vote and publicly announce it as in Figure 8 (c). The voter has an interface that contains the candidates information and the voter will choose one of them. Thereafter, the voter will choose his candidate and the voter choice in addition to all the voter information will be encrypted by using ElGamal encryption scheme with the authority public key that voter receives it in the voting time.

3.3.3 Counting Phase

In DynaVote the counting stage(as in Figure 9 (a)) begins at the end of the election time by an authority called Counter. Then, Ballot Generator, Key Generator and Counter will announce the following sub lists as SubBallotList, SubVotingKeyList and SubVoteList respectively. These three lists that show the lists of encrypted votes that belong to each voter will be viewed by everyone because it will be posted on the bulletin board. Each voter will make sure from the presence of his vote because he will compare the receipt that he has received it and from that on bulletin board. Then, the counter authority will ensure from the validity of each vote, each valid vote will be calculated and posted on bulletin board.

In End-2-End scheme the counting stage (Figure 9 (b)) begins at the end of the election time. The counter authority that called C rearranges the safe lists that have the greatest sequence and also has a valid secret number that is given previously to each voter. C authority will multiply all the votes as one block using ElGamal encryption scheme. Later, C will send the valid votes that have the greatest s and valid sn by using one way filter to another authority called DS. The DS authority will decrypt these votes by using threshold of Trusted Parties (TP). In DS the private key x of the election will built by using t of TP. Later, the election authority will announce the results. In receipt- free voting the counting stage (Figure 9 (c)) starts when the time of the election is finished. When all the votes are collected and encrypted by using homomorphic encryption with El-Gamal encryption scheme. Then, we get on T block of the votes. The block that collected is decrypted by using $t-1$ from authorities with proof of correct decryption.

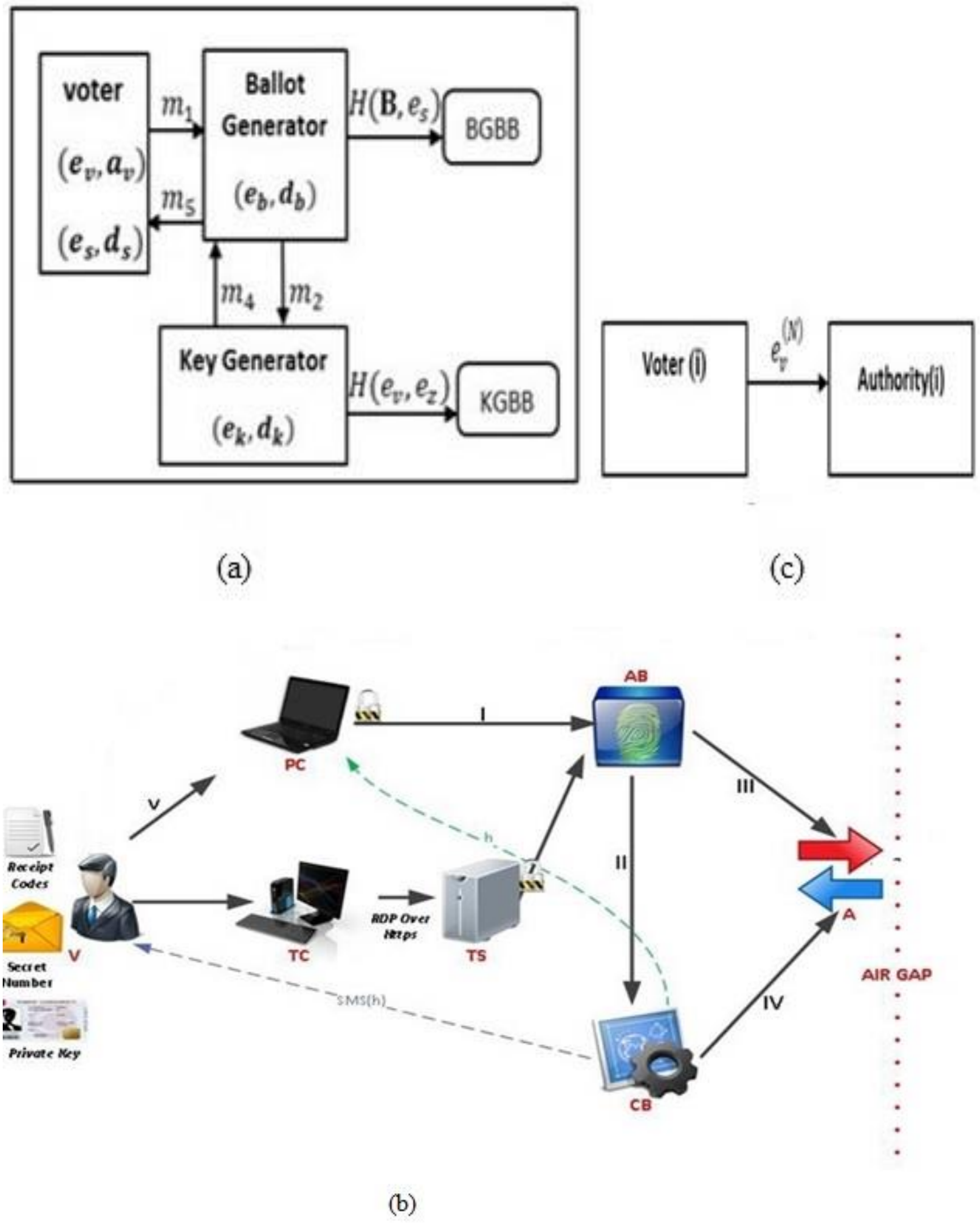

Figure 8 The Voting Schemes (a) DyneVote, (b) End-2-End, (c) Receipt-Free

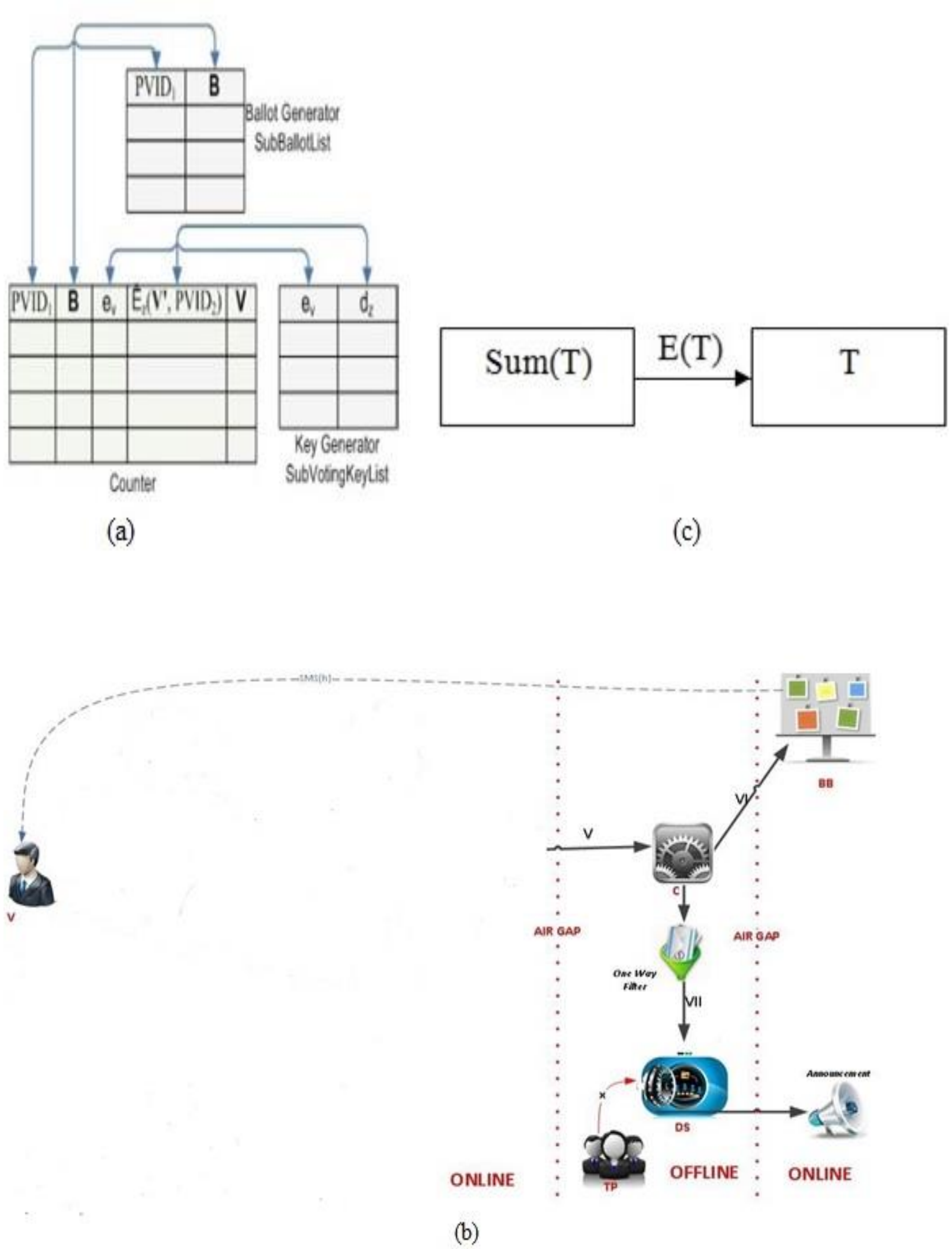

Figure 9 The Counting Schemes (a) DyneVote, (b) End-2-End, (c) Receipt-free

3.4 Robustness Against Attack

In the follows section, we will present various types of attacks that could be exposed to these protocols and we will show the extent of resistance and weaknesses according to these attacks.

DynaVote and End-2-End resist if the attacker intercepting the vote of the voter across network and replaced it by another one because the voter will receive a hash of the encrypted vote whereas the original vote will be presented on the Bulletin Board, so if the voter has noticed any change in his vote he can submit an objection to the voting authorities. Also Receipt-Free will not suffer from this type of attack, because it (as mentioned earlier) employed to work in controlled environment. For that it will not suffer from the intercepting of the vote through network because the environment in which it runs is small and controlled.

There is another type of attack occured to DynaVote and End-2-End which is send the same sn to more than one voter and that happen with End-2-End. In case of DynaVote this attack will send the same $PVID$ to more than one voter.

This attack does not work too, since the existence of the Bulletin Board that can be seen by all the voters and authorities, so the voters and authorities can easily discover that this PVID or this sn is belong to more than one voter and easily discard these votes and voter can cast his vote again (voter can cast his vote many times and only the last valid vote will be calculated). In Receipt-Free there is no secret number that can be helped to carry out the voting process. Thus, this kind of attack will not be used through whole the voting process.

The other type of attack that could have occured to DynaVote and End-2-End is when the attacker stop function the process of vote sending for a period of time during the election time. This attack will not succeed because in every voting process voter will receive a receipt at the transmitting time. This receipt shows the voting time. Thus, if the attacker stop function the transferring process so the voter will not see his vote on the Bulletin Board and he will submit an objection to the voting authority. However, for Receipt-Free this attack may work because there is no any receipt can be given to the voter in order to allow him to confirm that his vote has been transferred and counted correctly.

DynaVote and End-2-End will resist if the attacker transmit the vote more than once, where every time the system will registe new time on the Bulletin Board and only the last vote will be calculated. In the case of Receipt-Free scheme this attack could be succeed even in controlled environment. It can be if the authorities tried to calculate the vote more than once or to vote instead of absentee voters or instead of voters who abstaining from voting.

In this section we have assessed the performance of the algorithms that have been demonstrated and extent of its strength and weaknesses. These attacks, which were displayed is the bases of attacks that can be occured to the vote when transmitted from the voter PC to the authorities server.

3.5 Comparison according to cryptographic primitives used

Table 7 shows the encryption primitives that have used in the previous schemes and also described earlier.

3.6 Comparison according to security requirements that satisfied

Table 8 shows the requirements of the security that satisfied and unsatisfied by the schemes.

Our search has not been exposed to the problems facing every single vote or group of votes across the network, which may bring other problems. As shown in the Table 8 only DynaVote scheme has achieved all security requirements. Moreover, previous protocols do not mention to the identical security features and when they have not definite in the similar context. For instance, DynaVote means freedom of voter to choose his favored candidate without prove that practically. However, in DynaVote and End-2-End Verifiable Voting System, we see that the voter has received proof that he had voted, despite the fact that the voter cannot prove anything through it. As we see about the robustness in DynaVote, the inverters of End-2-End Verifiable Voting System and Efficient Receipt- Free Voting Based on Homomorphic Encryption had claimed that their systems are strong and robust without prove their claim. The inventor of Dynavote was the only one who had proved the robustness of his system. Therefore, our comparison about the safety properties can contain misjudgment and there is a real need to formalize to make a real comparison.

DynaVote			**End-2-End**	**Receipt-Free**
Cryptographic primitives used	**ZKP**		X	X
	HE		X	X
	BS	X		
	MN			X

Table 7: The Schemes and Their Primitives

The abbreviations that used: ZKP: Zero-Knowledge Proof, HE: Homomorphic Encryption, BS: Blind Signatures, MN: Mix-Nets.

DynaVote			**End-2-End**	**Receipt-Free**
Privacy		S	S	S
Receipt -Freeness		S	S	A
Robustness		S	S	A
Fairness		S	S	S
Accuracy		S	S	C
Verifiability	**I**	S	S	A
	U	S	S	S
Democracy	**PMV**	S	S	S
	E	S	S	S

Table 8: The Requirements Achieved by The Schemes

The abbreviations that used: U: Universal, I: Individual, E: Eligibility, PMV: Prevent Multiple Voting, C: claimed achieved requirement, S: supposed achieved requirement, A: attacks found.

CHAPTER 4

CONCLUSION

The development of e-voting has been introduced in many areas (governmental election, company manager) election according to different techniques since 1900. Therefore, it is considered a huge field of study. The process of counting votes electronically take place from a long time ago, but polling process was conducted in some countries in various political and non-political environments and in the other countries are still under discussion. Different voting techniques have been used such as criticized DRE voting machines and erroneous punch cards systems but the voting process via the Internet is a process that focus of attention and a large concern.

In this section we conclude about the three proposals. DynaVote has developed the blind signature scheme through which the identity of the voter cannot track or trace and thus ensure that the voter's identity remains reserved, which is one of the most important factors that help the voter to cast his vote freely. DynaVote have developed other factors for the integrity of the election process, such as the ability to vote more than once in order to overcome the problem of coercion, as well as relying on the bulletin board at each stage of the voting process, in order to ensure the safety of the transfer of voter ballot from one phase to another.

DynaVote can be used in electronic voting systems via the Internet because it achieves most of the electronic voting requirements that have been mentioned, in addition to the following requirements such as mobility, efficiency, dispute freeness and scalability.

But nevertheless it has some flaws or shortcomings that appear when the plots ballot generator with the key and the counter conspire and work together. However, in general DynaVote is an electronic voting system via the Internet offers a great success.

End-2-End is an electronic voting system via the Internet (as mentioned previously) uses homomorphic encryption with ElGamal algorithm, where all votes are collected after the casting process, so any attacker cannot see what the voter has chosen. Thus, you will keep on hiding the identity of the voter because the decryption process will be one of the mass of a single process and not every vote separately in addition to sending a message at every stage to voters via secure channels (like sms) to satisfy the voter that his vote had arrived counting stage safely and reliably. We can say that End-2-End has achieved a lot of electronic voting requirements including privacy, security and transparency.

Receipt-Free has been designed mainly to be another offer to the traditional voting systems because it did not contain any stage to ensure the identity of the voter. Also it has some disadvantages; some of the electronic voting requirements have not been achieved such as individual verifiability, democracy and coercion resistance. So we suggest one may not to use this algorithm in political elections, but he can benefit from it in small elections, such as student elections to choose their representatives in a college, where he can control the voting that takes place in this environment.

REFERENCES

1. **Adida B., Marneffe O., Pereira O., Quisquater J., (2000),** *"Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios. Electronic Voting Technology"*, Workshop on Trustworthy Elections Cambridge University Press, UK, vol. 1, no. 3, pp. 41-45.

2. **Website of federal voting assistance program us.** *"http://www.fvap.gov/"*. (Data Download Date: 01 April, 2014).

3. **Ronald L. R., (2000),** *"Perspectives on End-2-End Voting Systems"*, NIST E2E Workshop, Brazil, vol. 3, pp. 202-207.

4. **Lambrinoudakis C., Gritzalis D., (2002),** *"Building a Reliable E-Voting System"*, Functional Requirements and Legal Constraints. Proceedings of the 13th International Workshop on Database and Expert Systems Applications, Singapore, pp. 435.

5. **Okediran O. O., Omidiora E. O., (2011),** *"A Framework For A Multifaceted Electronic Voting System"*, International Journal of Applied Science and Technology, USA, vol. 1 pp. 4.

6. **Khasawneh M., Al-Jarrah O., (2001),** *"Modeling and Simulation of a Robust E- Voting System"*, In Communications of the IBIMA, Norway, vol. 2, pp. 228-234.

7. **Allen C. G., (2011**), "*http://www.elections.nsw.gov.au/__ data/ assets/ pdf_ file/0004/93766/July_2011_Final_ACG_iVote_Report_ELE01-C_Final.pdf "*, Evaluation of Technology Assisted Voting Provided at the New South Wales State General Election, USA, (Data Download Time 01 September. 2014).

8. **Avgerou C., Ganzaroli A., Poulymenakou A. and Reinhar N., (2007),** "*http://www.ifipwg94.org.br/ful lpapers/R0098-1.pdf"*, Proceedings of the 9th International Conference on Social Implications of Computers in

Developing Countries, Sao Paulo, Brazil, (Data Download Date 11 October. 2014).

9. **Avgerou C., Ganzaroli A., Poulymenakou A., Reinhard N., (2009),** *"Interpreting the Trustworthiness of Government Mediated by Information and Communication Technology: Lessons From Electronic Voting in Brazil"*, in Information Technology for Development, Brazil, vol. 15, no. 2, pp. 133-148.

10. **Douglas W. J., (2010),** *"Progress and Pitfalls of Election Technology"*, Indirect Recording Electronic Voting Systems, International Foundation for Electoral Systems, New York, USA, pp. 77-78.

11. **Clifford S., (2006),** *"www.osce.org/odihr"*, Expert Visit on New Voting Technologies: Local Elections, OSCE Office for Democratic Institutions and Human Rights, USA, pp. 144-160.

12. **Matt B., Arel C., Sophie E., Chris K., Naveen S., Micah S., Till S., Kaping Y., (2008),** *"Source Code Review of the Sequoia Voting System"*, Technical Report, University of California, USA, pp. 71-83.

13. **Kristian G., (2010),** *"Analysis of an Internet Voting Protocol"*, Technical Report, NTNU university, Norway, vol. 3, pp. 202-207.

14. **Atsushi F., Tatsuaki O., Kazuo O., (1992),** *"A practical Secret Voting Scheme for Large Scale Elections"*, Norway, pp. 244-251.

15. **Ghassan Z., Rani T., (2007),** *"Electronic Voting Systems: Requirements, Design and Implementation"*, Computer Standards & Interfaces, India, vol. 102, no. 1, pp. 77-86.

16. **Thomas H., Charles E., Ronald L., Clifford S. t., (2001),** *"Introduction to Algorithms, Second Edition"*, MIT Press and McGraw-Hill, pp.262-268 .

17. **Josh C., Michael F., (1985),** *"A Robust and Verifiable Cryptographically Secure Election Scheme"*, In Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science (FOCS), USA, pp. 372-382.

18. **Chaum D., (1981),** *"Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms"*, Communications of the ACM, USA, pp.84–88.

19. **Chaum D., (1982),** *"Blind Signatures for Untraceable Payments"*, USA, pp. 199-203.

20. **Boyd C., (1989),** *"A New Multiple Key Cipher and an Improved Voting Scheme"*, USA, pp. 617-625.

21. **Pfitzmann A. B., Waidner M., (1991),** *"Untraceable Communication With Very Small Bandwidth Overhead in Proc. GI/ITG Conf. Communication Distributed Systems, Informatik-Fachberichte"*, New York: Springer-Verlag, pp. 451–463.

22. **Boyd C., (1989),** *"A New Multiple Key Cipher and an Improved Voting Scheme"*, Springer-Verlag, USA, pp. 617-625.

23. **David C., (1981),** *"Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms"*, Communications of the ACM, USA, pp. 181-201.

24. **ElGamal T., (1985),** *"A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithm"*, Technical Report NIST, USA, vol. 8, no. 2, pp. 123-129.

25. **Thomas H. C., Charles E. L., Ronald L. R., Clifford S., (2001),** *"Introduction to Algorithms, Second Edition"*, MIT Press and McGraw-Hill, Norway, vol. 3, no. 5, pp. 241 – 255.

26. **Juang W., Lei C., (1997),** *"A Secure and Practical Electronic Voting Scheme for Real World Environment"*, Transactions on Communications Electronics Information and Systems, China, pp. 62-63.

27. **Anjana C., Ravinder T., Manish M., (2013),** *"A Review: Data Security Approach in Cloud computing by Using RSA Algorithm"*, International Journal of Advance Research in Computer Science and Management Studies, USA, pp.232-245.

28. **Richard K., Vincent C., Timothy W., Chang S., (2001),** *"Introduction to Public Key Encryption and the Federal PKI Infrastructure"*, Singapore, pp. 186-187.
29. **Stallings W., (2010),** *"Cryptography and Network Security"*, University Press, Cambridge, Great Britain, vol. 102, no. 1, pp. 77-86.

30. **Fips P., (1995),** *"Secure Hash Standard (SHS)"*, Technical Report NIST, USA, pp. 11-15.

31. **Blum M., Feldman P., Micali S., (1988),** *"Non-Interactive Zero-Knowledge and Its Applications"*, In 20th ACM Symposium on the Theory of Computing, Brazil, pp. 103-112.

32. **Benor M., Goldwasser S., Kilian J., Wigderson A., (1988),** *"Multi-Prover Interactive Proofs: How to Remove Intractability"*, In 20th ACM Symposium on the Theory of Computing, USA, pp. 113-131.

33. **Goldwasser S., Micali S., Rackoff C., (1989),** *"The knowledge Complexity of Interactive Proof Systems"*, SIAM Journal on Computing, India, vol. 18, pp. 186-208.

34. **Michael J. R., (1995),** *"An Untraceable, Universally Verifiable Voting Scheme"*, Brazil, pp. 46-48.

35. **Fujioka A., Tatsuaki O., Kazuo O., (1992),** *"A Practical Secret Voting Scheme For Large Scale Elections"*, Theory and Application of Cryptographic Techniques, London, UK, Springer-Verlag, pp. 244–251.

36. **Miyako O., Fumiaki M., Masayuki A., Atsushi F., Tatsuaki O., (1999),** *"An Improvement on a Practical Secret Voting Scheme"*, Lecture Notes in Computer Science, Springer, USA, vol. 1729, pp. 225–234.

37. **Kim K., Kim J., Lee B., (2001),** *"Ahn. Experimental Design of Worldwide Internet Voting System Using PKI"*, USA, pp. 1-5.

38. **Juang W., Lei C., (1997),** *"A Secure and Practical Electronic Voting Scheme For Real World Environment"*, Transactions on Communications Electronics Information and Systems, UK, pp. 1-4.

39. **David C., Peter Y. A., Steve A. S., (2005),** *"A practical, Voter-Verifiable Election Scheme"*, Lecture Notes in Computer Science, Springer Berlin, Germany, vol. 3679, pp. 118–139.

40. **Stefan W., (2006),** *"A coercion-Resistant Cryptographic Voting Protocol - Evaluation and Prototype Implementation"*, Master Thesis, University of Technology Germany, pp. 201-210.

41. **Martin H., Kazue S., (2000),** *" Efficient Receipt-Free Voting Based on Homomorphic Encryption"*, Lecture Notes in Computer Science, Berlin, Germany vol. 1807, pp. 539.

42. **Kazue S., Joe K., (1995),** *"Receipt-Free Mix-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth"*, Technical Report NIST, USA, pp. 393–403.

43. **Markus J., Kazue S., Russell I., (1996),** *"Designated Verifier Proofs and Their Applications"*, Lecture Notes in Computer Science, USA, vol. 1070, pp.143–145.

44. **Zuzana R., (2002),** *"Electronic Voting Schemes"*, Ph.D. Thesis, Comenius University, Bratislava, pp.76-85.

45. **Josh D. C. B., Dwight T., (1994),** *"Receipt-Free Secret-Ballot Elections"*, Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, New York, USA, pp. 544–553.

46. **Ronald C., Rosario G., Berry S., (1997),** *"A secure and Optimally Efficient Multi-Authority Election Scheme"*, Lecture Notes in Computer Science, New York, USA, vol. 1233, pp.103.

47. **Sako K., Kilian J., (1995),** *"Receipt-Free Mix-Type Voting Scheme: A Practical Solution to the Implementation of A Voting Booth"*, Ph.D. Thesis, Comenius University, Bratislava, pp. 393-403.

48. **Cranor L., (1997),** *"Sensus: A Security-Conscious Electronic Polling System for the Internet"*, In Proceedings of the 30th Annual Hawaii International Conference on System Sciences, Wailea, Hawaii, pp. 200-205.

49. **Karlof C., Sastry N., Wagner D., (2005),** *"Cryptographic Voting Protocols: A Systems Perspective"*, In Proceedings of the 14th Conference on Security Symposium, Baltimore, vol. 2312, pp.213.

50. **Cetinkaya O., Doganaksoy A., (2006),** *"A practical Privacy Preserving E-Voting Protocol Using Dynamic Ballots"*, 2nd National Cryptology Symposium, Ankara, Turkey, pp. 5-8.

51. **Schoenmakers B., (2014),** *"Lecture Notes over Cryptographic Protocols"*, Blackie & Son, Glasgow, Great Britain, pp. 46-48.

52. **Sınak A., Sabır M., Zkan S., Yıldırım H., (2013),** *"A Secure Internet Voting Protocol Based on Homomorphic Encryption"*, Proceedings of 6th International Conference on Information Security and Cryptology, USA, pp.142-148.

53. **Martin H., Kazue S., (2000),** *" Efficient Receipt-Free Voting Based on Homomorphic Encryption"*, Lecture Notes in Computer Science, Brazil, vol. 1807 pp. 539.

54. **Kazue S., Kilian J., (1995),** *"Receipt-Free Mix-Type Voting Scheme - a Practical Solution to the Implementation of a Voting Booth"*, Southampton, Boston, pp. 393–403.

55. **Markus J., Kazue S., Russell I., (1996),** *"Designated Verifier Proofs and Their Applications"*, Lecture Notes in Computer Science, USA, pp.143-144.

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: AL-GBURI, Omar

Date and Place of Birth: 20 March 1989, Diyala

Marital Status: Single

Phone: +90 5378823139

Email: algboryomer@ymail.com

Degree	**Institution**	**Year of Graduation**
M.Sc.	Çankaya University, Information Technology	2015
B.Sc.	Diyala University, Computer Science	2011
High School	Al-Tomuh	2007

FOREIN LANGUAGES

English, Beginner Turkish.

HOBBIES

Football , Reading, Travel, Swimming.