Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

A NOVEL E-VOTING SYSTEM WITH DIVERSE SECURITY FEATURES

by
Haijun Pan

Internet-based E-voting systems can offer great benefits over traditional voting machines in areas, such as protecting voter and candidate privacy, providing accurate vote counting, preventing voter fraud, and shortening the time of vote counting. This dissertation introduces, establishes and improves Internet-based E-voting systems on various aspects of the voting procedure. In addition, our designs also enable voters to track their votes which is a very important element in any elections.

Our novel Internet-based E-voting system is based on the following realistic assumptions: (1) The election authorities are not 100% trustworthy; (2) The E-voting system itself is not 100% trustworthy; (3) Every voter is not 100% trustworthy. With these three basic assumptions, we can form mutual restrictions on each party, and secure measurements of the election will not be solely determined and influenced by any one of them. The proposed scheme, referred to as Time-lock algorithm based E-voting system with Ring signature and Multi-part form (TERM), is demonstrated to achieve the goal of keeping votes confidential and voters anonymous, as well as reducing the risk of leaking the voters' identities during the election. In addition, TERM can prevent any possible clash attack, such as manipulating voting results or tampering voters' original votes by malicious election authorities or hackers. The security performance analysis also shows that TERM provides outstanding measurements to secure the candidates' manifest on each type of

ballots during the whole election duration. TERM provides a roadmap for future fair elections via Internet.

A NOVEL E-VOTING SYSTEM WITH DIVERSE SECURITY FEATURES

by
Haijun Pan

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

January 2017

.

APPROVAL PAGE

A NOVEL E-VOTING SYSTEM WITH DIVERSE SECURITY FEATURES

By
Haijun Pan

Dr. Sui-Hoi Edwin Hou, Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Nirwan Ansari, Co-Advisor — Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. John D. Carpinelli, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Mengchu Zhou, Committee Member — Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Vincent Oria, Committee Member — Date
Professor of Computer Science, NJIT

BIOGRAPHICAL SKETCH

Author: Haijun Pan

Degree: Doctor of Philosophy

Date: January 2017

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2017

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2008

- Bachelor of Science in Electrical Engineering,
  Tong Ji University, Shanghai, P. R. China, 1999

Major: Electrical Engineering

Presentations and Publications:

Haijun Pan, E. Hou, and N. Ansari, "TERM: An E-voting System with Diverse and Secure Measurements", submitted to IEEE Transactions on Information Forensics & Security, December 2016.

Haijun Pan, E. Hou, and N. Ansari, "E-NOTE: An E-voting System that Ensures Voter Confidentiality and Candidate Privacy", Security and Communication Networks, Vol 7, Issue 12, December 2014, pp. 2335-2344.

Haijun Pan, E. Hou, and N. Ansari, "M-NOTE: A Multi-part Ballot based E-voting System with Clash Attack Protection", Proc. IEEE International Conference on Communications, London, UK, June 8-12, 2015, pp.7433-7437.

Haijun Pan, E. Hou, and N. Ansari, "RE-NOTE: An E-Voting Scheme based on Ring Signature and Clash Attack Protection", Proc. IEEE Global Communication Conference, Atlanta, USA, December 9-13, 2013, pp. 867-871.

Haijun Pan, E. Hou, and N. Ansari, "E-NOTE: An E-voting System that Ensures Voter Confidentiality and Voting Accuracy," Proc. IEEE International Conference on Communications, Ottawa, Canada, June 10-15, 2012, pp. 825-829.

Haijun Pan, E. Hou, and N. Ansari, “Ensuring Voters and Candidates' Confidentiality in E-voting Systems,” Proc. 34th IEEE Sarnoff Symposium, Princeton, USA, May 3-4, 2011, pp.1-6.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Objective

Today's society is experiencing an explosive growth in Internet-based applications and systems where a large number of social activities are performed online. However, most of the political elections in either developed or developing countries are still carried out using paper-based or touch screen voting systems where the voters need to cast their votes at a precinct. This maybe a major contributing factor in low voter turnout in elections. A reliable and secure Internet-based Electronic voting (E-voting) system would provide voters with greater convenience and more accurate vote counting process. In the meantime, the significant demand from launching a reliable and secure E-voting system has driven efforts from researchers for years. Although paper-based voting systems have been successfully deployed for a long time, the U.S. government has been looking into improving the security and accuracy of E-voting in order to avoid disputes over paper-based voting system before/during/after an election. For instance, in Volusia County in Florida on election night November 2000, the main concern was with whether or how voters' votes would be accurately counted in an election. There was a big dispute in Florida and election authorities in many counties were called to recount the votes. As we know, traditional voting machines with paper ballots inevitably yield a certain rate of misreading on the ballot. Meanwhile, using paper ballots or machine readable paper ballots could incur huge delay and human errors of tallying results. For example, in Minnesota, there was a recount and ensuing court case after Election Day because many miscounted ballots were discovered among the voting machines that were jammed. So

far, in the United States, the concept of E-voting has only been applied in some local governments' electronic voting machines. Typical electronic voting procedures include setting up electronic voting machines and casting electronic ballots by touch screen devices. During recent U.S. presidential election held from the last decade, most states were still employing the traditional voting machines or ballot scanners with a few exceptions which adopted touch screen voting machines to let voters cast their ballots electronically.

The objective of my research is concentrated on developments of Internet-based E-voting systems in the following areas:

- Novel electronic ballot design
- Reliable vote counting and vote tracking algorithms
- Information security features in E-voting system
- Cryptograph based voter privacy protection scheme
- Clash attack prevention

1.2 Challenges

The E-voting procedure raises legitimate concerns about its reliability and trustworthiness when millions of voters cast their ballots over the Internet. Meanwhile, there are still many open issues about E-voting through the Internet. Other works have reviewed many concerns about the Internet E-voting, such as reliability of software, data transmission, database systems, confidentiality of electronic votes, detection on double voting, and vote buying (Wu, 2002). These concerns are aligned with the major issues of Internet attacks

around the globe.

Before Internet-based E-voting systems can completely replace those traditional paper- based voting systems, we need to make sure that the new system will perform as efficiently as the traditional ones without any security and technical concerns. Nonetheless, here are some major challenges for researchers and authorities to deploy the Internet-based E-voting system:

First, the trust from the public and the government is the cornerstone of any Internet-based E-voting system. It also serves as an important precondition to widely adopting this kind of E-voting system. During the 2000 U.S. presidential election, the recounting of votes and disputes in Florida demonstrated that it was time-consuming and could yield a certain error rate during the vote counting procedure for the machine-readable paper ballot based voting system. Therefore, it is critical that the future developed E-voting system must be secure, effective and flawless. To increase public confidence, many states have been considering E-voting systems that provide voter-verifiable paper audit trails. Some efficient E-voting systems with higher vote counting rates have been proposed recently.

Second, the reliability of online data transactions is worrisome to many users. Online data hacking and data breach incidents have been occurring at an alarming rate. Although the newly designed E-voting system can collect ballots quickly and count votes efficiently, it still relies on the Internet as the communication medium. Numerous concerns regarding communications among voters, authorities and E-voting systems through the Internet must be addressed. The new E-voting system must be able to prevent hackers from conducting online cyberattacks such as unauthorized access to the system

and votes, maliciously jamming the data traffic resources, etc. The Internet-based E-voting system must incorporate advanced data transmission technologies and improved security algorithms to detect and deter malicious online activities.

The third critical concern is the trustworthiness of E-voting system's designer and manufacturer. Although E-voting systems vendors advertise that the whole procedure of vote casting and ballot collecting is securely monitored and guaranteed during the election, but the public still lacks the confidence of these systems and they believe that the whole election procedure should be monitored and guarded by the public themselves. In this dissertation, our research works mainly focus on developing an E-voting system without fully relying on the trustworthiness of the election authorities and voting system. We have developed the measurements to place mutual restrictions among voters and authorities and vendors to ensure a fair election.

Despite the challenges listed above, why shall we continue to develop and deploy Internet-based E-voting systems? Besides having the benefit of highly efficient vote counting process and saving resource, it offers the opportunity to achieve an out-standing performance level over traditional voting in many aspects, such as the system can allow voters to self-correct on their own voting mistakes which may result in an invalid vote, the system will prevent multiple votes from the same voter, it also shorten the time for counting and retrieving the voting result.

1.3 Assumptions

Most importantly, the political election should be conducted fairly, transparently and honestly under the supervision of the election authority. But from past experiences, voters have concerns about whether their votes were accurately counted or have been

manipulated during the election (Cranor, 1997). At the same time, source codes embedded in the voting machines are usually proprietary, the programming and processes are always under a veil of secrecy.

In this dissertation, we developed several novel E-voting models, NOTE (Name and vOte separaTed E-voting system), E-NOTE (Enhanced Name and vOte separaTed E-voting system), RE-NOTE (Ring signature based Enhanced Name and vOte separated E-voting system) , M-NOTE (Multi-part Ballot based Name and vOte separated E-voting system) and TERM (Timed-lock algorithm based E-voting system with the Ring signature certificate and Multi-part ballot form), which can better address the voting issues discussed above.

Our research works make the following tenable assumptions:

1. The election authorities are not fully trustworthy, which is a reasonable and practical concern from the public.
2. The E-voting system itself is not fully trustworthy.
3. The voters are not fully trustworthy.
4. Data transmission through the Internet is reliable. Our research will only focus on the overall picture of designing the voting protocol and voting scheme.

With these four basic assumptions, we can easily form some mutual restrictions among each party involved in the whole voting scheme, and the secure measurement of the E-voting system will not be solely determined and influenced by any of them. In addition, our proposed E-voting system model will allow voters to have more auditing capabilities in terms of vote counting and tracking during and after the election.

We have gradually introduced, established and improved the Internet-based E-voting system, including the solution to maintain candidates' and voters' confidentiality,

to protect voters' privacy, and to empower voters to do vote tracking and verification, which are three most important elements in any political election. The corresponding E-voting system model has well been illustrated to mitigate underlying issues discussed above.

A typical voting model may endow several responsibility roles of an election such as registration, vote counting, and vote tally to a single entity. Since a political election is a complicated procedure which requires security measurements at every step of the process. We will introduce a voting protocol that contains several distinguished phases: voter registration, ballot distribution, voter casting, ballot collecting, vote counting, vote tally publishing, and vote auditing.

Any voting data breach that happens in any of phases in a political election will lead to an invalid election result. In the past, many researchers mainly focused on the cryptography design or its suitability for E-voting. Our works are mainly focused on finding a practical solution to improve the E-voting system and prevent it from being hacked or manipulated by different malicious parties. Our research goal is to design an E-voting system model that would incorporate existing cryptographic algorithms so that it is secure and provisions vote audit capability instead of inventing a new mathematical cryptography method. We expect such a secure E-voting system model will also draw high participating rate from voters in the future.

In Chapter 2, we will review the background of our research in terms of cryptography and protocol. We will also introduce attacks on the existing E-voting scheme, and we also generalize the basic requirements for setting up an E-voting system with fairness and transparency.

In Chapter 3, we introduce our previous works including NOTE, E-NOTE, RE-NOTE, M-NOTE with paper ballot mode for readers' better understanding. We also introduce useful definitions which are used in TERM for the preliminaries.

In Chapter 4, we will give a detailed illustration of every step in TERM to achieve a secure and accurate voting process. Each step will illustrate a specific measurement to guarantee the voting operation.

We also give details on the mathematical and security analysis of TERM and other proposed works in Chapter 5 and 6 to provide a full detailed description of the proposed E-voting system. In Chapter 7, we will conclude our works and discuss the different aspects of E-voting system that need to be considered for elections in the future.

CHAPTER 2

LITERATUR REVIEW

2.1 Cryptography Used In E-voting System

In this chapter, we summarize related works on E-voting systems in terms of encryption algorithm. Generally, researchers have categorized existing E-voting systems into three major types: blind signature based scheme, homorphic-based scheme, and mix-nets based scheme.

Blind signature is a kind of digital signature that can be used in an E-voting system to better protect a voter's privacy (Chaum, 1983), (Chien, 2001). Several blind signature based E-voting systems have been proposed and the common main idea is to allow a signer (voter) to transmit any important voting message (ballot) anonymously. However, messages are only sealed and encrypted in one direction. It is not traceable in the sense if it is used in elections that a voter cannot reverse the encryption process when vote audit is needed.

Homorphic cryptography is the second most popular encryption used in E-voting systems (Benaloh, 1987). It allows the cipher text to carry some specific computations before an encrypted message is generated. This message can later be decrypted and audited to see whether it matches the result of the same operations performed on the original plaintext.

Consider an example in which the voter has two options {1,-1}, which stand for two different candidates in the race, and there are several voters casting their ballots in the election. The Homomorphic cryptography based scheme will calculate the sum of votes to determine the final result. If the sum of the votes for a specific candidate is larger

than 0, then this candidate is considered as the winner. Otherwise, this candidate will be considered as to concede the election. If there are more than two candidates or options in the election, then it is more complicated to determine the final result of the election by using the homomorphic based voting scheme.

This type of scheme still exhibits drawbacks such as limited scalability; usually, it can accommodate only two options for voting, but a typical poll has more than two options.

Mixnets allows messages to be encrypted with different servers and random patterns (Chaum, 1981). Ideally, it does not enforce a final encryption form except for an original encryption since the message could be encrypted infinitely.

There are, however, various issues and technical challenges associated with each type of cryptographic methods mentioned above. They must be solved before they can be implemented in E-voting systems so that voters are confident enough to vote through the Internet. In addition, a summary of practical issues in E-voting procedure such as voting manipulation, voting fraud, data transferring through the Internet, and database maintenance are discussed (Jakobsson, 2004). Still, ideas from other designs may have the influence on the current and future trend of E-voting system development.

2.2 Other Researchers' Work

Over the past two decades, there have been many papers focusing on E-voting issues as technological advances seem matured enough to warrant transition from traditional voting methods to electronic ones. Currently, the most widely used "Direct Recording Electronic" (DRE) voting system focuses on facilitating voters to cast paperless ballots on specific voting machines. In general, this type of new machines essentially replaces

paper ballots with electronic ones. Other researchers are advocating for the Internet voting, and our works are also tailored for the Internet voting environment. Several research works have focused on the voting security issues during the data encryption process.

We will categorize other researchers' work by analyzing the voting mechanism used in the voting procedure. The typical design of a voting system can be categorized into paper ballot based, electronic machine based, and the Internet-based E-voting system. We focus on online E-voting activities because we believe that "vote through the Internet" will evolve into a normal life style soon in a similar way as online banking and online shopping that the public does nowadays. We also list several works from other researchers that may have some common properties regarding security and privacy concerns.

Helios is the first online E-voting system and it is web based and offers the great flexibility for voters to vote online with open audit function (Adida, 2008). Every voter will be provided a tracking number to audit the result.

The FOO system is composed of voter, authentication authority, and counting authority in the whole voting procedure (Fujioka, 1992). This protocol also contains four phases: Initialization, Registration, Voting, and Counting phases. The author also claimed the vote check function can be achieved to ensure the vote verifiability.

Punchscan is a kind of voting scheme while the ballot is designed with top and bottom sheet of a ballot (Chaum, 2006), (Popvenuic, 2006). This system provides a voter-verifiable scheme to allow voters audit. The Pret a Voter Verification Election System has also suggested the idea of using the mix server cryptography for voters and public to randomly check and audit the ballot (Ryan, 2009). In reality, if every voter appears at the

voting booth and manually verifies and checks the plain ballot before casting, the total duration time of one voter in a voting booth will be longer than normal and will cause more delay and energy consumption in a large scale presidential election.

The Threeballot system is a paper ballot based system that the whole voting system does not need any cryptography (Rivest, 2006). If a voter selects a candidate, he/she will randomly fill two out of all three ballots for this candidate. Any candidate, who receives only one vote out of three ballots, is considered vetoed by this voter. This scheme has simplified the voting process but its lack of traceability does not meet the in-time demand from voters in today's democratic society.

2.3 Clash Attack

Besides the cryptography and voting process concerns we discussed above, the public also has concerns about corruption, fraudulent and manipulation from authorities in the election. Potential attacks from election authorities could be a big threat to true democracy. Previously most researchers only consider risks from external factors such as voting system glitch, message transmission error, etc. We also consider issues related to the voting authorities that may happen.

The concept of Clash attack on E-voting system was first introduced by (Kuesters, 2011). It is a kind of attack that can undermine the verifiability in an election and it usually involves malicious authorities who want to manipulate the voting result. When an election is held through the Internet, the verifiability which is the basic requirement for running any modern E-voting system, becomes a very important security concern. Clash attack may occur and undermine the verifiability in the election and this kind of attack usually

involves malicious authorities that could manipulate the voting result without the public knowing. It is one of the highly concerned attacks that have drawn great interest from many researchers in this field. Suppose there is at least one malicious authority that exists and actively participates in the election, if this malicious authority plays an important role, it may impact the voting result significantly. The malicious authority can generate the same receipt to different voters during the vote casting phase. Meanwhile, it can safely replace any vote with its own favor and eventually can manipulate the election without being detected during the vote auditing phase. Thus, resolving this issue is another imperative requirement for setting up a modern E-voting system. To demonstrate this, we will give a detailed example later to show how the malicious authority uses this kind of attack to manipulate the election voting result. Another possible attack from malicious authority will be discussed in Chapter 3.

Consider the following scenario:

Malicious authority such as VCC counts the vote with its favorite candidate even when voters have their vote receipt (confirmation number or tracking number) from the authority, it still could not prevent malicious behaviors happening because the authority can generate the duplicated receipt more than once to different voters when their votes are being inquired.

When voters verify his/her own vote, since he/she has the same receipt as other voters, the result he/she checked might be that of another voter who has the exact same choices on candidates in the election, and his/her vote might not be actually counted.

We need to set up a mechanism to enable voters to track their own votes and audit the voting result with great flexibility as well as anonymity. Thus how to balance between

traceability and anonymity is what we need to focus on. In our system, we introduce the concept of voting receipt with a confirmation number to serve for both tracking and clash attack prevention purpose. As we mentioned above, a voter casts their votes to the authority, VCC, along with a unique confirmation on each ballot. The confirmations could be either generated from authorities' pre-set pool or voters could revise them. During the ballot distribution phase, every voter is assigned a set of unique confirmations along with his/her assigned ballot and they can either accept or modify these pre-set confirmations. The confirmation used by voters must be unique throughout the whole E-voting system, and the voter may apply that randomly on any assigned ballot. The unique confirmation chosen by the voters will be printed on every cast ballot upon casting.

2.4 Basic Requirements for Setting Up an E-voting System

Our integrated approach takes into account when there are untrustable authorities, untrustable systems and untrustable voters.

The following important features must be absolutely provisioned for setting up a fair election:

Anonymity: To maintain the anonymity of voters, the voting protocol must allow voters to request the ballot anonymously because authorities are not assumed to be trustworthy and in fact may violate the anonymity rule in the election. A malicious authority refers to any official entity such as Election Committee (EC), BDC or Vote Counting Committee (VCC) that might turn to be malicious in the election. With a malicious authority, if the voter requests the ballot by showing his/her identity, the ballot could be associated with this specific voter by the malicious BDC and eventually the voting content could be compromised. It is crucial that personal information exposure be

limited. In one of our E-voting system models, we use a ring signature based scheme during the ballot distribution phase. We also respect both candidates' and voters' anonymity, and design a voting scheme that will dissociate each candidate and its corresponding vote on the ballot to keep candidates anonymous to possible malicious authority (VCC) during the vote counting phase, and keep voters anonymous to the possible malicious authority and the public during the ballot distribution and auditing phase. The E-voting schemes with different features will be further elaborated in Chapter 3.

Confidentiality: Who a voter voted for should be known to this specific voter only. This principle must be mandatorily applied in any political election, and this basic election rule must be strictly obeyed by all authorities and E-voting system designers. In this dissertation, we have assumed untrusted or malicious authorities in the election, and they may unlawfully and secretly link a voter's identity with his/her assigned ballot so that they could track the votes. This is an obvious consequence of protocol breach. Our goal is to completely block any unauthorized association between voters and their corresponded votes to ensure confidential deliveries of the votes to the final tally.

Verifiability: This refers to the match between any cast ballot and the corresponding voter's record in the vote auditing phase. The step of vote verification and audit is much more stringent, as voters cast the vote and election authorities must accurately count the vote. Usually, two aspects of verifiability are defined, individual verifiability and universal verifiability. Our E-voting system models can satisfy these two aspects of verifiability to allow voters to audit and verify both their own votes and the system-wide voting result easily. We have further detailed discussion in the later chapters.

Invariability: Although our previous works have well illustrated a fair election in terms of various security measurements, there are many remaining challenges throughout the whole voting process. For example, the voting protocol will be breached if a malicious authority manipulates the manifest of candidates' permutation on each type of ballot to favor a certain candidate instead of changing the votes. It is important that in a Name and vote separated E-voting system, the permutation of candidates' identities on each type of ballot should be fixed to stabilize a vote counting process for mapping all types of ballot and candidate identities during the whole election. Hence, our E-voting system models have incorporated the measurement to ensure the manifest of the candidates' permutation is unchanged from the beginning to the end of the election. We will discuss this kind of attack in Chapter 3.

Efficiency: Every voter can independently and simultaneously obtain his/her ballot before voting, and thus the entire time frame for vote collecting and counting can be greatly shortened. In general, an Internet-based election held online can reduce required resources and complete the whole voting procedure easier than the one based on any other mediums.

Multiple or repeated vote casting prevention: As the election is held through the Internet, the voting process must be able to prevent a voter from voting multiple times or voting repeatedly in different states (batches). A watchdog device is introduced in our proposed scheme can record and monitor voting transactions. This solution will be further elaborated in Chapter 3.

From Chapter 3 onwards, the term "E-voting" specifically refers to the voting process being held through the Internet.

CHAPTER 3

PROPOSED WORK AND OUR CONTRIBUTIONS

This dissertation mainly focuses on safety concerns and practical problems arising from implementing an E-voting system. In this chapter, we will present our works by showing how different technologies and protocols are integrated into our designed E-voting system model. Since the E-voting system is a complicated system, any breach in any phase in terms of voting data, security, voter privacy, voter anonymity and voting accuracy will eventually ruin the whole election, and thus we have to consider every detail in the process to ensure the operation is secure. In each section, specific measurements are applied to ensure a secured and accurate voting process and will address all safety issues and concerns as discussed in the previous chapter.

We will first illustrate terms and technologies used in our proposed schemes in the paper ballot based mode. Our proposed idea will also work in electronic mode.

Note that the variable definitions used in each section are independent of other sections.

3.1 Name and Vote Separated E-voting system (NOTE)

It is necessary to disassociate the candidate and the voter's vote to ensure the anonymity of candidates because election authorities are considered potentially malicious. If a candidate is associated with a vote on the ballot received at the VCC side, there could be an obvious weakness that can be exploited by the malicious VCC in the vote counting phase. Punchscan and Voter Pret may have the similar ballot form with the one in our proposed system but our research work presents the ballot in a different way as it features

voter's self-decision power and ballot simplicity. We list a series of sample ballots in Figure 3.1. There are β candidates and φ types of ballot ($1 \leq |\varphi| \leq \beta!$), and β checkboxes. The checkbox on each row on the ballot indicates whether this associated candidate is voted for or not. Here, we set β to 3 and φ to 6 (3!). Then six types of ballots corresponding to six unique permutations of a three-candidate race. Every ballot is distinguished by a marker (indicated by the blackened area on the top right side of a ballot) that is not visible to voters nor VCC (who can only see the type of the ballot in an encrypted form). This marker represents the type of the ballot and it can be numbers or letters depending on the encryption method. Figure 3.2 shows the ballots after they have been marked and the ballots are torn into two parts. The main purpose of using this type of ballots is allowing candidates to be anonymous when the vote is collected and counted by VCC in the vote counting phase.

We will illustrate a kind of attack while all ballots in the election only contain a fixed permutation of candidates without applying the Name and Vote separated voting scheme: If VCC is malicious and even if the vote is not shown with a candidate, when VCC counts votes of a candidate who is not a favorite of VCC, since the candidate permutation is fixed, VCC can easily locate this specific candidate's vote on the casted ballot and might have a chance to change or manipulate it during the vote counting phase. No matter how many percentages of total votes could be successfully altered by malicious authority VCC, the true and fair democracy has been breached. This is a form of attack to be deterred by applying the Name and Vote Separated E-voting scheme.

This scheme is suitable for voting through the Internet since the ballot separation procedure can be easily achieved digitally. The proposed scheme is distributed, has a

collusion-resistant mechanism for E-voting and is capable of maintaining the voters' confidentiality. We require that EC and VCC are independent of each other. EC and VCC will never be in collaboration or coercion. For example, in a U.S. presidential election, the two major parties can take the responsibility of EC and VCC separately, or the two parties will supervise EC and VCC together. At this moment, we will discuss further serious situations in the later charters.

There are numbers of differences between NOTE and other existing schemes such as punchscan and three-ballots voting system:

NOTE is based on the Internet voting media where as punchscan is designed for the DRE voting system which uses machine-scan paper ballots. The three-ballot scheme is also based on the paper ballot mode.

In punchscan and three-ballot voting scheme, there is only one official Election Authority (EA) that supervises the whole voting procedure. EA is also in charge of distributing ballots, collecting vote and counting the final tally. NOTE has two different and independent authorities, EC and VCC, the role of EC and VCC is very important in the voting procedure. They are independent and can monitor to each other.

The voters and the candidates in the punchscan and three-ballot scheme are partial auditors for the election; EA will generate at least twice the number of ballots in the election. There is also a pre-election step allowing the voters and the candidates to check the ballot. This feature is probably not needed for the Internet voting which NOTE uses since the data transaction will be at least doubled than it is expected. The goal of NOTE is to protect candidate's anonymity, reduce internet traffic during voting and offer the same secure level as provided by the traditional paper ballot mode and DRE mode.

Another serious issue is that the most of existing E-voting machines and their source codes are not published. The public has concerns about whether the voting procedure is really fair and transparent or not, and how the particular type of E-voting system is chosen. Is there any political reason or pressure in the decision of the committee in choosing a particular voting machine? This is another important reason we need to introduce NOTE. Because the candidate identity can be well protected through this protocol since it is disassociated the relationship between the candidates in the race and their received vote during the vote counting phase.

To make it easier to follow, we will illustrate the operation of our model with the paper ballot mode first and then we will discuss the same procedure in the E-voting mode. Consider a small class election that Alice, Bob and Charlie are the candidates for the president of the student association. There are several identified students as voters in the classroom; two students are in charge of counting the votes. We do not know whether these two students are good friends of Alice, Bob, or Charlie. We assume these two students who play the role of VCC are not 100% trustable. Since we have three candidates, there are 3! permutations in ordering the candidates on the ballot. The teacher may generate more than 3! ballots, each displaying one of the 3! permutations with a marker. For illustrative purposes and to be able to resolve possible contention in the final tally counting process, the teacher generates 3! distinct ballots as shown in Figure 3.1, and gives each voter one plain ballot.

Figure 3.1 shows the six possible ordering of the candidates' names on each ballot; there is also a hidden marker (covered with the black area) on every ballot to indicate/index the type of the ballot. Each type of ballots may use a set of distinct

markers. If there are *n* candidates for the election, there will be *n*! different types of ballots. When a voter casts his/her votes, the voter must separate the ballot into two parts as shown in Figure 3.2 and 3.3. Then, voters should submit the right part of the ballot as shown in Figure 3.3 to the two students acting as VCC to count the votes (the types of the ballots are hidden, and still remain unknown at this time). Voters also give the left part of the ballot to the teacher to collect. After receiving all right-part of ballots, these two students who are in charge of counting will unveil the hidden marker of every ballot as shown in Figure 3.4. Since the ballots being counted do not contain the candidates' names, implying that they do not know the names of candidates 1, 2 and 3. These two students only tabulate how many votes are for each candidate (1, 2 and 3; names of candidates remain anonymous) in each type of ballots. When this step is completed, the two students will pass the tally results of votes for each type of ballots to the teacher. The teacher will reveal the hidden marker of each type of ballot, and calculate the final result for each candidate. Note that we may actually have more than six distinct hidden marker types while there are several ones referring to a same sequence of candidates which is one type of the six (3!) in total.

The novelty of this scheme is that the ballot is separated into two parts; one part contains the list of the candidates in some random order (the identification of this order is hidden), and the other one is for the choice from voters. Note that each ballot is counted by VCC on the voter's choice and the sequence type, and the candidate names and other information are not revealed to VCC.

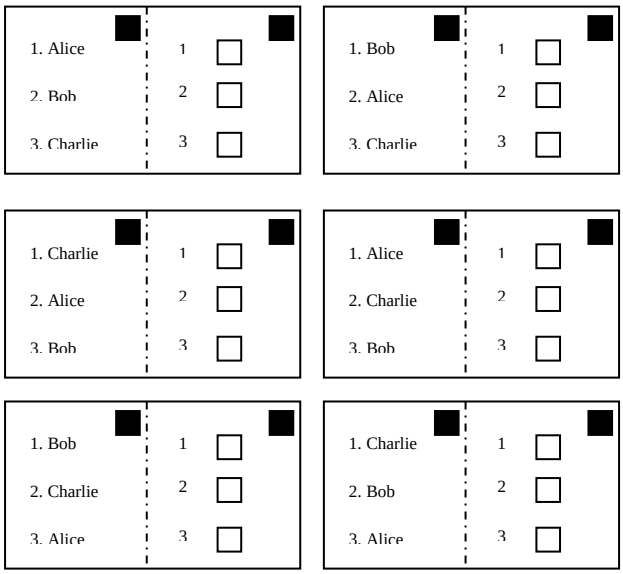

Figure 3.1 Sample ballots for a three candidate race.

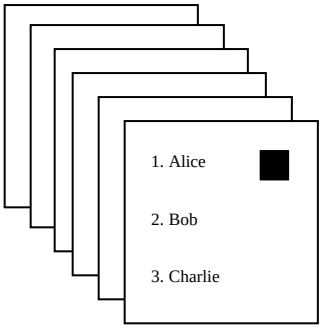

Figure 3.2 Left part of the ballots with candidate names.

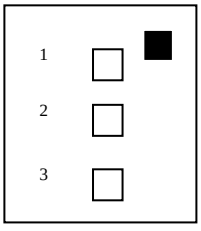

Figure 3.3 Right part of the ballots with vote choice.

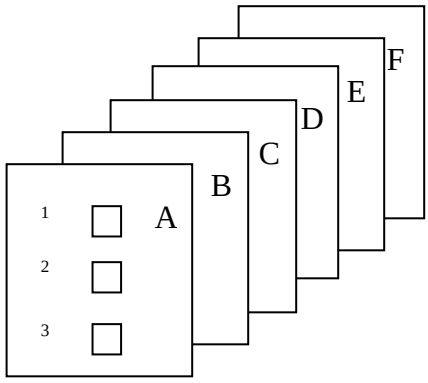

Figure 3.4 Ballots with marker type revealed when two students count the vote.

The procedure in the Internet voting mode of the proposed E-voting scheme (Figure 3.5) can be summarized as follows.

EC receives the registration from a voter; it will verify and determine whether the voter is eligible to receive a ballot to vote. After the verification, EC sends an encrypted ballot through the Internet to the voter. The original format of the ballot prior to encryption is $\{ \boldsymbol{c}, t \}$, where the array $\boldsymbol{c} = (c[1], c[2], ..., c[\mathrm{n}])$ denotes the names of the candidates, t denotes the marker which determines the type of the ballot, and n is the number of candidates running for the election.

We use the RSA algorithm (note that other encryption may be adopted) to encrypt all data for transmission (Chien, 2001). Then, EC sends the encrypted ballot $\{\boldsymbol{C}, T\}$ to the voter where $\boldsymbol{C} = (C[1], C[2], ...C[\mathrm{n}])$; at the voter side, they can use the public key to recover array $\boldsymbol{c}$ by decrypting $\boldsymbol{C}$, but the marker T remains hidden.

After casting the vote, the binary array $\boldsymbol{D} = (D[1], D[2], ..., D[\mathrm{n}])$ will be generated by the voter's choices. A "1" stands for voting for YES; a "0" stands for voting for NO. The

information of the voter ID (“voted” ack) will be sent to EC, and EC will record the status of the specific voter ID as “voted”; this is the most important information for the database at EC to prevent double voting during the election. Receipt *L* will be generated which is used for the voter to track his/her voting in the final voting tally.

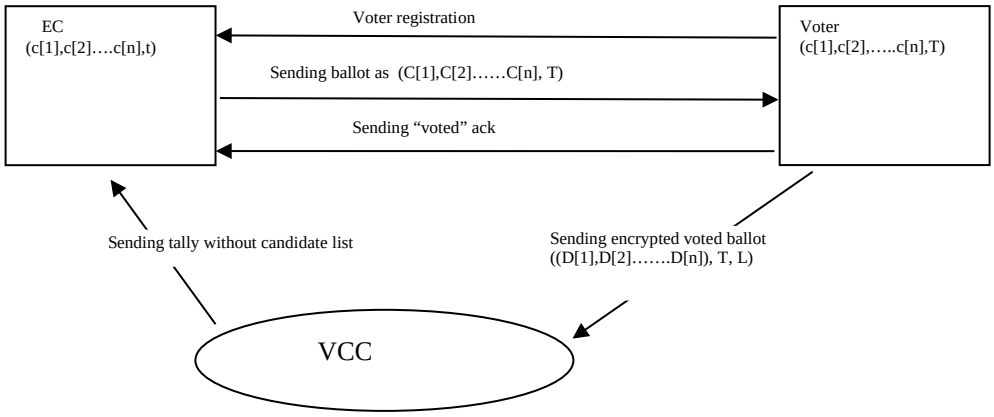

Figure 3.5 The block diagram of the voting procedure in NOTE.

The data array ***D***, marker *T,* and receipt *L* will be sent to VCC. VCC will use the key to decrypt the array ***D***, but they do not have the key to decrypt the hidden marker *T*. That is, they cannot identify the type of the ballot; the next step is to calculate the vote tally for each choice with the same type of ballot.

VCC submits the final tally of every type of ballots. After collecting the results of each type of ballots, the tally of each type of ballots will be published to the voters first. At this time, the voters still do not know the exact result since the candidates’ names are still unknown with the encrypted marker *T*. Then, the EC reveals the marker *T* on every ballot, and calculates the final tally of the votes according to the candidate names. In this case, nobody can change or undermine the result that is published already.

The numeric result of each type of ballot is shown in Table 3.1.

Table 3.1 Voting Result of Each Type of Ballots

	Choice 1	**Choice 2**	**Choice 3**
A	12	13	4
B	1	12	21
C	7	13	8
D	6	11	4
E	5	10	12
F	10	5	6

After this result is published, the teacher will reveal the exact name ordering as shown in Table 3.2.

The final election result shows that Alice gets 61 votes, Bob 36 votes, and Charlie 63 votes.

We will describe the mathematical formulation of our proposed method for voters and candidates' confidentiality.

Denote $\boldsymbol{c} = (c[1], c[2], ..., c[\mathrm{n}])$ as the list of candidates where $\boldsymbol{n}$ is the number of candidates, and $c[\mathrm{i}]$ is a string representing the name of the candidates.

Denote $t \in R$ as the marker on the ballot, where R is a set of sequence numbers, and R is greater than $n!$.

Finally, EC generates two sets of data, the ballot identification along with the corresponding list of candidates. Then, EC encrypts array **c** and marker $\boldsymbol{t}$ with different

keys.

Table 3.2 Voting Result of Each Type of Ballots after Revealing the Names of Candidates

	Alice	**Bob**	**Charlie**
A	12	13	4
B	12	1	21
C	13	8	7
D	6	4	11
E	12	5	10
F	6	5	10

Let

$$z = wy,\ p = qh,$$

where w, y, q and h are large numbers.

$$\text{Let } \alpha = (w-1)(y-1),\ \beta = (q-1)(h-1)$$

We will also find r >1 which is coprime to α, and s >1 coprime to β, and choose r' and s' satisfying the following:

$$(rr' \bmod \alpha) = 1,\ (ss' \bmod \beta) = 1,$$

where $(rr'-1)$ can be evenly divided by α, and $(ss'-1)$ can be evenly divided by β.

The data t and $c[m]$ are encrypted by the RSA algorithm as follows:

$$T = t^r \bmod z$$

$$C[m] = c[m]^s \bmod p,$$

where $m = 1,2,3,...,n$, T and $C[m]$ are the encrypted data received at the voter side, and its private key and public key are all held by EC; the system will automatically filter this factor and the voter can only read the vote option and cast his/her votes accordingly.

After casting the vote, the system will send the vote $(D\ [1], D\ [2],..., D\ [\mathrm{n}], T, L)$ with the value of marker T to VCC. $\boldsymbol{D} = (D[1], D[2],..., D[\mathrm{n}], T, L)$ is an array of binary value generated by the system at the voter side, where "0" means voting NO, and "1" means voting YES. *L* is the voter's verification key to let the voter verify his/her vote in the final result that whether the vote is counted already.

When VCC receives the array ***D***, it will count the tally of the votes with the same group of *T*.

It will be decrypted with

$$d[m] = D[m]^{s} \bmod p$$

While VCC does not have the private key of (r', z),

$$R_T = (\sum (d[1], d[2],..., d[n]), T)$$

$$= ((\sum d[1], \sum d[2],..., \sum d[n]), T)$$

$$Q_T = ((\sum L), T)$$

Here, R_T is the tally of the ballot whose marker is *T* with the same permutation. $\sum (d[1], d[2],..., d[n])$ means the sum of each option's votes. After the counting procedure, each R_T with the same group of marker *T* will be re-transmitted to the public board or EC; the public will get the tally results while *T* still remains encrypted. The next step is to publish the tally result; EC will use the private key of (r', z) to decrypt $t = T^{r'} \bmod z$. Then, it will count the final tally by comparing the value of the marker t with the matched

candidate list. Q_T is the database created for the voters' information, and the voters may check and verify the cast votes at EC.

When the marker of each type of ballots is clarified and published, the whole tally summary can be calculated accurately by EC under the public supervision.

As explained, we have addressed the problem exhibited during recent elections in the past few years. The proposed method ensures voters and candidates' confidentiality, safeguarding fair democratic election.

NOTE will prevent problems caused by malfunctions at the E-voting system with a centralized vote counting authority. It is a novel distributed election model. EC and VCC are independent of each other to ensure absolute fairness. In past elections, many contentions happened during the vote counting period, and people always have the concern on something unknown either in the voting machine or the software. NOTE has mitigated both of these concerns with decentralized counting procedure to ensure the absolute independence and voters and candidates' confidentiality.

3.2 Enhanced Name and Vote Separated E-voting system (E-NOTE)

E-NOTE provides an extra measurement to ensure the accuracy of the vote tallying results that would prevent VCC from malicious behaviors. Besides the issues addressed in NOTE, many other issues emerge while the election is held through the Internet.

Voter confidentiality is one of the biggest concerns in elections and many researchers have explored this issue for a long time. If EC colludes and shares the ballot distribution information to other authorities, then voter confidentiality may be compromised. If EC is corrupted, it can form the relationship between a ballot and a

specific voter who cast the vote. Typical voting model endows the responsibility of the election authority including registration, vote counting and vote tally. We delegate Ballot Distribution Center (BDC) to be solely responsible for ballot distribution, which is necessary to disassociate the link between voter's identity and his/her distributed ballot. BDC, VCC, and EC are independent of each other, and our proposed scheme prevents them from collusion. Besides BDC, we define several distinguished phases more specifically in the E-voting procedure as follows: voter registration, ballot distribution, voter casting, ballot collecting, vote counting, vote tally publishing and vote auditing.

Comparing with NOTE, E-NOTE improves two levels of privacy measurements to reduce the risk of voter privacy leakage, collusion among voting authorities, and mistaken vote counts. Meanwhile, E-NOTE provides a new platform to ensure a fair E-voting environment to address voter fraud issue. In order to prevent voters from disputing their votes, we introduce a hardware called watchdog device to record all voting transactions at voters' end. The watchdog device is issued by EC when the voter registers and get verified to vote at EC.

We illustrate E-NOTE scheme with an example of using paper ballots to help readers understand the concept of the scheme and the different from NOTE. Figure 3.6 shows the flow of information between voters and voting authorities. There are three election authorities: EC, BDC, and VCC. The responsibility of BDC is separating the ballot distribution duty from EC. The EC certifies the voter's eligibility and issues him/her an electronic certificate, and the voter can obtain the ballot from BDC using EC's electronic certificate instead of voters' identity. Other than the certificate, a voter does not need to show any identification to BDC and obtains the ballot anonymously. There is

no linkage between the certificate and the voter's identity. Hence, this method ensures voter confidentiality and privacy.

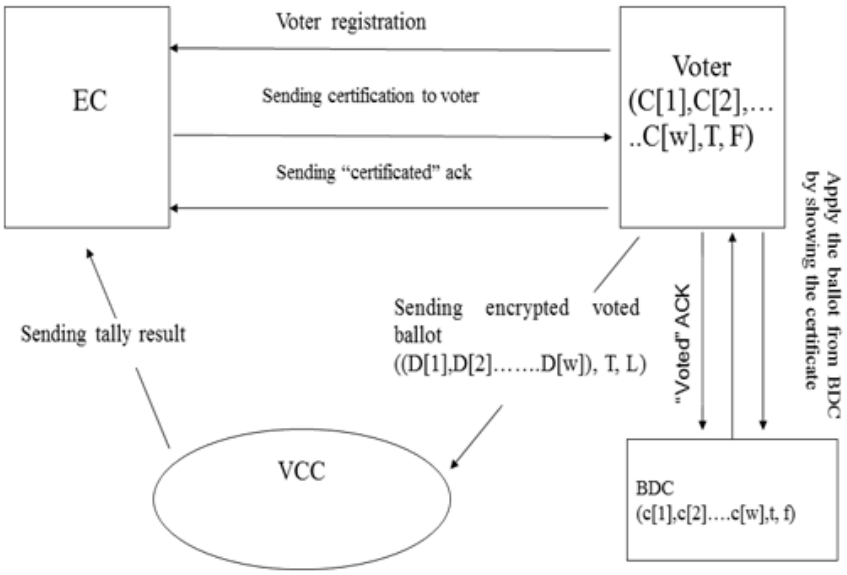

Figure 3.6 The block diagram of the voting procedure in E-NOTE.

The watchdog device records all data transactions carried out during the voting process (certificate request and issuance, ballot request and issuance, and vote submission), thereby eliminating the possibility of voter frauds, such as requesting the ballot or voting more than once.

For example, voter Jessica is one of the voters in a small class to vote; the voting procedure conducted in E-NOTE for her can be described as follows:

Step 1: Jessica goes to A1 for registration where she receives a certificate card from A1.

Step 2: Jessica goes to A2 and shows her certification card. Note that Jessica does not have to show her identity to A2. A2 generates more than 3! types of ballots (since we

have 3 candidates), and each ballot has one of these 6 (3!) permutations and a marker indicating its type. A2 checks Jessica's certification card to verify that she is eligible to vote and that her thumb is not marked (indicating she has not voted, and this is equivalent to recording the transaction in the watchdog device). When that is confirmed, Jessica receives the ballot and gets her thumb marked (the mark simply indicates Jessica has received the ballot and is assumed to remain intact during the whole election). The first two steps are used to protect voter confidentiality and privacy as there is no linkage between the certificate and the ballot.

Step 3: The ballot has a carbon copy, and when Jessica marks and submits her ballot, the carbon copy is retained by her (this is equivalent to recording the transaction in the watchdog device). The main purpose of this step is to prevent voter frauds. If Jessica claims there is a problem with her vote, she can use the carbon copy of her ballot to seek help from authorities for further investigation. In the E-voting mode, the watchdog device is used to record all the transactions in the voting process.

Step 4: This is similar to the step discussed in NOTE, Jessica tears the ballot into two parts and casts them into two different boxes. The part containing Jessica's vote is given to A3 and the part containing the candidate's name is given to A1.

Step 5: A3 counts the received ballots once the voting is completed. Since there are 3! types of ballots and the ballots do not contain the candidates' names, A3 only tallies the results based on the ballot type.

Step 6: A3 publishes the tally results of each type of ballots to the public on the blackboard and passes these results to A1.

Step 7: A1 reveals the candidate's name for each type of ballots, and the final

count for each candidate is obtained.

Next, we illustrate the mathematical formulation of E-NOTE. Suppose there are n voters and w candidates:

A_i: Voter i's batch code

B_i: Voter i's date of birth

U_i: Voter i's identity number

G_i: Voter i's gender

D_i: Voter i's choice

$C_{i:}$ Candidate's name where i is from 1 to n.

M: the certificate given to voters by EC; the voter needs to show this certificate to BDC to obtain the ballot from BDC.

Let $r\ r'$ mod $(p\text{-}1) = 1$ and $s\ s'$ mod $(p\text{-}1) = 1$

where p is a large prime number and r is in the range from 1 to $(p\text{-}1)$ with gcd$(r, p\text{-}1) = 1$. We denote r as the private key of EC and r' as the corresponding decryption key of EC. Denote s as the private key of voter i, and s is in the range from 1 to $(p\text{-}1)$ with gcd$(s, p\text{-}1) = 1$. s' is the corresponding decryption key of voter i.

Step 1: Voter i sends data array (U_i, G_i, B_i, A_i) to EC for registration. After EC's verification, EC uses the three-pass encryption algorithm and sends the message:

$$(E\ (r, M), F) = (\ M^{\ r} \bmod p, F\ )$$

to Voter i. Before the election, the authorities will initialize every watchdog device with a set of data. These data are used for securing and monitoring voters' online voting behaviors. F is used for the watchdog device to verify that this packet is really from EC, and $E(.,.)$ is the encryption function. This step shows that EC sends the certificate M to

Voter *i*. If *F* matches the data stored in the watchdog device which is used for verifying the packet authenticity, the voting process can proceed. The voter does not have access to information in the watchdog device; only the authority can review and check the watchdog device upon request.

Voter *i* receives the packet from EC, and encrypts it with his/her own private key *s*, resulting in the following data:

$$(E(s, E(r, M)), F) = (E(r, E(s, M)), F) = ((M^{r})^{s} \bmod p, F) = (M^{rs} \bmod p, F)$$

This is sent back to EC to decode the packet with its key *r'* by the decryption function $Z(.,.)$.

$$Z(r', E(r, (s, M))) = E(s, M) = M^{s} \bmod p$$

Denote Φ as the shared key from BDC to EC; Φ is in the range from 1 to p-1 with gcd(v, p-1) = 1 and v' is the corresponding decryption key of BDC. Then, EC uses the BDC's shared key v to encrypt the certificate again, and the certificate becomes.

$$E(\Phi, E(s, M)) = M^{s\Phi} \bmod p$$

Finally, EC sends this certificate to the voter again.

Step 2: Voter *i* receives and decodes the certificate with his/her key *s'* as

$$Z(s', M^{s\Phi} \bmod p) = M^{\Phi} \bmod p = E(\Phi, M)$$

and then sends $E(\Phi, M)$ to BDC to show that he/she has the certificate from EC, and it is encrypted with the key that is only shared between BDC and EC.

BDC uses the same method to decode

$$Z(\Phi', E(\Phi, M)) = M^{\Phi\Phi'} \bmod p = M$$

Then, BDC distributes one ballot to Voter *i*. This protects the voter's identity.

Step 3: Denote $c[i]$ as the name of the i^{th} candidate and the array $\boldsymbol{c} = (c[1], c[2], \ldots, c[w])$ as the packet containing the list of names of all the candidates. Denote $t \in X$ as the marker on the ballot, where $X = 1, 2, \cdots$ is a set of sequential numbers, and $|X|$ is greater than $w!$.

BDC generates two sets of data, the ballot identification along with the list of candidates. BDC encrypts the array $\boldsymbol{c}$ and marker $\boldsymbol{t}$ with different keys.

Let

$$\alpha = (k-1)(k'-1), \beta = (y-1)(y'-1)$$

where $z = kk'$, $q = yy'$, and k, k', y and y' are large numbers.

We choose $1 < h < \alpha$ and $1 < v < \beta$ such that $\gcd(h, \alpha) = 1$ and $\gcd(v, \beta) = 1$. h' and v' are chosen so that they satisfy the following:

$$(hh' \bmod \alpha) = 1 \text{ and } (vv' \bmod \beta) = 1,$$

where *h'* and *v'* are the multiplicative inverse of $h \bmod \alpha$ and $v \bmod \beta$, respectively.

The marker t and the array $\boldsymbol{c}$ are encrypted as follows:

$$T = t^h \bmod \alpha$$

$$C[i] = (c[i])^v \bmod \beta$$

where $i = 1, 2, \ldots, w$, T and $C[i]$ are the encrypted data received at the voter side with different keys, *h* and *v*, respectively. BDC holds both the private key exponent *h'* and the public key exponent *h*, i.e., VCC cannot decrypt the marker *T*.

Step 4: After the voter submits the vote, the system sends the vote packet $(D[1], D[2], \ldots, D[w], T, L)$ with the value of marker $\boldsymbol{T}$ to VCC. The array $\boldsymbol{D}$ is encrypted

with the public key exponent *v*. It is a binary array generated by the system at the voter side and can be decrypted at the VCC side with the VCC's private key *v'*. *L* is the voter's voting receipt to let the voter verify his/her vote to ensure that the vote is counted properly.

Step 5: VCC receives and decrypts the array ***D***, and then the votes are tallied for each type of ballot based on *T*. The array ***D*** is decrypted as follows:

$$d[i] = D[i]^{v'} \bmod \beta$$

where *i* is from 1 to *w*.

While VCC does not have the private key *h'*,

$$TALLY_T = (\sum (d[1], d[2], ..., d[w]), T)$$

$$= ((\sum d[1], \sum d[2], ..., \sum d[w]), T)$$

$$VER_T = ((\sum L), T)$$

Here, $TALLY_T$ is the tally of the ballots with the same marker *T*. $\sum (d[1], d[2], ..., d[w])$ represents the sum of the votes for every candidate, i.e.,

$$(\sum d[1], \sum d[2], ..., \sum d[w])$$

where $\sum d[j]$ represents the number of votes casted for candidate *j*. For example, if there are three types of ballots with the sequences of candidates "Alice, Bob, Charlie", "Alice, Charlie, Bob" and "Charlie, Alice, Bob". VCC will summarize the result of each individual type of ballots with the same sequence which is marked by *T*. $\sum d[1], \sum d[2], ..., \sum d[w]$ shows the number of votes that each candidate receives for one type of ballot. VER_T will be stored in a database used for the voting receipt storage. It allows voters to visit and track the votes they have voted.

Step 6: After VCC finishes counting the votes, each $TALLY_T$ with the same marker *T* is published to the public and transmitted to EC.

Step 7: EC uses the private key to decrypt $t = T^{h'} \bmod \alpha$ and tallies the final results.

When the content of the marker for each ballot type is reviewed and published, the final votes can be tallied accurately by EC under the public's scrutiny.

Since E-NOTE is set up to prevent VCC or other possible hackers from changing the vote results. Consider an example that the malicious authorities/hackers are trying to subvert or change the election voting results under the table.

We define *x* as the event that the malicious authority or the hacker guessed the location of their favorite candidate on the ballot correctly. Define *P*(*x*) as the probability that this can be done. If there are *w* candidates, then there are *w* outcomes for this event and *P*(*x*) is 1/*w*.

We define *y* as the event that the malicious authority or the hacker guessed the sequence of candidates on the ballot correctly. Define *P(y)* as the probability that this can be done and there are *w*! types of the ballot, and therefore

$$P(y) = \frac{1}{w!}$$

The entropy of *X* and *Y*,

$$H(X) = -\sum_w p(x) \log\ p(x)$$

$$H(Y) = -\sum_{w!} p(y) \log\ p(y)$$

Using the definition in information theory, we know that the mutual information relationship is as follows:

$$I(X,Y) = H(Y) - H(Y|X)$$

$$I(X,Y) = H(X) - H(X|Y)$$

From our definition and assumption, if the permutation of the list of candidates on the ballot is guessed successfully by VCC or other hackers, then VCC's favorite candidate's position on the ballot will be known. The conditional entropy is:

$$H(X|Y) = 0$$

This conditional entropy is 0 since the order of the candidates on the ballot is already known.

From the relationship between the mutual information and conditional entropy:

$$I(X,Y)=H(X)-H(X|Y)$$

$$I(X,Y)=H(Y)-H(Y|X)$$

Since $H(X|Y)=0$,

$$H(Y)-H(Y|X)=H(X)$$

$$H(Y|X)=H(Y)-H(X)$$

From above, we have:

$$H(Y|X) = H(Y) - H(X)$$

$$= -\sum_{w!} p(y) \log\ p(y) - \left(-\sum_{w} p(x) \log\ p(x) \right)$$

$$= w * \frac{1}{w} log \frac{1}{w} - w! * \frac{1}{w!} log \frac{1}{w!}$$

$$= log \frac{1}{w} - \log \frac{1}{w!} = log\ (w-1)!$$

Figure 3.7 shows the relationship between the number of candidates and the

conditional entropy. The conditional entropy can be interpreted as how likely the malicious authority VCC or the hackers can guess the permutation of the candidates set on the ballot if the position of their favorite candidate on the ballot is known. The number of candidates varies from 2 to 50 in Figure 3.7 and a larger value in the conditional entropy indicates there is more uncertainty in determining the permutation of the candidates on the ballot.

Consider the special case when there are only 2 candidates running in the election, i.e., w=2. When VCC or the hackers guessed the position of their favorite candidate successfully on one ballot, the position of the other candidate on the ballot is also known. This means that the conditional entropy $H(Y|X)$=0 as shown in Figure 3.7.

In large-scale elections where there are a large number of ballots, we can consider the probability of picking any ballot type as being equal. If we have η types of the ballot, the probability of any specific type ballot being selected is $1/\eta$.

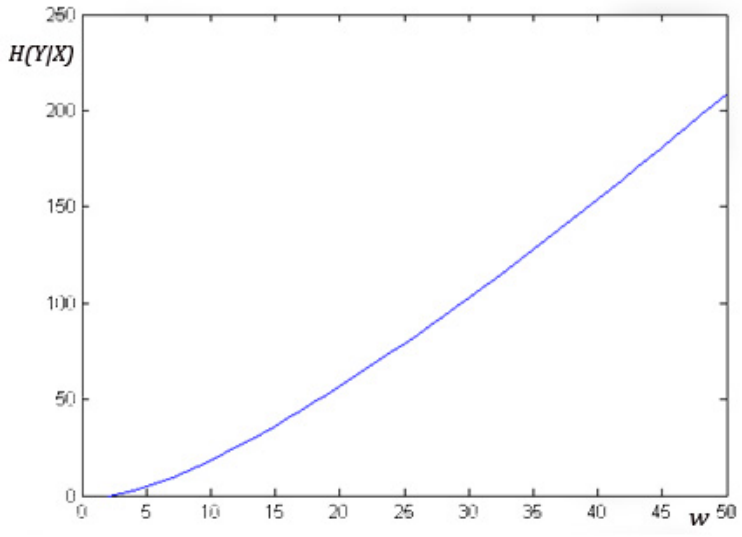

Figure 3.7 The relationship between the number of candidates and the conditional entropy.

Consider the case that the malicious authority VCC or the hacker guessed their favorite candidate on one of the ballots successfully. When the malicious authority or the hacker obtains the next ballot, the event τ is defined as the successful guess on the next ballot and the probability $P(\tau)$ of guessing successfully the same candidate is:

$$P(\tau) = \frac{1}{w!} * 1 + \left(1 - \frac{1}{w!}\right) * \frac{1}{w}$$

$$= \frac{1}{w!} + \left(\frac{w! - 1}{w!}\right) * \frac{1}{w}$$

$$= \frac{w + w! - 1}{w! * w}$$

A plot of Equation above is shown in Figure 3.8. The values of $P(\tau)$ for selected values of w are tabulated as Table 3.3.

In the worst case scenario where the cryptography method has been hacked by the malicious authority or the hacker, our E-NOTE scheme can still protect the candidate's privacy. $P(\tau)$ will drop from 1 to 0.04 if there are 25 candidates in the election.

Besides the candidates for the presidency, a national election ballot will have candidates of senators, governors and local officers. The total number of choices in a ballot can be around 25 and we can add another 25 more "virtual candidates" in the actual communication packet to decrease the probability of being guessed to 0.02. Voters will not see the "virtual candidate" while they vote and this will enhance the security in the election.

Table 3.3 Comparison Table on Different w Affecting to the Probability $P(\tau)$

W	***P(τ)***
5	0.26
10	0.10
20	0.05
25	0.04
30	0.03
50	0.02

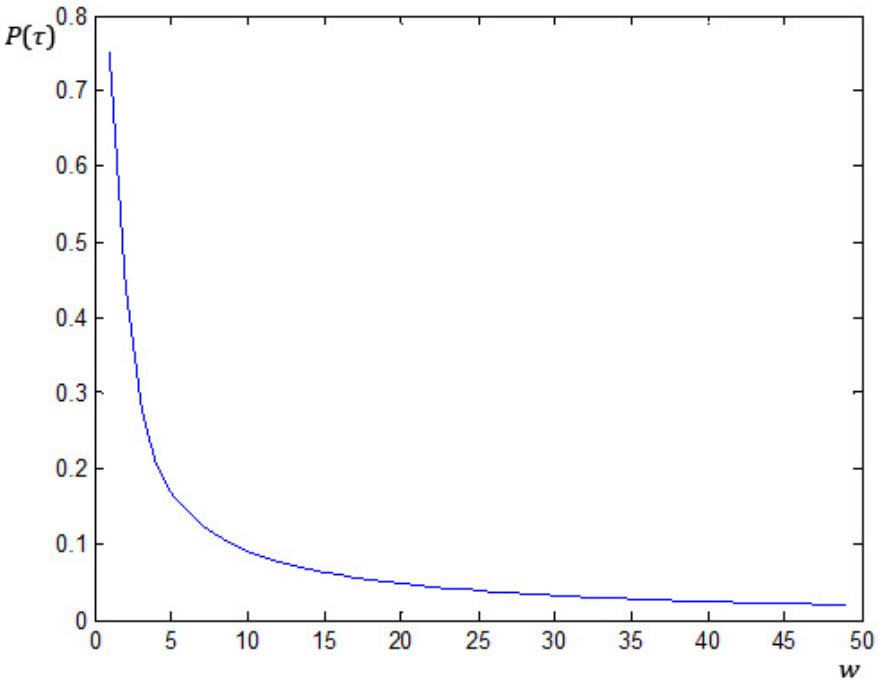

Figure 3.8 The relationship between $P(\tau)$ and w.

3.3 Three-pass Based Enhanced Name and Vote Separated E-voting System

The three-pass algorithm is a kind of encryption used for two parties' communication. In the voter registration phase, EC receives the registration request from the voter; EC verifies and determines whether the voter is eligible to receive the certificate. EC uses a three-pass cryptographic algorithm to encrypt the certificate and sends it to the voter. The voter uses a three-pass cryptographic algorithm to encrypt the certificate with his/her own key. Then, the voter sends the encrypted certificate (which is encrypted twice with different keys from EC and the voter) back to EC. When EC receives the encrypted certificate from the voter, EC decodes it with the voter's own key and then applies the BDC's key (which is only shared by EC and BDC) on the encrypted certificate, and sends it back to the voter again. The voter receives the certificate and uses her own key to decode the encrypted message. After this step, the certificate is only encrypted by the BDC's key. The voter sends the encrypted certificate which is only encrypted with the BDC's key to BDC. BDC uses its private key to decode the certificate and verify the certificate is from EC. If the verification is positive, the voter is eligible to receive a ballot.

This procedure could well protect voters' identity from being leaked if the voting authority is malicious.

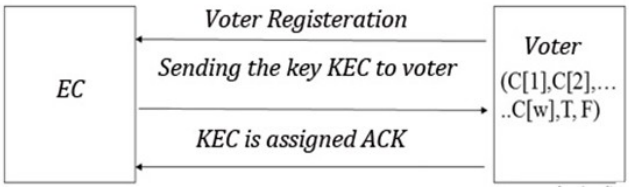

Figure 3.9 The block diagram of the ballot distributing process in three-pass cryptography based E-voting system.

3.4 Ring Signature Based Enhanced Name and Vote Separated E-voting System (RE-NOTE)

The ring signature algorithm is designed to allow a group member to sign a message anonymously among the whole group. There are several possible signers and one actual signer to form a valid ring signature. The actual signer is the member who signed the signature in the whole group, the possible signer is the member who did not sign the signature but belongs to this specific group with the actual signer. Although the actual signer is anonymous, he/she still can be identically recognized to this certain group. According to the definition of the ring signature, the verification function must be a one to one collision-free mapping function. In the ballot distribution phase in RE-NOTE, we set a group of voters to create a ring signature.

We need to emphasize that the ring signature scheme is similar to the group signature scheme except that the former one has no group manager. The advantage of the ring signature is that no centralized controller is involved in the signature. If we incorporate a group manager to form a signature in the scheme, there will be another potential concern about trustworthiness since we consider the reality that the election authorities are not fully trustable. Ring signature is pretty well suitable for our work RE-NOTE during the ballot distribution phase since it can provision voters' anonymity and overcome the complicated situation when there is a corrupted or malicious authority involved in the whole voting procedure during the election.

An E-voting scheme not equipped with the ring signature may result in the following attack scenario: Voter Jack is being verified at EC and is going to BDC to get a ballot. If BDC is malicious and Jack does not make himself anonymous, BDC could track Jack's assigned ballot and get to know the content of Jack's vote. This obviously

ruins the privacy and confidentiality policy of any political election. By applying ring signature, it is more secure for voter Jack to show his eligibility rather than his identity to request a plain ballot from BDC.

The Ring signature scheme is designed to allow a group of members to sign messages while remaining anonymous among the whole group. For the ballot distribution procedure in RE-NOTE, the following assumptions are made:

Suppose we have *l* members in a group to create the ring signature. Permutation on each input: for each *i*, $1 \leq i \leq l$, for any fixed and distinct value of all the inputs IN_i, the verification function $CK_i()$, and the value *rv* is one to one mapping plus collision-free from the input IN_i to the output OUT_i which means we will get a unique output with each different input. Neither voting authorities nor voters are fully trustable. This model can prevent authorities' fraud from among them, including the clash attack, and it has other advanced features such as voter identity recognition. The ring signature and RSA are the main cryptographic methods used in RE-NOTE.

The basic operation procedure of the ring signature scheme is described below:

There are *l* members forming a group to perform a ring signature. Each member of the group has a pair of keys, public key (Pki) and private key (Ski): (Pk_1, Sk_1), (Pk_2, Sk_2),…. (Pk_l, Sk_l). A group member A_s, can create a signature *rv* by using the ring signature cryptography. Under the definition of ring signature, anyone may check the validity of a ring signature by using the signature *rv*, the message *m*, and the public keys involved $Pk_1, Pk_2, \ldots, Pk_l$.

Here we need to clarify the notion of the actual signer and the possible signer. The actual signer is a signer of the signature on a message *m*, and the possible signers are

a set of members in the ring group who may sign the signature. Each signature will be signed by one actual signer only, and this actual signer belongs to a group of several possible signers.

Denote the group with *l* members to form a ring signature scheme. Each group member A_i has the public key PK_i to form a function $f(.)$. Since each member A_i has its own private key SK_i , this member is the only one who knows how to get the reverse function: $f^{-1}(.)$

1) Generating a group signature:

The actual signer: member A_s is given the message *m* and a set of public keys from the other members in the group: $Pk_1, Pk_2, \ldots . Pk_l$.

A_s picks a random value *rv,* uniformly-distributed in {0,1}, also randomly picks x_i as a generator for all the other ring members A_i (1≤i≤*l*, i≠*s*), then computes:

$$y_i = f\ (x_i)$$

Since there is a unique value y_s that satisfied the equation $CK_i(y_{1,}, y_2, y_s, \ldots, y_l) =$ *rv*, where $CK_i()$ is the function used for the verification step. *rv* is the initialization value since we already know y_i (1≤*i*≤*l*, *i*≠*s*).

Once y_s is calculated and found, an actual signer A_s can compute:

$$x_s = f^{-1}(y_s)$$

2) Form the ring signature:

The ring signature will be signed with the format below:

$$(Pk_1, Pk_2, \ldots . Pk_l, rv\ , x_1, x_2, \ldots x_s \ldots . x_l, m)$$

3) Check the signature:

Any verifier can verify an alleged signature on the message as follows:

Apply the function $f(.)$. to compute $y_i = f(x_i)$ for each i=1,2,…l,:

$$CK_i(y_{1,}, y_2, y_s, \dots, y_l) = rv$$

If the equation above is satisfied, the verifier accepts the signature as valid.

During the ballot distribution phase, we assume there is a group of several voters creating the ring signature, according to the definition of the ring signature, the verification function ψ(.) which is used for verification purpose is a one to one mapping with collision-free function.

We focus on one of the basic foundations of a fair election which is to maintained voters confidentiality and anonymity. In the past years, many researchers have explored this topic on different levels. The goal of our research is to setup an E-voting model for the future. The voter's confidentiality and vote verification are two of the most important issues and challenges in elections. If EC and BDC collude and share the ballot distribution information, then voter's confidentiality can be compromised. If the EC is corrupted, the relationship between voters and a specific ballot for each voter can be rebuilt and linked. At the final step during the election, if the authority made the faked or duplicated receipt to different voters, the clash attack could be applied successfully.

We will adopt ring signature's basic property and clash attack solution to achieve a better explanation to readers on our scheme RE-NOTE. Figure 3.10 shows the information flow between voters and voting authorities in the election. We still have three election authorities: EC, BDC, and VCC. Eligible voters will be divided into several groups to reduce the possible ring signature size and improve the calculation efficiency. After certifying the voter's eligibility, EC issues this voter a key of EC that is only shared among BDC and all other eligible voters to form the ring signature. Every voter will

choose a random number to sign a ring signature to keep himself anonymous by showing the signature. Once BDC verifies that the signature is valid, they will acknowledge that this voter is eligible to get a new plain ballot. Apart from the ring signature, each voter does not need to show any identification to the authority (BDC) and there is no linkage between the signature and voter's identity either. Therefore, this ring signature method definitely ensures voters confidentiality and privacy at the ballot distribution step.

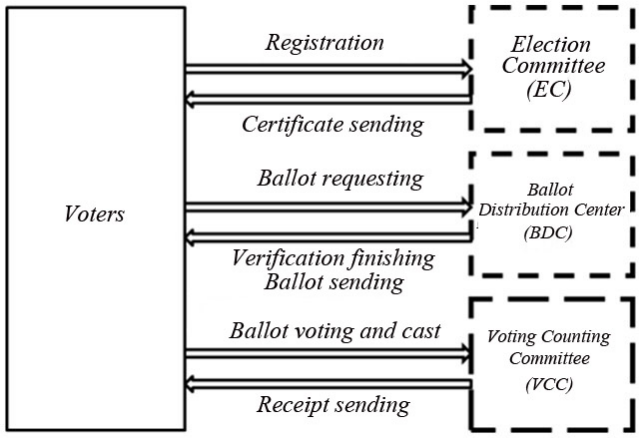

Figure 3.10 Flow diagram of how voters interact with election authorities.

Watchdog devices are also used for securing and monitoring the online voting behaviors at voter side. We still need to implement the special device (watchdog) to provide a reliable voter-reorganization through the Internet in RE-NOTE as an enhanced model.

We will discuss the ring signature based ballot distributing function along with clash attack prevention as below.

Step 1: EC will check each voter's eligibility. Once this voter is verified, EC will assign the key *KEC* and the certificate *M* to him. This shared key *KEC* will be only

transferred from BDC to those voters who passed the registration verification. And this key can be set to different value among a large number of groups. The voter uses the ring signature algorithm to encrypt the certificate with his/her own random pick. Then, the voter can request the ballot from BDC by sending the ring signature.

Step 2: When BDC receives the ring signature from the voter, BDC checks the ring signature to verify whether it comes from the authority EC since the signature must contain the key *KEC* and the certificate *M* from EC. If the ring signature passes the check by BDC, it means this voter is eligible to obtain a plain ballot. In the meantime, voter's identity is hidden and protected.

Step 3: The voter receives the ballot from BDC. The flag in the watchdog will be set to record this action to prevent the possible second ballot requesting or receiving action.

Step 4: The ballot mode and vote casting have been described in E-NOTE. We will continue with the solution to prevent clash attack on vote verification step.

Step 5: The voter casts the vote along with a random number chosen by herself/himself to VCC, and a receipt with tracking number will be generated with this random number and sent back to the voter. The voter can use this tracking number to review and track his/her vote, and this action will be also recorded in the watchdog at the voter side.

Step 6: While VCC collects all the data required for vote counting, VCC uses its own key to decrypt votes, and the marker still remains unknown since VCC does not have the key to decode. This step ensures that all types of ballots remain anonymous to VCC.

Step 7: VCC tabulates and publishes the results for each type of ballots and then sends them to EC. The candidate's name is still unknown to VCC because of the encrypted marker.

Step 8: BDC publishes the private key on the bulletin before EC reveals the value of marker on all ballots, and calculates the final tally of the votes according to the candidate names. Since those steps are open to the public during the vote counting, authority will have no opportunity to manipulate the voting tally in the public domain.

Step 9: Every voter can check his own vote with the vote tracking number. If the authority manipulates the process by giving the same receipt/tracking number to different voters, described as the clash attack above, voters will get the votes associated with a random number. Since this random number was picked personally by the specific voter, it is easy to detect the clash attack from voters if the attack has been applied.

Our proposed process is a method to protect voter's privacy better as well as offer a better way for voters to verify their own votes. Once the authority BDC receives the ballot request from the anonymous voter, they still cannot locate or track the voter's identity even if the ring signature is checked correctly. As described above, we have reduced communication steps from either BDC or EC to voters through the Internet, versus those steps described in E-NOTE. The voting authorities, EC, BDC and VCC, are independent of each other to ensure absolute fairness during the election. Our revised E-voting scheme RE-NOTE can mitigate both of these concerns by utilizing a decentralized counting process thus can better protect voter confidentiality.

Next, the same ballot distribution procedure will be calculated by ring signature in mathematic form.

We will illustrate the mathematical formula of our proposed method. Suppose l voters form a voting group and there are w candidates:

A_s: Voter s batch code or group number

B_s: Voter s basic information including date of birth, identity number and gender, etc.

D_s: Voter s choice

$C_{s:}$ Candidate's name where s is from 1 to n.

KEC: EC's key required for eligible voters to form a ring signature

REC: EC's private key in the group

M: the certificate given to voters by EC; the voter needs to show *M* with the ring signature to BDC to obtain the plain ballot from BDC.

Suppose we have a simple case where there is only one group of voters. In reality, all voters will be divided into several groups or batches based on the voters' registration county or state. According to the method discussed above, we denote (Pk_i, Sk_i) to be a pair of the public and private key of each ring group member except EC in the group, and (*KEC*, *REC*) to be the public and private key of EC. The public key *KEC* may be different in the different groups, and it is only shared between BDC and those voters who have negotiated with *EC*. We denote voter *s* as one of the group members.

Step 1: Voter *s* sends data array (B_s, A_s) to EC for registration. EC will not distribute the certificate *M* and the key *KEC* to voter *s* until voter *s* passes EC's voter registration check.

Step 2: After voter *s* gets the certificate *M* and key *KEC* from EC, voter *s* will have all public keys of each group member $(KEC, Pk_1, Pk_2, \ldots Pk_l)$,

Let

$$\Theta=(KEC, Pk_1, Pk_2, \ldots Pk_l)$$

be the list of public keys. Voter *s* calculates the signature by using some independent cryptographic hash functions:

$$K_s = Hash\ (\Theta)$$

Pick the random value rv_s from {0,1} and calculate every y_i with the equation:

$$y_i = f\ (x_i) \qquad (1 \leq i \leq l,\ i \neq s,)$$

$$y_{KEC} = f\ (x_{KEC})$$

$f(\,)$and $f^{-1}(\,)$ are a pair of function and its reverse function. After voter *s* calculates all y_i required to form a ring signature, voter *s* will get the specific y_s as below:

$$CK_s(\ y_{KEC,}\ y_1, y_2, \ldots y_s, \ldots \ldots y_l, M,\ K_s) = rv_s$$

Then computes:

$$x_s = f^{-1}(y_s)$$

$CK_s()$ is considered as the signature verification function.

Finally, the ring signature is generated as:

$$(\Theta\ , rv_s\ , M,\ x_{KEC,} x_1, x_2, \ldots x_s. x_l)$$

Step 3: After BDC receives this ring signature

$(\Theta\ , rv_s\ , M,\ x_{KEC,} x_1, x_2, \ldots x_s. x_l)$ from the voter, BDC will verify the value by using:

$$CK_s(\ y_{KEC,}\ y_1, y_2, \ldots y_s, \ldots \ldots y_l, M,\ K_s) = rv_s$$

Obviously, this ring signature contains EC's key information and the certificate

M. If this signature passes the check, it does mean that the voter is eligible since EC only distributes its own information above to the eligible voters.

Step 4: Voter *s* is eligible to receive the plain ballot from BDC. The ballot format can be described as: Denote $c[i]$ to be the name of the i^{th} candidate and array $\boldsymbol{c} = (c[1], c[2], \ldots, c[w])$ to be the packet containing the list of names of the candidates in the election.

BDC generates two sets of data for each eligible voter which is the ballot marker along with the list of candidates. BDC encrypts the marker $\boldsymbol{t}$ with the key that is known to BDC only.

We have

$$\alpha = (k-1)(k'-1),$$

where $z = kk'$, k and k' are large numbers.

We select 1<h<α such that $\gcd(h, \alpha) = 1$. h' and h are chosen so that they satisfy the following:

$$(hh' \bmod \alpha) = 1$$

and *h*' is the multiplicative inverse of $h \bmod \alpha$ respectively.

The marker t are encrypted as follows:

$$T = t^h \bmod \alpha$$

BDC holds both *h'* and *h*, i.e., VCC does not have the key for decrypting the marker *T* in the packet.

Step 5: When the voter casts the vote, they will send the voting data $(d[1], d[2], \ldots, d[w], T, z_s)$ with the marker $\boldsymbol{T}$ to VCC. The binary array *d* represents the choice

of voter *s*. z_s is the random number that enables voters to verify his votes.

Step 6: VCC receives array *d* from all voters, and the votes are tallied for each type of marker *T*. Then VCC generates unique confirmation number with each z_s and array *d* that will be used for voters' verification step. The confirmation ver_s will be sent back to voter *s*.

While VCC can gather all information as below:

$$TALLY_T = (\sum(d[1]_i, d[2]_i, ..., d[w]_i), T)$$

$$= ((\sum d[1]_i, \sum d[2]_i, ..., \sum d[w]_i), T)$$

$$Database_T = ((\sum z_i), T)$$

Here, $TALLY_T$ is the temporary tally result for the ballots with the same marker *T*. $\sum(d[1]_i, d[2]_i, ..., d[w]_i)$ represents the sum of the votes for every candidate, i.e., $\sum d[j]_i$ represents the number of votes received by candidate *j*. ver_i will be stored in a database for voting receipts. It allows voters to track and verify the votes they cast.

Step 7: After VCC finishes counting procedure, each $TALLY_T$ with the same marker *T* is unveiled to all voters and transmitted to EC. At the same time, BDC will unveil the private key to all the voters, EC uses its own key to decrypt the marker as $t = T^{h'} \bmod \alpha$ then tallies the final results.

Step 8: Voter *s* can make an inquiry whether his voter has been counted correctly and verify his own vote in the final tally. He sends his vote tracking confirmation ver_s to the authority. Since voter *s* is anonymous as we describe above, the authority will not discover the exact checker, but have to provide the votes along with the random number z_s that was picked up by this specific voter.

We need to emphasize that our proposed model RE-NOTE is based on the ring signature in ballot distribution process for voters to exchange the required information with authorities during the election. Comparing with the three-pass algorithm, we have following important advantages:

Simplifying the communication step: In E-NOTE, there are several steps between voters and EC or BDC through the Internet to locate the voters; in RE-NOTE, there is only one communication step between voters and each authority to identify the eligible voters. It will be beneficial for future network resource to adopt the large scale election through the Internet media.

Refining the size of voter groups: Since we use the ring signature to setup one part of election procedure, voters are required to be divided into different groups or batches, the communication packets will be fixed due to the fixed size of the ring signature. This would make it easier for authorities to manage the voters in a large scale. Since every county and every state have different candidates in different race, it is easier to divide millions of voters into smaller voting group to make the algorithm applicable.

Due to the fact that more communication steps are required for the three-pass algorithm in E-NOTE, using ring signature may reduce the communication steps between voters and the authorities during the ballot distribution procedure. Our new proposed scheme RE-NOTE will provide a better encryption scheme to ensure a real protection for ballot distributing that would prevent the authorities such as BDC, EC from conducting malicious activities.

Most voting election has the privacy requirement that there should be no association between the cast vote and the voters' identity. RE-NOTE goes further beyond

by disassociating the relationship between the voters and assigned ballots as well. Our other implemented methods such as the watchdog device can also be transplanted to RE-NOTE to build an E-voting model so that reliable and authentic voters can communicate with the authorities. If there are any voting disputes claimed by voters, it will be a good recorder for authority's further investigation.

The novelty of RE-NOTE is to create several groups to setup the mutual restrictive relationship between the voters and the voting authorities. The ring signature will secure the anonymity of the voters to the authorities. The outlined scheme also eliminates voter anonymity leakage and protects both voters' and candidates' confidentiality and privacy. The application of the ring signature scheme will increase the security and cryptography level on the voters' confidentiality and anonymity. If the number of the voters is large enough, the possibility for hackers to decode the message encrypted by ring signature will be greatly reduced.

We have illustrated the enhanced E-voting system, RE-NOTE, which protects the voters' anonymity. In addition, we have developed the framework and hardware method that can better protect voter confidentiality and keep voter anonymous.

3.5 Multi-part Ballot Based Name and Vote Separated E-voting System (M-NOTE)

Since our assumption is based on a no-fully trusted authority, it is important to consider the possible clash attack issue and find a solution in our proposed system. Consider the following scenarios:

Malicious authorities provide fake or duplicate receipts to different voters and then instigate the clash attack successfully.

A hacker obtains a portion of a ballot and finds out what the voter actually voted.

A hacker reconstructs a valid original ballot by collecting all portions that were assigned to a specific voter.

If any of the above ever happens, the relationship between voters and their votes can be exposed, and the anonymity aspect of the E-voting system will be compromised.

A multi-part ballot is defined as a kind of ballot containing several separable parts. Each candidate is listed in a permutation from CAN-1 to CAN-β on the ballot. Each separable single-part only contains one vote for one candidate out of all the candidates. The choice on the single-part could be "yes" or "no", and every single-part of the multi-part ballot contains a unique sequence number (as shown in Figure 3.11 as the blackened area). Every single-part (see Figure 3.12) contains one choice of a specific candidate no matter the vote is "yes or no" on it. All of the single-part ballots will be cast to VCC and counted independently. Since we have β candidates in the election, the number of single-parts is set to β.

Since a multi-part ballot contains β single-parts, where each single-part has a type marker because the total number of ballot type is φ ($1 \leq \varphi \leq \beta!$), a sequence number Se ($1 \leq Se \leq \beta$), and a checkbox is defined to indicate whether this candidate is voted or not. Our multi-part ballot design will add an extra security level to protect candidate's identity and the vote information from being hacked as compared with the scheme containing the integrated ballot information of a voter's vote in the election. Figure 3.12 illustrates a sample of this multi-part ballot scheme with β equal to 3. After the voter casts his/her ballot, it will be counted in a more secure way by disassociating candidates' identities with the corresponding permutation on an assigned ballot.

Instead of attacking the robust E-voting system, hackers may try to intercept the

voting data through the Internet. We need to reduce the voting data transmitted to a minimal so that the proposed E-voting scheme can be maximally protected and secure the whole E-voting procedure.

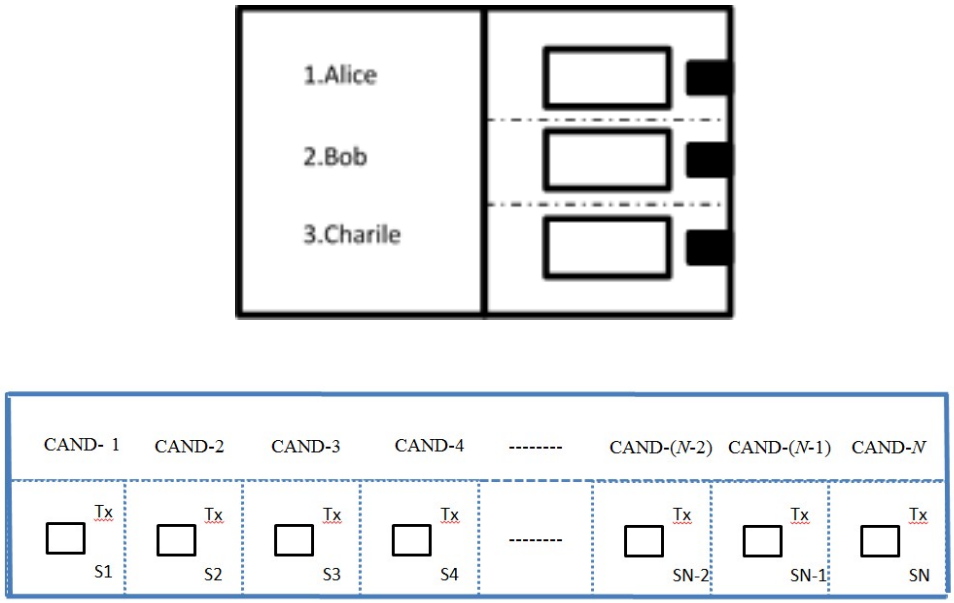

Figure 3.11 A sample of a plain multi-part ballot for a three candidates' race.

The operation procedure of M-NOTE can be described as follows:

EC verifies every voter's identity. Once a voter is verified, EC will authorize this voter's voting privilege and a ballot will be distributed to him/her by BDC. The detailed ballot distribution procedure is described in, in which we also describe how the voters' anonymity is well protected during the ballot distribution phase by using ring signature.

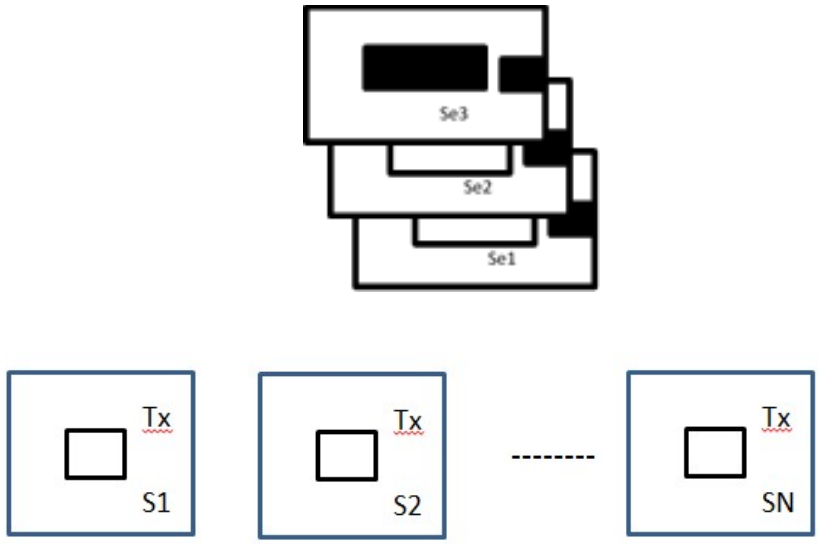

Figure 3.12 Single-part portions which show three candidates race with the marked ones on the top.

The voter receives a ballot from BDC and its format is shown in Figure 3.11.

As described in our previous works, a watchdog device is used to record and monitor the entire online voting transactions. Only the voting authorities have access to the data stored in the watchdog device. The ballot distribution transactions will be recorded in the watchdog device to avoid any multiple or duplicated ballots.

After making his/her choices on the ballot, the voter separates the ballot into parts (as shown in Figure 3.12) and casts them to VCC along with the set of trackers.

The voter can use these trackers to review and track his/her vote anonymously in the final tally. The whole procedure is also recorded in the watchdog device. This will prevent the receipt-based clash attack during the vote verification and ballot reconstruction phases.

If a voter wants to verify his/her vote, the voting authorities must show him/her the corresponding trackers that have been recorded from the multi-part ballot. These

trackers must match with the ones the voter has recorded earlier. This procedure ensures that each vote will be counted without being compromised in the final tally. Figure 3.13 shows the procedural flow of M-NOTE.

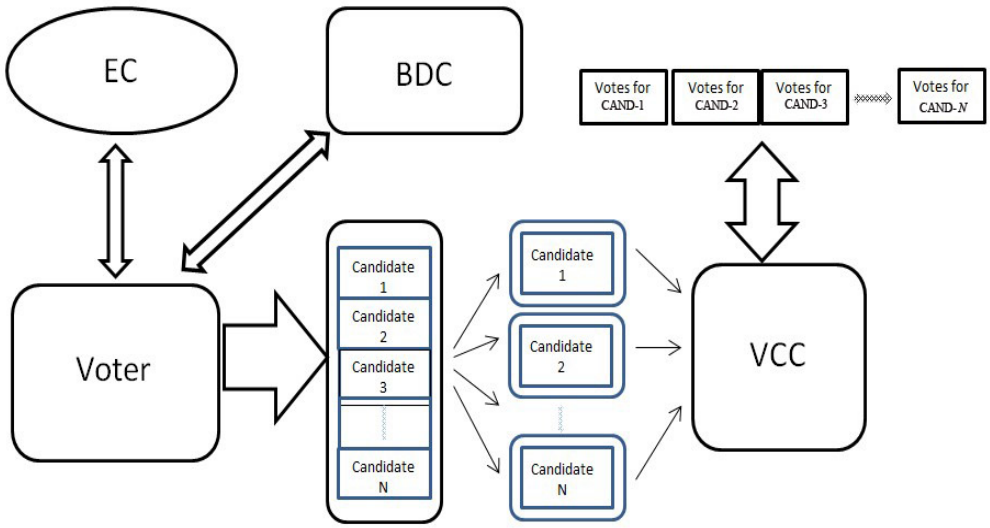

Figure 3.13 The block diagram of a voter interaction with election authorities by using a multi-part ballot.

Since the voter's identity is anonymous during the ballot distribution phase, the tracker chosen by the voter cannot be linked back to his/her identity. Note that each tracker chosen by the voter comes from a single database and it is unique for all voters during the election. In the vote verification step, if a voter makes an inquiry in the final tally, the authority cannot respond with a manipulated vote since the inquirer's identity is not known to the authorities. In the situation that the malicious authority instigates the clash attack to manipulate the vote count by generating exactly the same receipt/tracking number to different voters, it will be quite easy for the voters to detect the attack as well,

for the same reason that the authority cannot identify each voter and respond with a matched tracker.

We illustrate the mathematical formulation in our scheme M-NOTE by assuming there are *M* voters participating in the election with *N* candidates:

A_i: Voter *i's* group number;

B_i: Voter *i's* basic identification and other information;

$d_i[j]$: Voter *i's* choice (either 0 or 1), where *j* is from 1 to *N*. To improve the security level, every type of ballots may have the reversed definition of the choice. For example, type 1 may define 0 for voting "yes" and 1 for voting "no" while type 2 may do the opposite, 0 for voting "no" and 1 for voting "yes".

Step 1: Voter *i* sends data array (B_i, A_i) to EC for registration. The voter will get a ballot from BDC after passing the voter registration check by EC.

Step 2: When voter *i* casts the vote, the voting data

$(D_i[1], S[1], T_x, z_{i[1]})$, $(D_i[2], S[2], T_x, z_{i[2]})$, ..., $(D_i[N], S[N], T_x, z_{i[N]})$

will be sent to VCC. The binary array D_i represents the encrypted choices of d_i and $z_{i[j]}$ ($1<j<N$) is a set of unique random numbers used as trackers that were initially generated by the authorities and could be modified by voters. This set of tracker numbers will enable the voter to verify and audit his/her vote. $S[j]$ ($1<j<N$) is the sequence number of a single-part ballot. T_x ($1<x<N!$) is the encrypted marker used to represent the ballot type.

Step 3: Voter *i* will save all $z_{i[j]}$ as the receipt of casting the vote.

Step 4: VCC receives the array D_i from all voters, and these votes will be grouped according to the marker T_x. Then, VCC uses its private key to decrypt the entire encoded data array D_i to count the votes.

While VCC tallies the votes:

$$TALLY_{T_x} = \left((\sum d_i[1], \sum d_i[2], \ldots, \sum d_i[N], (z_{1[1]}, \ldots, z_{i[j]}), T_x\right)$$

At this time, the trackers will be stored separately along with the votes for each candidate in the election database as below:

$$Database_{C_1} = \left(z_{i[j]}, (C_1)_{z_{i[j]}}\right)$$

$$\ldots\ldots Database_{C_N} = \left(z_{i[j]}, (C_N)_{z_{i[j]}}\right)$$

Here, (C_1 C_2….C_N) is the set of candidates. $Database_{C_1}$ stores the data related to every vote and tracker for candidate C_1. $z_{i[j]}$ represents the collection of the trackers for every candidate, i.e., $(C_j)_{z_{i[j]}}$ ($1 \leq j \leq N$) is the final tally for candidate C_j accordingly with all votes. Each unique tracker is the key for voters to retrieve and locate their own votes anonymously.

Step 5: Voter *i* can make an inquiry to verify whether his/her votes have been counted correctly by checking the corresponding $z_{i[j]}$ in the candidate's database. Since voter *i* is anonymous, the authority will not be able to discover the identity of the actual inquirer but has to provide the votes along with the tracker $z_{i[j]}$ that was selected earlier (in Step 2) by this specific voter.

3.6 Time-lock and Timed-release Scheme

To prevent manipulation and alteration from malicious authorities such as EC in the final tally, we introduce the time-lock and timed-release protocol that will be used to secure the manifest of candidate orders on each specific ballot type during the whole election (May, 1993). The basic idea of the time-lock and timed-release crypto is to encrypt a

message and then decode it in a future time point in order to lock this important message for a certain period of time. In the new proposed model, the order of candidates on each type of ballot will be securely locked (unable to be decoded and changed) for a certain time. Normally we set this certain period to be the whole duration of the corresponding election. Once election ends, there is no need to hide the manifest of the ballot type. Hence, this time-lock could be used to restrict some other unauthorized access on the manifest in terms of time. The term "manifest" refers to a document, which provides comprehensive details of the ballot type and candidate sequence associated with each ballot type design in the election. Meanwhile, we will enhance and extend the existing framework of our research work by introducing a new method that can further restrict any malicious authority's activities

Suppose we have a political election with two candidates as shown in Table 3.4. There are two permutations of candidates, and therefore we have two types of ballots. After all ballots are collected and VCC begins to count votes, the voting result is published without releasing any candidate's identity or the permutations in Table 3.5. As compared with the type of ballot shown in Table 3.4, we can obtain the final voting result as shown in Table 3.6. Alice wins the election by receiving 35 votes vs Bob's 5 votes.

If Bob is the desired winner by the malicious authority EC, EC could alter the manifest of permutations to temper the voting results. Table 3.4 could be altered to the one shown in Table 3.7 and the manipulated voting results would be as shown in Table 3.8. Obviously, with the manipulated permutation, Bob eventually wins the election. This is a kind of attack we need to defend by applying the time-lock and timed-release scheme.

Table 3.4 Example of Two Type of Ballot with Different Candidate Permutations

Ballot Type A	**Ballot Type B**
Alice	Bob
Bob	Alice

Table 3.5 Voting Results without Releasing the Actual Candidate Permutation Info

Ballot Type A	**Ballot Type B**
Choice 1: 12	Choice 1: 4
Choice 2: 1	Choice 2: 23

Table 3.6 Authenticated Voting Results According to the Released Candidate Permutation

Ballot Type A	**Ballot Type B**
Alice : 12	Bob : 4
Bob : 1	Alice : 23

Table 3.7 Manipulated Permutations at Malicious Authority's Favor

Ballot Type A	**Ballot Type B**
Bob	Alice
Alice	Bob

Table 3.8 Tampered Voting Results with Manipulated Candidate Permutation

Ballot Type A	**Ballot Type B**
Bob : 12	Alice : 4
Alice : 1	Bob : 23

3.7 Voter Jury

The time-lock and timed-release protocol employs a trusted agent to operate the time-lock scheme and to release the time-lock after a certain time, which is called "time puzzle". In our research work, the original "trusted agent" defined in the time-lock and timed-release protocol will be replaced with a voter jury composed of a group of voters to supervise the time-lock. Similar to jury members in the court, every voter can be randomly chosen to be the voter jury member or pre-registered prior to the election to conduct the legal exercise. All jury members' identities can be published to the public. It is a basic requirement to incorporate the time-lock and timed-release scheme into our E-voting system model.

Since we apply time-lock and timed-release mechanism into our E-voting system model, we need a decentralized trusted agent instead of a single trusted agent to generate the time-lock and timed-release puzzle. It is more favorable to have more than one trusted agent to form a shared key by voter jury members to operate the time puzzle to ensure the fairness of this phase. As described in Section 3.6, the time-lock and timed-release mechanism is suitable to deter this type of attacks. It takes a certain time T for anyone to compute the encrypted message without knowing the key. We normally set the certain

time T to be the length of the election duration. So even if the manifest of the list of permutations is hacked after computing for a certain time, the election has ended. Then, the manifest of candidate permutations on each ballot type is regularized and any modification during the election is impossible.

CHAPTER 4

VOTING PROTOCOL WITH ANTI-ATTACK SOLUTIONS

The main features of our voting protocol include:

(1) voter anonymity throughout the process (after registration),

(2) vote verification in the final tally, and

(3) safe guard against malicious authorities from manipulating received votes.

The voting protocol consists of the following steps:

(1) The voter's eligibility is verified at EC and is assigned a digital certificate from EC.

(2) The voter requests ballot anonymously from BDC using the certificate.

(3) BDC verifies the voter eligibility and assigns a multi-part ballot (one of the several types).

(4) The voter designates tracking number on the vote and cast the vote to VCC.

(5) VCC counts ballots based on ballot types and releases the voting summary to the public.

(6) The ballot type manifest is released and the final result is tallied.

(7) The voter can anonymously inquire their vote with the confirmation number.

Figure 4.1 illustrates our E-voting system model, referred to time-lock algorithm based E-voting system with Ring signature and Multi-part form (TERM). The voting process and the various system functionalities will be explained through a voter (Tom) voting in the paper ballot mode so that the process can be easily understood.

We give an example that a voter, Tom, participates in the election, and the voting procedure will be introduced as below.

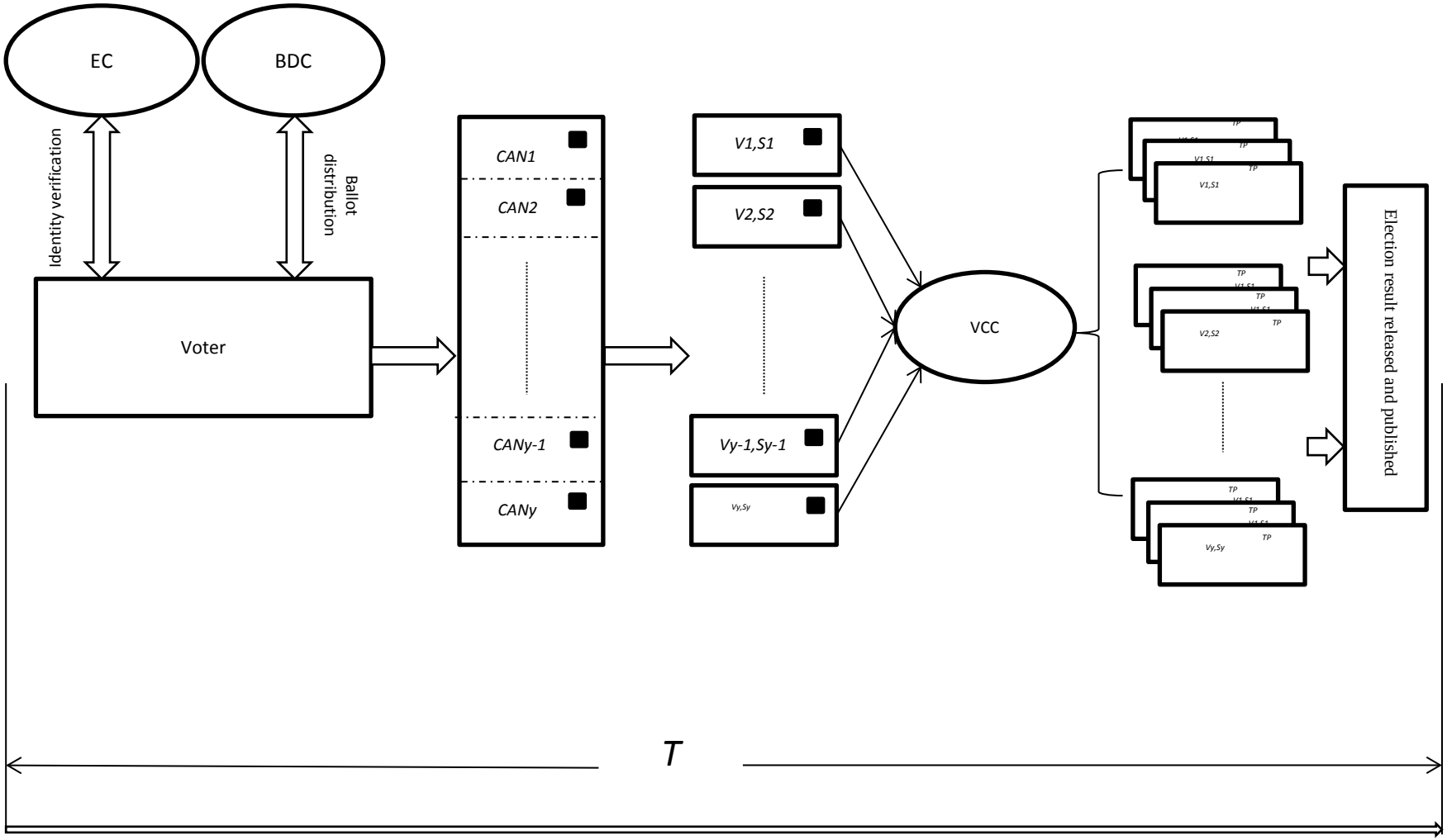

Figure 4.1 The whole voting procedure for a voter to send own ballot to VCC and tracking own vote by using the confirmation to inquire.

4.1 Registration

First, voter Tom will verify his voting eligibility with the election authority EC prior to the election. This is similar to the traditional voter registration process with the difference that Tom will now receive a certificate that is required when he requests his ballot from BDC. Note that the identity of the voter is not recorded in the certificate. When our proposed E-voting system is operated through the Internet, the issued certificate will be a digital certificate that can be pre-stored in a hardware flash card. This hardware flash card is called the watchdog device as mentioned in Chapter 2 and is used for identity recognition and recording online communication transactions between this corresponding specific voter and election authorities. The flash card does not contain any pre-identified information besides EC's certificate. Only EC can access the data stored inside for investigational purposes if there is a dispute by the specific voter on any voting transactions after the election. The function of the digital certificate stored in the watchdog is to prove voter eligibility and also to hide the voter's identity so that EC cannot link a voter to his/her ballot.

4.2 Time Lock Up the Manifest

Since a certain amount of ballot types are created in the election, we will incorporate the time-lock and timed-release mechanism introduced in the previous section into our proposed E-voting system model to prevent malicious authorities from manipulating the ballot type manifest during the election. All voter jury members will gather together to form a shared key to lock up the ballot type manifest for at least the duration of the election. This will prevent a malicious authority such as VCC from manipulating the ballot type manifest so as to elevate its own candidate to the leading position.

Here we define this message as the manifest of the whole candidate permutations, and the certain secure time *T* is set to be longer than the duration of the election. Since the candidate permutation is regularized with each ballot type, any modification during the election is impossible.

We will give a detailed mathematical formulation of the time-lock and timed-release measurement in next chapter.

4.3 Voter Identity Encryption

If BDC is a malicious authority, another serious attack from BDC may occur if Tom shows his identification instead of the certificate obtained from EC while he requests the plain ballot from it. BDC may use this opportunity to link the assigned ballot info with Tom's identification. This kind of attack will violate the voter privacy rules in a political election. To prevent that, we have to use certain measurement to make eligible voters such as Tom anonymous to BDC while requesting the plain ballots. Therefore, we apply ring signature to ensure voters' identity confidentiality. In another word, BDC will not be able to obtain any identity information from an anonymous ballot requester other than his/her voting eligibility. When we switch to the online E-voting environment, Tom will still use the watchdog plugged at his own computer to identify himself as an eligible voter without releasing any other personal information. The watchdog is also used to record and monitor the entire online E-voting transactions. Only voting authorities have access to the data stored in the watchdog if Tom disputes any voting transaction post the election.

4.4 Ballot Distribution

By using the certificate to verify eligibility from BDC, Tom will receive a plain ballot from BDC and he will be marked with an electoral ink (such as a semi-permanent ink or dye that is applied to the forefinger) to indicate that he has already been assigned a ballot. This measurement successfully protects the voters' anonymity, prevents voters from voting more than once, and isolates the assigned ballot and its traceability in the election. When we apply the proposed E-voting system model through the Internet, the entire ballot distribution transaction will be recorded in the watchdog device so that Tom cannot request more than one ballot. The transaction data on the watchdog device can be reviewed to resolve any dispute after the election.

4.5 Voting

When Tom votes on the ballot, besides the possible attack from malicious authorities regarding the manifest, we still need to consider possible clash attacks from malicious authority VCC. In a traditional election, the public bulletin board is the only way for voters to review and verify the voting result; in our proposed scheme, the public board is additionally endowed with voting confirmation inquiry and verification responsibility (shown in Figure 4.2). Voters will mark their own votes on every single part with a system-wide unique confirmation. These confirmations will be published on the public bulletin board for the public's inquiry and supervisory purpose. In our proposed E-voting system model, Tom first checks on the bulletin board to see whether there is a conflict between confirmations he chose and those already posted by other voters. If so, Tom has to pick up another new confirmation to replace the conflicted one until all his picks (confirmations on every single part) are unique. Then, those confirmations will be written on every single part

one by one and post on the bulletin at the same time for public inquiry, meanwhile, these confirmations are serving as inquiries and trackers post the election.

With those confirmations on every single part of the original multi-part ballot, the malicious authority VCC may not be able to manipulate the vote as easily as the way they did as described in the Clash attack scenario. Because Tom is anonymous after the ballot distribution phase, it is not traceable for authorities to link the assigned ballot and the corresponding voter. Back to our example, Tom is anonymous to VCC after he got the certificate. In the vote audit phase, when Tom sends his confirmations to the authority to inquire his vote in the final tally, VCC will not be able to locate Tom's identity but has to respond with Tom's actual original votes. If VCC generated more than two exact same confirmations to different voters to initialize a clash attack, at this time when Tom inquires the voting result by his confirmations, the response does not match his original vote and the attack will be detected right away since this is not a one to one mapping's reverse procedure.

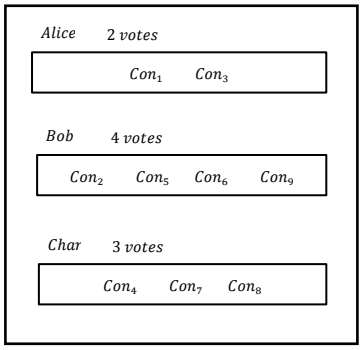

Figure 4.2 The published confirmations on the public bulletin.

4.6 Ballot Collecting

After Tom marked his choice on the multi-part ballot, he will tear it into individual single parts (as shown in Figure 4.3) and casts them to VCC. VCC will collect all ballots from all voters, and then group ballots by categorizing the same marker on each single part and then tallying the vote according to the type and the sequence number.

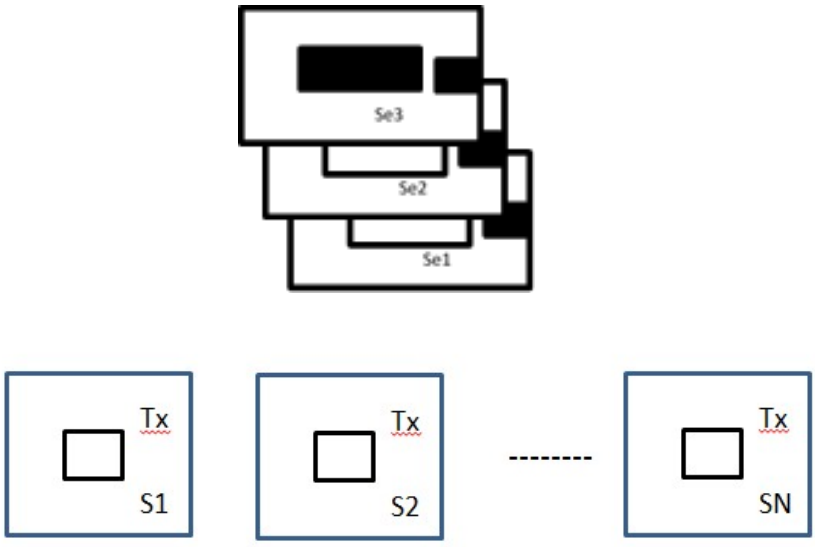

Figure 4.3 A sample of single-part portion which shows Three-candidate race with the marked ones on the top.

4.7 Releasing the Manifest

When the tally result including voters' confirmations is published on the public bulletin, the manifest of the ballot types is the only unknown factor in the election. The voter jury members will get together again and reform the manifest, and then EC will announce the final vote tally for each candidate according to the necessary ballot information release.

The public bulletin contains all confirmations from all voters. In our example, Tom can verify his own vote by checking his confirmations with those posted on the public

bulletin. Tom can also verify the voting result of the election in several different ways from the public bulletin (details on inquiry related to the security concern will be discussed in Chapter 4). In addition, a voter's whole voting process is recorded in this specific voter's watchdog device, and this can also prevent the duplicated voting in the election. If Tom wants to dispute the voting results, he must present his watchdog device containing all complete voting data transactions to the authority for further investigation.

Besides security features, our protocol also provides a great of diversity and flexibility for voters and candidates to achieve a fair election environment with self-audit and self-revise the confirmation. TERM has addressed these concerns by using several measurements mentioned above.

CHAPTER 5

MATHEMATICAL ANALYSIS

In this chapter, we will provide the mathematical analysis of our proposed system.

5.1 Pre-election

The time-lock and timed-release cryptography used to lock the candidate manifest can be generalized as below:

Since we have a voter jury to supervise the authority, and this jury is composed of several volunteer voters. Suppose we have *j* jury members in the jury. The manifest *M* represents the message to be encrypted by the timed-release cryptography. According to the Secret Sharing method by Shamir, it is divided by *j* jury members into *j* shares: $(M_1, M_2, \ldots M_j)$ Array *M* has the properties of Shamir's Secret Sharing theory as below:

1. Knowledge of *j* or more shares can easily reconstruct *M*.
2. Knowledge of less than *j* shares cannot reconstruct *M*.

Now we have *j* jury members to create the time puzzles to encrypt every own share. By definition, all *j* members must show up at the time *T* to decrypt *M* then the manifest could be reconstructed. The reason to have multiple members in this scheme instead of single trusted agent discussed in the Time-lock and Timed-release cryptography is that we assume any single voter or authorities are not fully trustable. This assumption is practical in reality as we have seen several cases of political elections disputes. Thus, we introduce this court-like jury composed of several either voluntary or selective voters to supervise this time-lock and Timed-released process. These *j* members will create the time puzzle as below:

1. Each jury member chooses a composite modulus

$$n_i = p_i q_i \quad i \in (1,2 \ldots j)$$

2. Calculates the Euler's totient function:

$$\Phi(n_i) = (p_i - 1)(q_i - 1) \quad i \in (1,2 \ldots j)$$

3. Chooses $e_{i,}(1 < e_i < \Phi(n_i)), \gcd(e_i, \Phi(n_i)) = 1$ randomly, where the function gcd(.,.) finds the greatest common divisor, such that the inverse exponent d_i that satisfies:

$$d_i\, e_i = 1\, mod \Phi(n_i) \quad i \in (1,2 \ldots j)$$

4. According to the time-lock puzzle definition, the puzzle factor *t* can be calculated by *t=TS*. We need to emphasize that the puzzle *t* can be applied to all jury members because all shares of the manifest *M* need to be released at the same time to reconstruct *M*. And we set T at least to be longer than the election duration.

5. Computes

$$r_i = 2^t mod\ \Phi(n_i)$$

and

$$d'_i = 2^t + \Phi(n_i) - r_i + d_i$$

6. Choose a random number a_i $(1 < a_i < n_i)$, encrypt d'_i as:

$$D'_i = d'_i + a_i^{2^t} mod\ n$$

7. Then every jury member will publish $(n_i,\ D'_i)$ instead of (n_i, d'_i) to the public. (n_i, e_i) is the private key of each jury member.

8. Here, we illustrate that each share of the manifest will be encrypted by every jury member through their own private key e_i as below:

$$M_i = m_i^{e_i} \, mod \, n_i$$

Then we have the time puzzle $(n_i, a_i\,, t, D'_i, M_i)$. All shares of the manifest have been encrypted by jury members safely. Neither the public nor authorities could see, reconstruct or manipulate them, since the public and authorities do not have information about (p_i, q_i) and it is very hard to factor them. Nobody can calculate the function $\Phi(n_i)$ to get the key of each jury member directly without (p_i, q_i). There is no faster way to compute $(a_i^{2^t} mod \, n_i)$ by sequentially starting with a_i and computing *t* squaring.

For practical purposes, we may ask jury members to form a shared key to encrypt the manifest or we could divide the manifest into *J* shares to be encrypted by jury members to reduce the complexity and time costing.

Table 5.1 Notations

a_i	The random number chosen by voter jury member *i*	M	The manifest of the candidate permutations
d_i	The original public key of voter jury member *i*	n_i	The modulus that voter jury member *i* gets
d'_i	The result of d_i by adding functions	p_i	Large prime that voter jury member *i* chooses
D'_i	The encrypted public key of voter jury member *i*	q_i	Large prime that voter jury member *i* chooses
e_i	The private key of voter jury member *i*	r_i	The reminder of each *i*'s computation
$F_{i'}$	Voter i' personal identification and other information	S	The processing speed of the server
$G_{i'}$	Voter i' voting group number	t	The time puzzle factor to create the puzzle
i	The index of voter jury member, $i \in (1,2 \dots j)$	T	The time-lock and release puzzle
i'	The index of voter $i' \in (1,2 \dots x)$	x	The number of voters
j	The number of voter jury members	y	The number of candidates
TP_w	The marker on every ballot representing the type	$s_e[]$	The sequence number on a single port
M_i	The encrypted share of manifest *M*	$TALLY_{TPw}$	The tally result for *TPw*
$con_{i'[j']}$	The confirmation for voter i' on candidate j'	$Data_{Cj'}$	The tally contains votes for$C_{j'}$
$v_{i'}[j']$	Voter *i'* choice (either 0 or 1)	j'	The index of candidates, $j' \in (1,2 \dots y)$
w	The number of types of ballots, $w \in (1,2 \dots y!)$	$CE_{i'}$	The certification of voter i'
$KEY_{i'}$	The group key of EC used for ring signature to voter i'	l	The number of voters in a ring signature
PK_l	The public key of each ring signature members	$K_{i'}$	Hash value of voter i''s calculation
$\delta_{i'}$	The random value chosen by voter i', $\delta_{i'} \in [0,1]$	$u_{i'}$	The random value chosen by voter i'
$b_{i'}$	The value calculated by *f*( ) for voter i'		

5.2 Voter Registration and Ballot Distribution

We will continue our mathematical formulation of the voting process after the time-lock and timed-release cryptography has been successfully applied. Suppose there are *x* voters participating in the election with *y* candidates:

$G_{i'}$: Voter i' group number; $i' \in (1,2 \dots x)$

$F_{i'}$: Voter i' personal identification and basic information required for the election registration;

$v_{i'}$ $[j']$: Voter i' choice (either 0 or 1), where j' is from 1 to *y*. To improve the security level, every type of ballot may have the reversed definition of its corresponding choice. For example, in the election, we have several types of ballots, some types of ballots may define "0" for voting "yes" and "1" for voting "no" while the others may reverse the definition, "0" for voting "no" and "1" for voting "yes".

The voting process can be generalized as below:

Step 1: Voter i' sends data array ($G_{i'}$, $F_{i'}$) to EC for registration. EC will authorize his/her voting privilege and a certification $CE_{i'}$ will be assigned to him/her by EC. Voters' anonymity is well protected with this ring signature cryptography during the ballot distribution phase. The reason why we use ring signature scheme instead of group signature is due to the advantage of ring signature's property. The group manager of a group signature may conspire and become corrupted to compromise voter anonymity. In a ring signature, rings are geometric regions with uniform periphery without center controlling behaviors, ring signature can be powerful once members of ring want to be independent.

Step 2: EC will not distribute the certificate $CE_{i'}$ or the key $KEY_{i'}$ used for the ring signature scheme to voter i' only after voter i' passes EC's voter registration check. After

voter i' receives $CE_{i'}$ and $KEY_{i'}$ from EC, voter i' will have all public keys of each group member ($KEY_{i'}, Pk_1, Pk_2, \dots, Pk_l$),

Let

$$\Theta = (KEY_{i'}, Pk_1, Pk_2, \dots, Pk_l)$$

be the list of public keys. Voter i' calculates the signature by using an independent cryptographic hash function:

$$K_{i'} = Hash\ (\Theta)$$

Voter i' chooses a set of random values $u_{i'}$ as a generator for all other ring members. A random value $\mathfrak{z}_{i'}$ is also selected from [0, 1], then $b_{i'}$ is calculated according to the following equation:

$$b_{i'} = f(u_{i'}) \quad (1 \leq i' \leq l,\ i' \neq i',)$$

$$b_{KEY_{i'}} = f(u_{KEY_{i'}})$$

$f(.)$ and $f^{-1}(.)$ are a pair of function/inverse function. After voter i' uses Eq. above to calculate all $b_{i'}$ ($1 \leq i' \leq l,\ i' \neq i',$) required to form a ring signature, voter s will solve for y_s to form:

$$\psi\left(y_{KEY_{i'}}, b_1, b_2, \dots, b_{i'}, \dots, b_l, CE_{i'}, K_{i'}\right) = \mathfrak{z}_{i'}$$

Furthermore,

$$u_{i'} = f^{-1}(b_{i'})$$

$\psi(.)$ is used to verify the signature according to the definition of the ring signature.

Finally, the ring signature is generated as:

$$(\Theta, \mathfrak{z}_{i'}, CE_{i'}, u_{KEY_{i'}} u_1, u_2, \dots, u_l)$$

Step 3: After BDC receives this ring signature

$(\Theta\ , \mathfrak{z}_{i'}\ ,\ CE_{i'},\ u_{KEY_{i'}}, u_1, u_2, \dots, u_l\ )$ from voter, BDC will verify the value by checking:

$$\psi\big(\ b_{KEY_{i'}}, b_1, b_2, \dots, b_l, CE_{i'},\ K_{i'}\big) = \mathfrak{z}_{i'}$$

Obviously, this ring signature contains EC's key information and the certificate $CE_{i'}$. If this signature passes the check, it means that this voter is eligible since EC only distributes its own certificate to the eligible voters.

5.3 Ballot Casting and Vote Counting

Step 4: After voter i' casts the vote, the voting data $(V_{i'}[1], S_e[1], TP_w, con_{i'[1]})$, $(V_{i'}[2], S_e[2], TP_w, con_{i'[2]})$, ..., $(V_{i'}[y], S_e[y], TP_w, con_{i'[y]})$ will be sent to VCC. The binary array $V_{i'}$ represents the encrypted choices of $v_{i'}[j']$ with authorities' public key, and $con_{i'[j']}$ $(1 \leq j' \leq y)$ is a set of unique random numbers defined as confirmations that voters may use for tracking purpose. This set of confirmation numbers also can help prevent the clash attack from malicious authorities as discussed above. $S_e[j']$ $(1 \leq j' \leq y)$ is the sequence number on every single-part portion of the multi-part ballot. TP_w $(1<w<y!)$ is the encrypted marker representing the ballot type.

Step 5: Voter i' saves all $z_{i'[j']}$ as the receipt for cast vote. Array $con_{i'}$ is used for vote tracking and inquiry purposes post the election. It also can be observed and checked by anyone beyond voters.

Step 6: After receiving all the ballots, VCC groups all portions according to TP_w. Then, VCC uses its private key to decrypt the entire encoded data array $V_{i'}$ to count the votes in plain data form without any candidate info.

While VCC tallies the votes:

$$TALLY_{TP_w}$$

$$= \left(\left(\sum\left(V_{i'}[1], con_{i'[1]}\right), \sum\left(V_{i'}[2], con_{i'[2]}\right), \ldots, \sum\left(V_{i'}[y], con_{i'[y]}\right)\right), TP_w\right)$$

The corresponding confirmations will be stored separately along with the votes for each candidate in the election database as below:

$$Database_{CAN_1} = \left(\sum con_{i'[j']}\right), \left(\sum (CAN_1)_{con_{i'[j']}}\right)$$

……..

$$Database_{CAN_y} = \left(\sum con_{i'[j']}\right), \left(\sum (CAN_y)_{con_{i'[j']}}\right)$$

Here, $(CAN_1\ CAN_2 \ldots . CAN_y)$ represents the set of candidates. $Database_{CAN_{j'}}$ stores those data related to every vote cast for candidate $CAN_{j'}(1 \leq j' \leq y)$ along with its respective confirmation. $\sum con_{i'[j']}$ represents the gathering of confirmations for every candidate, i.e., $\sum (CAN_{j'})_{con_{i'[j']}}(1 \leq j' \leq y)$is the final tally of all votes for candidate $CAN_{j'}$. We need to emphasize that i' $(1 \leq i' \leq x)$ presented in the final tally for $\sum (CAN_{j'})_{con_{i'[j']}}$ may vary as voters' choices may vary. $TALLY_{TP_w}$ will then be published on the public bulletin, but the mapping for each type of ballot or the actual candidate permutation on the manifest is still unrevealed.

Step 7: Since the time-lock and timed-release scheme has securely protected the manifest, each jury voter uses his/her own key to decrypt his share of the manifest. After all shares get decrypted and put together, they can be used to reconstruct the manifest and will be published to the public and the bulletin. Meanwhile, if any voter wants to verify

the manifest before it is released, it still takes *T* period to calculate the public key of each jury voter which is d'_i where $i \in (1,2 \ldots j)$.

5.4 Vote Tracking

If Voter *i'* wants to verify whether his/her vote have been counted correctly, he/she can check the corresponding $con_{i'[j']}$ in $Database_{CAN_{j'}}$. Since voter *i'* is anonymous during the ballot distribution phase, the authority will not be able to identify the actual inquirer but have to provide the confirmation $con_{i'[j']}$ received and its associated vote. Note, the response from the authority must match what voter *i'* has created and recorded earlier, or it will be detected by this anonymous inquirer (voter *i'*). This procedure ensures that each vote will be counted without being altered in the final tally.

CHAPTER 6

ADDITIONAL SECURITY FEATURES

Security is always one of the crucial factors when we evaluate any E-voting system. In this chapter, we will evaluate TERM in both security and performance aspects.

6.1 Security Analysis and Case Study

To counteract possible attacks from authorities or hackers through the Internet, we will introduce additional security features to our E-voting system.

6.1.1 The *T* time length in Time-lock and Timed-release scheme

We have described the Time-lock and Timed-release schemes in Section 3.6. The secure time period *T* is determined by the duration of the election, and then the length of the key is determined by *T* and the processing speed of the processor at the server side. We ask the voter jury to setup the period *T* together to secure the manifest, so that each voter jury member has the same *T* value to process and releases the Time-lock at the same time. Figure 6.1 is shown the time frame for the voter jury members to encrypt their own keys to time-lock a message. In reality, the processing speed at each voter jury member may vary. The situation will become more complicated if these jury members' time-lock cannot be released at the same time.

The key size of *K* equals to by $\lceil log_2(2ST) \rceil$ according to the definition. Suppose the secure period we want is 3 days which means *T*=259200 second, and we assume the processing speed is 3.4Gbps, then we get *K*= $\lceil log_2(2ST) \rceil = \lceil 53.95 \rceil$=54, which means

the lower bound of the size of *K* is 54 (Kim, 2010). For any key with a length of 128 bits, it will be long enough to be used as the time-lock and timed-release key.

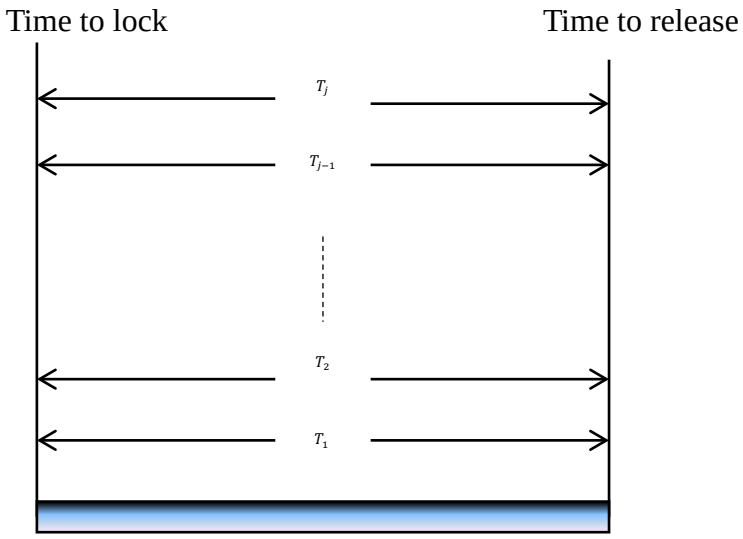

Figure 6.1 The time lock is locked by all voter jury members with same time puzzle.

The formulation defined in (Kim, 2010) gives us:

$$K=lg(2ST)$$

We can conclude from the above calculation that the longer the key size, the more complexity it will bring into the scheme. Therefore, we need to optimize the calculation and reduce the complexity of the time-lock and timed-release scheme. We have several options as below.

The first option is to set up several jury members, but each one has a different *T* secure period as shown in Figure 6.2. Since we have *j* voter jury members. As long as the first jury member j_1 begins to setup the period required for time-lock and timed-release scheme, other (*j*-1) jury members can set up their own time puzzles in turns rather than at the same time. The length of the time puzzle calculated by j_1 is T_1. Then when the second

jury member sets the key, the time puzzle of T_2 can be reduced to $(T - T_1)$. It's obvious that the computational complexity for the second jury member j_2 is reduced. Similarly, we have each jury member T_i's time puzzle is equal to $(T - \sum_{k=1}^{i-1} T_k)$ $(1 \leq i \leq j)$. Each jury member's time puzzle is decreased so that a faster computation can be achieved while the security level of the system remains same.

Another option is to have each jury member in charge of a certain length of the secured period time T_i. $(1 \leq i \leq j)$ and $\sum_{i=1}^{j} T_i > T$, and each jury member $T_i (1 < i < j)$'s secured period must overlap at least two neighbor members except that the first and last jury member' T_1 and T_j , respectively. These two jury members only have one overlap with his/her neighbor jury member. Figure 6.3 shows the time-lock and timed-release scheme with less complexity on the key size.

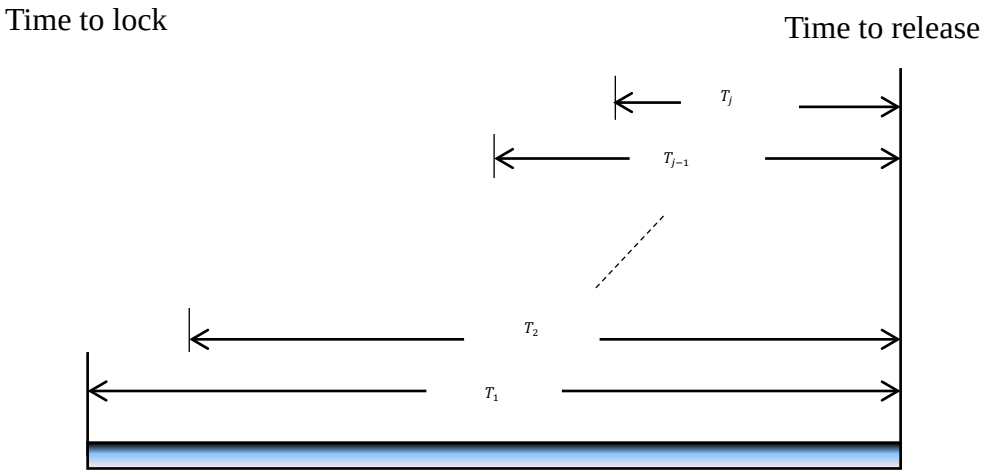

Figure 6.2 The time lock is locked by all voter jury members with decreased time puzzle.

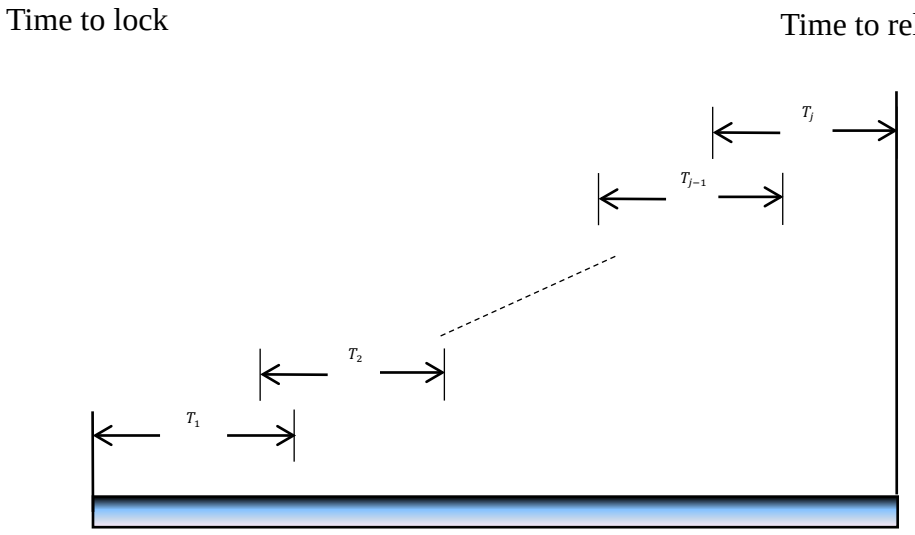

Figure 6.3 The time lock is locked by all voter jury members with overlapped time puzzle.

6.1.2 Inquiry from Voter and Through the Public Bulletin

This dissertation has discussed a clash attack prevention scheme, which could restrict manipulating activities from the malicious authority (EC, BDC or VCC). In this scenario, the confirmations are used by voters to track and verify the voting results in the final tally. Each confirmation actually can be composed of a set of numbers or characters. These confirmations can be originally generated from an authority's database, and they must have a one to one association with the corresponding assigned ballot. In our proposed E-voting system model, the voter must inquire whether those confirmations he/she will use conflict with other voters' in the system before casting his/her vote. At this moment, the election authority does not know what the voter will exactly vote. Then, it is very risky and unpredictable for the malicious authority to respond with a false inquiry result and

duplicate the same confirmation to different voters at this step. Since the malicious authority cannot predict what and who these voters will exactly vote for after having been assigned confirmations. As voters are anonymous at the ballot distribution phase, their identities will remain anonymous when they obtain these confirmations from the authority along with the plain ballots. The voters may either modify or keep those assigned confirmations as long as they remain unique in the election's database. It is the voters' responsibility to inquire the database again to make sure that the intended new confirmations are still unique after their modifications on confirmations. All these confirmations can be served as receipts for voters to track their votes after the ballot casting phase.

If any voter wants to inquire his/her own vote from the authority in the final tally, he/she only needs to provide and send the corresponding confirmation's information ($con_1, con_2, \ldots con_y$) to the authority, and the authority has to respond with the inquirer's original vote to the inquirer accordingly.

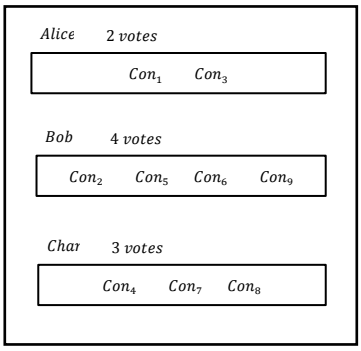

Figure 6.4 The published confirmations on the public bulletin.

The public bulletin is used to record and publish all votes along with each corresponding confirmation as shown in Fig 6.4. Voters may check from this bulletin to see if there are any duplicated confirmations or whether the total number of votes and confirmations match the number of actually participated voters. This method has empowered the voters to monitor and supervise the election.

6.2 Performance Analysis

In this section, we will evaluate the performance of our work to demonstrate how it can help provision the security features, reliability, and trust-worthiness of E-voting system. The proposed work can protect against the clash attack which falls into the category of a receipt-based attack by utilizing a multi-part ballot method along with voter-selected unique confirmations. We assume that if malicious authorities or hackers are able to intercept the packets transmitted through the Internet successfully, then our methodology will be the last protection measurement to stop hackers from obtaining the content of single-part portions to reconstruct the complete multi-part ballot. Thus, its proper design is very important to ensure the fairness and privacy for the whole political election. To demonstrate the performance of our methodology, let us calculate the probability that malicious authorities and hackers can successfully reconstruct the ballots and manipulate the votes during the election. Here any instance of a voter's vote that can be correctly revealed by a hacker from any single part of the multi-part ballot is considered as a successful attack.

The following three analysis methods are considered:

The probability of successfully reconstructing a single-part portion of a specific voter's vote among the entire voting pool:

We denote *y* as the number of candidates; *x* as the number of voters. If a hacker or a malicious authority wants to initiate an attack on the vote of a specific voter, it requires getting 3 factors: the probability of successfully identifying this specific voter is $\frac{1}{x}$. Then since we have *y!* types of a multi-part ballot and this voter must have used one of them, the probability of successfully identifying the correct multi-part ballot type is $\frac{1}{y!}$. The possibility of a successful hack on this voter's choice on a single-part portion of the ballot, either 0 or 1, is $\frac{1}{2}$. Thus the probability of successfully attacking a specific voter's vote is:

$$\frac{1}{y!} * \frac{1}{2} * \frac{1}{x} = \frac{1}{2x * y!}$$

The probability of successfully reconstructing a single-part portion of a specific voter's vote with respect to a specific candidate:

Here we still have *y* candidates and *x* voters. Compared with the previous case, we need an extra factor to locate this specific candidate, thus an extra $\frac{1}{y}$ (from *y* candidates) will be applied as shown below:

$$\frac{1}{y!} * \frac{1}{y} * \frac{1}{2} * \frac{1}{x} = \frac{1}{2xy * y!}$$

The probability of successfully reconstructing a valid ballot from all single parts of the ballots is:

$$\frac{1}{(y!)^y * y!}$$

If malicious authorities or hackers successfully intercept the data packet containing a single-part portion of a ballot, and want to intercept the second single-part portion from the same ballot to reconstruct the original multi-part ballot, the successful probability to

achieve this goal is $\frac{1}{y!}\frac{1}{(y-1)}$. We deduce this number as follows: the probability of obtaining the second single-part portion in the same type as the first one is $\frac{1}{y!}$, and the sequence number of this single-part portion must be different from that of the first single-part portion from the same multi-part ballot, thus a probability factor $\frac{1}{y-1}$ is applied after. With the same principle, we can determine that the probability of successfully reconstructing the third single-part portion from the same ballot to reconstruct the original multi-part ballot is $\frac{1}{y!}\frac{1}{(y-2)}$. Here $\frac{1}{y-2}$ is applied as the sequence number of the third single-part portion must be different from that of the first and second sequence number on the single-part portion. So on so forth thus the probability of successfully attacking the (y-1)th single-part portion that could be used to reconstruct the original multi-part ballot is $\frac{1}{y!}\frac{1}{(y-(y-1))} = \frac{1}{y!}$. Therefore, the probability of reconstructing an original multi-part ballot from any hacked or existing known single-part portion is $\frac{1}{(y!)^{y-1}*(y-1)!}$, deduced as below:

$$\frac{1}{y!} * \frac{1}{(y-1)} * \frac{1}{y!} * \frac{1}{(y-2)} * \dots * \frac{1}{y!} * \frac{1}{1} = \frac{1}{(y!)^{y-1} * (y-1)!}$$

A probability factor $\frac{1}{y*y!}$ also needs to be applied, as the malicious authority and hacker may randomly pick the first single-part portion, which can be among y! types of multi-part ballots, and can be among one of N sequence numbers in one multi-part ballot. Thus the final probability of successfully reconstructing an original multi-part ballot is shown below:

$$\frac{1}{y * y!} * \frac{1}{(y!)^{y-1} * (y-1)!} = \frac{1}{(y!)^{y} * y!}$$

Note that the calculated result refers to the probability of reconstructing a voter's valid multi-part ballot. This voter could be any participating voter; if the hacker wants to reconstruct a valid multi-part of a specific voter, the probability would be much lower than the one we presented. On the typical ballot used in presidential elections, the number of candidates will be around 10 to 20 including the local, state, and congressional races. We choose the number of candidates from 2 to 14 for illustration and analysis purpose, and summarize the corresponding probability for malicious authorities and hackers to successfully reconstruct an original multi-part ballot in Table 6.1. Figure 6.5 and 6.6, respectively plot the probability curve accordingly.

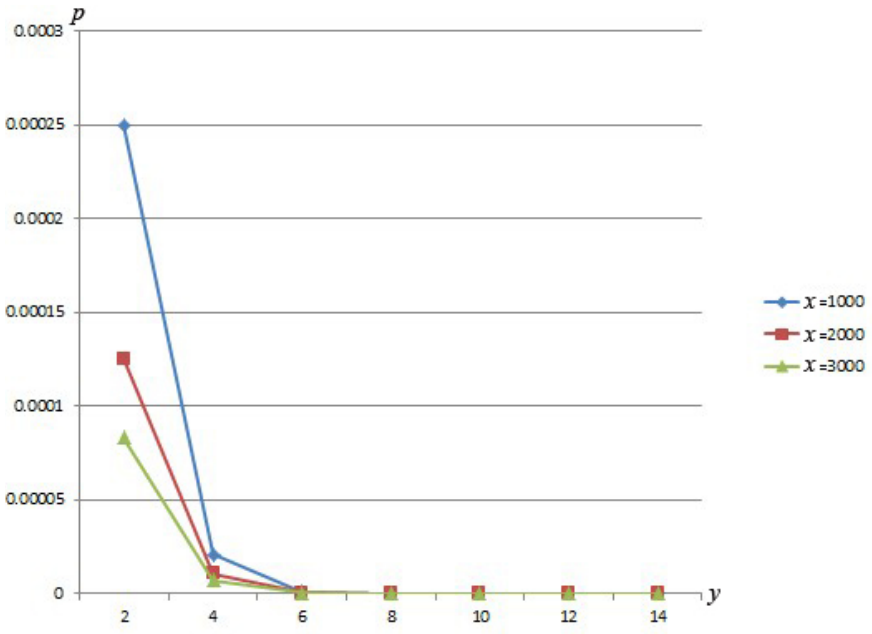

Figure 6.5 Probability of successfully attacking a specific voter's vote for x=1000, 2000, 3000 while y is from 2 to 14.

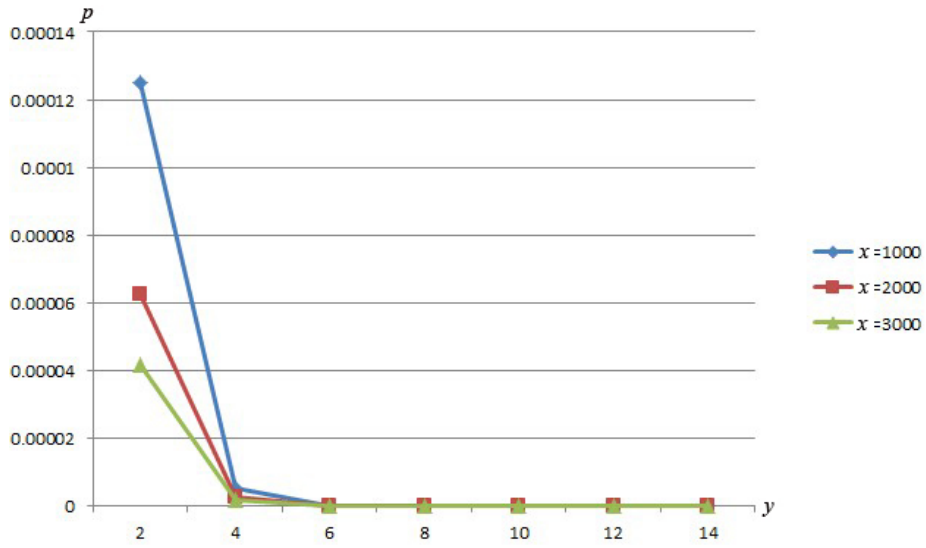

Figure 6.6 Probability of successfully attacking a specific voter's vote with the respect to a specific candidate
when x=1000, 2000, 3000 while y is from 2 to 14.

Table 6.1 Probability of Reconstructing a Valid Original Multi-part ballot with Different Number of Candidates

# of Candidates	**Successful Probability**
4	$5.02 * 10^{-7}$
6	$5.98 * 10^{-20}$
8	$2.84 * 10^{-41}$
10	$6.96 * 10^{-72}$
12	$1.71 * 10^{-112}$
14	$1.09 * 10^{-163}$

CHAPTER 7

CONCLUSION AND FUTURE WORK

Compared with previous works, TERM has exhibited the following important advantages:

(1) The probability of successfully reconstructing a ballot by malicious authorities or hackers is close to zero.

(2) Setting up a secured time period limits any possible manipulation by malicious authorities.

(3) Creating a confirmation can prevent possible clash attack carried out by malicious authorities.

Other advantages of TERM are summarized as below:

Anonymity: In the ballot distribution phase, the voter's identity is not associated with the ballot received from BDC.

Verification: The confirmations chosen by voter can be used to track and verify his/her vote in the final tally. These confirmations are unique and exclusive to this specific voter who is still anonymous to the authorities. Any un-matching result will be detected immediately during the inquiry process.

Privacy: Our performance analysis shows that the probability of reconstructing an original multi-part ballot by malicious authorities is close to zero as long as the number of candidates is above a certain number. For example, if the number of participating candidates in the election is 14 (a typical number for presidential elections in the USA including senate and house of representative, the probability of successfully reconstructing an original multi-part ballot from those single-part portions is $1.09 * 10^{-163}$.

Confidentiality: The manifest will remain undisclosed during the vote counting

procedure, to prevent any manipulation in the vote counting phase.

Security: Using a ballot that separates the candidates' name and voters' vote enhance the security requirement in the E-voting system.

Recordable and traceable transactions via the watchdog device: The watchdog device records the entire E-voting transactions. The authorities may use it to investigate any dispute such as mismatched voting choices claimed by the voters.

This dissertation describes a framework for developing an integrated E-voting system with diverse security features, several areas of this research can be expanded in the future:

(1) Voters choosing the ballot type: Before the election begins, election authorities will publish the types of the ballot that will be used in the election. Voters can select and choose one of the published ballot types randomly to cast the vote.

(2) Addition performance analysis are needed so that the E-voting protocol can be implemented for large-scale elections in the future.

In this dissertation, we have presented an overview of our proposed E-voting system model, TERM, which mitigates a number of security concerns such as ballot reconstruction, vote manipulation, tampering permutation list of candidates, and clash attack from malicious authorities; at the same time, it provisions a secured vote verification mechanism and can further mitigate those issues by utilizing a decentralized ballot collecting process along with a vote verification feature to better protect both candidates' and voter's confidentiality. The 2016 U.S. presidential election has raised awareness of many serious issues such as voter fraud, voting machines manipulation (Kaleem, 2016), software glitch (Durden, 2016), untrustable authority and hackers. Our proposed TERM can readily mitigate some of these issues, and provide a leap forward in ensuring a fair and democratic voting process for future elections.

REFERENCES

D. Bruschi, F. Cindio, D. Ferrazzi, G. Poletti, and E. Rosti, "Internet voting: do people accept it? Do they trust it?" *Database and Expert Systems Applications, 2002. Proc. 13th International Workshop*, Sep. 2002, pp.437.

The American Presidency Project, (2016, Nov). "Documents Related to the 2000 Election Dispute," [Online] Available: http://www.presidency.ucsb.edu/florida2000.php

C. Brown and S. Tribune. (2008, Dec.14) "Minnesota's vote: Cast into doubt," Startribune.com, Minneapolis, St. Paul, Minnesota, [Online]. Available: http://www.startribune.com/politics/national/senate/36093364.html

H. Pan, E. Hou, and N. Ansari, "Ensuring voters and candidates' confidentiality in E-voting systems," *Proc. 34th IEEE Sarnoff Symposium*, Princeton, NJ, May 3-4, 2011, pp.1-6.

H. Pan, E. Hou, and N. Ansari, "E-NOTE: An E-voting system that ensures voter confidentiality and voting accuracy," *Proc. 2012 IEEE International Conference on Communications*, Ottawa, Canada, Jun 10-15, 2012, pp.825-829.

H. Pan, E. Hou, and N. Ansari, "RE-NOTE: An E-voting Scheme based on Ring Signature and Clash Attack Protection," *Proc. 2013 IEEE Global Communication conference*, Atlanta, USA, Dec 9-13, 2013, pp. 867-871.

D. Chaum, "Blind signature for untraceable payments," *Advances in Cryptology, CRYPTO82*, Plenum Press, New York, 1983, pp.199-203.

L. Cranor and R. Cytron, "Sensus: a security-conscious electronic polling system for the Internet," *Proc. the Thirtieth Hawaii International Conference on System Sciences*, Wailea, Hawaii, Jan 7-10, 1997, vol.3, pp.561-570.

H. Chien, J. Jan, and Y. Tseng, "RSA-based partially blind signature with low computation," *Proc. Eighth International Conference on Parallel and Distributed Systems. (ICPADS 2001)*, KyongJu City, Korea, Jun 26-29, 2001, pp.385-389.

J. Benaloh, "Secret Sharing Homomorphisms: Keeping shares of a secret," *Proc. Advances in Cryptology (CRYPTO' 86)*, Springer-Verlag, New York, 1987, pp. 251-260.

D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *ACM Commun.*, Vol 24, no. 2, Feb. 1981, pp. 84-90.

C. Wu and R. Sankaranarayana, "Internet voting: concerns and solutions," *Proc. First International Symposium on Cyber Worlds*, Tokyo, Japan, Nov 6-8, 2002, pp. 261-266.

M. Jakobsson and A. Juels, “DIMACS workshop on electronic voting theory and practice,” DIMACS Center, Rutgers University, Piscataway, NJ, May 26- 27, 2004.

B. Adida, “Helios: web-based open-audit voting,” *Proc. the 17th Conference on Security Symposium*, USENIX Association, Berkeley, CA, USA, Jul 28- Aug 1, 2008, pp. 335–348.

A. Fujioka, T. Okamoto, and K. Ohta,. “A Practical Secret Voting Scheme for Large Scale Elections,” *Proc. the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology* (ASIACRYPT '92), Springer-Verlag, London, UK, 1992, pp. 244-251.

D. Chaum, (2006, Oct. 15). “Punchscan,” [Online]. Available: *http://www.punchscan.org.*

S. Popovenuic and B. Hosp, “An Introduction to Punchscan,” *Proc. Workshop on Trustworthy Elections,* June, 2006.

Y. Peter and A. Ryan. “Pr^et a voter with confirmation codes,” *Proc. the USENIX Electronic Voting Technology Workshop*, 2011, pp.611-627.

Y. Peter and Z. Xia. “Pr^et a voter: a voter-verifiable voting system,” *IEEE Transactions on Information Forensics and Security*, 2009, pp. 662-673.

R. Rivest. (2006, Oct. 15). “The ThreeBallot Voting System,” [Online]. Available: http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf

R. Rivest and W. Smith, “Three voting protocols: ThreeBallot, VAV, and twin,” *Proc. USENIX/ACCURATE Electronic Voting Technology Workshop* , Boston, MA, 2007, pp. 16.

R. Kusters, T. Truderung, and A. Vogt, “Clash Attacks on the Verifiability of E-Voting Systems,” *Proc. IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, May 20-23, 2012, pp. 395-409.

R. Küesters, T. Truderung, and A. Vogt, "Verifiability, Privacy, and Coercion-Resistance: New Insights from a Case Study," *Security and Privacy (SP), 2011 IEEE Symposium*, May 22-25.2011, pp.538-553.

T. May. (1993, Feb. 10). “Timed release crypto,” [Online]. Available: http://www.hks.net/cpunks/cpunks.html.

R. Rivest and Y.Tauman, “How to leak a secret,” *Proc. Advances in Cryptology*, ASIACRYPT’01, 2001, pp. 552–565.

A. Shamir, “How to share a secret,” *Communications of the ACM*, November 1979.

K. Kim and S. Cho, “A 3.4Gbps transmitter for multi-serial data communication using pre-emphasis method,” *Proc. the 4th WSEAS international conference on Circuits, systems, signal and telecommunications* (CISST'10), Stevens Point, Wisconsin, USA, 2010, pp.153-156.

N. Gupta, P. Bala, and V.Singh, “Area & Power Efficient 3.4Gbps/Channel HDMI Transmitter with Single-Ended Structure,” *Proc. 26th International Conference on VLSI Design and 12th International Conference on Embedded Systems* (VLSID '13). IEEE Computer Society, Washington, DC, USA, 2013, pp.142-146.

R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, Feb. 1978, pp. 120-126.

J. Kaleem. (2016, Oct. 26). “Here’s what we know about so far about voter fraud and the 2016 elections,” *Los Angeles Times*, [Online]. Available: http://www.latimes.com/politics/la-na-pol-voting-irregularities-snap-story.html

T. Durden. (2016, Oct.26). “Texas County enacts ‘Emergency paper ballots’ after ‘software glitch’ in voting machines,” [Online]. Available: http://www.zerohedge.com/news/2016-10-26/texas-county-enacts-emergency-paper-ballots-after-software-glitch-voting-machines

Johngibbs. (2016, Oct. 13). “Voter Fraud is Real. Here is the proof,” [Online]. Available: http://thefederalist.com/2016/10/13/voter-fraud-real-heres-proof/