A Voting Scheme Based on Revised-SVRM and Confirmation Numbers

By

S. M. Saifur Rahman

A Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Engineering in Computer Science and Engineering

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh

February, 2019

Declaration

This is to certify that the thesis work entitled "**A Voting Scheme Based on Revised-SVRM and Confirmation Numbers**" has been carried out by S. M. Saifur Rahman in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh. The above thesis work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

13.02.19

Signature of Supervisor
Dr. Kazi Md. Rokibul Alam
Professor,
Dept. of Computer Science and Engineering,
Khulna University of Engineering & Technology

13.02.19

Signature of Candidate
S. M. Saifur Rhman
Roll: 1507508
Dept. of Computer Science and Engineering,
Khulna University of Engineering & Technology

Approval

This is to certify that the thesis work submitted by S. M. Saifur Rahman entitled "**A Voting Scheme Based on Revised-SVRM and Confirmation Numbers**" has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh in February, 2019.

BOARD OF EXAMINERS

1. ______________________ Chairman (Supervisor)
Dr. Kazi Md. Rokibul Alam
Professor
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology,
Khulna, Bangladesh

2. ______________________ Member
Dr. Muhammad Aminul Haque Akhand
Professor & Head
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology,
Khulna, Bangladesh

3. ______________________ Member
Dr. K. M. Azharul Hasan
Professor
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology,
Khulna, Bangladesh

4. ______________________ Member
Dr. Pintu Chandra Shill
Professor
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh

5. ______________________ Member (External)
Dr. Kamrul Hasan Talukder
Professor
Computer Science and Engineering Discipline,
Khulna University, Khulna

Acknowledgement

At the start, I would like to thank Almighty for his blessings on me. Then I would like to express my sincere gratitude to my supervisor Prof. Dr. Kazi Md. Rokibul Alam, Professor, Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET) for the continuous guidance, encouragement , continuous assistance and immense knowledge. His inspiration and infinite dedication has enabled me to complete the thesis successfully. I could not have imagined having a better supervisor and mentor for my M. Sc. study. I am especially grateful to him for giving me his valuable time whenever I need and always providing continuous support in my effort.

I am also grateful to Prof. Shinsuke Tamura, Graduate School of Engineering, University of Fukui, Japan for his continuous suggestions and guidelines for this research.

I am especially grateful to all the faculty members of the Department of CSE, KUET to have their privilege of intensive, in-depth interaction and suggestions for the successful completion of my master degree.

Finally, I am grateful to my family members specially my mother and my wife for all their patience, support and encouragement during this period.

February, 2019 Author

Abstract

This thesis improves the performance of the revised-simplified verifiable re-encryption mixnet (R-SVRM) based e-voting scheme by introducing confirmation numbers (CNs) that are used in the CN based e-voting scheme. Although CN based and R-SVRM based schemes had made e-voting schemes more practical by excluding zero knowledge proof (ZKP) that requires large volume of computations, still they are not efficient enough. Namely, the CN based scheme adopts RSA encryption functions that are not probabilistic or commutative, therefore to satisfy essential requirements of elections, extra random factors are necessary for individual votes, election authorities must sign on votes and they must keep encryption keys as their secrets. On the other hand, although the R-SVRM based scheme uses ElGamal encryption functions that are probabilistic and commutative, vote forms in it is complicated, i.e. vote forms consist of at least 3 items and they include information about candidates as exponents. The improved scheme simplifies vote forms by exploiting CNs, and extensively reduces the number of operations required for individual votes. Also, the scheme successfully satisfies all essential requirements of e-voting systems, i.e. it is endowed with features about privacy, robustness, accuracy, incoercibility and fairness as CN based and R-SVRM based schemes are.

Contents

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

B	Booth manager
BBs	Bulletin Boards
CN	Confirmation Number
C_n	Confirmation number assigned to V_n
DRE	Digital-recording electronic
E-voting	Electronic Voting
g	Publicly known appropriate integers used for vote construction
ID_n	n-th voter identifier
$k_{(n,i)}$, $d_{(n,i)}$	Secret integers of M_i to re-encrypt v_n
M_i	i-th mix-server
N	Numbers of voters
P	Mutually independent mix-servers
Q	Publicly known appropriate integers used for vote construction
RSVRM	Revised-Simplified Verifiable Re-encryption mixnet

$r_{(n,i)}$	Secret integer of M_i to encrypt C_n
$s_{(n,i)}$, $e_{(n,i)}$	Secret integers of M_i to conceal $v_{(n,i)}$ and v_n
T_n and W_n	Anonymous credential of V_n, and a secret integer for concealing T_n
U and $\underline{U}$	Publicly known integers for generating used seals
U^{Z_n}, $\underline{U}^{Z_n}$	1st and 2nd used seals calculated by V_n
V_n	n-th voter
$X_{(i)}$	Private key of M_i for vote decryption
ZKP	Zero Knowledge Proof
Λ	A publicly known integer to encrypt and verify v_n

CHAPTER I

Introduction

1.1 Background

In any society, people possess in their mind to select a good leader. Precisely voting is the best way to do this and to sustain democracy. Voting is the process, in which voters cast their votes where a group of authorities collects the votes and outputs the final tally. One of the most important tasks of a government is the planning and the execution of the election that designates its successor. Voting authorizes an official mechanism for people to express their views to the government. Not surprisingly, it is also one of government's most challenging tasks one whose requirements and constraints are remarkably strict. Thus doubtful results, failing technology, and ingenious methods of fraud have been noted throughout election history [1].With the advancement of the cryptographic protocols and networks, electronic voting system is now a day's an important research topic. The recent advances in cryptographic voting system, a type of election system that provides mathematical proofs of the results rather than the machines is very promising. An electronic voting system, like other automated information systems, can be judged on several bases, including how well is design provides for security, accuracy, ease of use, and efficiency, as well as its cost.

1.2 Motivation

Conventional voting systems comprises of papers, mechanical levers, optical-scan machines, punch cards etc. In a paper ballot voting system, recording and counting votes of voters cast on paper sheets which are produced by voters themselves, by political parties or by election authorities for making a decision of successful candidates. Cards and a small clipboard-sized device are provided recording votes in a punch card voting system. Before the cards are placed in a ballot box for tabulation, voters punch holes in cards at positions consistent with their selected candidate using punch devices. In a mechanical lever voting system, every candidate's name is consigned to a particular lever in a rectangular array of levers on the front of the machine. A set of printed strips visible to the voters indicate the lever assigned for each candidate and issue the choice. In an optical-scan machine voting system, voters mark their choices in locations consistent with their choices usually by filling rectangles, circles, ovals, or by completing arrows. For reading marked paper ballots and tallying the results, optical scanners are used in an optical-scan machine voting system.

However, these conventional schemes can't satisfy a truly secure and verifiable election while maintaining privacies of voters since they cannot demonstrate their honest operations without revealing individual votes. Likewise, these systems are not competent as they are conducted manually and hence very often they are inaccurate and it takes huge time for final tally.

On the other hand, electronic voting (e-voting) systems improves the limitations of conventional voting systems and enable accurate, efficient, verifiable and convenient elections. Electronic voting is basically based on computers, computer networks and cryptographic protocols. Likewise the resources of e-voting schemes (*e.g*. the software, the communication mechanisms and the computing devices) are reusable, therefore e-voting based elections become cheap and economic. Furthermore, any geographical proximity of voter is not necessary (*e.g*. employees or soldiers working abroad can participate in elections) and they deliver better scalability for large scale public elections [2]. The number of people those who usually do not participate in elections because of the inconveniences of conventional voting systems may be encouraged by the above conveniences of e-voting systems and thereby the number of vote castings can be maximized in elections.

However e-voting schemes have potential problems which may degrade their credibility. For instances, issuing of a unique identification number to each voter to the verification of the accuracy of elections smoothly would enable the authority (or authorities) identifying the linkages between voters and their votes and disclosing the privacy of the voters [3]. When election authority issues receipts to voters to prove its honesty, coercers can force voters to follow their intentions more easily. On the contrary, complicated mechanisms that achieve complete anonymity of voters while maintaining verifiability of their votes make e-voting systems non-scalable and non-practical. For instance, many election schemes include zero knowledge proof (ZKP) (either interactive or non-interactive) proving the correct behavior of entities *e.g*. for the confirm of only eligible votes are accepted and all eligible votes are counted, however ZKP involves complicated computations and communications which make e-voting schemes impractical [4]. Also in many existing schemes, reliability of authorities is expected conducting the election *e.g*. to generate and distribute tokens while registering the legitimate voters for the election, which lead to sacrifice privacy of voters and incoercibility. Moreover the assumption of the existence of trusted or absolutely trusted authority (or authorities) is impractical. Likewise, the vote formats of many prevailing e-voting schemes are rigid, *e.g*. some of them can support only yes/no votes, or simple one out of two candidate elections or some other schemes can support only pre-specified candidates elections.

E-voting scheme [3] must fulfil extensive requirements related to flexibility, privacy, integrity, implementation, verifiability of vote formats and the assumptions about credibility of involved authorities. It is highly challenging that e-voting schemes is to satisfy even mutually contradictory requirements, in addition to satisfying all of them altogether.

So, in this proposed e-voting scheme in this research, following e-voting characteristics are also considered with our main objective.

1) Fulfils all the security requirements of e-voting systems *i.e*. privacy, integrity, universal verifiability, fairness, accuracy, incoercibility, receipt-freeness, practicality, robustness, scalability, integrity and dispute-freeness; usually found as traded in existing e-voting schemes [2, 5],
2) The scheme is centered on the weaker assumptions about credibility of entities, *i.e*. none can make the scheme untrustworthy if at least one authority is authentic among multiple authorities, and
3) It assists flexible candidate selection *i.e*. accommodating freely chosen write-in ballots, votes for pre-specified or t out of l choices as well as yes/no votes.

1.3 Overview of the field

Existing e-voting schemes (based on adopted cryptographic techniques) can be classified into three categories: (i) blind signature based schemes [3, 6], (ii) homomorphic encryption based schemes [7, 8, 9] and (iii) mixnet based schemes [5, 10]. A lot of hybrid of homomorphic encryption and mixnet based schemes [11, 12] are also available. In addition to these schemes, paper based cryptographic voting schemes [13, 14] relying on visual cryptography have been projected. However, existing schemes are unable to satisfy all the essential requirements of e-voting systems at the same time because the tradeoffs among the individual requirements and constraints are remarkable.

Similarly, achieving the verifiability of votes or proving the truthful behaviors of voting authorities, practically all of these schemes extensively deploy ZKP, which the costly one, not efficient and not practical enough, as it involves complicated computations and communications. For instance, homomorphic encryption based schemes use ZKP that is to prove the validity of votes also their correct decryptions, and mixnet based voting schemes use ZKP that is to prove the correctness of operation of each mix-server. Consequently existing e-voting systems which are available currently can fulfill only a part of the requirements of voting and they are non-scalable also non-practical.

To achieve verifiability rather than deploying ZKP, e-voting scheme proposed in [15] involves confirmation numbers (*CNs*) to votes individually disabling anyone to know the link between a vote and its voter. CN based e-voting scheme [15] successfully satisfies essential requirements of voting. However, because RSA encryption functions are not probabilistic or commutative, additional mechanisms and constraints become necessary which degrade its performance and the cryptosystem proposed in [15] that cannot ensure the use of public keys for encryption and verification purposes. Herein, multiple mutually independent tallying authorities and the voter need to keep their individual keys as secret while encryption-decryption and/or signing verification operations. Then along with authorities, it is mandatory for the voter to encrypt and decrypt the vote by itself. Also, it requires a pair of signatures of authorities to ensure the verifiability of the involved mixnet.

Another e-voting scheme proposed in [20] avoids ZKP, decomposes the vote to protect its voter from the coercer and exploits R-SVRM [20] to ensure the verifiability of vote construction as well as of mixnet. ElGamal encryption functions are probabilistic and commutative, the R-SVRM based scheme becomes simpler and efficient than the CN based one as above. Namely, extra random factors are not necessary for encrypting votes, honest behaviors of mix-servers can be verified without authorities' signatures. In addition, encryption keys of mix-servers can be made public. But the encrypted form of vote still has high complexity. However, as discussed later on, its verification process is still colossal. Thus, these schemes are also unscalable.

1.4 Overview of the proposed e-voting scheme

Key mechanisms of the proposed e-voting scheme in this thesis are R-SVRM with confirmation numbers (*CNs*). Here *CNs* are publicly disclosed and registered unique numbers and they are attached to votes of individual voters, and R-SVRM is a mechanism of mixnet also used to develop an electronic voting (e-voting) scheme. A mixnet shuffles, encrypts and decrypts given data so that no one can know the correspondences between inputs and outputs of mixnet.

CNs involved in individual votes make votes verifiable while disabling all entities including voters themselves to know the linkages between voters and their votes. *CNs* are unique registered numbers and they are encrypted by multiple entities independently, so that no one knows their exact values. Therefore anyone can convince itself the authenticity of votes when attached *CNs* are the registered ones. Nevertheless any link between voters and their votes is removed because no one knows the decrypted forms of *CNs* attached to voters. Also publicly disclosed encrypted *CNs* ensure that all votes from eligible voters are counted, and thereby maintain the total accuracy of the election while protecting all privacies of voters. Different from ZKP, a mechanism for *CNs* is simple enough, it requires much less computations for individual entities without assuming any absolutely trustworthy election authority. Because of *CNs* this scheme requires much more simple computations for election entities in comparison with other existing schemes. The proposed scheme does not need any extra proof of correctness of votes.

The R-SVRM based scheme uses ElGamal encryption functions which are probabilistic and commutative and makes simpler and efficient e-voting system. Mix-servers re-encrypt and shuffle votes, so no one can link between CN (confirmation no) of voters and vote. Namely, extra random factors are not necessary for encrypting votes, honest behaviors of mix-servers can be verified without their signatures. In addition, encryption keys of mix-servers can be made public. So, it is possible to develop e-voting systems that satisfy all the requirements including scalability and practicality.

So, we conclude that this paper will improve the efficiency and simplicity of the R-SVRM based scheme by introducing CNs that were used in the CN based scheme. By decreasing the numbers of items in individual vote forms from 3 to 2 and excluding items that include information about candidates as exponents, it reduces the required number of cryptographic operations and simplifies verification procedures of individual operations. Hence, essential requirements of elections are satisfied more easily.

1.5 Thesis Organization

The rest of the thesis is organized as follows.

- **Chapter II** discusses requirements of ideal e-voting schemes and represents the existing works in the related field and focuses on the advantages and drawbacks of existing works**.**
- **Chapter III** discusses the *CN* based, the R-SVRM based e-voting schemes, and the mechanism of anonymous credential.
- **Chapter IV** discusses our proposed e-voting scheme based on R-SVRM while exploiting *CN*s.
- **Chapter V** evaluates the scheme by comparing the computation volume (time, efficiency) and security requirements among the proposed scheme, *CN* based and R-SVRM based scheme.
- **Chapter VI** concludes this thesis together with the outline of probable future directions of research opened by this work.

CHAPTER II

Requirements and Related Works

This chapter discusses requirements of ideal e-voting schemes and some existing e-voting schemes that have been proposed till now.

2.1 Requirements of e-voting schemes

E-voting schemes require to fulfil widespread requirements, among them some of the requirements are at odds with others where there are compromises. Since these sorts of requirements, voting is one of the most challenging applications of information security technologies. Perfect e-voting schemes should fulfil the following requirements [2, 5, 14].

- **Eligibility:** The most basic requirement to conduct reliable elections is that only persons who fulfils certain pre-determined criteria *e.g.* only citizens are allowed to cast permitted number of votes. For achieving this, eligibility of voters, authority requires to verify as well as record their casting votes.
- **Privacy:** Voters typically do not want others to know their casting votes including election authorities. So, except anyone’s own vote, it must not be able to know others votes. In order to achieve this, during the whole election any traceability between voters and their votes must be removed, *i.e.* at every stage of the election it is essential to hide the identity of voters or votes.
- **Integrity**: Integrity of vote refers to protecting vote from being modified by unauthorized parties. Voter may verify the correctness of encryption of all mix-servers.
- **Accuracy:** Voters expect their votes are correctly casted and all eligible votes are properly tallied in elections. It should be noted that accuracy is the degree of satisfactions of the voters’, and can be maintained by the verifiability mentioned below.
- **Verifiability:** It is the ability for the determination whether only and all lawful votes are tallied in final tally or not *i.e.* for the determination of the correctness of the election. Correctness of the election can be verified in two ways, *individual verifiability* is the one where only voters can verify their own votes in the tally. When

there are less than or equal to *n* votes and all *n* voters verify their votes, correctness of the election consists of *n* voters is ensured. *Universal verifiability* is the other one which enables any third party verifying the correctness of the election.

- **Fairness:** For conducting the neutral election, none is allowed for the computation of the partial tally before the end of the election that may influence the remaining voters and the voting result may be affected. Some voting schemes belief if the authorities will not disclose partial tally *e.g.* [5, 9], but this kind of assumptions must be exclude.

- **Receipt-freeness:** Receipt-freeness incapacitates anyone including voters themselves linking voters to their votes, for protecting voters from being coerced following intentions of other entities. Achieving receipt-freeness, the voting system should not leave any information about votes to voters. Likewise, votes should not consist of any information peculiar to the voters. Though a vote includes some traceable information about the corresponding voter, this information can work as the receipt. E-voting systems permit entities in gathering data easily about voters and their votes and link them each other when the receipt-freeness is not confirmed, consequently e-voting schemes cannot be used for real political elections without ensuring receipt-freeness. Authorities consign random numbers to voters to be put in their votes *e.g.* [7, 8, 9] in some voting schemes and is not able to achieve receipt-freeness completely because it is easy for the authorities to link voters to their votes based on these random numbers. The same conception with privacy is shared by receipt-freeness.

- **Incoercibility:** Incoercibility safeguards voters against coercers who can communicate with the voters actively. Incoercibility must cope with randomization, forced-abstention, and simulation attacks.

  - Randomization attacks force voters submitting invalid votes by manipulating the manner in which votes are cast.
  - Forced-abstention attacks enable coercers to force voters abstaining from casting their votes, and
  - Simulation attacks let coercers impersonate valid voters at some stage of the voting scheme and surrender to votes on their behalf.

  Though receipt-freeness property does not imply incoercibility, incoercible schemes must be receipt-free.

- ✧ **Dispute-freeness:** Conducting elections in environments where even dishonest voters are involved, involving relevant entities disagreements between entities should be solved. The conception of universal verifiability is similar to dispute-freeness is limited to the voting and tallying stages.
- ✧ **Robustness:** Any entity does not supposed to be able to disrupt the voting, *i.e*. the voting system should have the capacity detecting dishonest entities also completing the voting process without the help of detected dishonest entities.
- ✧ **Scalability:** To enable large scale elections, a scheme has to be prolonged easily satisfying computation, storage requirements, and communication of the scheme.
- ✧ **Practicality**: A scheme ought not have assumptions and requirements which are difficult for the implementation.

Some requirements are usually satisfied among these and their implementation is simple, but some are difficult to satisfy. Especially satisfying some hard requirements altogether at the same time is really difficult to tradeoff among them. For instance, achieving incoercibility leads sacrificing universal verifiability and henceforth accuracy since incoercible schemes hides the links between voters and their votes while vote submission. As another example, satisfying dispute-freeness lets schemes complex [2] for the reason that for every stage of the election, dispute-free schemes is to prove the legitimacy of all actions of all involved entities and consequently schemes become impractical or unscalable. Similarly write-in ballots rattle with the properties of receipt-freeness of universally verifiable schemes and randomization attacks (previously discussed, that means to force a voter to vote in a certain way). At this point write-in ballot is a ballot in which a voter can insert a freely chosen message a right protected in certain legislations and jurisdictions [16]. Here, peculiar information inserted within write-in ballots can be used as receipts of their corresponding voters, also by this means coercers can mount randomization attack by manipulating voters submitting invalid votes.

Conversely sacrificing one requirement sometimes also leads to sacrificing another one or more requirements for the reason that they are mutually interrelated and dependent. For instance, the maximal level of fairness and privacy preservation has the same notion against corrupt authorities. For the reason that maximal privacy suggests the privacy of a voter to be penetrated only with a consent of all remaining entities *e.g*. authorities and voters, also while desirable, requires all the voters to either participate in the post-vote-casting stage or mandatorily cast their votes (*i.e.* no abstaining). In this situation, breaching the privacy of voters enables corrupted authorities to reveal or modify the partial tally. E-voting schemes (existing many) can satisfy only a part of the above requirements. For example, voting scheme proposed in [17] can fulfil accuracy, privacy, fairness, dispute-freeness, universal verifiability, and practicality, but it cannot satisfy either of receipt-freeness, robustness, scalability or incoercibility. Nevertheless e-voting systems must adjust with intrinsic tradeoffs among these requirements.

2.2 Related works

2.2.1 Schemes with ZKP and specialized hardware, software

A lot of widespread researches on e-voting schemes till now have been organized. In recent times, a number of blind signature (BS), homomorphic encryption, and mixnet based voting schemes have been projected along with different cryptographic techniques. By using specialized hardware like tamper resistant randomizer (TRR) [5], several schemes accomplish receipt-freeness. Likewise, ensuring the correctness of votes, they use zero knowledge proof (ZKP), which needs weighty computations. Once more, authorities may discover the random number of a voter and use it to link the voter using specialized devices in these schemes which shows that these schemes are not completely receipt-free. Though the principle/criterion of TRR suggested in [5] is such that the voter exploiting it eventually loses her knowledge on randomness, here TRR has impaired the practicality of this scheme.

The scheme proposed in [15] fulfills major security requirements, and its deployment of cryptosystem supports homomorphic, probabilistic and commutative [19] properties altogether. However, engaged entities keys are required for both encryption and decryption because of its exploited cryptosystem and it is required to keep as secret that is signing verification. So a voter is to interact with authorities whereas encrypting his or her vote and/or confirming the correctness of encryption and signing operations. These increase involved entities' computation and communication overheads, also make the scheme unscalable. The 'proxy e-voting scheme' is proposed in [21] as exploits proxy signature enabling a voter to envoy a proxy to cast her vote. However, the authority can detect the responsible voter because of its 'double voting detection' capability, while double voting takes place. Thus the link between the vote and its voter is exposed which sacrifices the privacy of the voter. Another scheme known as Helios [22], that is the first web based open auditing system, however cannot provide a strong guarantee of privacy, which satisfies both individual and universal variability. As a client program, it runs a browser, and by using the browser a voter can submit his vote. In conclusion, it shuffles all encrypted votes disabling the link between a vote and its voter while vote submission closes, and produces a non-interactive ZKP proving the perfection of shuffling.

2.2.2 Schemes based on blind signature (BS)

E-voting schemes based on BS are simple and efficient in implementation, not exploiting complicated ZKP and supporting flexible vote formats. But the voter's striking factors can be used as a receipt of the vote and in that way the receipt-freeness is sacrificed. Similarly, as every vote is blinded and unblinded only by its corresponding voter, this produces universal veriability [23, 24]. A scheme suggested in [26] is based on Chaum's BS. In this voting system, a registered voter submits her unblinded signed vote anonymously while voting. Then, it is duty to publish a list of received ballots that is accessible by all voters. Lastly in order for decrypting the vote, each voter requires to interact with the tallying authority by sending her private key. Though the scheme satisfies privacy, scalability, fairness, etc; its' major limitation is the registration authority can identify the nonparticipating registered voters and can add votes for them. The scheme proposed in [27] abuses a uniquely threshold BS to get blind threshold votes, and lets any registered voter abstaining from vote submission. To guarantee the fairness among the candidates campaign, it also uses threshold cryptosystem. It can achieve fairness and accurateness conditionally, though it satisfies scalability, robustness and practicality.

Another scheme suggested in [28] for the deployment of pseudo-voter identity (PVID) developed by Chaum's BS ensuring the voter's privacy. It doesn’t use other complex cryptographic algorithms like homomorphic encryption or ZKP, and has not any physical assumptions such as untappable channels. However, it has some shortcomings, i.e., while key generator, ballot generator and counter work together and contrive, they can alter casted votes. Likewise there is possibility of corrupted authority may trace the voter's IP over the internet. Furthermore, the scheme is not so robust and can satisfy practicality and fairness conditionally. Likewise, it involves multiple mutually independent signing authorities; thereby nothing can make the scheme untrustworthy while at least a single authority is authentic. Additionally here, as data about interactions among entities are publicly verifiable; disputes are resolvable.

2.2.3 Some recent schemes to attain incoercibility

Recently some other schemes are proposed like Civitas [29], UVote [18], Cobra [31, 32] etc. Civitas [29] is based on the mechanism suggested in [30] and aims to fulfil both incoercibility and veriability. Nevertheless to achieve incoercibility, it lets the voter to submit multiple votes where multiple votes with the same token are excluded during the tallying. Here, each voter requires to include ZKPs indicating which earlier votes to be erased as well as showing the knowledge of the choice as well as the token used in earlier votes. The scheme suggested in [30] also exploits ZKP. Though here incoercibility is achieved; unfortunately accuracy and scalability are compromise.

UVote [18] permits a registered voter submitting multiple votes from which only the last vote is counted, and thus satisfies incoercibility. Initially a voter needs to register her primary account, also later on can add multiple accounts. Any notification and message is sent only to the primary account and it cannot be deleted from online for verification. If verifiability is achieved, receipt- freeness is given up since a receipt is provided to the voter. A registered voter's encrypted credential is attached with an encrypted bloom filter in Cobra [32]. The voter selects certain number of candidate passwords and registers anyone of them. Then, the voter encrypts his/her vote using the registered password regenerating the credential. Here, as the voter can deliver a fake or a panic password to the coercer and thus he is not able to manipulate the voter; incoercibility is achieved but thereby verifiability is compromise. Some schemes known as paper based cryptographic voting schemes those are based on visual cryptography [33]. Yet here; a voter has to envoy her computations in the voting booth. So, the booth can easily detect the vote of a voter. The process of prepare paper ballots in advance do not ensure privacy against its creators' [33]. Sandler et al. [34] have developed voting scheme, considering commercial e-voting scheme that is based on cryptographic techniques and hardware/machines, like digital-recording electronic (DRE), optical scan voting machine, etc.

2.2.4 Mixnet based Schemes that do not used ZKP

E-voting scheme proposed in [15] involves confirmation numbers (CNs) to votes individually disabling anyone to know the link between a vote and its voter. CN based e-voting scheme [15] successfully satisfies essential requirements of voting. However, because RSA encryption functions are not probabilistic or commutative, additional mechanisms and constraints become necessary which degrade its performance and the cryptosystem proposed in [15] that cannot ensure the use of public keys for encryption and verification purposes. Herein, multiple mutually independent tallying authorities and the voter need to keep their individual keys as secret while encryption-decryption and/or signing verification operations. Then along with authorities, it is mandatory for the voter to encrypt and decrypt the vote by itself. Also, it requires a pair of signatures of authorities to ensure the verifiability of the involved mixnet.

Another e-voting scheme proposed in [20] avoids ZKP, decomposes the vote to protect its voter from the coercer and exploits R-SVRM [20] to ensure the verifiability of vote construction as well as of mixnet. ElGamal encryption functions are probabilistic and commutative, the R-SVRM based scheme becomes simpler and efficient than the CN based one as above. Namely, extra random factors are not necessary for encrypting votes, honest behaviors of mix-servers can be verified without authorities' signatures. In addition, encryption keys of mix-servers can be made public. But the encrypted form of vote still has high complexity. However, as discussed later on, its verification process is still colossal. Thus, these schemes are also unscalable.

CHAPTER III

Allied E-voting Schemes and Security Components

This chapter summarizes the CN based [15], and the R-SVRM based [20] e-voting schemes that exploit RSA and ElGamal based mixnets respectively. Also briefly, it states the mechanism of anonymous credential [35] that enables to authenticate voters anonymously. In the followings, it is assumed that both mixnets consist of P-mutually independent mix-servers $M_1, M_2,\ldots,M_P$, and there are N-voters $V_1, V_2,\ldots,V_N$.

3.1 *CN* Based Scheme

Encryption functions of mixnets in e-voting systems must be probabilistic; if not probabilistic, the encrypted forms of same candidates are always same. Therefore, a voter can know votes of other voters even if they are encrypted when multiple voters choose the same candidates. They must be commutative also in cases where mix-servers $M_1,\ldots,M_P$ sign on encrypted votes to convince others that the votes were correctly handled. When they are not commutative, the signed form of encrypted vote $S_{K_s}(E_{K_e}(v))$ cannot be decrypted to plain signed form $S_{K_s}(v)$. Here, $E_{K_e}(v)$ is the encrypted form of vote v using encryption key K_e and $S_{K_s}(v)$ is the signed form of v using signing key K_s.

Although the CN based e-voting scheme uses RSA encryption functions, they can be made probabilistic and commutative [19]. Firstly, to make RSA encryption functions probabilistic, each mix-server M_i encrypts v while adding a secret random factor as described in fig. 3.1, *i.e.* M_i mixes v with a secret integer r to encrypt v, and calculates $E_{\{K_{(i)}, H_{(i)}\}}(r, v) = \{E_{K_{(i)}}(vr) = (vr)^{K(i)} (\mathrm{mod}\, p_1), E_{H_{(i)}}(r) = r^{H(i)} (\mathrm{mod}\, p_2)\}$ by encryption keys $K_{(i)}$ and $H_{(i)}$. While decryption, $E_{\{K_{(i)}, H_{(i)}\}}(r, v)$ is decrypted to pair $\{vr, r\}$ by decryption keys $F_{(i)}$ and $G_{(i)}$, and v is obtained as $v = vr / r$. Fig. 3.1 shows encrypted form Data part and Randomization part of vote.

Data part	Randomization part
$E_{K_{(i)}}(vr) = (vr)^{K(i)}$ (mod p_1)	$E_{H_{(i)}}(r) = r^{H(i)}$ (mod p_2)

Fig.3.1 Encrypted form of Vote.

In the above, $\{K_{(i)}, F_{(i)}\}$ and $\{H_{(i)}, G_{(i)}\}$ are encryption and decryption key pairs of RSA encryption functions owned by M_i. In detail, provided that $p_{1(+)}$, $p_{1(-)}$, $p_{2(+)}$ and $p_{2(-)}$ are large prime numbers and $p_1 = p_{1(+)}p_{1(-)}$, $p_2 = p_{2(+)}p_{2(-)}$, relations $u^{K(i)F(i)}$ (mod p_1) = u (mod p_1) and $w^{H(i)G(i)}$ (mod p_2) = w (mod p_2) hold for any integer u and w. In the followings, notations (mod p_1) and (mod p_2) are omitted.

RSA encryption functions $E_{K_{(i)}}(v)$ and $E_{K_{(j)}}(v)$ become also commutative when $M_1,\ldots,M_P$ share the same modulo arithmetic, *i.e.* $E_{K_{(i)}}(E_{K_{(j)}}(v)) = v^{K(j)K(i)}$ is decrypted to v in either way as $((v^{K(j)K(i)})^{F(j)})^{F(i)} = v$ or $((v^{K(j)K(i)})^{F(i)})^{F(j)} = v$. But different from usual RSA encryption scheme, any M_i cannot disclose encryption keys $K_{(i)}$ or $H_{(i)}$ to keep decryption keys $F_{(i)}$ and $G_{(i)}$ as its secrets, *i.e.* disclosure of $K_{(i)}$ by M_i facilitates other M_j to guess $F_{(i)}$ from relation $K_{(i)}F_{(i)}$(mod φ(p_1)) = $K_{(j)}F_{(j)}$ (mod φ(p_1)) where φ(p_1) = ($p_{1(+)}$ – 1)($p_{1(-)}$ – 1).

Under the above settings, $M_1,\ldots,M_P$ in the CN based e-voting scheme handle vote v_n of voter V_n as follows.

Re-encryption: Firstly to conceal v_n from $M_1,\ldots,M_P$; V_n generates secret integers r_n and L_n, calculates $v_n r_n$ and $r_n{}^{L_n}$, and shows pair $\{v_n r_n, r_n{}^{L_n}\}$ to 1st mix-server M_1. Then, $M_1,\ldots,M_P$ sequentially encrypt it into $\{E_{K_{(P)}}(\ldots(E_{K_{(2)}}(E_{K_{(1)}}(v_n r_n)))\ldots) = (v_n r_n)^{K(1)K(2)\ldots K(P)}$, $E_{H_{(P)}}(\ldots(E_{H_{(2)}}(E_{H_{(1)}}(r_n{}^{L_n})))\ldots) = (r_n{}^{L_n})^{H(1)H(2)\ldots H(P)}\} = \{E_{K_*}(v_n r_n), E_{H_*}(r_n{}^{L_n})\}$. After that V_n calculates $E_{\{K_*, H_*\}}(r_n, v_n) = \{(v_n r_n)^{K(1)K(2)\ldots K(P)}, r_n{}^{H(1)H(2)\ldots H(P)}\}$ from $\{E_{K_*}(v_n r_n), E_{H_*}(r_n{}^{L_n})\}$.

In the above, $r_n{}^{L_n}$ is considered as RSA encryption form of $E_{L_n}(r_n)$ that are commutative with each $E_{H_{(i)}}(x)$, *i.e.* $E_{L_n}(x)$ and $E_{H_{(i)}}(x)$ are calculated under the same modulo arithmetic. Therefore V_n that knows L_n can easily calculate $E_{H_*}(r_n)$ from $E_{H_*}(r_n{}^{L_n})$. Also, actually integer r_n is composed as the product of integers that are secrets of V_n and $M_1,\ldots,M_P$. If V_n knows r_n, coercers can know v_n by asking V_n to disclose r_n and L_n.

Re-signing: Then $M_1,\ldots,M_P$ generate 2 different signed forms of $E_{\{K_{(*)}, H_{(*)}\}}(r_n, v_n)$ by their signing keys, thereby later on anyone can verify correct decryptions of votes. These signatures can be generated in the same way because RSA encryption functions are signing functions at the same time. Also V_n can verify the correctness of $E_{\{K_{(*)}, H_{(*)}\}}(r_n, v_n)$ and their signed forms without knowing encryption, decryption or signing keys because RSA encryption functions are homomorphic. To verify $E_{\{K_{(*)}, H_{(*)}\}}(r_n, v_n)$ for example, V_n generates secret integers $\{\delta_n, \sigma_n\}$ and asks M_P,…,M_1 to decrypt $E_{K_*}((v_n r_n)^{\delta_n})$ and $E_{H_*}(r_n{}^{\sigma_n})$. As M_P,…,M_1 do not know δ_n, σ_n, $v_n r_n$ or r_n, they fail to retrieve $(v_n r_n)^{\delta_n}$ and $r_n{}^{\sigma_n}$ from $E_{K_*}((v_n r_n)^{\delta_n})$ and $E_{H_*}(r_n{}^{\sigma_n})$ if they are not correct.

Re-decryption: In the decryption stage, M_P,…,M_1 simply decrypt and shuffle signed forms generated in the above, and they disclose their verification keys to convince anyone that decryption results are legitimate (authenticity of the decryption results will be discussed in the verification stage). Therefore, although decrypted results reveal each pair $\{v_n r_n, r_n\}$, no one except V_n can know the correspondence between $E_{\{K_{(*)}, H_{(*)}\}}(r_n, v_n)$ and decrypted result v_n. Also, because each $E_{\{K_{(i)}, H_{(i)}\}}(r_n, v_n)$ and signing functions are commutative, signatures on encrypted form $E_{\{K_{(P)}, H_{(P)}\}}(r_n, v_n)$ are decrypted to signatures on plain form $\{v_n r_n, r_n\}$.

Verification: To make decryption results verifiable, voter V_n actually constructs an encrypted form of v_n as triplet $\{E_{K_*}(v_n C_n r_n), E_{H_*}(r_n), E_{K_*}(C_n)\}$. Where, integers C_1,…,C_N are registered unique confirmation numbers, and M_1,…,M_P jointly encrypt and shuffle them to generate encrypted forms $E_{K_*}(C_1) = E_{K_{(P)}}(\ldots(E_{K_{(2)}}(E_{K_{(1)}}(C_1)))\ldots)$,…,$E_{K_*}(C_N)$ and disclose them publicly in advance. Then, V_n calculates $E_{K_*}(v_n C_n r_n)$ as the product of $E_{K_*}(v_n r_n)$ and $E_{K_*}(C_n)$ *i.e.* $E_{K_*}(v_n C_n r_n) = E_{K_*}(v_n r_n) E_{K_*}(C_n)$ (as RSA encryption functions are homomorphic) where $E_{K_*}(C_n)$ is assigned to it.

As a result, anyone can confirm that votes are correctly decrypted. Namely, mix-server that does not know signing keys of other mix-servers cannot forge decryption forms consistently so that their 2 signed forms become consistent. Because each C_n is unique, and C_n and $E_{K_*}(C_n)$ are publicly disclosed, anyone can convince itself that only and all votes of legitimate voters are decrypted when decrypted results are accompanied by different registered numbers C_{h1},…,$C_{h\prod}$. Each V_n can maintain v_n as its secret of course because no one knows the correspondence between C_n and $E_{K_*}(C_n)$.

CN based e-voting scheme successfully satisfies essential requirements of voting as discussed above. However, because RSA encryption functions are not probabilistic or commutative, additional mechanisms and constraints become necessary which degrade its performance.

3.2 R-SVRM Based Scheme

In the R-SVRM based e-voting scheme, each mix-server M_i maintains secret decryption key $X_{(i)}$ and public encryption key $Y_{(i)} = g^{X_{(i)}}$ (mod Q) of an ElGamal encryption function. Also, to encrypt the vote v_n of voter V_n, M_i generates secret integers $r_{(n,i)}$, $s_{(n,i)}$, $u_{(n,i)}$, $k_{(n,i)}$, $t_{(n,i)}$ and $w_{(n,i)}$. Where, g is an appropriate integer and Q is a large prime number, and they are publicly known. In the followings, notations Y_*, X_*, $r_{(n*,i)}$, $s_{(n*,i)}$, $u_{(n*,i)}$, $k_{(n*,i)}$, $t_{(n*,i)}$, and $w_{(n*,i)}$ represent $Y_* = Y_{(1)}\ldots Y_{(P)} = g^{X_{(1)}+\cdots+X_{(P)}} = g^{X_*}$, $r_{(n*,i)} = \prod_{i\in P} r_{(n,i)} = r_{(n,1)}.r_{(n,2)}\ldots r_{(n,i)}$, $s_{(n*,i)}= \sum_{i\in P} s_{(n,i)} = s_{(n,1)}+\ldots+ s_{(n,i)}$, $u_{(n*,i)} = \sum_{i\in P} u_{(n,i)}= u_{(n,1)}+\ldots+ u_{(n,i)}$, $k_{(n*,i)} = s_{(n*,P)}+\sum_{i\in P} k_{(n,i)} = s_{(n*,P)}+k_{(n,1)}+\ldots+k_{(n,i)}$, $t_{(n*,i)} = u_{(n*,P)}+\sum_{i\in P} t_{(n,i)} = u_{(n*,P)}+t_{(n,1)}+\ldots+t_{(n,i)}$ and $w_{(n*,i)}= (u_{(n*,P)}.v_n.(v_n + \Lambda))+ \sum_{i\in P} w_{(n,i)} = (u_{(n*,P)}.v_n.(v_n + \Lambda))+w_{(n,1)}+\ldots+w_{(n,i)}$, respectively (therefore although no one knows X_*, Y_* is publicly known). In addition, provided that v_n is decomposed into products as $v_n = \prod_{i=1}^{P} v_{(n,i)} = v_{(n,1)}.v_{(n,2)}\ldots v_{(n,P)}$; $v_{(n*,i)}$ and $\underline{v}_{(n*,i)}$ represent $v_{(n*,i)} = \prod_{i\in P} v_{(n,i)} = v_{(n,1)}.v_{(n,2)}\ldots v_{(n.i)}$ and $\underline{v}_{(n*,i)} = \prod_{i\in P} {v_{(n,i)}}^2 = {v_{(n,1)}}^2.{v_{(n,2)}}^2\ldots {v_{(n,i)}}^2$ respectively. Votes in the R-SVRM based e-voting scheme are handled as below.

Vote decomposition: In order to conceal its vote from mix-servers $M_1,\ldots,M_P$; firstly V_n decomposes v_n into products as $v_n= \prod_{i=1}^{P} v_{(n,i)}= v_{(n,1)}.v_{(n,2)}\ldots v_{(n,P)}$ (mod Q) and informs each M_i of $v_{(n,i)}$. Then, each M_i generates secret integers $s_{(n,i)}$, $u_{(n,i)}$ and $r_{(n,i)}$, and mix-servers $M_1,\ldots,M_i$ calculates $E_{Y_{*(i)}}\{(s_{(n*,i)}, v_{(n*,i)}), (u_{(n*,i)}, r_{(n*,i)})\} = \{(g^{s_{(n*,i)}}$ (mod Q) $= g^{s_{(n*,(i-1))}}g^{s_{(n,i)}}$ (mod Q), $v_{(n*,i)}{Y_{*(i)}}^{s_{(n*,i)}}$ (mod Q) $= v_{(n*,(i-1))}{Y_{*(i-1)}}^{s_{(n*,(i-1))}}v_{(n,i)}{Y_{(i)}}^{s_{(n,i)}}$(mod Q)), $(g^{u_{(n*,i)}}$ (mod Q) $= g^{u_{(n*,(i-1))}}g^{u_{(n,i)}}$(mod Q), $r_{(n*,i)}{Y_{*(i)}}^{u_{(n*,i)}}$ (mod Q) $= r_{(n*,(i-1))}{Y_{*(i-1)}}^{u_{(n*,(i-1))}}r_{(n,i)}{Y_{(i)}}^{u_{(n,i)}}$(mod Q))$\}$ from $E_{Y_{*(i-1)}}\{(s_{(n*,(i-1))}, v_{(n*,(i-1))}), (u_{(n*,(i-1))}, r_{(n*,(i-1))})\}$ calculated by M_{i-1}.

After that, M_P that calculates $E_{Y_*}\{(s_{(n*,P)}, v_n), (u_{(n*,P)}, r_n)\} = \{(g^{s_{(n*,P)}}, v_n{Y_*}^{s_{(n*,P)}}), (g^{u_{(n*,P)}}, r_n{Y_*}^{u_{(n*,P)}})\}$ (for simplicity, notation (mod Q) is omitted in the followings) sends $(g^{u_{(n*,P)}}, r_n{Y_*}^{u_{(n*,P)}})$ to M_1, and mix-servers $M_1,\ldots,M_i$ calculate $E_{Y_{*(i)}}\{(u_{(n*,P)}\cdot\underline{v}_{(n*,i)}, {r_n}^{\underline{v}_{(n*,i)}}), (u_{(n*,P)}\cdot\Lambda\cdot v_{(n*,i)}, {r_n}^{\Lambda\cdot v_{(n*,i)}})\}$, *i.e.* pairs $\{g^{u_{(n*,P)}\cdot\underline{v}_{(n*,i)}} = \left(g^{u_{(n*,P)}\cdot\underline{v}_{(n*,(i-1))}}\right)^{\underline{v}_{(n,i)}}$, ${r_n}^{\underline{v}_{(n*,i)}}{Y_{*(i)}}^{u_{(n*,P)}\cdot\underline{v}_{(n*,i)}} = ({r_n}^{\underline{v}_{(n*,(i-1))}}{Y_{*(i-1)}}^{u_{(n*,P)}\cdot\underline{v}_{(n*,(i-1))}})^{\underline{v}_{(n,i)}}$ and $\{g^{u_{(n*,P)}\cdot\Lambda\cdot v_{(n*,i)}} = (g^{u_{(n*,P)}\cdot\Lambda\cdot v_{(n*,(i-1))}})^{v_{(n,i)}}$, ${r_n}^{\Lambda\cdot v_{(n*,i)}}{Y_{*(i)}}^{u_{(n*,P)}\cdot\Lambda\cdot v_{(n*,i)}} = ({r_n}^{\Lambda\cdot v_{(n*,(i-1))}}{Y_{*(i-1)}}^{u_{(n*,P)}\cdot\Lambda\cdot v_{(n*,(i-1))}})^{v_{(n,i)}}\}$ from $E_{Y_{*(i-1)}}\{(u_{(n*,P)}\cdot \underline{v}_{(n*,(i-1))}, {r_n}^{\underline{v}_{(n*,(i-1))}})$,

$(u_{(n*,P)}\cdot\Lambda\cdot v_{(n*,(i-1))}, r_n{}^{\Lambda\cdot v_{(n*,(i-1))}})\}$ sent by M_{i-1}. As a result, M_P calculates $E_{Y_*}\{(u_{(n*,P)}\cdot v_n{}^2, r_n{}^{\underline{v}_n}), (u_{(n*,P)}\cdot\Lambda\cdot v_n, r_n{}^{\Lambda\cdot v_n})\}$, and finally V_n that receives $E_{Y_*}\{(s_{(n*,P)}, v_n), (u_{(n*,P)}, r_n)\}$ and $E_{Y_*}\{(u_{(n*,P)}\cdot v_n{}^2, r_n{}^{\underline{v}_n}), (u_{(n*,P)}\cdot\Lambda\cdot v_n, r_n{}^{\Lambda\cdot v_n})\}$ from M_P constructs triplet $E_{Y_*}\{(s_{(n*,P)}, v_n), (u_{(n*,P)}, r_n), (u_{(n*,P)}\cdot v_n\cdot(v_n+\Lambda), r_n{}^{v_n\cdot(v_n+\Lambda)})\}$ as its vote form. Here, Λ is a publicly known integer and $\underline{v}_n = v_n{}^2$, also $E_{Y_*}\{(u_{(n*,P)}\cdot v_n\cdot(v_n+\Lambda), r_n{}^{v_n\cdot(v_n+\Lambda)})\}$ is calculated as the product of $E_{Y_*}\{(u_{(n*,P)}\cdot v_n{}^2, r_n{}^{\underline{v}_n})\}$ and $E_{Y_*}\{(u_{(n*,P)}.\Lambda.v_n, r_n{}^{(\Lambda.v_n)})\}$.

As a result, no one except V_n itself can know v_n unless all mix-servers conspire because each M_i does not know secrets of other mix-servers. Here, V_n can conceal v_n by simply encrypting it by itself, but in this case coercers can ask V_n to disclose its encryption parameters to know v_n. Provided that erasable state voting booths that disable V_n to memorize all information that it had generated and received are available, V_n in the above can protect itself from coercers because it cannot tell others sufficient information for vote re-construction. About verification of $E_{Y_*}\{(s_{(n*,P)}, v_n), (u_{(n*,P)}, r_n), (u_{(n*,P)}\cdot v_n\cdot(v_n+\Lambda), r_n{}^{v_n\cdot(v_n+\Lambda)})\}$, V_n can confirm its correctness without knowing secrets of $M_1,\ldots,M_P$ by exploiting homomorphic property of ElGamal encryption functions as same as in the CN based scheme.

Re-encryption: By using its secret integers $k_{(n,i)}$, $t_{(n,i)}$ and $w_{(n,i)}$, each mix-server M_i re-encrypts and shuffles encrypted form $E_{Y_*}\{(s_{(n*,P)}, v_n), (u_{(n*,P)}, r_n), (u_{(n*,P)}\cdot v_n\cdot(v_n+\Lambda), r_n{}^{v_n\cdot(v_n+\Lambda)})\}$ constructed by voter V_n to $E_{Y_*}\{(k_{(n*,P)}, v_n), (t_{(n*,P)}, r_n), (w_{(n*,P)}, r_n{}^{v_n\cdot(v_n+\Lambda)})\} = \{(g^{k_{(n*,P)}}, v_nY_*{}^{k_{(n*,P)}}), (g^{t_{(n*,P)}}, r_nY_*{}^{t_{(n*,P)}}), (g^{w_{(n*,P)}}, r_n{}^{v_n\cdot(v_n+\Lambda)}Y_*{}^{w_{(n*,P)}})\}$. In detail, M_i that receives $E_{Y_*(i-1)}\{(k_{(n*,(i-1))}, v_n), (t_{(n*,(i-1))}, r_n), (w_{(n*,(i-1))}, r_n{}^{v_n\cdot(v_n+\Lambda)})\}$ from M_{i-1}, calculates $\{(g^{k_{(n*,(i-1))}}g^{k_{(n,i)}} = g^{k_{(n*,i)}}, v_nY_{*(i-1)}{}^{k_{(n*,(i-1))}}Y_{(i)}{}^{k_{(n,i)}} = v_nY_{*(i)}{}^{k_{(n*,i)}}), (g^{t_{(n*,(i-1))}}g^{t_{(n,i)}} = g^{t_{(n*,i)}}, r_nY_{*(i-1)}{}^{t_{(n*,(i-1))}}Y_{(i)}{}^{t_{(n,i)}} = r_nY_{*(i)}{}^{t_{(n*,i)}}), (g^{w_{(n*,(i-1))}}g^{w_{(n,i)}} = g^{w_{(n*,i)}}, r_n{}^{v_n\cdot(v_n+\Lambda)}Y_{*(i-1)}{}^{w_{(n*,(i-1))}}Y_{(i)}{}^{w_{(n,i)}} = r_n{}^{v_n\cdot(v_n+\Lambda)}Y_{*(i)}{}^{w_{(n*,i)}}) = E_{Y_{*(i)}}\{(k_{(n*,i)}, v_n), (t_{(n*,i)}, r_n), (w_{(n*,i)}, r_n{}^{v_n\cdot(v_n+\Lambda)})\}$. Therefore, anyone including V_n itself cannot identify the correspondence between $E_{Y_*}\{(s_{(n*,P)},v_n), (u_{(n*,P)}, r_n), (u_{(n*,P)}\cdot v_n\cdot(v_n+\Lambda), r_n{}^{v_n\cdot(v_n+\Lambda)})\}$ and $E_{Y_*}\{(k_{(n*,P)}, v_n), (t_{(n*,P)}, r_n), (w_{(n*,P)}, r_n{}^{v_n\cdot(v_n+\Lambda)})\}$.

Re-decryption: Mix-servers $M_P,\ldots,M_1$ decrypt encrypted vote $E_{Y_*}\{(k_{(n*,P)}, v_n), (t_{(n*,P)}, r_n), (w_{(n*,P)}, r_n{}^{v_n\cdot(v_n+\Lambda)})\} = \{(g^{k_{(n*,P)}}, v_nY_*{}^{k_{(n*,P)}}), (g^{t_{(n*,P)}}, r_nY_*{}^{t_{(n*,P)}}), (g^{w_{(n*,P)}}, r_n{}^{v_n\cdot(v_n+\Lambda)}Y_*{}^{w_{(n*,P)}})\}$ by their decryption keys $X_{(P)},\ldots,X_1$. Namely, provided that $Y_{*(i)} =$

$g^{X_{(1)}+\cdots+X_{(i)}}$, each M_i decrypts $E_{Y_{*(i)}}\{(k_{(n*,P)}, v_n), (t_{(n*,P)}, r_n), (w_{(n*,P)}, r_n{}^{v_n\cdot(v_n+\Lambda)})\}$ received from M_{i+1} to $\{(g^{k_{(n*,P)}}, v_nY_{*(i)}{}^{k_{(n*,P)}} / g^{k_{(n*,P)}X_{(i)}}), (g^{t_{(n*,P)}}, r_nY_{*(i)}{}^{t_{(n*,P)}} / g^{t_{(n*,P)}X_{(i)}}), (g^{w_{(n*,P)}}, r_n{}^{v_n\cdot(v_n+\Lambda)}Y_{*(i)}{}^{w_{(n*,P)}} / g^{w_{(n*,P)}X_{(i)}})\} = \{(g^{k_{(n*,P)}}, v_nY_{*(i-1)}{}^{k_{(n*,P)}}), (g^{t_{(n*,P)}}, r_nY_{*(i-1)}{}^{t_{(n*,P)}}), (g^{w_{(n*,P)}}, r_n{}^{v_n\cdot(v_n+\Lambda)}Y_{*(i-1)}{}^{w_{(n*,P)}})\} = E_{Y_{*(i-1)}}\{(k_{(n*,P)}, v_n), (t_{(n*,P)}, r_n), (w_{(n*,P)}, r_n{}^{v_n\cdot(v_n+\Lambda)})\}$. Therefore, triplet $\{v_n, r_n, r_n{}^{v_n\cdot(v_n+\Lambda)}\}$ is extracted from the final decryption result $E_{Y_{*(0)}}\{(k_{(n*,P)}, v_n), (t_{(n*,P)}, r_n), (w_{(n*,P)}, r_n{}^{v_n\cdot(v_n+\Lambda)})\} = \{(g^{k_{(n*,P)}}, v_n), (g^{t_{(n*,P)}}, r_n), (g^{w_{(n*,P)}}, r_n{}^{v_n\cdot(v_n+\Lambda)})\}$.

Verification: Provided that triplet $\{\Gamma, \Omega, \Phi\}$ is a final decryption result in the above, relation $\Phi = \Omega^{\Gamma(\Gamma+\Lambda)}$ must hold if it is legitimate, therefore anyone can determine that $\{\Gamma, \Omega, \Phi\}$ is incorrect when the relation does not hold. Here, although $M_1,\ldots,M_P$ that know public encryption keys can easily forge encryption or decryption forms so that decryption result $\{\Gamma, \Omega, \Phi\}$ satisfies $\Phi = \Omega^{\Gamma(\Gamma+\Lambda)}$. But $M_1,\ldots,M_P$ are disabled to behave dishonestly when each M_i discloses the sum of its secret integers. In other words,$M_1,\ldots,M_P$ can prove their honest encryptions and decryptions without revealing secrets of honest entities.

In detail, to convince any entity A that M_i encrypted pairs $(g^{k_{1*(i-1)}}, v_1Y_{*(i-1)}{}^{k_{1*(i-1)}}),\ldots,(g^{k_{N*(i-1)}}, v_NY_{*(i-1)}{}^{k_{N*(i-1)}})$ in encryption forms $E_{Y_{*(i-1)}}\{(k_{1*(i-1)}, v_1), (t_{1*(i-1)}, r_1), (w_{1*(i-1)}, r_1{}^{v_1\cdot(v_1+\Lambda)})\},\ldots,E_{Y_{*(i-1)}}\{(k_{N*(i-1)}, v_N), (t_{N*(i-1)}, r_N), (w_{N*(i-1)}, r_N{}^{v_N\cdot(v_N+\Lambda)})\}$ honestly to $(g^{k_{1*(i)}}, v_1Y_{*(i)}{}^{k_{1*(i)}}),\ldots, (g^{k_{N*(i)}}, v_NY_{*(i)}{}^{k_{N*(i)}})$, firstly M_i discloses sum $K_{(i)}= \sum_{n=1}^{N} k_{(n)(i)} = k_{1(i)}+\ldots+k_{N(i)}$. After that A calculates products $D_{1(i-1)} = g^{k_{1*(i-1)}}g^{k_{2*(i-1)}}\ldots g^{k_{N*(i-1)}} = g^{K_{(1)}+\cdots+K_{(i-1)}}$, $D_{2(i-1)} = v_1Y_{*(i-1)}{}^{k_{1*(i-1)}}v_2Y_{*(i-1)}{}^{k_{2*(i-1)}}\ldots v_NY_{*(i-1)}{}^{k_{N*(i-1)}} = (v_1v_2\ldots v_N)Y_{*(i-1)}{}^{K_{(1)}+\cdots+K_{(i-1)}}$, $D_{1(i)} = g^{k_{1*(i)}}\ldots g^{k_{N*(i)}} = g^{K_{(1)}+\cdots+K_{(i)}}$, and $D_{2(i)} = v_1Y_{*(i)}{}^{k_{1*(i)}}\ldots v_NY_{*(i)}{}^{k_{N*(i)}} = (v_1v_2\ldots v_N)Y_{*(i)}{}^{K_{(1)}+\cdots+K_{(i)}}$. Then, M_i was honest when relations $D_{1(i)}/D_{1(i-1)} = g^{K_{(i)}}$ and $D_{2(i)}/D_{2(i-1)} = Y_{(i)}{}^{K_{(i)}}$ hold. Namely, because $D_{1(i)}/D_{1(i-1)} = g^{K_{(1)}+\cdots+K_{(i)}} / g^{K_{(1)}+\cdots+K_{(i-1)}}$ and $D_{2(i)}/D_{2(i-1)} = (v_1v_2\ldots v_N)Y_{*(i)}{}^{K_{(1)}+\cdots+K_{(i)}} / (v_1v_2\ldots v_N)Y_{*(i-1)}{}^{K_{(1)}+\cdots+K_{(i-1)}}$, $D_{1(i)}/D_{1(i-1)}$ and $D_{2(i)}/D_{2(i-1)}$ must coincide with $g^{K_{(i)}}$ and $Y_{(i)}{}^{K_{(i)}}$ if M_i was honest. On the other hand if encrypted results are incorrect, because solving discrete logarithm problems is difficult, M_i that does not know decryption key X_* cannot find the value of $K_{(i)}$ that satisfies relations $D_{1(i)}/D_{1(i-1)} = g^{K_{(i)}}$, $D_{2(i)}/D_{2(i-1)} = Y_{(i)}{}^{K_{(i)}}$ in addition to $\Phi = \Omega^{\Gamma(\Gamma+\Lambda)}$ for each

$\{\Gamma, \Omega, \Phi\}$. Also, M_i can maintain integers $k_{1(i)},\ldots,k_{N(i)}$ as its secrets even after it had disclosed $K_{(i)}$.

Because ElGamal encryption functions are probabilistic and commutative, the R-SVRM based scheme becomes simpler and efficient than the CN based one as above. Namely, extra random factors are not necessary for encrypting votes, honest behaviors of mix-servers can be verified without their signatures. In addition, encryption keys of mix-servers can be made public. But the encrypted form of vote $E_{Y_*}\{(s_{(n*,P)}, v_n), (u_{(n*,P)}, r_n), (u_{(n*,P)}{}^{v_n \cdot (v_n + \Lambda)}, r_n{}^{v_n \cdot (v_n+\Lambda)})\}$ still has high complexity, *e.g.* it includes vote v_n as exponent.

3.3 Anonymous Credential

Although mixnets conceal correspondences between voters and their votes as discussed above, mechanisms to make voters anonymous are also essential for e-voting schemes. If voters are not anonymous, anyone can know whether a voter abstained from the election or not. Hence, the proposed scheme exploits anonymous credentials to make voters anonymous.

In detail, while disclosing its identity voter V_n receives credential T_n from an election authority B and provided that b is a publicly known appropriate integer, V_n shows $T_n{}^{W_n}$ (mod b) to others without revealing its identity while generating secret integer W_n (in the followings, notation (mod b) is omitted). Where, Z_n is a secret integer that V_n includes in T_n, and V_n convinces others of its eligibility by calculating values that become consistent with $T_n{}^{W_n}$ only by integer Z_n without disclosing Z_n itself. Therefore together with the fact that no one except V_n can know the correspondence between T_n and $T_n{}^{W_n}$, V_n can preserve its anonymity. In addition, in the course V_n calculates the above values, entities can force V_n to calculate used seal U^{Z_n} that is unique to T_n from given integer U while using Z_n in T_n honestly. This means that entities can use U^{Z_n} as an evidence that V_n had shown T_n. Here, no one except V_n can calculate Z_n from U^{Z_n} of course.

CHAPTER IV

DEVELOPMENT OF R-SVRM BASED E-VOTING SCHEME WITH *CN*S

This chapter discusses development of an e-voting scheme based on R-SVRM while exploiting CNs, where notations used in this section are summarized in Table 4.1.

4.1 Entities and Their Roles

The scheme consists of N voters V_n ($n \in \{1,\dots,N\}$), P (at least 2) mutually independent mix-servers M_i ($i \in \{1,\dots,P\}$), booth manager B, and 5 public bulletin boards (BBs) [2, 23] *i.e.*, VoterList, ConfNoList, VotingPanel, ShufflingPanel and TallyingPanel. Their roles are described below. Where $E_{Y_{(i)}}$, $E_{Y_{*(i)}}$ and E_{Y_*} denote encryption by public key/keys of mix-server M_i, mix-servers $M_1,\dots,M_i$, and mix-servers $M_1,\dots,M_P$, respectively.

Voter V_n: Every voter V_n is characterized uniquely by its identifier ID_n, and V_n obtains anonymous credential T_n that includes unique secret integer Z_n from booth manager B by showing ID_n. From now, V_n proves its eligibility to others including B without revealing its identity.

Mix-server M_i: Mutually independent mix-servers $M_1,\dots,M_P$ ($P >= 2$) re-encrypt and shuffle votes submitted by voters on VotingPanel to disclose on ShufflingPanel, and decrypt encrypted votes to disclose on TallyingPanel. Each M_i maintains secret decryption key $X_{(i)}$ and public encryption key $Y_{(i)} = g^{X_{(i)}}$, where g is a publicly known integer common to all votes. M_i also has secret integers $s_{(n,i)}$, $e_{(n,i)}$, $k_{(n,i)}$, $d_{(n,i)}$, and $r_{(n,i)}$, where $s_{(n,i)}$ and $e_{(n,i)}$ are used to conceal voter V_n's vote v_n jointly with V_n. Integers $k_{(n,i)}$ and $d_{(n,i)}$ are used to re-encrypt v_n, and $r_{(n,i)}$ is used to re-encrypt C_n.

Booth manager B: B is responsible for issuing credentials, generating CNs, authenticating anonymous voters and accepting votes submitted by voters. It also identifies liable entities when inconsistent votes are detected. For issuing credentials and accepting encrypted votes, B maintains publicly known integers U and $\underline{U}$ so that each voter V_n can calculate 1st and 2nd used seals U^{Z_n} and $\underline{U}^{Z_n}$ as its approval.

VoterList: VoterList consists of ID and credential parts as shown in Fig. 4.1 (a). The ID part maintains IDs of legitimate voters, and booth manager B puts credential T_n on the credential part when B gives it to voter V_n.

ConfNoList: It consists of CN, encrypted CN and used seal parts as shown in Fig. 4.1(b). N different confirmation numbers $C_1,\ldots,C_N$ generated by B for N voters are disclosed on the CN part. Then, mix-servers $M_1,\ldots,M_P$ re-encrypt and shuffle each C_n to $E_{Y_*}(r_n, C_n)$ so that no one knows correspondences between C_n and $E_{Y_*}(r_n, C_n)$ to be disclosed on the encrypted CN part. When encrypted C_n *i.e.* $E_{Y_*}(r_n, C_n)$ is assigned to voter V_n, it calculates 1st used seal U^{Z_n} by its credential to attach it with $E_{Y_*}(r_n, C_n)$ as the value of used seal part. Then, anyone can confirm that CNs are assigned only to legitimate voters. Because used seals are unique, anyone can confirm that each V_n obtained only one CN also. But no one except V_n can know the voter to whom $E_{Y_*}(r_n, C_n)$ is assigned. In addition, no one including V_n can know the C_n assigned to V_n. To make notations simple, it is assumed that $E_{Y_*}(r_n, C_n)$ is assigned to V_n, although it may be different.

ID	credential
ID_1	T_1
…	
ID_n	T_n
…	
ID_N	T_N

a) VoterList

CN	encrypted CN	used seal
C_1	$E_{Y_*}(r_h, C_h)$	U^{Z_1}
…	…	…
C_n	$E_{Y_*}(r_n, C_n)$	U^{Z_n}
…	…	…
C_N	$E_{Y_*}(r_u, C_u)$	U^{Z_N}

b) ConfNoList

vote	used seal	
<$E_{Y_*}\{s_{(h*,P)}, v_h\}$, $E_{Y_*}\{e_{(h*,P)}+r_h$, $(v_h+\Lambda)C_h\}$>	$\underline{U}^{Z_1}$	
…		
<$E_{Y_*}\{s_{(n*,P)}, v_n\}$, $E_{Y_*}\{e_{(n*,P)}+r_n$, $(v_n+\Lambda)C_n\}$>	$\underline{U}^{Z_n}$	
…		
<$E_{Y_*}\{s_{(u*,P)}, v_u\}$, $E_{Y_*}\{e_{(u*,P)}+r_u$, $(v_u+\Lambda)C_u\}$>	$\underline{U}^{Z_N}$	

c) VotingPanel

vote
<$E_{Y_*}\{k_{(n*,P)}, v_n\}$, $E_{Y_*}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}$>
…
<$E_{Y_*}\{k_{(u*,P)}, v_u\}$, $E_{Y_*}\{d_{(u*,P)}, (v_u+\Lambda)C_u\}$>
…
<$E_{Y_*}\{k_{(h*,P)}, v_h\}$, $E_{Y_*}\{d_{(h*,P)}, (v_h+\Lambda)C_h\}$>

d) ShufflingPanel

vote	CN
v_u	$(v_u+\Lambda)C_u$
v_h	$(v_h+\Lambda)C_h$
v_n	$(v_n+\Lambda)C_n$

e) TallyingPanel

Fig. 4.1Configurations of Bulletin Boards.

VotingPanel: This panel consists of vote and used seal parts, and convinces anyone that encrypted votes on it are legitimate ones because corresponding voters approve them. Booth manager B puts vote v_n concealed by voter V_n on the vote part, and after confirming that its vote is correctly posted on the panel, V_n calculates 2nd used seal $\underline{U}^{Z_n}$ by its credential as its approval to post on the used seal part as shown in Fig. 4.1 (c).

ShufflingPanel: To conceal correspondences between votes put by individual voters and finally decrypted votes, mix-servers re-encrypt and shuffle votes of VotingPanel, and post results on this panel as shown in Fig 4.1(d).

TallyingPanel: As shown in Fig 4.1(e), it is the decrypted form of VotingPanel (and ShufflingPanel) and consists of vote and CN parts which correspond to the 1st and 2nd parts of vote forms respectively.

4.2 Individual Stage

Votes in the proposed scheme are processed through 5 stages, *i.e.* CN generation, registration, voting, tallying and disruption detection stages as follows.

a. CN Generation

In preparations for the election, booth manager B generates integers U, $\underline{U}$, Λ and N-unique confirmation numbers $C_1,\ldots,C_N$; and discloses them publicly. Also, $M_1,\ldots,M_P$ sequentially encrypt and shuffle $C_1,\ldots,C_N$ after they are disclosed on the CN part of ConfNoList. B and $M_1,\ldots,M_P$ behave as follows:

1. B generates integers U, $\underline{U}$, Λ and N-unique confirmation numbers $C_1,\ldots,C_N$ and discloses them on the CN part of ConfNoList.
2. $M_1,\ldots,M_P$ sequentially encrypt and shuffle each C_n on ConfNoList to $E_{Y_*}(r_n, C_n) = \{g^{r_n}, C_n Y_*^{\ r_n}\}$, and put $E_{Y_*}(r_n, C_n)$ on the encrypted CN part of ConfNoList.

About Step 2, each M_i maintains secret integer $r_{(n,i)}$ and r_n is calculated as sum *i.e.* $r_n = \sum_{i=1}^{P} r_{(n,i)} = r_{(n,1)}+\ldots+r_{(n,P)}$. Namely, each M_i calculates $\{g^{r_{(n*,(i-1))}} g^{r_{(n,i)}}, C_n Y_{*(i-1)}^{\ r_{(n*,(i-1))}} Y_{(i)}^{\ r_{(n,i)}}\} = \{g^{r_{(n*,i)}}, C_n Y_{*(i)}^{\ r_{(n*,i)}}\} = E_{Y_*(i)}(r_{(n*,i)}, C_n)$ from $E_{Y_{*(i-1)}}(r_{(n*,(i-1))}, C_n)$ received from M_{i-1}, where $r_{(n*,i)} = \sum_{i \in P} r_{(n,i)} = r_{(n,1)}+\ldots+r_{(n,i)}$. Therefore, no one can know the correspondence between C_n and $E_{Y_*}(r_n, C_n)$.

TABLE-4.1

List of Notations Used in the Proposed Scheme

Notation	Description
N, P (> = 2)	Numbers of voters and mutually independent mix-servers
Q and g	Publicly known appropriate integers used for vote construction
V_n, ID_n	n-th voter and its identifier
M_i	i-th mix-server
B	Booth manager
$X_{(i)}$ X_*	Private key of M_i for vote decryption $X_* = X_{(1)}+\ldots+X_{(P)}$
$Y_{(i)} = g^{X_{(i)}}$ Y_*	Public key of M_i for vote encryption $Y_* = Y_{(1)}\ldots Y_{(P)}$
E_Y(k, ●)	ElGamal encryption form of ● using encryption key Y and secret integer k *i.e.*, $E_Y(k, \bullet) = \{g^k, \bullet Y^k\}$
C_n	Confirmation number assigned to V_n
T_n and W_n	Anonymous credential of V_n, and a secret integer for concealing T_n
U and $\underline{U}$	Publicly known integers for generating used seals
U^{Z_n}, $\underline{U}^{Z_n}$	1st and 2nd used seals calculated by V_n
$r_{(n,i)}$ r_n	Secret integer of M_i to encrypt C_n $r_n = \sum_{i=1}^{P} r_{(n,i)} = r_{(n,1)}+\ldots+r_{(n,P)}$
$s_{(n,i)}$, $e_{(n,i)}$	Secret integers of M_i to conceal $v_{(n,i)}$ and v_n
$s_{(n*,P)}$, $e_{(n*,P)}$	$s_{(n*,P)} = \sum_{i=1}^{P} s_{(n,i)} = s_{(n,1)}+\ldots+s_{(n,P)}$, $e_{(n*,P)} = \sum_{i=1}^{P} e_{(n,i)} = e_{(n,1)}+\ldots+e_{(n,P)}$
$s_{(n*,i)}$, $e_{(n*,i)}$	$s_{(n*,i)} = \sum_{i\in P} s_{(n,i)}$, $e_{(n*,i)} = \sum_{i\in P} e_{(n,i)}$
$k_{(n,i)}$, $d_{(n,i)}$	Secret integers of M_i to re-encrypt v_n
$k_{(n*,P)}$, $d_{(n*,P)}$	$k_{(n*,P)} = s_{(n*,P)}+\sum_{i=1}^{P} k_{(n,i)}$ (= $k_{(n,1)}+\ldots+k_{(n,P)}$), $d_{(n*,P)} = e_{(n*,P)}+r_n+\sum_{i=1}^{P} d_{(n,i)}$ (= $d_{(n,1)}+\ldots+d_{(n,P)}$)
$k_{(n*,i)}$, $d_{(n*,i)}$	$k_{(n*,i)} = s_{(n*,P)}+\sum_{i\in P} k_{(n,i)}$, $d_{(n*,i)} = e_{(n*,P)}+\sum_{i\in P} d_{(n,i)}$
Λ	A publicly known integer to encrypt and verify v_n

A security problem in this stage is mix-server M_i can encrypt CNs incorrectly. Especially the last mix-server M_P can forge consistent encrypted forms so that it can know correspondences between CNs and their encrypted forms. The reason is CNs and encryption keys are publicly known and no one adds operations to encrypted results calculated by M_P. Booth manager B removes these threats as below.

Namely, to detect dishonesties B asks each M_i to disclose the sum of its secret integers $R_{(i)} = \sum_{n=1}^{N} r_{(n)(i)} = r_{1(i)}+\ldots+r_{N(i)}$, and calculates $\Theta_{(i)} = \prod_{n=1}^{N} g^{r_{n*(i)}} = g^{r_{1*(i)}} g^{r_{2*(i)}} \ldots g^{r_{N*(i)}} = g^{r_{1*(i)}+\cdots+r_{N*(i)}}$ and $\Delta_{(i)} = \prod_{n=1}^{N} C_n {Y_{*(i)}}^{r_{n*(i)}} = (C_1 {Y_{*(i)}}^{r_{1*(i)}})(C_2 {Y_{*(i)}}^{r_{2*(i)}})\ldots(C_N {Y_{*(i)}}^{r_{N*(i)}}) = (C_1 C_2 \ldots C_N){Y_{*(i)}}^{r_{1*(i)}+\cdots+r_{N*(i)}}$, *i.e.* products of all CNs encrypted by mix-server M_i ($\Theta_{(0)} = 1$ and $\Delta_{(0)} = C_1 C_2 \ldots C_N$). After that it determines that M_i is dishonest when relation $\Theta_{(i)} / \Theta_{(i-1)} = g^{R_{(i)}}$ or $\Delta_{(i)} / \Delta_{(i-1)} = {Y_{(i)}}^{R_{(i)}}$ does not hold, and simply asks M_i to encrypt CNs again. As discussed in chapter III section 3.2, M_i that does not know secrets of other mix-servers cannot calculate $R_{(i)}$ so that the relations do not hold when it encrypted CNs incorrectly. Also while M_i was determined as a dishonest entity, later on it must encrypt CNs correctly because its' dishonesty is publicly known.

Here, by defining an arbitrary integer T, M_i can forge $\{g^{r_{(n*,i)}}, C_n {Y_{*(i)}}^{r_{(n*,i)}} / T\}$ and $\{g^{r_{(h*,i)}}, C_h {Y_{*(i)}}^{r_{(h*,i)}} T\}$ from correct encrypted forms $\{g^{r_{(n*,i)}}, C_n {Y_{*(i)}}^{r_{(n*,i)}}\}$ and $\{g^{r_{(h*,i)}}, C_h {Y_{*(i)}}^{r_{(h*,i)}}\}$ while satisfying relations $\Theta_{(i)}/\Theta_{(i-1)} = g^{R_{(i)}}$ and $\Delta_{(i)}/\Delta_{(i-1)} = {Y_{(i)}}^{R_{(i)}}$. But different from section 3.2, where individual votes are secrets of voters, CNs are publicly disclosed in their plain forms. Therefore M_i cannot behave as above. Actually, it is possible that C_n / T and $C_h T$ accidentally coincide with C_h and C_n (as a result $\Theta_{(i)}/\Theta_{(i-1)} = g^{R_{(i)}}$ and $\Delta_{(i)}/\Delta_{(i-1)} = {Y_{(i)}}^{R_{(i)}}$ hold). But this modification does not bring any benefit to anyone, *i.e.* correspondences between CNs and their encrypted forms are still unknown.

b. Registration

Booth manager B assigns anonymous credential T_n to each voter V_n so that later on V_n can prove its eligibility without revealing its identity. B and V_n interact as follows:

1. V_n shows its identifier ID_n to B.
2. If V_n is eligible, B asks V_n to include its secret integer Z_n in credential T_n and authorizes the credential by its signature.
3. B discloses the pair $\{ID_n, T_n\}$ on VoterList.
4. V_n generates its receipt if T_n is legitimate.

In the above, V_n cannot obtain multiple credentials, because V_n shows its identifier ID_n and gives its receipt to B. On the other hand B is forced to issue a credential to V_n, *i.e.* B cannot prove that it had issued a credential to V_n without V_n's receipt.

c. Voting

In this stage, each legitimate voter V_n conceals its vote v_n and posts it on VotingPanel through 2 sub-stages, *i.e.* voter acceptance and vote construction. They proceed as follows:

1) Voter Acceptance

The objective of this sub-stage is to authenticate anonymous voters and to allow only eligible ones to participate in voting.

1. V_n generates secret integer W_n and proves its eligibility to B by showing ${T_n}^{W_n}$ and calculating a value that is consistent with ${T_n}^{W_n}$ without revealing its identity, T_n or Z_n. It also calculates 1st used seal U^{Z_n}.
2. If the calculated value is consistent with ${T_n}^{W_n}$ and used seal U^{Z_n} is not registered on ConfNoList yet, B allows V_n to proceed to the next stages while assigning an unused encrypted C_n *i.e.* $E_{Y_*}(r_n, C_n)$ to it.
3. B puts U^{Z_n} on the used seal part of ConfNolist corresponding to $E_{Y_*}(r_n, C_n)$.

In the above, anonymous credential ensures that V_n is anonymous and it can obtain $E_{Y_*}(r_n, C_n)$ only when it is eligible. Therefore, no one other than V_n can know whether V_n had abstained from the election or not. But it must be noted that V_n cannot protect itself from forced abstention. Namely if a coercer asks V_n to calculate used seal U^{Z_n} by credential T_n again, definitely V_n calculates U^{Z_n} honestly. This threat will be discussed later. Other security problems at this stage are:

- B does not assign any CN to V_n or V_n obtains multiple CNs: Because V_n possesses credential T_n and used seal U^{Z_n} can be calculated only from T_n, B cannot refuse to assign C_n to V_n if U^{Z_n} does not exist on ConfNoList and U^{Z_n} is consistent with T_n. Here, Z_n is a secret of V_n, therefore no one except V_n can know U^{Z_n} until V_n calculates it. In contrast, V_n cannot obtain multiple CNs, because only V_n can calculate U^{Z_n}.
- B assigns same encrypted CNs to different voters: As encrypted CNs and used seals are accompanying and already assigned encrypted CNs are disclosed on ConfNoList, B cannot assign the same encrypted CN to multiple voters.
- Any coercer may obtain V_n's used seal: Registered V_n residing within the voting booth

can only interact with B, therefore cannot pass its' U^{Z_n} to any other entity.

2) Vote Construction

In this sub-stage to conceal vote v_n from others voter V_n encrypts v_n, and puts encrypted v_n on VotingPanel. Here, if V_n knows the encryption parameters, coercers can know v_n by asking V_n to encrypt v_n again. Therefore, encryptions are carried out jointly with $M_1,\ldots,M_P$ as in chapter III section 3.2. Fig. 4.2 depicts this sub-stage, *i.e.* interactions between V_n and $M_1,\ldots,M_P$ proceed as follows.

1. At first, V_n decomposes v_n and $(v_n+\Lambda)$ into products as $v_n = \prod_{i=1}^{P} v_{(n,i)} = v_{(n,1)} \cdot v_{(n,2)} \ldots v_{(n,P)}$ and $(v_n+\Lambda) = \prod_{i=1}^{P} \underline{v}_{(n,i)} = \underline{v}_{(n,1)} \cdot \underline{v}_{(n,2)} \ldots \underline{v}_{(n,P)}$ respectively, and sends each $v_{(n,i)}$ and $\underline{v}_{(n,i)}$ to M_i.
2. Each M_i generates secret integers $s_{(n,i)}$ and $e_{(n,i)}$, and calculates pair $<E_{Y_{(i)}}\{s_{(n,i)}, v_{(n,i)}\}, E_{Y_{(i)}}\{e_{(n,i)}, \underline{v}_{(n,i)}\}> = <\{g^{s_{(n,i)}}, v_{(n,i)}{Y_{(i)}}^{s_{(n,i)}}\}, \{g^{e_{(n,i)}}, \underline{v}_{(n,i)}{Y_{(i)}}^{e_{(n,i)}}\}>$.
3. Here, in order to verify the correctness of encrypted form of any ElGamal pair *e.g.* $E_{Y_{(i)}}\{z, m\} = \{g^z, m{Y_{(i)}}^z\}$, V_n calculates $\{g^{\delta 1}, {Y_{(i)}}^{\delta 2}, (gY_{(i)})^{\delta 3}\}$ and sends to M_i from which M_i calculates $\{\Upsilon_1 = g^{z.\delta 1}, \Upsilon_2 = {Y_{(i)}}^{z.\delta 2}, \Upsilon_3 = (gY_{(i)})^{z.\delta 3}\}$ and sends back to V_n, where δ_1, δ_2 and δ_3 are secret integers of V_n. Now, by calculating $\chi_1 = g^{z.\delta 1}$, $\chi_2 = {Y_{(i)}}^{z.\delta 2}$, $\chi_3 = (g^z {Y_{(i)}}^z)^{\delta 3}$ from $E_{Y_{(i)}}\{z, m\}$, V_n confirms the correctness while relations $\Upsilon_1 = \chi_1$, $\Upsilon_2 = \chi_2$ and $\Upsilon_3 = \chi_3$ holds. Thus without knowing secrets of any mix-server, V_n can verify the correctness of encryption through the scheme of Deffie and Hellman [36] as in chapter III section 3.2.

4. Then, V_n that receives $<E_{Y_{(i)}}\{s_{(n,i)}, v_{(n,i)}\}, E_{Y_{(i)}}\{e_{(n,i)}, \underline{v}_{(n,i)}\}>$ from each M_i, calculates $<E_{Y_*}\{s_{(n*,P)}, v_n\}, E_{Y_*}\{e_{(n*,P)}, (v_n+\Lambda)\}>$ and converts it to pair $<E_{Y_*}\{s_{(n*,P)}, v_n\}, E_{Y_*}\{(e_{(n*,P)} + r_n), (v_n+\Lambda)C_n\}>$. Here through $s_{(n*,P)} = \sum_{i=1}^{P} s_{(n.i)}$, $E_{Y_*}\{s_{(n*,P)}, v_n\} = \{g^{s_{(n*,P)}}, v_n {Y_*}^{s_{(n*,P)}}\}$ is calculated as $\{g^{s_{(n,1)}} g^{s_{(n,2)}} \ldots g^{s_{(n,P)}} = g^{s_{(n,1)}+\cdots+s_{(n,P)}}, (v_{(n,1)}{Y_{(1)}}^{s_{(n,1)}})(v_{(n,2)}{Y_{(2)}}^{s_{(n,2)}})\ldots(v_{(n,P)}{Y_{(P)}}^{s_{(n,P)}}) = (v_n {Y_*}^{s_{(n,1)}+\cdots+s_{(n,P)}})\}$. Then, through $e_{(n*,P)} = \sum_{i=1}^{P} e_{(n.i)}$, $E_{Y_*}\{e_{(n*,P)}, (v_n+\Lambda)\}$ is also calculated in the same way. About $E_{Y_*}\{(e_{(n*,P)} + r_n), (v_n+\Lambda)C_n\}$, V_n calculates it as $E_{Y_*}\{e_{(n*,P)}, (v_n+\Lambda)\}. E_{Y_*}\{r_n, C_n\} = \{g^{e_{(n*,P)}+r_n}, (v_n+\Lambda)C_n {Y_*}^{e_{(n*,P)}+r_n}\}$ (as ElGamal encryption functions are homomorphic).

5. V_n calculates 2nd used seal $\underline{U}^{Z_n}$ and puts <$E_{Y_*}\{s_{(n*,P)}, v_n\}$, $E_{Y_*}\{(e_{(n*,P)} + r_n)$, $(v_n+\Lambda)C_n\}$> on VotingPanel with $\underline{U}^{Z_n}$.

V_n decomposes v_n and $(v_n+\Lambda)$ as $v_n = \prod_{i=1}^{P} v_{(n.i)} = v_{(n,1)} \cdot v_{(n,2)} \cdots v_{(n,P)}$ and $(v_n+\Lambda) = \prod_{i=1}^{P} \underline{v}_{(n.i)} = \underline{v}_{(n,1)} \cdot \underline{v}_{(n,2)} \ldots \underline{v}_{(n,P)}$ respectively

⇩

V_n sends each $v_{(n,i)}$ and $\underline{v}_{(n,i)}$ to M_i

⇩

Each M_i calculates pair <$E_{Y_{(i)}}\{s_{(n,i)}, v_{(n,i)}\}$, $E_{Y_{(i)}}\{e_{(n,i)}, \underline{v}_{(n,i)}\}$> from $v_{(n,i)}$ and $\underline{v}_{(n,i)}$ by secret integers $s_{(n,i)}$ and $e_{(n,i)}$, and sends the pair to V_n

⇩

V_n calculates the products of pairs received from $M_1,\ldots,M_P$, *i.e.* <$E_{Y_*}\{s_{(n*,P)}, v_n\}$, $E_{Y_*}\{e_{(n*,P)}, (v_n+\Lambda)\}$>

⇩

V_n multiplies <$E_{Y_*}\{s_{(n*,P)}, v_n\}$, $E_{Y_*}\{e_{(n*,P)}, (v_n+\Lambda)\}$> *by* $E_{Y_*}\{r_n, C_n\}$, *i.e.* it calculates <$E_{Y_*}\{s_{(n*,P)}, v_n\}$, $E_{Y_*}\{(e_{(n*,P)}+r_n), (v_n+\Lambda)C_n\}$>

Fig. 4.2. Vote Construction Sub-stage.

Then, because each M_i does not know secret integers of other mix-servers, no one except V_n can know v_n unless all mix-servers conspire. Provided that erasable state voting booths that disable voters to memorize all information that they had generated and received are available, coercers cannot ask voters to disclose their votes either, *i.e.* voters themselves do not know all encryption parameters. But without erasable state voting booths, coercers that conspire with some mix-server M_i, which is not known to V_n, can know v_n. In detail, when V_n is asked to tell all $v_{(n,1)},\ldots,v_{(n,P)}$ it must tell them honestly, *i.e.* if V_n tells $v_{(n,i)}$ dishonestly conspiring M_i notices that. On the other hand when erasable state voting booths are available, V_n can forget some of <$E_{Y_{(1)}}\{s_{(n,1)}, v_{(n,1)}\}$, $E_{Y_{(1)}}\{e_{(n,1)}, \underline{v}_{(n,1)}\}$>,…, <$E_{Y_{(P)}}\{s_{(n,P)}, v_{(n,P)}\}$, $E_{Y_{(P)}}\{e_{(n,P)}, \underline{v}_{(n,P)}\}$>.

Security problems in this sub-stage are:

- Mix-servers encrypt vote v_n incorrectly*:* $M_1,\ldots,M_P$ cannot encrypt v_n incorrectly, because V_n verifies encrypted results. Although V_n itself can construct its encrypted vote incorrectly or it can accept incorrect encryption results intentionally without being detected, V_n must compensate corresponding losses by itself (*e.g.* V_n's vote may be determined as invalid but V_n cannot claim that election authorities are dishonest). Namely, encrypted votes on *VotingPanel* are approved by voters themselves.

- Mix-servers may not put V_n*'s* vote on VotingPanel: Because V_n did not reveal its used seal yet, it can claim that $M_1,\ldots,M_P$ are dishonest.
- B may add votes: Anyone can detect illegitimate additions, *i.e.* numbers of items on ConfNoList and VotingPanel become inconsistent.
- B may modify or delete votes on VotingPanel: Because *VotingPanel* is publicly disclosed, no one can modify votes once they are put on VotingPanel.
- V_n may submit votes repeatedly: No one can submit multiple votes because voters leave used seals that are unique to them and can be calculated only for legitimate credentials.

d. Tallying

To conceal correspondences between encrypted votes on VotingPanel and finally decrypted votes, in this stage $M_1,\ldots,M_P$ sequentially re-encrypt and shuffle votes of VotingPanel, disclose results on ShufflingPanel and decrypt votes of ShufflingPanel finally to be disclosed on TallyingPanel. This sage proceeds as follows:

1. $M_1,\ldots,M_P$ sequentially re-encrypt and shuffle each pair $<E_{Y_*}\{s_{(n*,P)}, v_n\}, E_{Y_*}\{(e_{(n*,P)} + r_n), (v_n+\Lambda)C_n\}>$ of VotingPanel. In detail, provided that $k_{(n,i)}$ and $d_{(n,i)}$ are secret integers of M_i and $k_{(n*,i)}= s_{(n*,P)}+\sum_{i\in P} k_{(n,i)}$, $d_{(n*,i)} = e_{(n*,P)}+r_n+\sum_{i\in P} d_{(n,i)}$, from $<E_{Y_{*(i-1)}}\{k_{(n*,(i-1))}, v_n\}, E_{Y_{*(i-1)}}\{d_{(n*,(i-1))}, (v_n+\Lambda)C_n\}> = <\{g^{k_{(n*,(i-1))}}, v_n Y_{*(i-1)}{}^{k_{(n*,(i-1))}}\}, \{g^{d_{(n*,(i-1))}}, (v_n+\Lambda)C_n Y_{*(i-1)}{}^{d_{(n*,(i-1))}}\}>$ forwarded by $M_{i\text{-}1}$, each M_i calculates $<\{g^{k_{(n*,(i-1))}} g^{k_{(n,i)}}, v_n Y_{*(i-1)}{}^{k_{(n*,(i-1))}} Y_{(i)}{}^{k_{(n,i)}}\}, \{g^{d_{(n*,(i-1))}} g^{d_{(n,i)}}, (v_n+\Lambda)C_n Y_{*(i-1)}{}^{d_{(n*,(i-1))}} Y_{(i)}{}^{d_{(n,i)}}\} = <E_{Y_{*(i)}}\{k_{(n*,i)}, v_n\}, E_{Y*(i)}\{d_{(n*,i)}, (v_n+\Lambda)C_n\}>$. Thus, finally M_P calculates $<E_{Y_*}\{k_{(n*,P)}, v_n\}, E_{Y_*}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}>$ for each n, and discloses it on ShufflingPanel.
2. Mix-servers sequentially decrypt each pair $<E_{Y_*}\{k_{(n*,P)}, v_n\}, E_{Y_*}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}>$ of ShufflingPanel. Namely, each M_i decrypts $<E_{Y_{*(i)}}\{k_{(n*,P)}, v_n\}, E_{Y_{*(i)}}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}>$ received from M_{i+1} to $<E_{Y_{*(i-1)}}\{k_{(n*,P)}, v_n\}, E_{Y_{*(i-1)}}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}>$ by its secret key $X_{(i)}$ to forward it to M_{i-1}.
3. 1st mix-server M_1 discloses each decryption result $<E_{Y_{*(0)}}\{k_{(n*,P)}, v_n\}, E_{Y_{*(0)}}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}> = <\{g^{k_{(n*,P)}}, v_n\}, \{g^{d_{(n*,P)}}, (v_n+\Lambda)C_n\}>$ on TallyingPanel.

In Step 2, notation $Y_{*(i)}$ represents $Y_{(1)}Y_{(2)}\ldots Y_{(i)} = g^{X_{(1)}+\cdots+X_{(i)}}$, and by decryption key $X_{(i)}$, M_i decrypts $E_{Y_{*(i)}}\{k, v\} = \{g^k, vY_{*(i)}{}^k\} = \{g^k, vg^{k.(X_{(1)}+\cdots+X_{(i)})}\}$ to $\{g^k, vg^{k.(X_{(1)}+\cdots+X_{(i)})} / g^{k.X_{(i)}}\} = \{g^k, vY_{*(i-1)}{}^k\} = E_{Y_{*(i-1)}}\{k, v\}$. As a consequence, M_1 finally decrypts $E_{Y_{*(P)}}\{k, v\}$ to $\{g^k, v\}$. Here, although mix-servers may encrypt or decrypt votes dishonestly, incorrect results are detected and entities liable for them are identified in the disruption detection stage.

e. Disruption Detection

This stage detects incorrect operations of mix-servers in the tallying stage and identifies entities liable for the dishonesties. In the following notation $A(CN)$ represents a set of all used CNs on ConfNoList. Each $\{\Omega_i, \Gamma_i\}$ is a pair of products $\Omega_i = \prod_{n=1}^{N} g^{k_{n*(i)}} = g^{k_{1*(i)}}g^{k_{2*(i)}}\ldots g^{k_{N*(i)}} = g^{k_{1*(i)}+\cdots+k_{N*(i)}}$ and $\Gamma_i = \prod_{n=1}^{N} v_n Y_{*(i)}{}^{k_{n*(i)}} = (v_1Y_{*(i)}{}^{k_{1*(i)}})\ (v_2Y_{*(i)}{}^{k_{2*(i)}})\ldots(v_NY_{*(i)}{}^{k_{N*(i)}}) = (v_1\ldots v_N)Y_{*(i)}{}^{k_{1*(i)}+\cdots+k_{N*(i)}}$, *i.e.* they are products of the 1st items in all forms <$E_{Y_{*(i)}}\{k_{1*(i)}, v_1\}$, $E_{Y_{*(i)}}\{d_{1*(i)}, (v_1 + \Lambda)C_1\}$>,…,<$E_{Y_*(i)}\{k_{N*(i)}, v_N\}$, $E_{Y_*(i)}\{d_{N*(i)}, (v_N + \Lambda)C_N\}$> re-encrypted by M_i at Step 1 in the tallying stage. The disruption detection stage proceeds as follows:

1. Each mix-server M_i discloses the sum of its secret integers as $K_{(i)} = \sum_{n=1}^{N} k_{(n)(i)} = k_{1(i)}+\ldots+k_{N(i)}$. Mix-servers also sequentially decrypt used encrypted CNs on *ConfNoList* to construct set $A(CN)$.
2. When $A(CN)$ includes non-registered numbers or same numbers, booth manager B determines mix-servers are dishonest and identifies liable entities.
3. For each i, B confirms that relations $\Omega_i/\Omega_{i-1} = g^{K_{(i)}}$ and $\Gamma_i/\Gamma_{i-1} = Y_{(i)}{}^{K_{(i)}}$ hold or not. If the relations do not hold, B asks mix-servers to carry out the tallying stage again until $\Omega_i/\Omega_{i-1} = g^{K_{(i)}}$ and $\Gamma_i/\Gamma_{i-1} = Y_{(i)}{}^{K_{(i)}}$ hold. Here, if M_i is honest Ω_i/Ω_{i-1} and Γ_i/Γ_{i-1} necessarily satisfy the relations as $\Omega_i/\Omega_{i-1} = g^{k_{1*(i)}+\cdots+}\ g^{k_{N*(i)}}/g^{k_{1*(i-1)}+\cdots+}\ g^{k_{N*(i-1)}} = g^{K_{(i)}}$ and $\Gamma_i/\Gamma_{i-1} = (v_1\ldots v_N)Y_{*(i)}{}^{k_{1*(i)}+\cdots+k_{N*(i)}}/(v_1\ldots v_N)\ Y_{*(i-1)}{}^{k_{1*(i-1)}+\cdots+k_{N*(i-1)}} = Y_{(i)}{}^{K_{(i)}}$. Also, M_i must behave honestly, when it is asked to encrypt votes on VotingPanal again, because B already had identified M_i as a liable mix-server.
4. B calculates $K = K_{(1)}+\ldots+ K_{(P)}$ and $\Phi = v_1\ldots v_N$ from $K_{(1)},\ldots,K_{(P)}$ reported by mix-servers and decryption result on TallyingPanel. From each encrypted form <$E_{Y_*}\{s_{(n*,P)}, v_n\}$, $E_{Y_*}\{(e_{(n*,P)} + r_n), (v_n+\Lambda)C_n\}$> on VotingPanel and <$E_{Y_*}\{k_{(n*,P)}, v_n\}$, $E_{Y_*}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}$> on ShufflingPanal, B also calculates $(v_1Y_*{}^{S(1,P)})(v_2Y_*{}^{S(2,P)})\ldots(v_NY_*{}^{S(N,P)})$ /

$\Phi = Y_*{}^{S(1*,P)+\cdots+(N*,P)} = \Psi$, and $(v_1 Y_*{}^{k(1,P)})\ldots(v_N Y_*{}^{k(N,P)}) / \Phi = Y_*{}^{k(1*,P)+\cdots+k(N*,P)} = \Sigma$. Then, B determines mix-servers dishonestly decrypted votes if relation $\Sigma = \Psi Y_*{}^K$ does not hold.

5. B can identify liable mix-servers incorrectly decrypted votes by calculating $\Psi_i = \prod_{n=1}^{N} v_n Y_{*(i)}{}^{k_{n*(P)}} = (v_1 Y_{*(i)}{}^{k_{(1*,P)}})(v_2 Y_{*(i)}{}^{k_{(2*,P)}})\ldots(v_N Y_{*(i)}{}^{k_{(N*,P)}}) = (v_1\ldots v_N)Y_{*(i)}{}^{k_{(1*,P)}+\cdots+k_{(N*,P)}}$, and identifies M_i as a dishonest mix-server if relation $\Psi_i/\Psi_{i-1} = Y_{(i)}{}^K$ does not hold. Then, B asks mix-servers to decrypt votes on ShufflingPanel again until relation $\Psi_i/\Psi_{i-1} = Y_{(i)}{}^K$ holds for every i. Here, apparently $\Psi_i/\Psi_{i-1} = Y_{(i)}{}^K$ must hold if M_i is honest, also M_i must behave honestly when it is asked to decrypt votes again, as same as in Step 3.
6. For each decrypted result $<E_{Y_*(0)}\{k_{(n*,P)}, \alpha_n\}, E_{Y_*(0)}\{d_{(n*,P)}, \beta_n\}>$ on TallyingPanel, B calculates the value $\gamma_n = \beta_n/(\alpha_n+\Lambda)$, and when $A(CN)$ does not include γ_n or γ_n appears multiple times, it determines mix-servers are dishonest and identifies liable entities.

In the above, mix-servers that do not know all secret values of other mix-servers cannot dishonestly encrypt or decrypt votes without violating relation $\Omega_i/\Omega_{i-1} = g^{K(i)}$, $\Gamma_i/\Gamma_{i-1} = Y_{(i)}{}^{K(i)}$, $\Sigma = \Psi Y_*{}^K$ or $\gamma_n = \beta_n/(\alpha_n+\Lambda)$ ($i \in \{1,\ldots,P\}$ and $n \in \{1,\ldots,N\}$) as discussed in section 3.2. Also, mix-servers must behave honestly when they are asked to encrypt or decrypt votes again. Therefore, booth manager B can detect dishonesties of mix-servers as the violation of relation $\gamma_n = \beta_n/(\alpha_n+\Lambda)$. In addition, M_i can maintain integers $k_{1(i)},\ldots,k_{N(i)}$ as its secret even after it discloses $K_{(i)}$, *i.e.* secrets of honest entities are not revealed.

Here, if Λ is removed from each vote form $<E_{Y_{*(i)}}\{k_{(n*,i)}, v_n\}, E_{Y_{*(i)}}\{d_{(n*,i)}, (v_n+\Lambda)C_n\}>$ as $<E_{Y_{*(i)}}\{k_{(n*,i)}, v_n\}, E_{Y_{*(i)}}\{d_{(n*,i)}, v_nC_n\}>$, M_i can modify pair $<E_{Y_{*(i)}}\{k_{(n*,i)}, v_n\}, E_{Y_{*(i)}}\{d_{(n*,i)}, v_nC_n\}>$ and $<E_{Y_{*(i)}}\{k_{(h*,i)}, v_h\}, E_{Y_{*(i)}}\{d_{(h*,i)}, v_hC_h\}>$ to $<E_{Y_{*(i)}}\{k_{(n*,i)}, v_n/T\}, E_{Y_{*(i)}}\{d_{(n*,i)}, v_nC_n/T\}>$ and $<E_{Y_{*(i)}}\{k_{(h*,i)}, v_hT\}, E_{Y_{*(i)}}\{d_{(h*,i)}, v_hC_hT\}>$ by using arbitrary integer T without violating the relations. Integer Λ protects vote forms from these dishonesties.

After inconsistent encryption or decryption results are detected, B identifies mix-servers liable for dishonesties also without revealing secrets of honest entities by tracing each inconsistent decryption result $<E_{Y_*(0)}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(0)}}\{d_{(n*,P)}, \beta_n\}>$ back to the corresponding initial encryption form on VotingPanel as below. Firstly B asks M_1 in the decryption stage to show $<E_{Y_*(1)}\{k_{(n*,P)}, \alpha_n\}, E_{Y_*(1)}\{d_{(n*,P)}, \beta_n\}>$ from which it had calculated $<E_{Y_{*(0)}}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(0)}}\{d_{(n*,P)}, \beta_n\}>$ and M_1 proves its correct decryption without revealing its secret. In the same way, each M_i shows $<E_{Y_{*(i)}}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(i)}}\{d_{(n*,P)}, \beta_n\}>$ from which it had calculated $<E_{Y_{*(i-1)}}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(i-1)}}\{d_{(n*,P)}, \beta_n\}>$ and proves its correct decryption without revealing its secret. Then, B determines M_i is dishonest when it cannot show consistent pair $<E_{Y_{*(i)}}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(i)}}\{d_{(n*,P)}, \beta_n\}>$ and $<E_{Y_{*(i-1)}}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(i-1)}}\{d_{(n*,P)}, \beta_n\}>$. B identifies mix-servers that dishonestly encrypted votes in the same way.

Here, M_i can convince B that pair $<E_{Y_{*(i)}}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(i)}}\{d_{(n*,P)}, \beta_n\}>$ and $<E_{Y_{*(i-1)}}\{k_{(n*,P)}, \alpha_n\}, E_{Y_{*(i-1)}}\{d_{(n*,P)}, \beta_n\}>$ is consistent without revealing its secret key $X_{(i)}$ through the scheme of Diffie and Hellman [36]. Firstly, B generates secret integer π, and calculates $g^{k_{(n*,P)}\cdot\pi} = \Theta$ and $\{\beta_n {Y_{*(i)}}^{k_{(n*,P)}} / \beta_n {Y_{*(i-1)}}^{k_{(n*,P)}}\}^{\pi} = \{{Y_{(i)}}^{k_{(n*,P)}}\}^{\pi}$. After that it asks M_i to calculate $\Theta^{X_{(i)}}$ by showing Θ, and determines the pair is consistent when $\Theta^{X_{(i)}}$ coincides with $\{{Y_{(i)}}^{k_{(n*,P)}}\}^{\pi}$.

When dishonest mix-servers are identified, B asks them to encrypt and decrypt incorrectly handled votes again to generate correct election results. Here, a coercer that is coercing voter V_n may know V_n's vote v_n if 1st mix-server M_1 that is conspiring with the coercer encrypted v_n dishonestly. Namely, the above identification procedure finally reaches $<E_{Y_*}\{s_{(n*,P)}, v_n\}, E_{Y_*}\{(e_{(n*,P)} + r_n), (v_n+\Lambda)C_n\}>$ on VotingPanel, it is re-encrypted and decrypted again to $<E_{Y_{*(0)}}\{k_{(n*,P)}, v_n\}, E_{Y_{*(0)}}\{d_{(n*,P)}, (v_n+\Lambda)C_n\}>$, and the coercer can know V_n's vote on VotingPanel by asking V_n to calculate used seal $\underline{U}^{Z_n}$. But actually, mix-servers do not behave dishonestly because dishonest mix-servers are necessarily identified. Namely, in a real sense, the disruption detection stage is not for detecting dishonesties, instead it is for convincing entities about honest conduction of elections. Thus, an adversary can succeed to execute dishonesty with a negligible probability.

Finally, it must be noted that although booth manager B in the above detect dishonesties of mix-servers, apparently any entity including voters can detect them by itself.

CHAPTER V

Evaluation of the Scheme

This chapter evaluates the scheme by comparing the computation volume (time, efficiency) and security requirements among the proposed scheme, CN based [15] and R-SVRM [20] based scheme.

5.1 Computation Volume

To measure computation times required for the registration, voting and tallying stages, a simulation system that includes 3 mix-servers and 1000 voters was developed using 2.40 GHz core $i7$ CPU with 8 GBytes of RAM and GMP 1024 bit modulus running on Windows 8 while considering all plaintext as 40 digits. Because the simulation system consists of a single computer, communication delays among voters, booth manager B and mix-servers were not measured. Computation time required for the CN generation stage was not measured either, because it can be carried out in advance as an off-line process.

TABLE 5.1

Computation Time Required by the Proposed Scheme

<table>
<tr><th colspan="2">Stage</th><th>Processing time (ms/vote)</th></tr>
<tr><td colspan="2">Registration</td><td>21</td></tr>
<tr><td rowspan="2">Voting</td><td>Voter acceptance</td><td>24.5</td></tr>
<tr><td>Vote construction</td><td>17.4</td></tr>
<tr><td colspan="2">Tallying</td><td>69.1</td></tr>
<tr><td colspan="2">Total</td><td>132</td></tr>
</table>

Table 5.1 shows the measuring results. For the registration stage, 21ms is required *i.e.* to issue a signed credential the booth manager B requires 10.4ms, and to generate a receipt the voter V_n requires 10.6ms, respectively. The voting stage consists of voter acceptance and vote construction sub-stages, where for the former sub-stage 24.5ms is required *i.e.* to authenticate the V_n through its anonymous credential requires 17.0ms and to calculate the 1st used seal U^{Z_n} requires 7.5ms. The 2nd sub-stage requires 17.4ms *i.e.* for encrypting v_n and (v_n+Λ), and multiplying encrypted forms of (v_n+Λ) and C_n requires 9.9ms and to calculate 2nd used seal $\underline{U}^{Z_n}$ requires 7.5ms. Here, the time is measured while considering the fact that components $\{v_{(n,1)}, \underline{v}_{(n,1)}\},\ldots,\{v_{(n,3)}, \underline{v}_{(n,3)}\}$ of $\{v_n, (v_n+\Lambda)\}$ are encrypted by $M_1,\ldots,M_3$ in parallel. About the tallying stage, it requires 69.1ms, *i.e.* 44ms for re-encryption and shuffling, and 25.1ms for decryption, executed by $M_3,\ldots,M_1$ sequentially.

TABLE 5.2
Computation Time Comparisons with CN Based and R-SVRM Based Schemes

Schemes	CPU (GHz)	Memory	Processing time (ms/vote)		
			Registration	Voting	Tallying
Proposed scheme	2.4	8 GBytes	21	41.9	69.1
CN based[15]	1.6	504 MBytes	47.1	164	133
R-SVRM based[20]	2.4	8 GBytes	21	133	191

Table 5.2 is the comparison of computation volumes of the proposed scheme, the CN based scheme [15] and the R-SVRM based scheme [20] where all schemes adopt 1024 bit modulus, involve same mix-servers and voters; although used CPUs are distinct.

The registration stage in the proposed scheme and the R-SVRM based one are different from the CN based scheme that adopts blind signature based authentication, comprises of blinding, signing and unblinding of a token by $M_1,\ldots,M_3$ in 2 different forms, and requires 0.3ms, 45ms and 1.8ms, respectively. Thereby the computation time for the registration stage in the proposed scheme and the R-SVRM based one that adopt anonymous credential based authentication is reduced from 47.1ms to 21ms. About the voting stage, the proposed scheme that does not require signatures on votes is different from the CN based scheme also, CN based one requires 164ms to encrypt a vote v_n comprises of the encryption by voter V_n, encryption by M_1,…,M_3, decryption by V_n, and generation of 2 different signatures by M_1,…,M_3 that require 3.0ms, 17.0ms, 9.0ms and 135.0ms, respectively. As the R-SVRM based scheme generates the encrypted form of v_n as ${r_n}^{v_n \cdot (v_n + \Lambda)}$ through 2-round re-encryption, its computation time is 133ms. Where in this stage, V_n calculates a used seal to enter the voting booth, V_n encrypts v_n in an initial form and M_1,…,M_3 perform first round re-encryption, at last V_n calculates another used seal to approve v_n and they require 7.5ms, 118ms and 7.5ms, respectively. Thus, it is also larger than the proposed scheme. In the same way, because signatures of mix-servers are not required or vote forms are simpler, computation time of the tallying stage in the proposed scheme is less than those in the CN based and R-SVRM based ones that require 133ms and 191ms, respectively. Here in the CN based scheme, this stage consists of decryption and shuffle by M_3,…,M_1 sequentially. But in the R-SVRM based scheme, this stage consists of second round re-encryption and shuffling, pre-tallying, and final tallying of votes through decryption that require 89ms, 51.3ms and 50.7ms, respectively. Here for the schemes, the verification time of voting and tallying stages are not considered in Table 3 to maintain uniformity.

When compared with ZKP based schemes, the computation volume of the proposed scheme is significantly less than that of them as discussed in [15].

Table 5.3 shows a comparison for cryptographic schemes and numbers of operations used and required (*i.e.* the efficiency aspects) among the proposed scheme, the CN based and R-SVRM based ones. Each mix-server M_i in the proposed scheme encrypts or decrypts 2 items for each vote in the voting and the tallying stages. On the other hand in the CN based scheme, M_i encrypts 3 items, re-encrypt 2 items, and decrypts 5 items for each vote because vote forms include 2 different signatures (actually due to 2 parts of data, the required signature is $5 \times P$). About the R-SVRM based scheme, because each vote form includes 3 items, M_i encrypts 6 items for each vote in the voting stage (3 items are encrypted through 2-rounds), and re-encrypts and re-decrypts 3 items in the tallying stage. Here, vote forms in the R-SVRM actually consist of 6 items, but 3 of them are used to protect voter V_n from coercers in cases where coercers force V_n to choose a candidate unique to it.

TABLE 5.3
Comparison for Cryptographic Schemes and Efficiency Aspects Among the Proposed Scheme, CN based and R-SVRM based Ones

Stage			Proposed scheme	R-SVRM based[20]	CN based[15]
Methods to authenticate voters in the registration stage			anonymous credential	anonymous credential	blind signature based token
Methods to conceal vote v_n in the voting stage			vote decomposition, parallel encryption	vote decomposition, 2-rounds re-encryption	random factor, re-encryption & re-signing
Methods to verify correct encryptions and decryptions in the tallying stage			R-SVRM and CNs	R-SVRM	CNs
Number of operations	Voting	vote encryption	$2 \times P$	$6 \times P$	$3 \times P$
		verification	$2 \times P$	$6 \times P$	$3 \times P$
		Signing	No	No	$5 \times P$
	Tallying	Re-encryption	$2 \times P$	$3 \times P$	$2 \times P$
		Decryption	2 × P	$3 \times P$	$5 \times P$

*requires other cryptographic operations and extra data also.

Therefore, to make comparisons fair these 3 items are removed in Table 5.3.

5.2 Achieved Security Requirements

The proposed scheme satisfies essential security requirements [2, 37] as follows. Besides, based on major security requirements, a comparison with allied e-voting schemes is presented in Table 5.4.

Privacy: While voting, Booth manager B authenticates voter V_n anonymously by anonymous credential T_n, and V_n approves its vote v_n by used seal $\underline{U}^{Z_n}$; thereby no one except V_n knows the link between the V_n and its v_n. No one can identify whether V_n abstains from the election or not

either.

TABLE 5.4
Comparison Among Schemes based on Security Requirements

Schemes	Privacy	Accuracy	Integrity	Fairness	Robustness	Receipt-free	Incoercible (*i.e.* free from) Randomization	Simulation	Forced abstention	Scalable	Practical
Proposed	Y	Y	Y	Y	Y	Y	Y	Y	C	Y	Y
Scheme [2]	Y	Y	Y	Y	Y	Y	Y	Y	N	NH	LP
Scheme [6]	Y	Y	Y	Y	Y	Y	Y	Y	C	NH	LP
Y: Yes; N: No; NH: Not Highly; C: Conditionally; LP: Less Practical;											

Accuracy: While registration, voter V_n obtains anonymous credential T_n from Booth manager B by showing its' identifier ID_n which disables entities to impersonate the V_n. Thus only the legitimate V_n can cast its' single vote formally. In addition, uniqueness of registered CNs along with used seals, and publicly open BBs ensure that all and only votes approved by their corresponding voters are finally posted on TallyingPanel.

Integrity: Voter V_n verifies the correctness of encryption of all mix-servers. Now V_n approves its vote on VotingPanel using used seal $\underline{U}^{Z_n}$ which ensures that the vote is casted as intended. Publicly open BBs disables any entity to modify data posted on it which ensures that the vote is recorded as casted. Finally, registered CN attached with each vote ensures that only recorded votes are counted.

Incoercibility: Mix-servers conceal correspondences between voter V_n and its vote v_n from anyone including V_n. Anyone does not know encryption parameters that are used to construct v_n. Also prior to encryption, V_n decomposes v_n into P products to be encrypted by mix-servers. Therefore V_n does not need to tell v_n correctly to coercers. But it must be noted that a coercer can know whether V_n chooses its designating $v_n{}^*$ or not if $v_n{}^*$ is unique to V_n. This difficulty can be removed by introducing the pre-tallying stage as in the R-SVRM based scheme. Also as same as in other schemes, voters cannot be protected from forced abstention. Namely, when a coercer asks voter V_n to calculate used seal $\underline{U}^{Z_n}$ again by its credential T_n, V_n must calculate it honestly. As a consequence the coercer can know whether V_n abstained or not. Forced abstention can be disabled when a regulation that forces all voters to register for example.

Fairness: Encrypted form of vote v_n is jointly calculated by voter V_n and mix-servers

$M_1,\ldots,M_P$, thereby no one can know the interim voting results until tallying results are disclosed.

Robustness: Because dishonest entities are identified in the disruption detection stage, even when incorrectly handled votes are detected, correct tallying results can be re-calculated without re-election. Here, privacy of honest voters still can be maintained and secrets of honest entities are not revealed as discussed in chapter IV in section 4.2 (e).

CHAPTER VI

Conclusions

This chapter draws the summary of the thesis and also discusses some possible future works based on the outcome of the present work.

6.1 Summary of the Work

By introducing CNs the proposed e-voting scheme improves the performance of the R-SVRM based scheme. Because it reduces the number of items in each vote form and excludes items that include information about candidates as exponents from vote forms, the scheme becomes simple and efficient. Also it satisfies all essential requirements of e-voting systems, *i.e.* it is endowed with features about privacy, robustness, accuracy, integrity, incoercibility and fairness. As a consequence, the scheme becomes practical and scalable.

6.2 Future Perspectives

Some potential future directions of works are available from the present study.

In this study, only booth voting is considered. In future it might be improved so that it can support remote voting.

Another future plan of improvement is to incorporate in more realistic environments where multiple authorities are distributed over different places, and many voters are involved.

This proposed mechanism may evaluate with features of additive and multiplicative homomorphic properties of Paillier cryptosystem.

REFERENCES

[1] D. Chaum, “Untraceable electronic mail, return address, and digital pseudonyms,” Communications of the ACM, Vol. 24, No. 2 , pp. 84 -88,1981.

[2] K. Sampigethaya and R. Poovendran, “A framework and taxonomy for comparison of electronic voting schemes,” Computers and Security, Elsevier Vol. 25, pp. 137-153, 2006.

[3] J. Wen-Shenq, L. Chin-Laung and L. Horng-Twu, “A verifiable multi-authority secret election allowing abstention from voting,” The Computer Journal, Vol. 45(6), pp. 672–82, 2002.

[4] K. M. R. Alam, and S. Tamura, “Electronic Voting Using Confirmation Numbers,” Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, pp. 4672–77, 2009.

[5] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang and S. Yoo, “Providing receipt-freeness in Mixnet-based voting protocols,” in Proceedings of the information Security and Cryptology (ICISC ’03), pp. 245–258, 2004.

[6] D. Chaum, “Elections with unconditionally- secret ballots and disruption equivalent to breaking RSA”, Advances in Cryptology – Eurocrypt’88, LNCS 330, Springer-Verlag, pp. 177–182, 1988.

[7] J. Benaloh and D. Tuinstra, “Receipt-free secret-ballot elections,” Proceedings of 26th Symposium on Theory of Computing, pp. 544–553, 1994.

[8] M. Hirt and K. Sako, “Efficient Receipt-Free Voting Based on Homomorphic Encryption,” Proceedings of EUROCRYPT, LNCS, Vol. 1807, pp. 539-556. Springer, 2000.

[9] B. Lee, and K. Kim, “Receipt-free electronic voting scheme with a tamperresistant randomizer”, ICISC 2002, LNCS 2587, Springer-Verlag, pp. 389–406, 2002.

[10] A. Neff, “A verifiable secret shuffle and its application to E-voting”, ACM CCS 2001, ACM Press, pp. 116–125, 2001

[11] J. Schweisgut, “Coercion-resistant electronic elections with observer,” 2nd International Workshop on Electronic Voting, Bregenz, August 2006.

[12] A. Juels and M. Jakobsson, “Coercion-resistant electronic elections,” Cryptology ePrint Archive, Report 2002/165, <http://eprint.iacr.org/>; 2002.

[13] D. Chaum, “Secret-ballot receipts: true voter-verifiable elections,” IEEE Security & Privacy Magazine, Feb 2004.

[14] B. Riva and A. Ta-Shma, “Bare-Handed Electronic Voting with Pre-processing,” Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop, Boston, MA, 2007.

[15] K. M. R. Alam, S. Tamura, S. Taniguchi, and T. Yanase. “An anonymous voting scheme based on confirmation numbers.” IEEJ Transactions EIS, Vol. 130, No. 11, pp. 2065-2073, 2010.

[16] A. Acquisti, “Receipt-free homomorphic elections and write-in voter verified ballots,” Cryptology ePrint Archive, Report 2004/105, <http://eprint.iacr.org/>; 2004.

[17] K. Aggelos and Y. Moti. “Self-tallying elections and perfect ballot secrecy,” Proceedings of public key cryptography, 5th international workshop on practice and theory in public key cryptosystems, LNCS, vol. 2274. Springer-Verlag, pp.141–58, 2002.

[18] R. Abdelkader and M. Youssef, “Uvote: A ubiquitous e-voting system,” in 3rd FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing (MUSIC'12), pp. 72-77, 2012.

[19] N. Islam, K. M. R. Alam, and S. S. Rahman, “Commutative re-encryption techniques: Significance and analysis," Information Security Journal: A Global Perspective, vol. 24, no. 4, pp. 185-193, 2015.

[20] S. Tamura, H. A. Haddad, N. Islam, and K. M. R. Alam, “An Incoercible E-Voting Scheme based on Revised Simplified Verifiable Re-encryption Mix-nets,” Information Security and Computer Fraud, Science and Education Publishing, Vol. 3, No. 2, pp. 32-38, 2015.

[21] C. C. Lee, T. Y. Chen, S. C. Lin, and M. S. Hwang,”A new proxy electronic voting scheme based on proxy signatures," Lecture Notes in Electrical Engineering, vol. 164, pp. 3-12, 2012.

[22] B. Adida, “Helios: Web-based open-audit voting," in Proceedings of 17th USENIX Security Symposium, Aug. 2008.

[23] L. Huian, A. R. Kankanala, and X. Zou, “A taxonomy and comparison of remote voting schemes," in 23rd International Conference on Computer Communication and Networks (ICCCN'14), pp. 1-8,2014. W. Diffie and M. Hellman, “New directions in cryptography,” IEEE Transactions Information Theory, Vol. IT-22, pp. 472-492, 1976.

[24] K. M. R. Alam and S. Tamura, “Electronic voting: Scopes and limitations," in Proceedings of International Conference on Informatics, Electronics & Vision (ICIEV12), pp. 525-529, May 2012.

[25] S. Tamura, and S. Taniguchi. "Simplified Verifiable Re-encryption Mix-nets." Information Security and Computer Fraud 1, vol. 1 (2013): pp. 1-7.

[26] A. Fujioka, T. Okamoto, and K. Ohta, “A practical secret voting scheme for large scale elections," in Advances in Cryptology (AUSCRYPT'92), pp. 244-251, 1993.

[27] W. S. Juang, C. l. Lei, and P. lingYu, “A veri_able multi-authorities secret election allowing abstaining from voting," Computer Journal,vol. 45, no. 6, pp. 672-682, 2002.

[28] O. Cetinkaya and M. L. Loc, “Practical aspects of dynavote e-voting protocol," Electronic Journal of E-government, vol. 7, no. 4, pp. 327-338, 2009.

[29] M. R. Clarkson, S. Chong, and A. C. Myers, “Civitas: Toward a secure voting system," in Proceedings of the 2008 IEEE Symposium on Security and Privacy, pp. 354-368, 2008.

[30] A. Juels, D. Catalano, and M. Jacobsson, “Coercion-resistant electronic elections,” Towards trustworthy Elections, pp. 37-63, 2010.

[31] R. Araujo, A. Barki, S. Brunet, and J. Traore, “Remote electronic voting can be efficient, verifiable and coercion-resistant," in International Conference on Financial Cryptography and Data Security, pp. 224-232, 2016.

[32] A. Essex, J. Clark, and U. Hengartner, “Cobra: Toward concurrent ballot authorization for internet voting," in International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections (EVT/WOTE'12), 2012.

[33] B. Riva and A. Ta-Shma, “Bare-handed electronic voting with pre-processing,” Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop, pp. 15, 2007.

[34] D. Sandler, K. Derr, and D. S. Wallach, “Votebox: atamper-evident veri_able electronic voting system,” in Proceedings of the 17th USENIX Security symposium, pp. 349-364, 2008.

[35] S. Tamura and S. Taniguchi, “Enhancement of Anonymous Tag based Credentials,” Information Security and Computer Fraud, Science and Education Publishing, Vol. 2, No. 1, pp. 10-20, 2014.

[36] W. Diffie and M. Hellman, “New directions in cryptography,” IEEE Transactions Information Theory, Vol. IT-22, pp. 472-492, 1976.

[37] P. Salini and S. Kanmani, “Security requirements engineering for specifying security requirements of an e-voting system as a legitimate solution to e-governance,” International Journal of Wireless and Mobile Computing, Vol. 7, No. 4, pp. 400-413, 2014.