INTERNET VOTING BASED ON HOMOMORPHIC ENCRYPTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAKAN YILDIRIM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

APRIL 2014

Approval of the thesis:

INTERNET VOTING BASED ON HOMOMORPHIC ENCRYPTION

submitted by **HAKAN YILDIRIM** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** ______________

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** ______________

Dr. Muhiddin Uğuz
Supervisor, **Department of Mathematics,METU** ______________

Dr. Mehmet Sabır Kiraz
Co-supervisor, **TÜBİTAK, GEBZE** ______________

Examining Committee Members:

Prof.Dr. Ersan AKYILDIZ
Department of Mathematics, METU ______________

Dr. Muhiddin UĞUZ
Department of Mathematics, METU ______________

Dr. Mehmet Sabır KİRAZ
TÜBİTAK, GEBZE ______________

Assoc.Prof.Dr. Ali DOĞANAKSOY
Department of Mathematics, METU ______________

Dr. Murat CENK
IAM, METU ______________

Date: ______________

ABSTRACT

INTERNET VOTING BASED ON HOMOMORPHIC ENCRYPTION

Yıldırım, Hakan
M.S., Department of Cryptography
Supervisor : Dr. Muhiddin Uğuz
Co-Supervisor : Dr. Mehmet Sabır Kiraz

April 2014, 44 pages

Some of the countries use classical methods for elections. In such voting systems, people are forced to be in a predefined place and have to cast their votes in a limited time interval. Parallel to economic growth, priority of time on people's life increased and such kind of limitations decreases the number of people who cast their vote. In order to make people's lives easier, we need to add usable automated systems to elections such as Internet Voting (i-voting) systems. However, constructing an i-voting system needs comprehensive knowledge of various sciences like information technology and cryptography. Electronic voting is in general very sensitive in the sense that if any problem occurs in the designed system, then the society will have a question mark in their mind and this will diminish the percentage of people who use voting systems. In this thesis, many cryptographic protocols and i-voting systems are studied. Norway and Estonia are the countries who use i-voting solutions successfully for some years. We focused on their solutions and studied on a modified version of the Norwegian system.

Keywords : Voting Systems, Homomorphic Encryption, Privacy, Threshold Cryptography

ÖZ

HOMOMORFİK ŞİFRELEMEYE DAYANAN İNTERNET SEÇİMİ

Yıldırım, Hakan
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Dr. Muhiddin UĞUZ
Ortak Tez Yöneticisi : Dr. Mehmet Sabır Kiraz

Nisan 2014, 44 sayfa

Bazı ülkeler seçimler için klasik yöntemleri kullanırlar. Bu tip seçim sistemlerinde, insanlar belirli bir yerde bulunmaya zorlanırlar ve oylarını limitli zaman aralığında kullanmalıdırlar. Ekonomik gelişmeye paralel olarak, insanların zamanlarına verdikleri öncelik artmıştır ve bu tip kısıtlamalar oyunu kullanan insan sayısını azaltır. İnsanların hayatlarını kolaylaştırmak için seçimlere internet seçim sistemleri (i-seçim) gibi kullanılabilir otomatize sistemleri eklememiz gerekir. Ancak, internet seçim sistemlerinin kurulması bilgi teknolojileri ve kriptoloji alanları gibi çok farklı bilimlerde geniş bilgi gerektirir. Elektronik seçim sistemleri genel olarak o kadar hassastır ki dizayn edilen sistemde herhangi bir sorun olması durumunda, toplumun kafasında bir soru işareti doğar ve bu seçim sistemlerinin kullanım oranını düşürür. Bu tezde, birçok kriptoloji protokolü ve internet seçim sistemleri incelenmiştir. Norveç ve Estonya internet seçim sistemlerini birkaç yıldır başarıyla kullanan ülkelerdir. Biz onların çözümlerini ve Norveç sisteminin değiştirilmiş versiyonunu inceledik.

Anahtar Kelimeler: Seçim Sistemleri, Homomorfik Şifreleme, Gizlilik, Eşik Kriptografisi

To My Little Daughter

ACKNOWLEDGMENTS

Till the cryptographic protocols course in the department, I have not decided on any subject to work in my thesis. In this course, my supervisor Dr. Mehmet Sabır Kiraz taught subjects that are new in cryptography like homomorphic encryption. This thesis is the result of ongoing work started at his course till now and includes many cryptographic primitives studied in his course. Thanks to him about all his help and guidance.

I know my supervisor Dr. Muhiddin Uğuz from the Calculus courses when I was an undergraduate student. He is the man who make me understand and love mathematics. When I started graduate education, I have taken two courses from him which contain the building blocks of cryptography. I enjoyed his courses and thanks to him about his guidance, comments and help. I hope to study with him for a long time in the future.

Special thanks to both Ahmet Sınak and Seçil Özkan from the Cryptography Program about their support and tolerance while we are working together with internet voting subjects.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 History

Elections are done in a predefined time and place. Voters and their voting locations are announced before the elections and among them who would like to cast their vote are expected to be in that place on the election day. People may be on a vacation or they may be in a business trip. So, they should stop their occupation and go to the vote casting place. Being in a specific place for a voting procedure is a problem for some people. They get stuck with loosing time or money and they will try to decide on to cast their vote. It is obvious that this kind of limitation will decrease the percentage of ballot submitters in the citizens.

Privacy of polling booth is emphasized in [42] for having a proper election and it is stated that casting votes should be done in a controlled environment. Note that in real life, expatriates should cast their vote in customs or embassy. In some countries, embassies may not be physically secure or an embassy may be several hours away from the residence of the expatriate by plane which is not convenient. Therefore, it may be hard for a country to satisfy physical security of the voting place and casting a vote does not take a little time or effort for most of the expatriates in the country.

Moreover, there are handicapped or old people. In most of the classical election systems, there is not an easiness for them. It is expected from those people to be in an election center in a given period of time, which is not logical. In that case, people could not cast their vote or they may require the help of people around them to go to that place. Vote casting is their free will and they cannot do that process without the help other people, which is a contradiction. In some countries, in order to diminish the complexity of vote casting procedure, postal voting is provided for voters. Postal voting has been used in Germany since 1956 [21]. In 1956, they used that method first in the third Federal Electoral Act. In 1957, 4.9% of the voters and in the elections held in 2009 21.4% of the voters cast their votes from their home using postal service. In [37], online questionnaire is made to understand the voters' attitude towards the verifiability in vote casting using postal service, for the results of this survey please refer to [37]. In addition to that, to make election procedure more flexible, voters are allowed to cast their votes in any preferred election center in Germany [21]. Notice that, there should be a mechanism to authenticate a voter and to check whether if voter cast her vote elsewhere before. Besides those actions, i-voting solutions are chance for everybody to make their lives easier.

In classical elections, vote counting errors can be performed since it is done manually. If designed correctly, automated systems decrease the rate of this error. Measured error rates for manual tallying range from 0.5% to 1.5%. Vote tallying have an error rate of 0% in a well designed automated system [8].

In most of the countries, which will be given in detail later, many electronic voting solutions are used in addition to the classical method. In [33], electronic voting solutions are divided roughly into two types. One is distant electronic solutions in which voters have chance to cast their votes from their personal computers and the other one is voting machines. Internet voting systems which can be abbreviated i-voting systems can be stated as an example of distant electronic solutions. In designing i-voting systems security and privacy are main concerns. I-voting systems cannot be considered as just a computer system but it includes many cryptographic systems behind. Automated systems for elections are indeed facilitate voting procedure but there are crucial issues to be solved that should be considered carefully like privacy, transparency, vote buying, coercion and receipt-freeness.

1.2 Problem Fields

In his work [30], Ka-Ping Yee made a comparison between banking and voting machines. He states that roughly both machines seem similar, on both devices selections can be made to run transactions. On the other hand, he differentiates those machines with respect to their auditing mechanisms and the consequences of a mistake seen in both devices. In banking devices, customers may get a detailed confirmation from the system that their transactions are done correctly, all the steps done in running transactions are audited, even user's actions on the machine may be recorded by a camera as a log. Not depending on the time elapsed, if there is a mistake in the actions done by the bank and suppose that customer gets less money than she has to take, she can get her money back from the bank. If she wants, she can leave the bank and can start working with the other one. But when we come to a voting machine, things have changed. Consequence of a mistake may cause an election to be run again. Moreover, in voting systems privacy is the main concern which makes those confirmation processes harder than the banking systems. This example is useful to understand the importance of voting systems in our lives and the difficulty in designing a voting system for system designers.

Privacy is one of the major doubts for the voters. If there is a chance for the system to match the vote and the voter at the end, you can get the list of people's freewill. In such a case, people will be separated from each other and some of them will be alienated from the community. Moreover, people may be afraid of casting their votes or they may change their candidate to whom community supports at the elections.

Correctness can be stated as a problem field in designing a voting system. System must collect all the votes without any alteration or error. After collecting ballots, system should store and count them correctly without any modification. Encryption/decryption and all computations done in the system should be verified at each step and logs should be collected carefully.

Transparency is another feature that system designers take into account when designing an i-voting system. All the steps and modules in the system should be known by the community or by the experts in order to increase trustiness. Therefore, community will not have gap in their minds about the system and huge amount of people should examine the system. If they find a problem in the system, then system designers will have a chance to fix it without any serious event occurs.

Vote buying and coercion are problems both for the conventional system and automated systems. In some regions of a country, there may be some people who have political or economical power on the community. Those people may have chance to influence the community, furthermore they may buy their freewill. Voters may be forced to cast their vote to a candidate by physical force or by personal relationships. This coercion can be done individually or in some places, group of people's freewill may be canalized to a candidate. If a system designer should find a solution for these types of problems, then it will become the prominent feature of his system.

Verifiability is a property that secure voting systems should possess. In [42], verifiability has divided into three parts which are individual verifiability, public verifiability, end-to-end verifiability. Individual verifiability is defined as the ability of individual voters to confirm their selection has been encoded in their receipt correctly. Public verifiability means everyone has chance to verify that the receipts posted on Bulletin Board have correctly decrypted and tallied. End-to-end verifiability means all the steps of the election system should be controlled and verified. In some systems, like Norwegian or customized one represented in this thesis, receipt is given with a piece of paper having numbers each representing a candidate for each voter. Voters have chance to check the system using this receipt code papers. In [2], as a receipt a new technique based on human verifiability named as visual sharing of shape descriptions is presented.

Receipt freeness is another security requirement in which voter cannot prove that she has cast her vote to a candidate. If a voting system has receipt freeness property, then vote buying and coercion cases can be hindered.

Elections should not possess difficulty for voters. Governments should find automated systems to decrease the potential difficulty in voting procedure. It is clear that automated voting systems help voters to cast their ballots easily. In the observations done by Electoral Council of Australia and New Zealand, it is stated that classical systems will sooner or later replaced by automated systems in the future [10]. If designed and implemented properly, it will increase the people who cast their vote in the community. Therefore, election results will indicate the freewill of more people in the country at the end. However, system owners and designers should spend time to consider problem fields mentioned here to make their system used by the community. If any notable error occurs in the automated system, it will decrease the trust of citizens in the elections, moreover it may estrange people from the community.

1.3 E-Voting Experiences

In reference [19], you can find a map which shows the countries' usage of e-voting systems. On the left bottom of the map it gives the colours which shows the usage type. Notice that e-voting solutions are commonly used by most of the countries all over the world. Besides that in [14] you can find another map which shows the usage of i-voting solutions in USA states.

CHAPTER 2

PRELIMINARIES

In this chapter, we present fundamental topics required to understand general voting systems.

2.1 Diffie-Hellman Key Distribution Scheme

2.1.1 Discrete Logarithm Problem, Diffie-Hellman Problem

Discrete Logarithm Problem was first declared by Diffie and Hellman in 1976 [52]. Let G be a multiplicative group, for $g \in G$, $G =< g >$ is a group generated by g. At this point, discrete logarithm problem for the group G can be defined as [31]:

Definition 2.1. Let $G =< g >$ be a group generated by g. Given $g \in G$ and $a \in< g >$, discrete logarithm problem is finding an $x \in Z$ such that $g^x = a$. Motivation of the problem comes from fast computation of g^x when g and x are given. But reverse operation, which is finding x when g^x is hard.

Definition 2.2. Diffie-Hellman Problem can be defined as [31], given g, g^x, g^y such that $x, y \in Z$ it is infeasible to compute g^{xy}.

Definition 2.3. Decisional Diffie-Hellman (DDH) assumption for group $G =< g >$ can be defined as [7], $x, y, z \in Z$ it is hard to distinguish g^{xy} from a random group element g^z when g, g^x, g^y are given.

2.1.2 Diffie-Hellman Key Exchange Protocol

Diffie-Hellman Key Exchange Protocol is commonly used to negotiate and interchange a key between two parties on an insecure channel. Plain Diffie-Hellman protocol which is not secure against some attacks, especially against man in the middle attacks [24]. Thus, in today's applications ephemeral Diffie-Hellman or Diffie-Hellman with certificate protocols are used. Let's try to narrate plain Diffie-Hellman protocol by giving a daily life illustration. Suppose that Alice and Bob are two parties who want to communicate on an authenticated channel. Using Diffie-Hellman Key Exchange Protocol

[39], they first agree on a big prime number p and a base element $g \in {Z_p}^*$ in a public channel. Therefore, any eavesdropper (Catherine) can listen communication between Alice and Bob. So, Catherine also knows this prime number p and the base element $g \in {Z_p}^*$.

Alice next chooses a random number K_{Alice} and she computes A which is the least positive residue modulo p of $g^{K_{Alice}}$. At the same time, Bob also chooses a random number $K_{Bob}<p$ and he also computes B which is the least positive residue modulo p of $g^{K_{Bob}}$. Alice keeps K_{Alice} secret and she sends element $A \in {Z_p}^*$ to Bob and Bob keeps K_{Bob} secret and sends element $B \in {Z_p}^*$ to Alice. Here, Catherine learns the elements A and B.

After this agreement and communication, Alice and Bob computes $g^{K_{Alice}K_{Bob}} \in {Z_p}^*$ by using A, B and their secret keys.

$(g^{K_{Alice}})^{K_{Bob}} = (g^{K_{Bob}})^{K_{Alice}} = g^{K_{Alice}K_{Bob}}$.

This element will be used as a secret key between Alice and Bob. Keep in mind that for Catherine, knowing $g^{K_{Alice}}$, $g^{K_{Bob}}$ it is infeasible to calculate $g^{K_{Alice}K_{Bob}}$ (Diffie-Hellman Problem).

2.2 ElGamal Encryption Scheme and Homomorphic Encryption

In 1984 Taher ElGamal proposed his algorithm [48]. ElGamal encryption algorithm is the first encryption scheme that is based on discrete logarithm problem [12]. In his algorithm a random number is used and with the help of this random number, two encryptions of the same plaintext is different which makes this algorithm probabilistic. This is an advantage for a voting system designer, because there should be multiple votes for the same party in the system. If your encryption algorithm is probabilistic, the same vote's encrypted data will be different, so that nobody can understand voter's will, till it is being decrypted. To obtain semantic security, RSA-OAEP [43] or even block cipher encryption algorithm AES-CBC (with padding) [35] can be used. However, homomorphic property of ElGamal makes this algorithm inevitable among others. Homomorphic property of an algorithm is important for voting systems because votes can be tallied without decrypting them. It is possible to decrypt only the tallied sum. Note that here ciphertext will be double the size of plaintext [48], which is a disadvantage.

2.2.1 ElGamal Encryption Algorithm

ElGamal encryption algorithm can be expressed in three phases: Key Generation, Encryption and Decryption.

Key Generation: $G =< g >$ of order p is used where g is a primitive element in the group, i.e. $g^{p-1} \equiv 1 \mod p$.

Receiver chooses a random integer x where $x \in \{1, 2, \ldots, p-1\}$. He calculates $h = g^x$ such that $h \in G$. Finally, secret key is x which will be kept by receiver and public key is h which will be sent to sender to encapsulate data.

Encryption:

In this phase, sender chooses a random integer $r \in \{1, 2, \ldots, p-1\}$. He calculates his public key share h^r and encrypts the plaintext m as mh^r where $mh^r \in G$. Ciphertext C which is $Enc(m, r) = (g^r, mh^r)$ calculated by sender where m is the plaintext.

Decryption:

At the decryption phase, ciphertext $C = Enc(m, r) = (g^r, mh^r)$ and private key x will be used by the receiver. Receiver calculates $(g^r)^x$, he gets the inverse of it in the group and he multiplicates mh^r with the inverse of $(g^r)^x$ to find m. Mathematical operations for decryption can be summarized as follows:

$m = \frac{mh^r}{(g^r)^x} = \frac{mh^r}{h^r}$

2.2.2 Homomorphic Encryption

It is possible to store sensitive data into a database or into a storage hardware in an encrypted form. When you want to make a query on the saved encrypted data or if you want to make a calculation using this data, first you have to decrypt them and you should get them to a volatile place. You should decrypt it and finally, you can make calculations or queries on the decrypted data. This leads to a security leakage since one has to decrypt and see the data to process it.

Notice that, homomorphic encryption's precious importance cannot be ignored. Homomorphic encryption allows us to operate on ciphertexts instead of decrypting and making operations on plaintexts. It is a good opportunity for the people who performs computations on sensitive data.

Definition 2.4. Homomorphic encryption can be defined as [32]:

Let M denote the plaintext set and C denote the ciphertext set. In order to express an encryption system homomorphic, this encryption system should satisfy

$\forall m_1, m_2 \in M, E(m_1 \Theta_M m_2) = E(m_1) \Theta_C E(m_2)$.

" $=$ " means that it can be computed from $E(m_1)$ and $E(m_2)$ without intermediate decryption. By homomorphic encryption, given two ciphertexts, one can calculate the encrypted form of multiplication or addition of plaintexts. If M or C is an additive group than this encryption system will be additively homomorphic system. Goldwasser-Micali and Paillier cryptosystems have additively homomorphic property. Therefore, algebraic operation on ciphertexts leads to an addition operation on the plaintexts such that:

$\forall m_1, m_2 \in M, E(m_1 + m_2) = E(m_1)\Theta_C E(m_2)$

If M or C is multiplicative group than encryption system will be multiplicatively homomorphic system. In that case, algebraic operation on ciphertexts leads to a multiplication operation on the plaintexts such that:

$\forall m_1, m_2 \in M, E(m_1 \cdot m_2) = E(m_1)\Theta_C E(m_2)$

Unpadded RSA and ElGamal cryptosystems have multiplicatively homomorphic property.

2.2.3 Homomorphic Properties of ElGamal Encryption Algorithm

Original ElGamal encryption algorithm allows us to have multiplicatively homomorphic encryption property as follows:

$\forall m_1, m_2 \in M, E(m_1, r_1) = (g^{r_1}, m_1 h^{r_1}), E(m_2, r_2) = (g^{r_2}, m_2 h^{r_2})$

If you multiply $E(m_1, r_1)$ and $E(m_2, r_2)$, you will obtain the ciphertext of multiplied plaintexts.

$E(m_1, r_1).E(m_2, r_2) = (g^{r_1+r_2}, m_1 m_2 h^{r_1+r_2}) = (g^r, m_1 m_2 h^r) = E(m_1.m_2, r)$ where $r = r_1 + r_2$.

Moreover, with some modifications ElGamal encryption algorithm can have additively homomorphism property [44]. In this form, ciphertext is calculated as follows:

$E(m, r) = (g^r, g^m h^r)$

In this form, for $E(m_1, r_1) = (g^{r_1}, g^{m_1} h^{r_1})$ and $E(m_2, r_2) = (g^{r_2}, g^{m_2} h^{r_2})$,

$$\begin{aligned} E(m_1, r_1) \cdot E(m_2, r_2) &= (g^{r_1} g^{r_2}, g^{m_1} g^{m_2} h^{r_1} h^{r_2}) \\ &= (g^{r_1+r_2}, g^{m_1+m_2} h^{r_1+r_2}) \\ &= (g^r, g^{m_1+m_2} h^r) = E(m_1 + m_2, r_1 + r_2) \end{aligned}$$

where $r = r_1 + r_2$. It is obvious that if you multiply two ciphertexts, you will obtain the addition of plaintexts. Notice that the main difference between the original algorithm and modified one is the discrete logarithm problem in the modified one. At the decryption process, from g^m it will be hard to obtain the m. To recover discrete logarithm problem it will be useful to use a look up table.

2.3 PKI and Digital Signatures

Public Key Infrastructures (PKI) is a platform which makes us to construct public key cryptography based services. PKI enables us to share data on an insecure public network such as internet, securely, using public and private key pair obtained from a trusted party. PKI is the technology that is behind very well known secure protocols used for secure data exchange such as Secure Sockets Layer (SSL) and Hypertext Transfer Protocol Secure Sockets (HTTPS) protocols [25].

To assure that a public key belongs to a claimed party, Certification Authorities (CA) are used in PKI. Any certificate can be verified by everybody who owns CA's public key. Role of CA is to sign and distribute the public key of an entity.

Digital signatures are created by public key cryptography and signatures can be verified by using CA's in the PKI [3]. Digital signatures are used for authorization, authentication and non-repudiation. When a user signs a document, he binds his signature with the document and generates a file which is the signature itself. This procedure can be expressed in detail like below [1]:

Let M represent the set of messages to be signed. S be the set of signatures generally binary strings which have a fixed length. S_A represents the transformation from the message to the signature set S. V_A represents the transformation from the set MXS to the set $\{true, false\}$. V_A is a publicly known verification function to verify signatures created by A.

Signing Procedure:

At this step A creates a signature for the message $m \in M$ by using S_A transformation(signing transformation) as follows $s = S_A(m)$. A sends the pair (m, s) to the verifier by keeping S_A transformation secret.

Verification Procedure:

To verify a message coming from entity A, B uses the verification function V_A as follows $u = V_A(m, s)$. At this step if $u = true$ B accepts the signature, else she rejects the signature. In Figure 2.1 sample illustration of digital signature scheme is given.

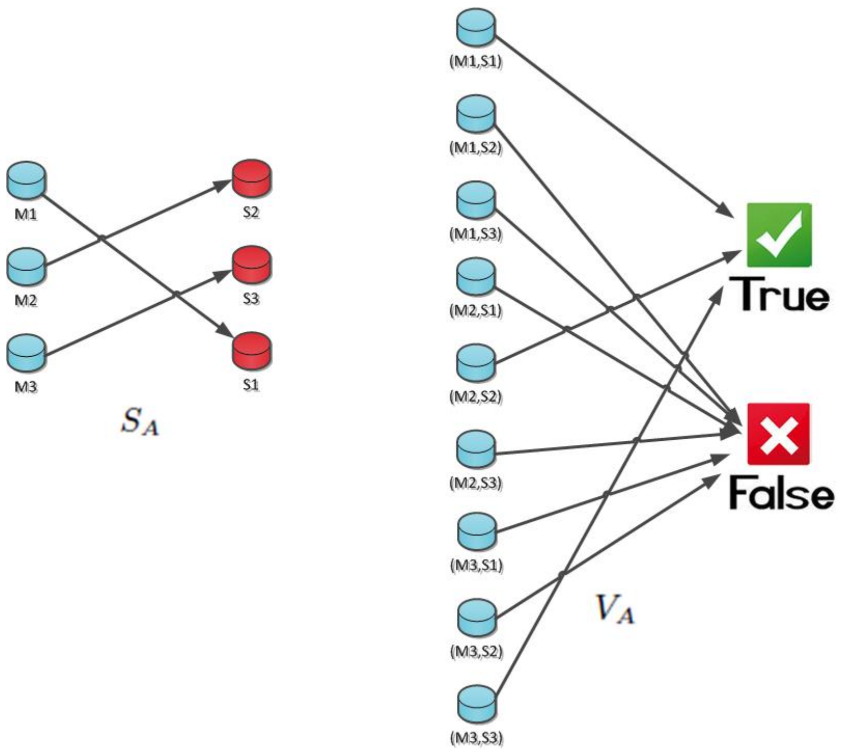


Figure 2.1: Digital Signature Scheme [1]

2.3.1 Digital Signature Standard (DSS)

In 1991 United States government decided to use DSS as a signature standard and they started to use Digital Signature Algorithm (DSA) to sign electronic documents which is similar to ElGamal digital signature algorithm [41]. Federal Information Processing Standards Publications (FIPS PUBS)are done by the National Institute of Standards and Technology (NIST). DSS is a standard which states algorithms appropriate for applications requiring digital signature [50]. In DSS document, three algorithms which are DSA, RSA signature algorithm, Elliptic Curve Digital Signature Algorithm (ECDSA) are stated as suitable for DSS and it is stated that the security of a digital signature system is valued as secure whenever security of the users' private key kept as secret. In this thesis, ElGamal Digital Signature algorithm is studied, for more information about DSS digital signature algorithms we refer to [50].

2.3.2 ElGamal Digital Signatures

In ElGamal encryption algorithm which is based on discrete logarithm problem, digital signatures can be created as follows [48]:

In cyclic multiplicative group $< g >= G$ of order p, $x \in Z_p$ is the secret key of the signer, h can be generated as $h = g^x$ which is the public key of the signer known by everybody. Let m be a document to be signed which is between 0 and $p-1$. The signature for the document m is the pair (r, s), $0 \leq r, s < p-1$, chosen such that the equation $g^m = h^r r^s \mod p$ is satisfied.

Signing Procedure:

Choose a number k uniformly between 0 and $p-1$, such that $gcd(k, p-1) = 1$.

Compute $r = g^k \mod p$. Now, $g^m = h^r r^s = g^{xr} g^{ks} \mod p$ can be obtained. Notice that $m = xr + ks \mod (p-1)$. The signature for the document m is the pair (r, s).

Verification Procedure:

In the verification procedure, given m, r and s it is possible to compute both sides of $g^m = h^r r^s \mod p$. If both sides of the equation are the same than verifier accepts else he rejects this statement. Notice that without knowing the secret of the signer (which is x), nobody can forge a valid signature.

2.4 Zero Knowledge Proofs

Zero knowledge proofs let prover to convince verifier about the correctness of a statement without revealing any information beyond the truth of the statement. In order to prove a statement using zero knowledge proofs, it is necessary that verifier puts some interactive input which can be named as a challenge. When verifier inputs some challenge, prover will respond him with a statement which ensures that the statement is true.

There is an example that can explain the fundamentals of zero knowledge proofs [17]. As a circular cave which is depicted below, which has one entrance and in front of the entrance there is a barrier inside the cave. There are two people whose names are Peggy and Victor. Victor knows that there is a barrier inside the cave and Peggy states that she knows the way to pass this barrier without explaining or showing the way to Victor. She has to persuade Victor about this statement without giving him information beyond the truth of his statement. In order to do that, Peggy goes inside the cave from the entrance as in the first figure below, she chooses a path A or B and she goes that way. After that Victor goes in the cave and he knows that Peggy has already entered the cave but Victor does not know the way that Peggy selected. Randomly, Victor chooses an exit path A or B like in the second figure below and wants Peggy to come out from that path. Peggy goes out from the selected path. Therefore, the probability of Victor's choice is the same as the path of Peggy's selection is $\frac{1}{2}$. When Victor and Peggy repeats this game n times, Peggy always goes out from the path which Victor selects. At the end, probability of being coincidence without knowing the way to pass the barrier is $\frac{1}{2^n}$, which is too low to occur as a chance. Finally, Victor becomes convinced that Peggy knows the way to pass that barrier.

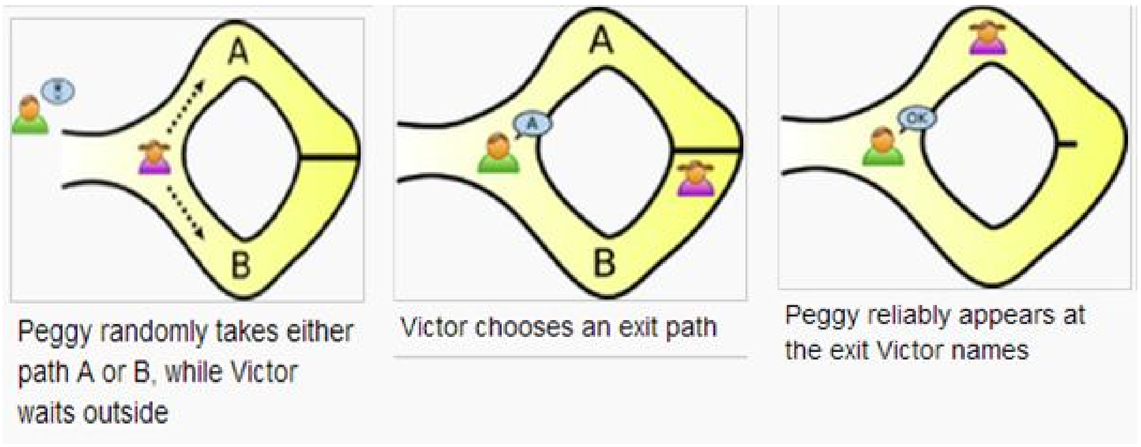

A
B
Peggy randomly takes either path A or B, while Victor waits outside
A
A
B
Victor chooses an exit path
OK
Peggy reliably appears at the exit Victor names

Figure 2.2: Zero Knowledge Proofs [17]

A zero knowledge protocol between prover and verifier should satisfy three properties [53]:

Completeness: If both prover and verifier are honest and if they follow (compute messages according to the protocol) the protocol, verifier always accepts.

Soundness: If prover is malicious or computationally unbounded, given a false statement to verifier, verifier always rejects the statement except some negligible probability.

Zero Knowledge: For any malicious PPT verifier, there is a PPT simulator such that when verifier interacts with prover, given two inputs from the real prover and the simulator, inputs are computationally indistinguishable for the verifier.

With the help of zero knowledge proofs, provers have chance to convince verifiers about the truth of a statement without giving any information to the verifiers. The well known zero knowledge proof type that is used in i-voting protocols is $\sum$-protocol that will be explained now.

2.4.1 $\sum$ Protocol

A sigma protocol lets a prover show to a verifier for some statement $v \in V$, he knows the witness $w \in W$ (w is the private input of the prover) such that $(v, w)\ \epsilon R$ and $R = \{(v; w)\} \subseteq VxW$ be a binary relation ($R \subseteq \{0,1\}^* \times \{0,1\}^*$). Language corresponding to the relation R can be defined as $L_R = \{v \in V : \exists_{w \in W}(v; w) \in R\}$ [7]. A Σ protocol is a zero-knowledge protocol if and only if there is a honest verifier.

Definition 2.5. Σ protocols are composed of three steps which are announcement, challenge and response. As an announcement, prover generates a random number $u \in_R {Z_n}^*$. He generates the announcement a and he sends this announcement to verifier. Verifier generates a challenge $c \in_R C$ and sends that challenge to prover. Prover calculates the response and sends it to verifier. Verifier checks the response, if it holds verifier accepts the conversation else he rejects it. This conversation flow is depicted below.

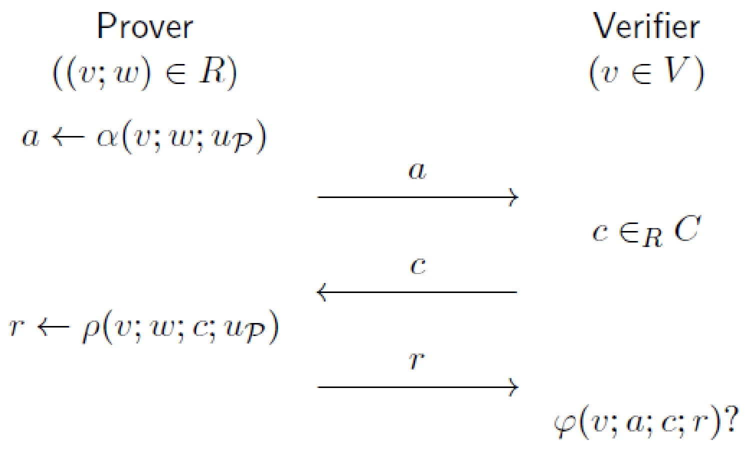

Prover
$((v; w) \in R)$
Verifier
$(v \in V)$
$a \leftarrow \alpha(v; w; u_{\mathcal{P}})$
a
$c \in_R C$
c
$r \leftarrow \rho(v; w; c; u_{\mathcal{P}})$
r
$\varphi(v; a; c; r)?$

Figure 2.3: Sigma Proofs [7]

A protocol between prover and verifier that can be defined as Σ protocol should satisfy the following properties:

Completeness: If prover and verifier follow the protocol truly then verifier always accepts.

Special Soundness: There exists a p.p.t. algorithm given any $v \in V$ and a pair of conversations $(a; c; r)$ and $(a; c'; r')$ with $c \neq c'$ computes a witness w satisfying $(v; w) \in R$. This is the witness extraction property. Using this property, it is known that with more than negligible probability, malicious prover can not convince verifier.

Special Honest-Verifier Zero Knowledge: There exist a simulator S which can produce the same conversations $(a; c; r)$ with the same probability distribution as conversations between honest prover and verifier on common input v and challenge c, where prover uses any witness w satisfying R. The main difference between zero knowledge and special honest-verifier zero knowledge property is in special case, simulator takes the challenge c as input [51]. The special honest-verifier property protects withness from only honest verifier who chooses random challenge.

A Sample Σ Protocol:

Schnorr identification protocol is a Σ protocol which is used to prove knowledge of a witness x satisfying $h = g^x$. Using this protocol, suppose that prover wants to prove that he knows the result of discrete logarithm problem $x = log_g h$. The sample protocol below indicates the steps of Σ protocol used for this process. In this process firstly, prover generates a random number $u \in_R {Z_n}^*$ and calculates $a = g^u \ mod\, n$ which is announcement. He sends it to the verifier. Verifier generates a random number c which is the challenge. Verifier sends it to prover and since prover knows x, he calculates $r = cx + u \ mod\, n$ and he sends it to verifier. At the end, since h, a, c is known by the verifier, he checks whether $g^r = h^c a \ mod\, n$. If it is right, he accepts it, else, he rejects.

Sample Σ Protocol

$$\begin{array}{lcr}
\textbf{Prover} & & \textbf{Verifier}\\
a = g^u & \xrightarrow{\qquad a \qquad} & \\
 & \xleftarrow{\qquad c \qquad} & c \in_R Z_n \\
r = cx + u & \xrightarrow{\qquad r \qquad} & g^r \overset{?}{=} h^c a
\end{array}$$

As it is been said before, to define a protocol as a Σ protocol, it has to satisfy completeness, special soundness and special honest-verifier zero knowledge properties. Let's try to show how this protocol satisfies given properties.

Completeness: If prover and verifier are honest, it is clear that given a true statement, verifier always accepts. In this example, completeness clearly holds,
$g^r = g^{cx+u} = (g^x)^c g^u = h^c a.$

Special Soundness: Given accepting conversations $(a; c; r)$ and $(a; c'; r')$,
$g^r = h^c a$ and $g^{r'} = h^{c'} a$
$\frac{g^r}{g^{r'}} = \frac{h^c a}{h^{c'} a}$
$g^{r-r'} = h^{c-c'}$
$g^{r-r'} = (g^x)^{c-c'}$
$h = g^{\frac{r-r'}{c-c'}}$
Therefore, witness can be obtained as $x = \frac{r-r'}{c-c'}$.

In other words, prover commits the same value using the same random number, verifier gives him different challenges, using this property, prover can extract secret x from $(a; c; r)$ and $(a; c'; r')$. However, in real world, prover does not use an old commit phase.

Special Honest-Verifier Zero Knowledge: To show special honest-verifier zero knowledge property real and simulated conversations are as follows:

$\{(a; c; r) : u \in_R Z_n; a \leftarrow g^u; r \leftarrow_n u + cx\}$,
$\{(a; c; r) : r \in_R Z_n; a \leftarrow g^r h^{-c}\}$.

These two distributions are identical, hence an attacker cannot learn any information about x during protocol run.

2.5 Commitment Schemes

Suppose that you want to play coin tossing game with one of your friend on the phone. How can you play that game without any alteration or cheating? One easy solution can be using a telephone book. Assume that both of you have the same telephone book and you planned to use the last telephone number at each page. If the result is head you will tell a page number which have an odd number at the last row, if the result is a tail, you will tell a page number which have an even number at the last row. After tossing the coin, you will tell a page number and ask your friend to tell him head or tail. If he is wrong than he has chance to check your respond by looking at the page from the telephone book. This kind of a mechanism can be defined as a simple commitment. Thus, commitment schemes enable us to commit something while keeping it secret.

The key point here is the time limit. Your friend should not find time to look for the page before he will tell head or tail. If he will have time to check the page, he will see your selection. Thus, there are some concepts that we have to cover while giving information about commitment schemes, these are binding/hiding properties of commitment schemes. Let's start with the definition of commitment schemes.

Definition 2.6. In commitment scheme there are two people, one is committer the other is the receiver [13]. The commitment scheme have two steps, one is commit phase, second is reveal or open phase.

1. **Commit Phase:** The sender sends the bit b to the receiver in an encrypted form.
2. **Reveal Phase:** Sender sends some additional data to the receiver in order to check the commitment.

There are three requirements that a protocol should satisfy in order to define it as a commitment scheme. These are [13];

Hiding Property: When receiver gets the commitment, he cannot learn anything about the committed value.

Binding Property: After committing the result, sender cannot change the committed value.

Viability: If the committer and receiver follows the protocol truly, receiver always recovers the committed value.

If the resources (e.g., computational power, time) of adversary is unlimited but he cannot demolish hiding/binding property than this scheme is *information theoretically* hiding/binding.

On the other hand, if the resources of adversary is limited and for that reason he cannot demolish hiding/binding property, moreover, if he would have more resources or more time he may destroy hiding/binding property, then this scheme is *computationally* hiding/binding.

2.5.1 Example: Pedersen Commitment Scheme

Trusted third party T chooses a multiplicative cyclic group G which have a large prime order p. Therefore the Diffie-Hellman problem is hard in G. T selects two generators h and g of the group G such that $h = g^x \mod p$. Discrete logarithm of h with respect to g is difficult. Here $x \in_R Z_p$ is a secret key. T publishes (G, p, g, h) which will be public [38].

Commit: When committer wants to commit a value $\alpha \in F_p$ he selects a random $\beta \in F_p$ and calculates $c = g^\alpha h^\beta \ mod\, p$.

Reveal: Committer shows the β and α to receiver, receiver checks whether $c \overset{?}{=} g^\alpha h^\beta \ mod\, p$.

Since the secret x is not known by the committer and for him to find this secret x is a discrete logarithm problem, this scheme is computationally binding. Receiver does not know the numbers α and β in advance, so he can find infinitely many such numbers. Even if the receiver will have infinitely powerful resources the result will not change. Therefore, this scheme is information theoretically hiding.

2.6 Threshold Cryptography

The main aim of threshold cryptography is to divide a secret S into n pieces $\{S_1, S_2, \ldots, S_n\}$ in such a way that [4];

1. If k or more secret shares come together, they can easily reconstruct S.
2. If less than k pieces come together then it should be impossible to obtain the secret S.

This kind of a scheme is called (k, n) threshold scheme.

2.6.1 Shamir's Threshold Scheme

We will now present one of the most famous (k, n) threshold schemes which is the Shamir's Threshold Scheme [49]. Let $P = \{P_1, P_2, \ldots, P_n\}$ be the set of participants, let K be the set of possible secrets, let S be the set of possible shares. In Shamir's scheme both K and S are equal to Z_p where $n + 1 \leq p$. Now, let's try to explain the steps to prepare the shares of a secret.

Initialization: Firstly, n distinct elements of Z_p which will be denoted as x_i are chosen by the one who wants to share the secret where $1 \leq i \leq n$. Here, each participant P_i obtains their coordinate x_i's which are not the shares. Without the function or their corresponding shares coordinates does not make any sense, therefore x_i's might be public.

Share Distribution: In order to share $K \in Z_p$, randomly chosen $k - 1$ elements of Z_p denoted as $a_1, a_2, \ldots, a_{k-1}$ and $a_0 = s$. A polynomial which have at most degree $k - 1$ is being constructed as follows:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1} = \sum_{j=0}^{k-1} a_j x^j modp$$

Shares are being constructed using x_i's, $y_i = f(x_i)$. Therefore, at the end y_i's which are the shares are constructed. Now, try to explain a method to reconstruct the polynomial to get the secret $a_0 = s$ back.

2.6.1.1 Lagrange Interpolation Formula

In order to reconstruct the polynomial $f(x)$, at least k points $y_1 = f(x_1), y_2 = f(x_2), \ldots, y_k = f(x_k)$ should be given. Lagrange interpolation formula is used here as follows [49]:

$$f(x) = \sum_{i=1}^{k} y_i \prod_{1 \le j \le k, j \ne i} \frac{x - x_j}{x_i - x_j}.$$

By using at least k shares, polynomial f can be reconstructed and secret s can be obtained by calculating $f(0) = s$.

2.7 Mix Network

Mix Networks (also known as mixnet), a phrase first introduced by David Chaum in 1981 [9]. Mix networks play role in many areas especially in anonymous web browsing, payment systems, election systems [26]. The main idea behind mix networks is to hide correspondence between input and output in a system. In mix networks, there are servers which take multiple messages from senders, permute them and send permuted messages to the next destination in random order [16]. In some architectures, re-encryption of messages at each mix network server is done besides permutation of messages like in Norwegian i-voting system which will be discussed later. In Figure 2.4, sample illustration of mix networks is given.

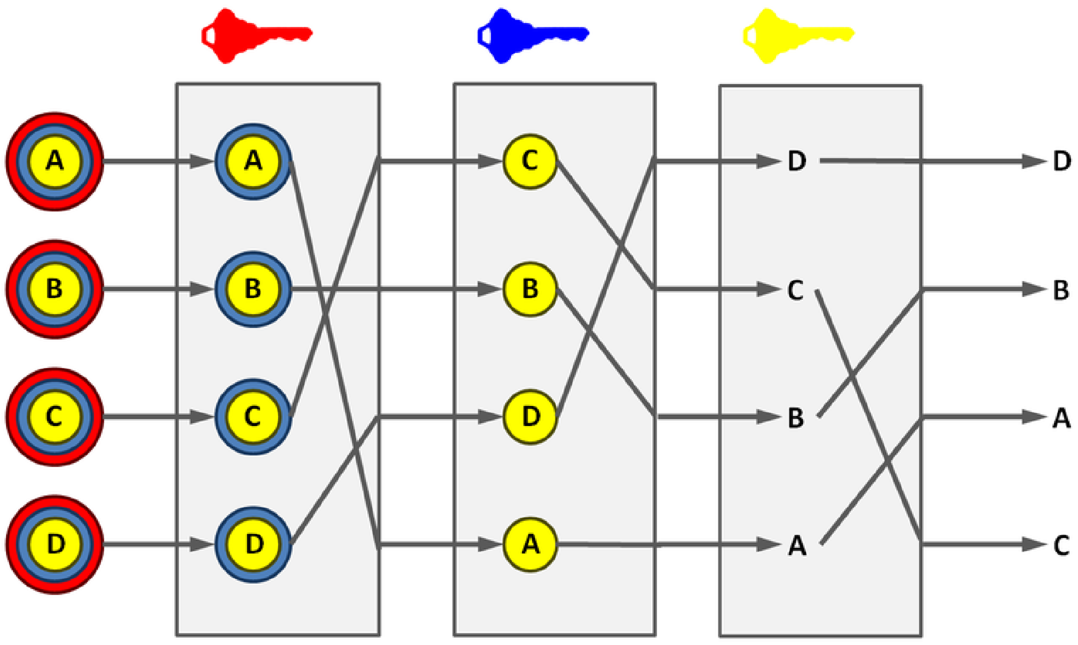


Figure 2.4: Mix Networks [16]

2.8 Thin Client-Terminal Server Technology

Thin client is a limited computer which depends terminal servers to accomplish its computational roles. They do not have a standard operating system and most of them does not have a hard drive. Therefore, most of them do not save data locally.

They are establishing remote desktop protocol (RDP) with the terminal server, it receives the screen of the user and it does not have any other functionality. All the computations will be done and all the programs will run in the terminal server side. Terminal server can serve almost 50 thin clients simultaneously depending on the hardware and complexity of the applications requested from users.

Some limitations on thin clients and management opportunities of it makes thin client usage advantageous and secure such as central management opportunity. Thin clients can be controlled centrally using central management software. All operating system installations and log/alert collection tasks can be done from the center. All the plugs of the thin client such as CD/DVD readers, USB ports can be restricted by central policies. If some specialized USB devices' usage is required then the serial numbers of those devices can be set to the system so that only allowed devices can be used. It is possible to show user only the web browser without address bar. Moreover, it is viable not to allow user to open any other application on the thin client device. He may not have ability to get the screen picture either. Those kind of restrictions stated above are crucial for critical systems in common areas.

All the connections between terminal server and thin client will be in secure channel which is using RDP over HTTPS protocol (e.g., they can be done using Terminal Services Gateway Protocol, for more information please refer to [20]). Therefore, anybody listening the channel will not obtain any information from communication. Simple illustration for the thin client-terminal server architecture usage in i-voting systems can be found in Figure 2.5.

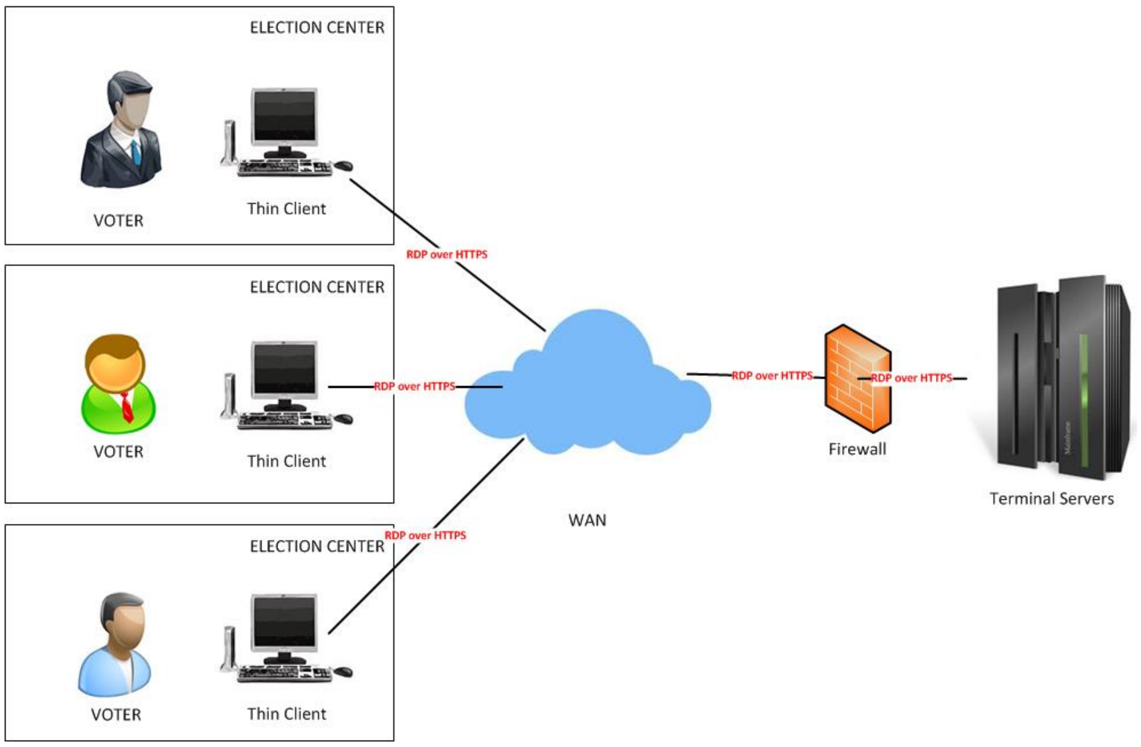


Figure 2.5: Thin Client-Terminal Server Usage in I-Voting Systems

Thin clients do not have a complex configuration. If any problem occurs in the device, since they do not store data and do not have complex configuration, they can be changed easily without loosing any time. At this point, remote installation of operating system and central log collection makes those operations more easy for system administrators.

In places where a system is used by many people and where you don't have any system operator next to computers and in which physical security of personal computers cannot be established, all of the advantages of thin client-terminal server technology like easy management, data storage only at terminal side, central management and remote software installation; it will be better to use thin clients instead of normal computers. It will bring security, efficiency and simplicity. Any voter, who does not have a PC or who has a compromised PC can easily use thin clients instead of them which will make their lives easier.

CHAPTER 3

INTERNET VOTING

Internet voting is an automated election system where people uses internet connection to submit their marked ballots [23]. Those systems are composed of some modules (for front-end and back-end) which are interconnected and in some cases there are air gaps between them for the back-end systems [5]. All computations done by the modules are audited and controlled by several independent authorities. Those modules can be seen as not only computer systems but also cryptographic modules. In this chapter, we examined Norwegian and Estonian voting systems and propose a new i-voting system [5]. Moreover, some modifications in [5] are expressed in this chapter.

3.1 Estonian System

I-voting solutions in Estonia is used in 2005 in Tallinn as a non-binding pilot and it is the first country which used country wide e-voting systems [36]. In October 2005, it is used officially in Local Government Council Elections. In this elections, there were 9.317 i-voters that makes almost 1.9% of participating voters [46]. In March 2011, number of i-voters become 140.846 and the percentage of i-voters become 24.3%. Actually, this statistical information indicates the concrete progress and trust of society on the Estonian i-voting system.

3.1.1 Enhancements of Estonian System

In Estonia ID-cards carrying a chip which includes two RSA keys for authentication, signing are used since 2002.

In this section, modules which are the building blocks of Estonian voting system are examined. The first module to be discussed here is Internet-Voting Client Application (IVCA). This module runs on the voter's PC and is responsible from executing all the commands taken from voter which are submitting, encrypting vote using the public key of the voting system with RSA algorithm and signing the vote using the private key of the voter using the key inside the ID-card [47]. Second module to be discussed here is Vote Forwarding Server (VFS). VFS's main role is to verify whether if the voter is an eligible voter or not. If the voter is an eligible voter then it returns the list of candidates to the IVCA. All the communication between the IVCA and VFS is done encrypted using HTTPS protocol. After vote casting is done, it gets the signature and encrypted vote from IVCA and it verifies the signature. If the signature is valid then VFS transmits received data to Vote Storing Server (VSS) [46]. VSS is responsible for the temporarily storing received ballots and anonymization of the valid votes. Here, VFS and VSS are connected to internet. But, VSS is behind the firewall that means all the communication between VSS and other things are controlled by a security filter. The last module that should be discussed here is Vote Counting Server (VCS). It is responsible for decrypting the votes and tallying them. This module is not online. Modules and the general picture of this system is depicted in Figure 3.1.

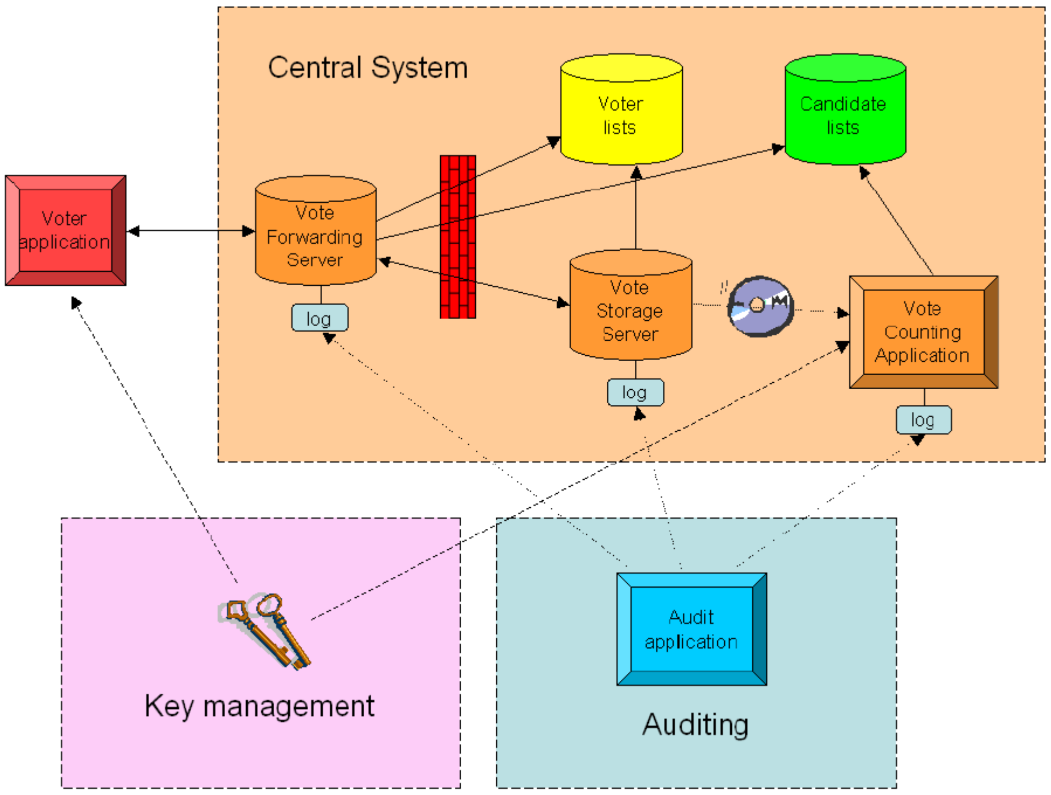


Figure 3.1: Estonian System [47]

When casting multiple votes from the i-voting system, only the last ballot will be counted and if the voter will use the classical system then this vote will be counted as a final vote. For details of this system, we refer to [46, 47].

3.1.2 Security Issues of Estonian System

Estonian i-voting system has many security considerations inside. But there is one gap that they have to spend time on to make the system more secure. This gap is the compromised PC’s. Since IVCA is a program that runs on voter’s PC and in case it is compromised, it may then be possible for a malicious software to change the ballot or it will be possible to learn the candidate that voter selected or it may compromise the integrity of the vote.

At first blush, it is understood that this is a crucial gap. This gap was announced in TV and newspapers and some people implemented computer programs to change the candidate in vote. Details of those events can be read from the reference [46].

Thus, fundamental problem in Estonian system was the compromised personal computers and maybe some verification procedures should be added to the system to let voters to check their votes in the system. In the next section, we examine Norwegian system and customized system in which we'll see such verification procedures.

3.2 Norwegian System

In 2008 Norwegian Government decided to use i-voting in local government elections as a trial. Ministry of Local Government and Regional Development selected 10 of their 439 municipalities and in elections which held on 11-12 September 2011, they used i-voting system [22]. In 9 September 2013, Norway Parliament selected 12 municipalities and total of 250.159 voters cast their votes via the internet [40]. Norway is the second country that used i-voting solutions for local governmental elections and they are planning to use i-voting system in 2017 for national election [34]. The Norwegian i-voting solution is prepared by Spanish on-line voting company Scytl [45]. It can be seen as an ameliorative version of Estonian system.

3.2.1 System Overview

To describe Norwegian i-voting system briefly, let's start with giving their high level architecture below.

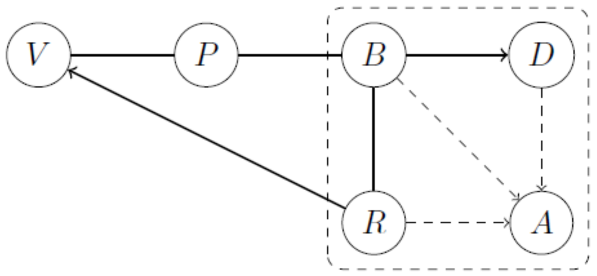


Figure 3.2: Norwegian System [28]

The letters in the Figure 3.2 represents the players in Norwegian system. Here, V denotes voter, P denotes PC, B is ballot box, R is the receipt generator, D is the decryption service, A is the auditor. The arrow from the player R to player V represents the receipt that is generated by R in cooperation with the B. This receipt code is the main mechanism for V to check his vote from the system.

V selects his candidate from the list at i-voting system and his selection is encrypted by P using the public key of the election. P also signs the vote using the private key of the V using the key inside V's ID-card. P sends encrypted and signed ballot to B. B checks the signature of V (it checks it whether if voter is a valid voter or not) and B calculates receipt code in cooperation with R [28]. After receipt code generation, R sends this receipt code to V by using an independent channel. SMS channel is used as an additional channel in Norway.

If V and vote are valid, R signs the hash of the encrypted vote and sends it to B. B transfers this data to P and P informs voter about the vote acceptance. All the computations of players are proved by zero knowledge proofs and auditor controls all logs during voting process.

When the ballot box closes, only the last ballots of each voter is collected (remember that voters have chance to cast their votes multiple times) and sent to decryption service in random order. Decryption service shuffles(permutes) the encrypted votes and re-randomizes them by using homomorphic property of ElGamal encryption which is a mix-net scheme. Decryption service decrypts ballots and announces the final ballots in random order.

3.2.1.1 Protocol Description

Here we will cover protocol explained above using mathematical descriptions and explanations. The protocol used in Norwegian system consists of three phases which are key generation, vote casting and tallying.

Key Generation:

Trusted party performs key generation process. It chooses two random values a_1 and a_2 and he computes a_3 such that $a_3 = a_1 + a_2$ mod q of a cyclic group G which has order q and generator g. B gets a_2, R gets a_3 and D gets the secret parameter a_1. Trusted party calculates public parameters $y_1 = g^{a_1}$,$y_2 = g^{a_2}$ and $y_3 = g^{a_3} \ mod\, q$. For every voter V, s is selected randomly from $\{0, 1, 2, \ldots, q-1\}$ and d is selected from the pseudorandom function family F. For each j that represents the candidate number, trusted party computes $\gamma = g^s$ and receipt code list $RC = \{(v_j, d(f(v_j)^s))\}$ [28]. Those receipt codes will be sent to each V before the elections as stated before.

Vote Casting:

V chooses vote v = $(v_1, v_2, \ldots, v_k)$ from web browser, he sends to P.

1. P sets $v_i = 0$ for $i = k + 1, \ldots, k_{max}$ where k_{max} represents the number of candidates in the election.

2. P gets a number t_i between $0 \leq t_i \leq q-1$ where $1 \leq i \leq k_{max}$ and he computes $(x_i, w_i) = (g^{t_i}, {y_1}^{t_i} f(v_i))$. He sends those k_{max}-tuple (x_i, w_i) to B.

3. B computes $\varrho_i = {x_i}^s$ and $\sigma_i = {w_i}^s {\varrho_i}^{a_2}$. He sends (ϱ_i, σ_i) with voter names to R.

4. R computes $r_i = d(\sigma_i {\varrho_i}^{-a_3})$ then he sends those r_is to V.
5. V checks that whether if those (v_i, r_i) is in the receipt code list or not.

Tallying: In tallying procedure, final ballots of each user will be counted, superceded ballots will be discarded. Here B sends the final encrypted ballots in random order to D. D decrypts all ballots as $w_i {x_i}^{-a_1}$ then he announces the resulting ballots in random order.

For more information and details, we refer to [28].

3.2.2 Security Issues of Norwegian System

Norwegian System is designed vigorously by using homomorphic encryption and verification mechanisms. Many security considerations have been taken into account by the system designers but there are still some open parts in their protocol. In this section, we will mention briefly about those problems.

3.2.2.1 Compromised or Cooperated Players

Since D knows a_1 which is the private key of the encryption algorithm, he opens all the ballots that he receives from B. If B does not apply MIX-NET operations properly, moreover if he gives the list of V and their corresponding encrypted votes to D, then privacy of the election will be destroyed.

When B and R cooperates, since B owns a_2, R owns a_3, they can also obtain secret key a_1 by subtracting a_2 from a_3 [34]. This time B and R can decrypt votes which violates the privacy of the voters. Therefore, there should be a mechanism to prevent those security violations.

3.2.2.2 Insecure Pre and Post Channel

As we stated before, security of Norwegian System mostly depends on verification mechanisms. Here, independent channel used for verification and the way that verification done are very important. For the pre-channel verification mechanism, printed receipt codes may be generated wrongly or they may change while they are transported to V. In [34], these pre-channel problems are stated and it is advised to use zero knowledge proofs during generation procedure. It is given in [34] that zero knowledge proofs can be verified by V by using a secure machine dedicated to do zero knowledge verification which does not have any connection with P or different computer can be used for verification procedure.

For the post-channel verification process SMS is used in Norwegian System. They used it as a secure channel since it does not have any connection to internet. But in

[11], it is stated that SMS is no longer a secure independent channel and in [11] three different SMS post-channel attack methods are given as follows:

1. Blocking Verification Method

In this method fake base transceiver station (BTS) is used. When V submits a vote, a verification message is sent to V by an SMS as normal. But in the same session, a malware in the browser (MITB) of the V casts another vote. Then fake BTS blocks the second SMS message from the receipt generator. At the end V will not notice that a vote has been cast by the malware on behalf of him. In Figure 3.3, this type of attack is shown.

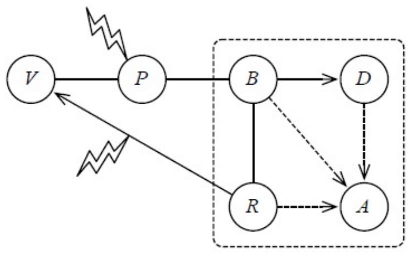


Figure 3.3: Blocking Verification Method [11]

2. Attacking Two Devices

In this scenario, it is assumed that both P and V's mobile device are infected by a malware. Like in the first scenario, in this case V casts his vote and a verification message is sent by R as usual. In the same session, MITB casts another vote and malware in the V's mobile device blocks receipt sent by R. Finally, V will not be aware of the last vote. Figure 3.4 illustrates this attack on modules.

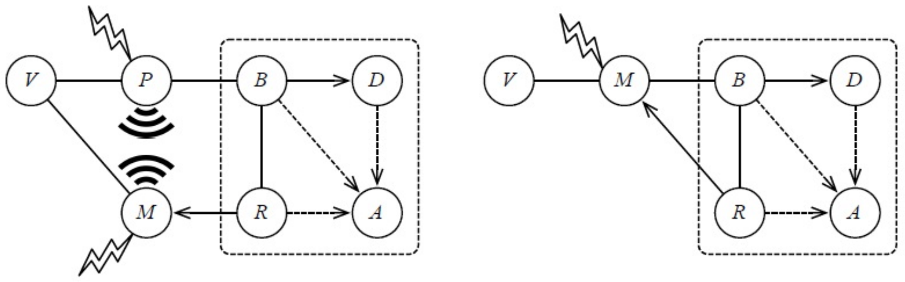


Figure 3.4: Attacking Two Devices [11]

3. Attacking A Single Device

In this case, it is assumed that V uses a mobile device both for vote casting and verification procedure. Therefore, it will be easier than other scenarios to make a post-channel attack. V casts a vote using his mobile device. A verification message is received and it is displayed to V as usual. Not to attract much attention, with some delay, another vote is being cast by the malware on the mobile device and verification message is being blocked and deleted.

In order to prevent from attacks given above, in [11] some limitations and precautions are given. As a first method, they offered to disable vote updating. Since, if vote updating is not possible, V will definitely be aware of a malicious attempt. Second method is given as vote updating in different sessions. Attack scenarios given above are done in the same session, if they make impossible to cast multiple votes in the same session, then they think that they make the system more secure. Third method is using transaction authentication number (TAN) which will be generated for each V in advance. They propose to generate a list of TAN's for each V before the elections and they advise to print them on a paper and send the list to V by postal service. It is given in the Norwegian System that V has opportunity to cast his vote multiple times. If this is the n'th time that V casts his ballot, then he will use n'th TAN in the web application to prove that he is the real voter. This cure seems that it is a simple solution and could make the system secure. But remember that it will limit the number of vote casting operation for each voter.

CAPTCHA is an acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart" [15]. It is a kind of challenge response test to understand that the user of a computer application is a human or not. Alternative method given in is using a CAPTCHA to understand that whether if the voter is a real voter or is a malicious software.

It is also proposed to use a trusted SMS receiver. This device is a limited machine and have ability just to show the coming SMS. Alternative method to recover from the first attack is to use signed verification SMS from V's mobile device to R. Using a trusted hardware token instead of using a device that does not use SMS as a verification method is asserted in the paper which can receive and display encrypted messages and return signed acknowledgement. They also state that by analyzing vote update frequencies one may detect an attack.

In [11], solutions mentioned briefly above are given. Among them, using a CAPTCHA seems more practical and it can prevent votes from malware.

3.2.3 Enhancements of Norwegian System

Most crucial feature of a cryptographic system is transparency. If someone thinks that his system is secure, he makes all parts of his system explicit to make other people to test his system against any attack. If analysts find any flaw in his system he will have chance to fix it before any serious security violation. Moreover, by making all parts of the solution transparent, trust of community on the system will increase. To build trust in their system, Norwegian government decided to make their system public. They revealed their system in the site of Ministry of Local Government and Regional Development [18]. They revealed not only the infrastructure but also source code of the system in order to increase trust to the system. In Norwegian system, they added two checking mechanisms for security. These are pre and post channel information sent to V. In their system, as a pre-channel information receipt code is used. Before the elections a receipt is prepared which includes candidate names and their corresponding verification codes. This receipt is prepared for each voter and it is sent to V by postal service before the elections. When V casts a vote, receipt is calculated again and sent to V by R using SMS. V gets the receipt and checks it from the receipt code paper which is sent in advance. If the received code is the true verification code for his candidate then it shows that his computer encrypted his vote truly and sent to B, else voter has chance to cast his vote again [36]. Therefore, SMS verification is used mainly against compromised computers. Remember that in Estonian System, there is no security mechanism which controls compromised computer problem.

3.3 Customized System

In this section, we will focus on an i-voting system which can be studied attentively in [5]. The protocol mentioned in [5] is similar to Norwegian Voting System which uses mainly homomorphic encryption, receipt codes and verification SMS messages. On the other hand, because of the problems expressed in the previous section, notable changes have been made on the existing system. You can find the information about the customized system below.

3.3.1 System Overview

Customized system is similar to Norwegian System but it has more modules than the previous one. Below you can find a table that shows modules and their abbreviations. Among them bulletin board (BB) is an important module that does not exist in Norwegian scheme. Bulletin board is a public channel, such as a website in the internet, holds information for the voters or for the public to verify the election without losing secrecy [27]. With the help of BB, voter has chance to check his vote till the end of the system. In Norwegian scheme, remember that voter may have chance to follow his vote till receipt generator. After receipt generator, voter cannot be sure that whether his vote is tallied as intended. Therefore, this feature is a good opportunity to control the system from the beginning to the end of it. In Table 3.1, symbols used in customized system and their corresponding modules are given.

Table 3.1: Modules [5]

Symbol	Module	Symbol	Module
V	The Voter	v	Ballot
B	Classical Ballot Box	PC	Voter's Computer
TC	Thin Client	TS	Terminal Server
AB	Authentication Box	CB	Control Box
A	Authority	DS	Decryption Service
BB	Bulletin Board	C	Counter
TP	Trusted Parties	CA	Certification Authority

Authentication box (AB) is the party whose main role is to authenticate voters. Control box's (CB) role is to control authentication box and send a receipt code to voter for verification. Authority (A) is the module that checks AB and CB, gathers final votes and transfers them to counter C. C's role is to tally votes without decrypting them. Decryption service DS is the module that decrypts all tallied votes which is far from other modules, because it will obtain the private key of the election at the end of the election period.

Before the elections, private and public key of the election/all users are generated by a trusted party by using threshold cryptography. The private key which voters use for signing encrypted votes will be in their ID-Cards. Public key of users will be given to authentication box whose main role is to authenticate voters.

Receipt code is used as a verification mechanism in customized system like in the Norwegian solution. But in customized one it is simpler. All the encryptions in customized system are done by using ElGamal algorithm. Each voter will use one random number in encryption process. For each voter, receipt codes are generated by using hash function. Receipt code includes all hash values of possible encrypted ballots for all candidates using voter's random number. Receipt codes are sent to election precincts or to the voters with random numbers. In addition to receipt codes, each voter has secret numbers which they will use when casting ballots. Those secret numbers are needed to protect users from coercion.

3.3.1.1 Protocol Description

Brief explanation of customized system is given above. In this part, we will cover important parts of this system one by one and give details of the protocol. In Table 3.2 variables sent between modules are given and Figure 3.5 indicates modules and their communication between each other.

Table 3.2: Sending Variables [5]

1	$Sign, c, ID, ZK_{PC}, \alpha$	2	$c, ID, ZK_{PC}, ZK_{AB}, \alpha, s, u$
3	$c^{a_1}, ID, ZK_{PC}, ZK_{AB}, \alpha, s, u$	4	$c^{a_2}, ID, ZK_{PC}, ZK_{CB}, \alpha, s, u$
5	$c^{a_1+a_2}, ID, ZK_{PC}, ZK_{AB}, ZK_{CB}, ZK_A, \alpha, s, u$	6	h, ID, α
7	$E(v_1 + v_2 + \cdots + v_l; r_1 + r_2 + \cdots + r_l)$		

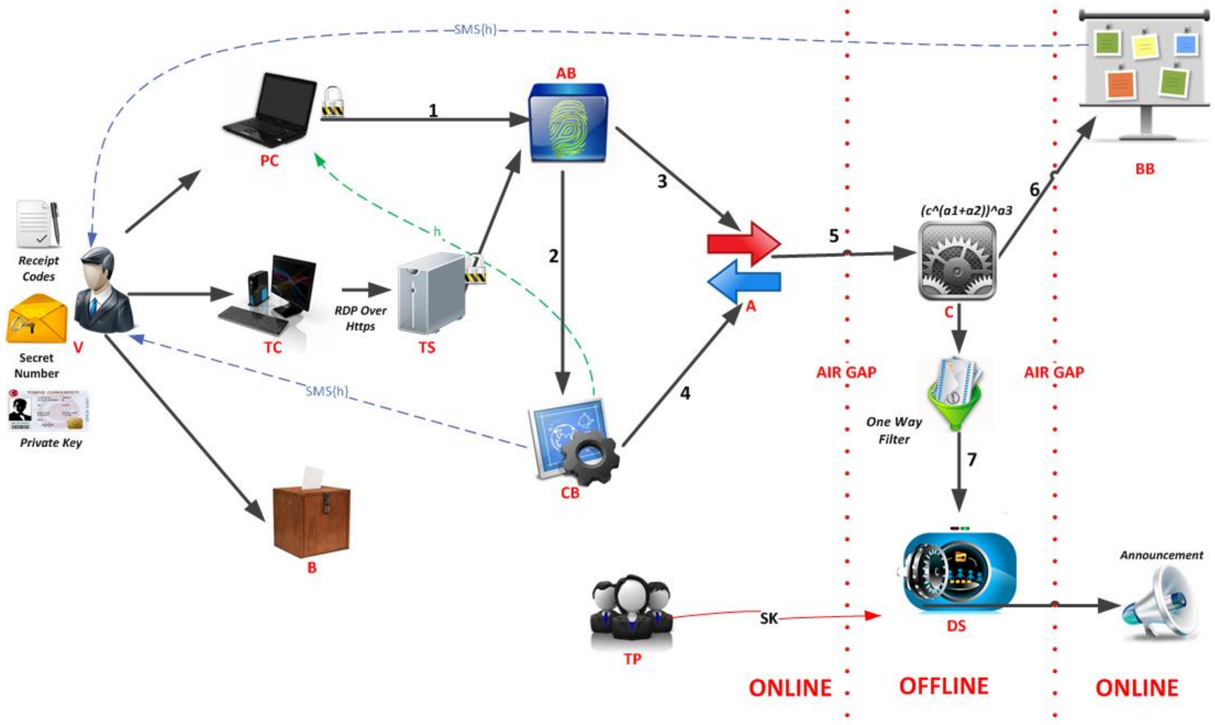


Figure 3.5: Customized System [5]

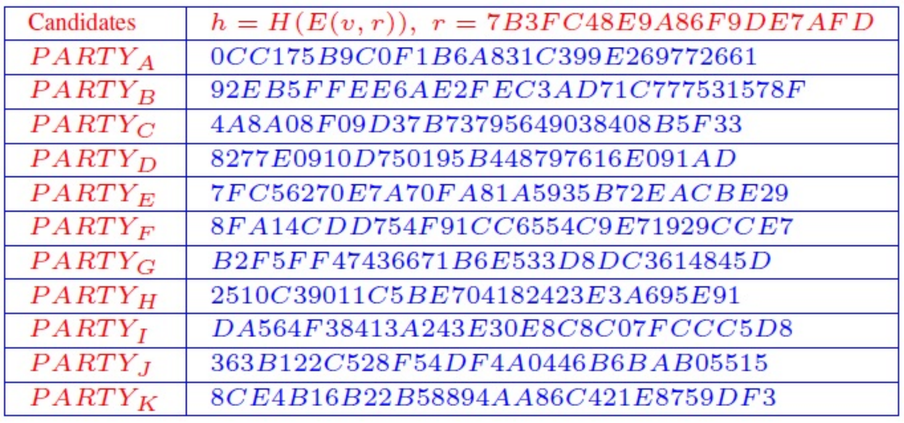
Secret Number:

As it is stated before, each user has a secret number. Those secret numbers are sent to voters by SMS before the elections with commitment of it. But the time of sending secret number is not definite. Voter may receive this message at any time in a month. Since it is not logical for a coercer to handle voters' mobile device for a long time, learning secret number for a coercer is not that easy because of the undefined time interval. To withstand coercion in the system, voter may request new secret numbers after real secret number's arrival. Those extra secret numbers are all fake and they will not be used in the system.

When voter selects candidates from the system, before casting their ballots, system will require his secret number. If voter inputs correct secret number, he will receive SMS from CB and vote is tallied. If voter inputs a fake secret number then system will not give any warning to voter, he will receive SMS from CB as normal, vote will be transported till the counter and counter will give this log to bulletin board (To protect voter from coercion). At the end, this vote will not be tallied. Note that, this secret number will be user friendly, for example four digits number. Voters can easily memorize this number without writing somewhere which endangers security.

Receipt Code:

It is given before that in Norwegian system a special function is used to generate a receipt. In customized system, simple hash function is used which gets hash of encrypted vote. Since each user have different random numbers then voters who will cast their votes to the same candidate will get different hash values. Each receipt code paper includes hash values of all encrypted votes using the random number on the same paper. Therefore, the number of hash values on the receipt code paper is equal to the number of parties attended to the election. Those receipt code papers are generated by High Election Board (HEB) before the elections and they are sent to election precincts and distributed to voters randomly. Suppose that there are 100 voters in a village, then more than 100 receipt code papers are generated and those papers will be sent to election precinct. Voters will get their paper from election precinct randomly. It is given that the probability of finding the link between each voter and receipt code paper in this scenario will be $1/100$ which is very low. Note that if there is a coercion, voters have chance to get a new receipt code paper from the election precinct. Below you can find a table taken from [6] which shows a sample receipt code paper in this system.



Candidates	$h = H(E(v,r)),\ r = 7B3FC48E9A86F9DE7AFD$
$PARTY_A$	$0CC175B9C0F1B6A831C399E269772661$
$PARTY_B$	$92EB5FFEE6AE2FEC3AD71C777531578F$
$PARTY_C$	$4A8A08F09D37B73795649038408B5F33$
$PARTY_D$	$8277E0910D750195B448797616E091AD$
$PARTY_E$	$7FC56270E7A70FA81A5935B72EACBE29$
$PARTY_F$	$8FA14CDD754F91CC6554C9E71929CCE7$
$PARTY_G$	$B2F5FF47436671B6E533D8DC3614845D$
$PARTY_H$	$2510C39011C5BE704182423E3A695E91$
$PARTY_I$	$DA564F38413A243E30E8C8C07FCCC5D8$
$PARTY_J$	$363B122C528F54DF4A0446B6BAB05515$
$PARTY_K$	$8CE4B16B22B58894AA86C421E8759DF3$

Figure 3.6: Sample Receipt Code Paper [6]

If voter trusts both his PC and the system, he does not have to make this verification process. Therefore, it is not mandatory. It is given that 5% of the voters are sufficient for verification to understand the correctness of the system.

Since the hash value also comes from CB to PC, if voter thinks that his PC is not compromised and if he does not trust the voting system, he may not use the random number on the receipt code paper. He may use the random number given by the system.

If voter does not trust his PC, it is better to use the random number on the receipt code paper. With this paper, voters can understand any problem in the PC or in the system. If voter wants to cast his vote multiple times using the random number on the receipt code paper, in order to prevent anyone to understand the link between the votes and their encrypted data, it is advised to get a new receipt code paper for the new vote casting attempt.

Key Generation:

In this system, private key of the election is produced by using threshold crypto system with the shares of governmental institutions, political parties, universities etc. which are named as Trusted Party in the infrastructure. Let x is the private key of the election,

$h = g^x \mod p$ is be the public key of the election (Used for encrypting votes). Beside keys, some masking parameters are defined in this system as follows. Secret masking parameter a_1 is produced for AB, a_2 is produced for CB and a_3 for C. There is a relation between those parameters which is $a_1 + a_2 = {a_3}^{-1} \mod p - 1$ where a_3 is invertible in $\mod p - 1$.

Note that, parameters a_1, a_2, a_3 are not used for encrypting/decrypting votes as in Norwegian solution, aim of using those parameters in the protocol is to prevent CB and/or A to create votes by itself. Moreover, those parameters are used to be sure that every vote passes over AB and CB.

Voter's Items:

When voter wants to use the system, he has his ID-Card which contains his private key, he has a receipt code paper with random number on it and he has a secret number as shown in the ”Voter's Items” figure below. A link in one of a governmental portal web site is given for voters who would like to use i-voting system. People use their ID-Cards or ID numbers and passwords to login to that page.

Figure 3.7: Voter's Items [5]

We are now ready to present this protocol:

Vote Submission:

1. At vote casting stage voter have 2 different opportunities. One is using i-voting system with PC the other one is with TC. Voter connects to i-voting web site, he selects his candidates and inputs his secret number.

2. Firstly, computer encrypts the vote by using public key of the election which is h as follows, $E(v, r) = (g^r, g^v h^r) = c$. Secondly, it gets the hash of encrypted vote, $H = Hash(c)$ and it signs vote using the private key of the voter which is in the ID-Card of the user, $Sign = Sign(h)$. Computer sends signature, encrypted vote, his ID number, zero knowledge proof of his own computations, commitment of secret number($\alpha = commit(r, n)$) to authentication box. It is

shown in the Vote Submission figure below which is indicated with the arrow 1, $PC \xrightarrow{Sign,c,ID,ZK_{PC},\alpha} AB$.

3. At this step, AB checks voter's identity by using the public key of voter and checks computer's calculations by looking at the ZK_{PC}. Since voter has chance to cast his vote multiple times, AB puts a sequence number (s) to each vote in order to understand the last vote of user and he gives a unique number (u) to each vote. AB sends zero knowledge proof of his own calculations, u, s and all received data from PC except $Sign$ to CB. This is shown in the figure below with arrow 2, $AB \xrightarrow{c,ID,ZK_{PC},ZK_{AB},\alpha,s,u} CB$.

4. CB controls ZK_{AB}, ZK_{PC} and gets the hash of c. He sends this hash value to voter by an independent channel (SMS). He also sends that value to PC which is used by him as an indication for approval. Now, voter has opportunity to control his vote from CB. If the received receipt code is true than voter should be aware that all the computations and transportations till CB is done truly. Process done in this step is given in Figure 3.8 below.

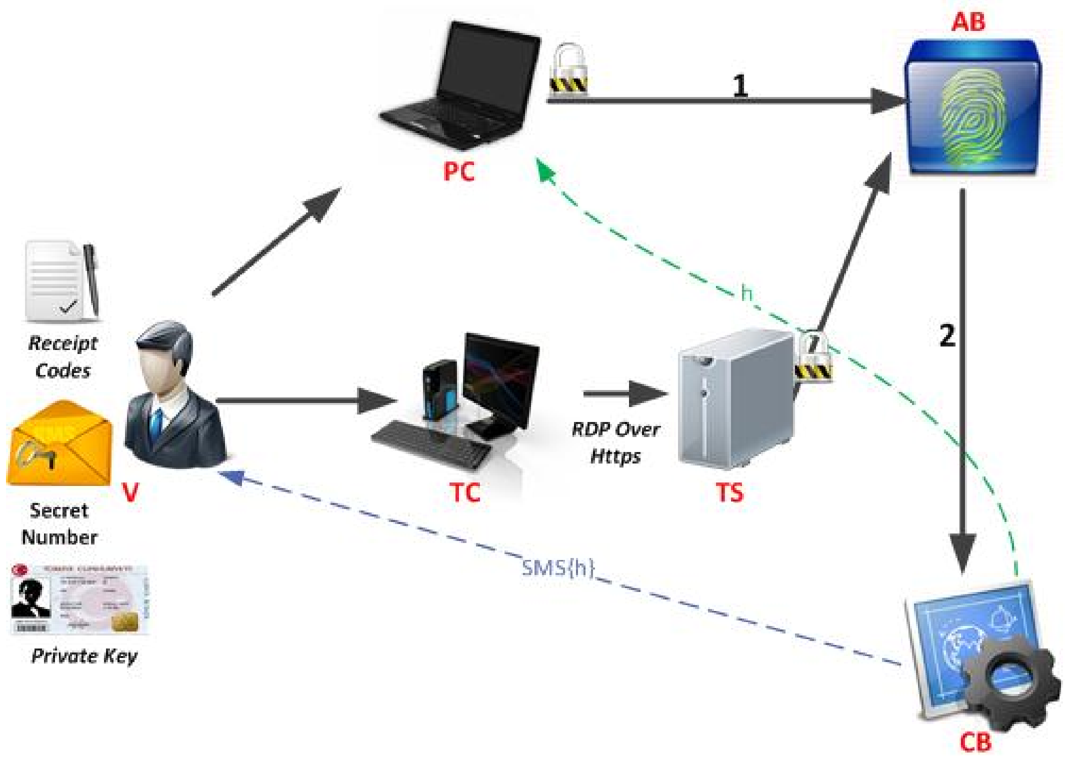


Figure 3.8: Vote Submission [5]

5. After verification procedure, CB and AB masks encrypted vote using their secret parameters as follows, c^{a_1}, c^{a_2}. Remember that those masking parameters are used to be sure that no vote is created by A and CB itself and vote from V is passed through AB and CB. They send data to A shown in the figure below which are shown with arrows 3 and 4.

$AB \xrightarrow{c^{a_1}, ID, ZK_{PC}, ZK_{AB}, \alpha, s, u} A.$
$CB \xrightarrow{c^{a_2}, ID, ZK_{PC}, ZK_{CB}, \alpha, s, u} A.$

6. A gathers vote pairs coming from AB and CB by using unique number u and multiplies them.
$c^{a_1}c^{a_2} = E(v^{a_1}, ra_1)E(v^{a_2}, ra_2) = E(v^{a_1+a_2}, r(a_1 + a_2)) = (g^r, g^v h^r)^{a_1+a_2} = c^{a_1+a_2}$
After calculating this, he verifies ZK_{PC}, ZK_{AB}, ZK_{CB} and he saves ballot to its database. Masking operations in the figure below shows those data flow. Notice that all the modules till that point are online.

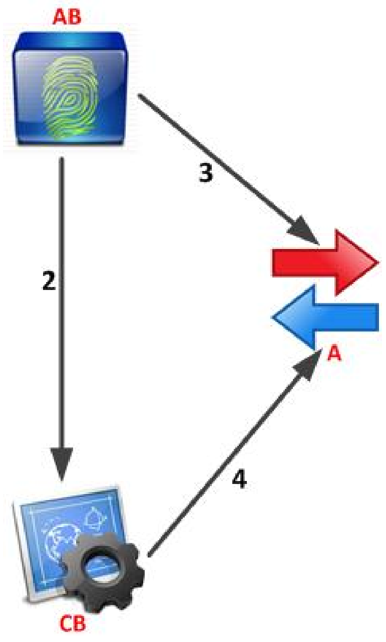


Figure 3.9: Masking Operations [5]

7. Since C is an offline module in the system, from A, collected votes are transported to C periodically. 5'th arrow below shows the data flow from A to C which is:
$A \xrightarrow{c^{a_1+a_2}, ID, ZK_{PC}, ZK_{AB}, ZK_{CB}, ZK_A, \alpha, s, u} C.$

8. C controls the proofs received from A and decrypts $c^{a_1+a_2}$ by using his secret parameter a_3. $(c^{a_1+a_2})^{a_3} = c \mod p$. He controls commitment of secret number, if it is true than he marks vote as valid.

9. C gets the hash of c, sends hash, ID number and commitment to BB by using an external HDD. Arrow 6 in the figure below shows this data flow.
$C \xrightarrow{h, ID, \alpha} BB$

10. BB sends h by SMS to voter and besides that voter has chance to check his vote from BB by making query from it. Queries on BB will be done using users ID number and hash of the encrypted vote.

Tallying:

1. At the end of the election, C selects last ballot of each user which are cast using correct secret number. Firstly, C gathers randomly l ballots, the number l is limited because of the discrete logarithm problem. Using homomorphic property of ElGamal, when he multiplies those ballots, he adds them as follows: $E(v_1, r_1)E(v_2, r_2)\cdots E(v_l; r_l) = E(v_1 + v_2 + \cdots + v_l; r_1 + r_2 + \cdots + r_l)$

2. Since DS will obtain the private key of the election at the end, in order to preserve privacy, DS should be kept far away from the other modules and there should be a mechanism which controls traffic between DS and other modules. In order to control traffic, one way filter device is used as can be seen in the figure below. C sends multiplied l ballots to DS which is represented by the 7'th arrow below in the figure, $C \xrightarrow{E(v_1,r_1)E(v_2,r_2)...E(v_l;r_l)} DS$.

3. Finally, at least t of trusted parties gather and they reproduce private key and give it to DS. DS will decrypt each bunch which contains l ballots' result.

4. Results will be announced.

Offline operations done in the system is depicted in Figure 3.10 below.

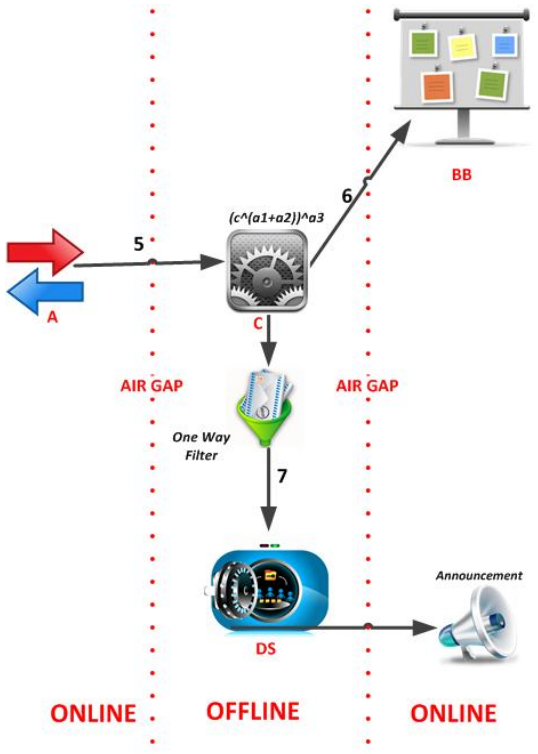


Figure 3.10: Offline Operations [5]

3.3.2 Security Issues of the Customized System

In this section, we will study some scenarios in which there are some members who cooperate or who are compromised. Let's start with compromised computer case.

1. Compromised PC: If PC is compromised, it may send a wrong ballot or it can send a vote using a wrong secret number. Suppose that it casts a vote to a different candidate. In that case voter can notice that change when he receives verification SMS from CB or when he makes a query on BB. If any problem occurs in the SMS verification then he may check any vote alteration from BB online.
2. Compromised AB: If AB is compromised and it generates a vote on behalf of voter, an SMS will be sent to the voter. This SMS message is an indication that shows a module generated a fake vote. By looking at the logs, compromised module should be found.
3. Compromised CB: If CB is compromised and generates fake votes, A will notice this security violation since there is a masking mechanism to understand this.
4. Cooperation Between AB, CB and A: In case of a cooperation of AB and CB or between AB, CB and A, if any fake vote exists then it could be noticed from the BB and also by looking at the audits, this kind of a security violation can be understood.
5. Cooperation Between BB and PC: In case of a cooperation between BB and PC, any security violation should be noticed by checking zero knowledge proofs.

All the information and figures about customised system above are gathered from reference [5]. From now on, we will cover some security considerations not from that article.

3.3.2.1 Insecure Pre and Post Channel

Wrongly created receipt codes is also problem in customized system. In such a case, many people may think that there is a problem in the i-voting system. If there are a lot of people in this group then community's trust on the i-voting system will diminish which will decrease the number of people who uses automated system. Therefore, here recommendation in [34] should be considered about the usage of zero knowledge proofs in receipt code generation. About the post channel (SMS) problem, same problems in Norwegian case exist in the customized system. But, in customized system there is a verification mechanism that is not related with SMS verification which is using BB. This opportunity certainly decreases the damage of wrong information or any blockage on the SMS communication on the system since finally voters may see the true receipts from online system. But we know that it will take some time to transport vote information to BB. Within this time interval, notice that if any problem occurs in the SMS verification system then people may be panic. In order to prevent this problem, latency of announcement in the BB should not be long.

3.3.2.2 Possible Amendments For The Customized System

Coercion and receipt-freeness are crucial points to be considered for i-voting solutions. Receipt-freeness is satisfied in this system because whenever a voter wants to prove to a vote buyer that he has cast his vote to a given candidate, he may make queries on the BB using all the hash values on the receipt code paper. However, there is one other case to be considered. Voter might cast his vote without using the random number on the receipt code paper. He might cast his vote using the random number given by the system. In that case, vote buyer cannot be sure whether if voter has cast his vote using the random number on the receipt code paper or using the random number given by the system. This is a very good property of this system. On the other hand, in customized system, some precautions are taken into account to resist against any coercion attempt. Multiple vote casting ability, secret number usage, multiple secret number generation can be stated as a precaution for this system. But suppose that there is a coercer who forces a voter to cast his vote to a candidate. Voter by force casts his vote to that candidate and suppose that this is the first time he casts his vote in that election. After that, coercer have chance to make query for all the candidates using hash values in the receipt code paper and voter's ID number. By making queries on BB, coercer have opportunity to check whether if voter cast his vote again or not. Notice that, this is possible whenever voter casts his vote using the random number on the receipt code paper, it is recommended to cast vote using the random number given by the system or multiple receipt code papers can be used to prevent from coercion case.

In order to make this system more robust against coercion problem, it may be limited to make query on BB. Maybe if voter casts his vote then he may have 3 opportunities to make a query on the BB. With the help of this limit, voter may not have ability to make queries for the whole list in the receipt code paper.

Another point that we can add to new system is using CAPTCHA for vote submission. With the help of CAPTCHA usage, some malware attacks in the system could be inhibited.

3.3.3 Enhancements of the Customized System

In Estonian solution, they did not put any verification mechanism to their system. Therefore, when there is a compromised computer case, voters couldn't notice any security violation. In Norwegian scheme, voters may have chance to follow their votes till receipt generator. In customized model, verification ability till counting step has been given to voters which brings almost end-2-end security.

Remember that both Norwegian and customized systems are using homomorphic property of ElGamal algorithm. In Norwegian case, they use homomorphic encryption to re-randomize ballots. But in customized system, they use it to tally ballots. Thus, in Norwegian system they decrypt votes one by one and tally them, but in customized system l of ballots are multiplied and result is decrypted which brings more privacy.

Main aim of this system is to recover secret key problem in Norwegian scheme. Remember that when R and B cooperates in Norwegian scheme they obtain secret key of election. So that they can decrypt and learn the voter's will. In customized scheme, votes are encrypted at PC and it continue its way till to DS in an encrypted form, DS only gets data which is the multiplication of l of random votes. Except DS no module in the system can possess the private key of the election. Then they don't have chance to decrypt votes. In addition to that, at the end of the elections, DS have the private key of the elections but since he will get the multiplication of l random votes, it is also impossible for DS to decrypt any votes and learn any voter's will.

Secret number usage is another enhancement in the system since it increases system's resistivity against any coercion attack. Using hash function instead of a special function (in Norwegian Scheme) is a simplification in this model.

CHAPTER 4

CONCLUSION

It is evident that in the near feature, percentage of i-voting solutions' usage will increase. Since election is a chance for every citizen to feel themselves as a part of government, it possesses a great importance to design an i-voting system. In constructing an i-voting system, one should have adequate knowledge of information technology and cryptography. If any security violence occurs in the system, it will not only diminish the i-voting usage but also it will make people to think themselves as not a part of community. Therefore, this might differentiate public. Thus, in order to build high-level democracy, election systems play a crucial role.

One has to take core issues into account for electronic voting systems such as correctness, privacy, verifiability, transparency, vote buying and coercion. Cryptographic protocols and encryption algorithms play an important role to assure these properties. Moreover, any developed automated system which will be used as an i-voting system should also be tested against all possible scenarios by different and independent organizations in order to ensure full trust and reliability. The system should also be transparent in such way that all parts of the system should be made public to increase the reliability of the system.

Homomorphic encryption has a key role for cryptographic protocols used in i-voting solutions. There should be a balance between the security measures and practicability. Namely, if you increase the security level of your system you may decrease its performance or usage. We know that today homomorphic encryption is not efficient for large data sets [29]. Since we do not play with large data in i-voting solutions, this does not add a significant overhead in order to make computations on encrypted votes.

In this thesis, we studied cryptographic protocols for i-voting solutions. We revisited Estonian, Norwegian and customized i-voting schemes and point out security and efficiency remarks for these proposals. It is not easy to find the best i-voting solution for every country since each country's or each election's requirements can be different. On the other hand, every solution has its best features and some security leakage. Risks and possible results of their effects should be thought and most suitable solution for election should be selected.

REFERENCES

[1] A.J.Menezes, P.C. van Oorschot, S.A.Vanstone, *Handbook of Applied Cryptography*, CRC Press, pp.103-118, pp.22-24, 1997.

[2] A. Kiayias, A. Orfanou, *Scaling Privacy Guarantees in Code-Verification Elections*, 4th International Conference, Vote-ID 2013, Guildford, UK, July 2013.

[3] A. M. Al Khouri, *PKI Technology: A Government Experience*, March 2012, International Journal of Computer Science, Engineering and Information Technology Research.

[4] A. Shamir, *How to share a secret*, Communications of the ACM, 22, pp.612-613, 1979.

[5] A. Sınak, M. Sabır Kiraz, S. Özkan, H. Yıldırım, *A Secure Internet Voting Protocol Based on Homomorphic Encryption*, ISCTURKEY 2013, Proceedings of 6th International Conference on Information Security and Cryptology, pp.142-148, Sep 20-21, 2013.

[6] A. Sınak, S. Özkan, H. Yıldırım, M. Sabır Kiraz, *End-2-End Verifiable Internet Voting Protocol Based on Homomorphic Encryption*, International Journal of Information Security Science (Accepted in April 2014).

[7] B. Schoenmakers, *Lecture Notes on Cryptographic Protocols*, http://www.win.tue.nl/ berry/2WC13/LectureNotes.pdf, Accessed:21 April 2013.

[8] D. Cochran, J. R. Kiniry, *Formal Model-Based Validation for Tally Systems*, 4th International Conference, Vote-ID 2013, Guildford, UK, July 2013.

[9] D. L. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, University of California-Berkeley, February 1981, https://mirror.robert-marquardt.com/anonbib/cache/chaum-mix.pdf, Accessed:23 April 2013.

[10] Electoral Council of Australia and New Zealand, *Internet voting in Australian election systems*, September 2013, http://www.eca.gov.au/research/files/internet-voting-australian-election-systems.pdf, Accessed:23 October 2013.

[11] *Final results from the e-voting in the trial municipalities in Norway*, 2011, http://www.regjeringen.no/en/dep/krd/prosjekter/ e-vote-2011-project/final-results-from-the-e-voting-in-the-t.html? id=654212, Accessed:21 May 2013.

[12] G.Castagnos, B.Chevallier-Mames, *Towards a DL-based Additively Homomorphic Encryption Scheme*, ISC 2007, Vol. 4779 of Lecture Notes in Computer Science, pp. 362-375, Springer-Verlag, 2007.

[13] H. Delfs, H. Knebl, *Introduction to Cryptography*, Springer, Second Edition, pp.100-101, 2007.

[14] https://www.verifiedvoting.org/resources/internet-voting/, Accessed:21 December 2013.

[15] http://en.wikipedia.org/wiki/CAPTCHA, Accessed:13 July 2013.

[16] http://en.wikipedia.org/wiki/Mix_network#cite_note-chaum-4, Accessed:14 May 2013.

[17] http://en.wikipedia.org/wiki/Zero-knowledge_proof, Accessed:19 May 2013.

[18] http://evalg.dep.no, Accessed:23 January 2014.

[19] http://neu.e-voting.cc/wp-content/uploads/2012/03/e-voting_map_2011.pdf, Accessed:23 December 2013.

[20] http://technet.microsoft.com/en-us/library/cc731264(WS.10).aspx, Accessed:21 December 2013.

[21] J. Budurushi, M. Henning, M. Volkamer, *Vote Casting in Any Preferred Constituency:A New Voting Channel*, 4th International Conference, Vote-ID 2013,Guildford, UK, July 2013.

[22] J. B. Esteve, B. Goldsmith, J. Turner, *International Experience with E-voting:Norwegian E-vote Project*,IEFS, June 2012.

[23] J. Epstein, *Internet Voting, Security, and Privacy*, 19 Wm. Mary Bill Rts. J. 885,2011, http://scholarship.law.wm.edu/wmborj/vol19/iss4/4, Accessed:11 June 2013.

[24] J. F. Raymond, A. Stiglic, *Security Issues in the Diffie-Hellman Key Agreement Protocol*, Zeroknowledge Inc, Technical Paper, September 2000, Accessed:17 July 2013.

[25] J. R. Vacca, *Public Key Infrastructure-Building Trusted Applications and Web Services*, CRC Press, 2004.

[26] J. U. Hjalmarsson, *Implementing a Mix-net For Use in Electronic Voting Systems*, KTH Computer Science and Communication,Bachelor's Thesis, http://www.csc.kth.se/utbildning/kth/kurser/DD143X/dkand13/Group2Douglas/report/uddholm.pdf, Accessed:23 December 2013.

[27] K. Braunlich, R. Grimm, *A Formal Model for the Requirement of Verifiability in Electronic Voting by Means of a Bulletin Board*, 4th International Conference, Vote-ID 2013, Guildford, UK, July 2013.

[28] K.Gjosteen, *Analysis of an Internet Voting Protocol*, Cryptology ePrint Archive: Report 2010/380, March 9, 2010.

[29] K. Lauter, M. Naehrig, V. Vaikuntanathan, *Can Homomorphic Encryption be Practical?*, Proceedings of the 3rd ACM Workshop On Cloud Computing Security Workshop, CCSW, 2011.

[30] K. P. Yee, *Building Reliable Voting Machine Software*, PHD Thesis, University of California, Berkeley, Fall 2007.

[31] K. S. Mccurley, *The Discrete Logarithm Problem*, Proceedings of Symposia in Applied Mathematics Volume 42, pp.49-51, 1990.

[32] M. Akinwande, *Advances in Homomorphic Cryptosystems*, Journal of Universal Computer Science, vol. 15, no.3, 2009.

[33] M. Arnaud, V. Cortier, C. Wiedling, *Analysis of an Electronic Boardroom Voting System*, 4th International Conference, Vote-ID 2013, Guildford, UK, July 2013.

[34] M. A. Bingol, F. Birinci, M. S. Kiraz and S. Kardas, *Norwegian Internet Voting Protocol Revisited:Security and Privacy Enhancements*, In proceedings of International Conference BulCrypt, Sofia, Bulgaria, September 2012.

[35] M. Gagne, P. Lafourcade, Y. Lakhnech, R. Safavi-Naini,*Automated Security Proof for Symmetric Encryption Modes*, December 2009, 13th Asian Computing Science Conference, Seul, Korea.

[36] M. J. M. Chowdhury, *Comparison of e-voting schemes: Estonian and Norwegian solutions*, 2010.

[37] M. M. Olembo, S. Bartsch, M. Volkamer, *Mental Models of Verifiability in Voting*, 4th International Conference, Vote-ID 2013, Guildford, UK, July 2013.

[38] M. Nabeel, S. Kerr, X. Ding, E. Bertino, *Authentication and Key Management for Advanced Metering Infrastructures Utilizing Physically Unclonable Functions*, IEEE SmartGridComm 2012 Symposium, 2012.

[39] N. Koblitz, *Algebraic Aspects of Cryptography*, Springer, Volume 3, pp.8, 1999.

[40] Office for Democratic Institutions and Human Rights, *OSCE/ODIHR Election Assessment Mission Final Report*, 16 December 2013, http://www.osce.org/odihr/elections/109517, Accessed:28 December 2013.

[41] *Proposed Federal Information Processing Standard for Digital Signature Standard (DSS)*, Federal Register, 1991.

[42] P. Y. A. Ryan, D. Bismark, J. Heather, S. Schneider, Z. Xia, *The Pret a Voter Verifiable Election System*, Information Forensics and Security, IEEE Transactions on Volume:4, Issue:4, 2009.

[43] RSA Laboratories, *RSAES-OAEP Encryption Scheme, Algorithm specification and supporting documentation*, 2000.

[44] R. Cramer, R. Gennaro, B. Schoenmakers, *A Secure and Optimally Efficient Multi-Authority Election Scheme*, EUROCRYPT, vol. 1233, pp.103-118, 1997.

[45] Scytl, *A Secure Electronic Voting Company*, http://www.scytl.com/, Accessed:23 April 2013.

[46] S. Heiberg, P. Laud,J. Willemson, *The Application of I-voting for Estonian Parliamentary Elections of 2011*, VoteID 2011, LNCS, vol. 7187, pp.208-223, Springer, Heidelberg, 2012.

[47] The Estonian National Electoral Committee, *General Description of the E-Voting System*, 2010.

[48] T. ElGamal, *A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms*, CRYPTO, 1985.

[49] T. Lepoint, *Secret Sharing*, http://www.cryptoexperts.com/tlepoint/pub/misc/Lep10-ssharing.pdf, Accessed:23 January 2014.

[50] U.S.Department of Commerce/NIST, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication, Federal Register, January 2000.

[51] V. Shoup, *Advanced Cryptography Lecture Notes*, 2007, http://cs.nyu.edu/courses/ spring07/G22.3220-001/lec4.pdf, Accessed:20 April 2013.

[52] W. Diffie and M. E. Hellman, *New Directions in Cryptography*, IEEE Transactionson Information Theory, 1976.

[53] Y. Ishai, E. Kushilevitz, R. Ostrovsky, A. Sahai, *Zero-Knowledge from Secure Multiparty Computation*, 2007, SIAM Journal on Computing.