A Secure Internet Voting Protocol Based on Homomorphic Encryption

Ahmet Sinak*, Mehmet Sabir Kiraz*'**, Secil Ozkan*, and Hakan Yildirim*

***Abstract*—In this paper, we focus on Internet voting protocol. Our protocol is similar to the Norwegian scheme which has been used in local elections in 2011. The primary focus of this paper is to prevent a possible cooperation between Ballot Box and Receipt Generator in the Norwegian scheme. The other purpose of this research is to present the alternative solution of coercion which is one of the most important problem in Internet voting. In our protocol, a voter can verify whether her vote is in the counting process.**

***Index Terms*—Electronic Voting, Internet Voting, Homomorphic Encryption**

I. Introduction

ELECTRONIC voting solutions are used by most of the countries around the world besides the classical solutions for elections. In particular, Internet voting (i-voting) is popular and used lately in some countries. Estonia is the first country which is using i-voting for general elections since 2005. Estonian i-voting system and the analysis of the elections are presented in [8], where more than 140 000 people cast their votes in the new system that makes the almost quarter of the whole population. Internet voting is believed to be useful for the voters in the country and for the expatriates since they spend too much time to go to the polling station. In Estonia, about half of the i-voting system users stated that they were spending about half an our to go the place in order to cast their votes. Therefore, it is obvious that Internet voting system simplified the voting process. Norway is another country who is using i-voting for local governmental elections in September 2011 and preparing a new and enhanced i-voting system for the national elections in 2017. The original cryptographic protocol for Norwegian elections was prepared by a Spanish company (Scytl) and later has been modified in order to suffice certain requirements. In order to create society's trust to the new system, Norwegian government made all parts of the technical details of the solution public [4]. Moreover, Norwegian Internet Voting System is analyzed and some improvements has been accomplished in [6], [7]. While Estonia and Norway are the pioneer countries using the i-voting, it is also used in Switzerland for several years. France also used Internet voting for expatriates in the general elections in 2012. Subsequently, more countries got interest to apply for the local and general elections.

Main drawback in an Internet voting is coercion. It is obvious that coercion can manipulate the election results and harm the voters freewill. Coercion can be done not only by force or threat, but also with personal relationships. Moreover, in some parts of the world, coercion might not only be personal but a group can also be forced or threatened to vote for a given candidate. Therefore, the effect of coercion on the election results cannot be ignored and some precautions should be taken. In order to prevent from coercion, the voters, in the previous works, are allowed to cast their votes more than once and only the final vote is counted. In addition to this, we propose to use secret number while casting a vote. One of the most important findings of our work is to find a solution to avoid coercion. In addition to coercion, there are many other issues to be solved but the most crucial ones are listed below: authentication, vote buying/selling, vote alteration, vote privacy, wrong tallying. Similar to coercion, preventing from vote buying is also another important problem. This can also be done personally or a group's votes can be directed to a specific candidate. Integrity is also need to be ensured while casting a vote, and voter must be sure whether if her vote has been stored, transported and tallied correctly. If there exist some voters who think that their votes are being tallied dishonestly, then their confidence to the ruling authority will diminish. Vote privacy is also related with the voters freewill; without privacy, voters can be divided into groups or maybe some of them can be alienated from the community. Therefore, voting should be thought as not only an election but also a chance for a voter to use her own will and to feel as a member of a community. If no precautions can be taken against those crucial problems, it cannot then be stated to have made an election at the end.

In our i-voting protocol, we tried to take some precautions against those problems. In addition the problems stated above, elections in most of the countries have a concrete date (one day). Most of the time, voters have to cast their vote in a specific place during the elections. Thus, in classical election method, there can be rush in airports, terminals in which people are trying to go to the place where they cast their votes. Some people have to stop their holidays or change their places for a specific time interval (during elections) which lead to a loss of time and money. Moreover, obligation to cast vote in a certain place decreases the percentage of people who casts their vote. This can be seen as a side effect of this problem. Therefore, i-voting scheme not only makes people's lives easier, but also increases voter participation.

One other key subject that has to be discussed here is receipt freeness. Receipt freeness is the case where voter cannot prove some other people that she casts her vote to a certain party or a candidate. If she cannot prove her election preference to the vote buyer then there does not exist other way to convince

**TUBITAK BILGEM UEKAE, Kocaeli, Turkey. * Department of Cryptography, Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey.

E-Mails:ahmet.sinak@metu.edu.tr, mehmet.kiraz@tubitak.gov.tr, {seilzkan, hakyildirim}@gmail.com

vote buyer. Receipt freeness is a crucial problem for i-voting systems since it is hard to construct a system which satisfies this property.

A. Contributions

The main concern of this paper is robustness. Our aim is to present a secure, almost end-2-end Internet voting system. There are four motivations of this paper. The first one is to simplify *receipt code* based paper. In the Norwegian system, receipt codes are constructed by a receipt code generator. This generator enables that if two parties in the system cooperate, they can obtain private key of election. We try to exclude a possible cooperation of them. In our protocol, even if they cooperate, the system ensures the privacy of vote. In order to prevent this problem, we use the hash values of the encrypted vote instead of their receipt code algorithm. Next motivation is to use *voter secret number* to protect any type of coercion. We propose to use secret number of the voter while casting a vote. When the entered secret number is right, the vote is tallied; otherwise, it will be canceled. Namely, if the voter is under coercion, he can enter a wrong secret number. A secret number is generated for each voter and they are sent to voters by SMS before the election. Third one is to use *Bulletin Board* in our protocol. The voter can verify whether his vote reaches to the counting process from Bulletin Board. It increases the credibility of the our protocol. Finally, our protocol is *new and almost End-2-End Secure*. The design approach of our system enables the easy usage with considering all the security and privacy conditions. Moreover, our i-voting solution is almost end-2-end secure, in which the system's security is considered from the initial point of system to the final point. In our i-voting system, the voter can verify that her vote was counted as cast.

B. Organization of the Paper

The rest of this paper is organized as follows. Section II describes the basic algorithms and definitions which are used in our protocol. In Section III, we explain the main infrastructures of our protocol. Key generation is also explained in this section. Section IV presents the vote submission and counting processes in our protocol. Section V evaluates security issue. Section VI gives summary of our work. Finally, our protocol is depicted in Appendix.

II. Preliminaries

In this section, we present fundamental definitions and algorithms used in our protocol. In general, e-voting solutions can be stated as a computer system which uses pure mathematics and algorithms. The following algorithms are required for our protocol. For the completeness of the paper, we summarized the underlying cryptographic primitives.

A. Discrete Log Assumption

Let $G =< g >$ be a cyclic group of prime order p (p is a very large prime). Discrete Log Assumption states that it is hard to calculate x by using the generator g and the random group element $y := g^x$.

$$G = \langle g \rangle = \left\{ g^0, g^1, \cdots, g^{p-1} \right\}$$

B. ElGamal Encryption Scheme

ElGamal encryption is used to encrypt the votes in our protocol. ElGamal signature is also used to sign the votes in our protocol. The ElGamal scheme [12] is a public-key cryptographic system based on the discrete logarithm problem. It consists of both encryption and signature algorithms. The encryption algorithm is similar in nature to the Diffie-Hellman key agreement protocol. Security of the ElGamal algorithm depends on the difficulty of computing discrete logs in a large prime number. ElGamal has the disadvantage that the ciphertext is twice as long as the plaintext. It has an advantage such that the same plaintext gives a different ciphertext each time it is encrypted (because of randomized encryption).

1) Homomorphic Encryption: Homomorphic encryption is an encryption scheme with a special property that allows operations applied to ciphertext be preserved and carried over to the plaintext. Let M be the set of plaintexts, C be the set of ciphertexts, and K be the set of keys. An encryption scheme is said to be homomorphic if for any given encryption key $k \in K$ the encryption function E satisfies:

$$E_k(m_1 \odot_M m_2) = E_k(m_1) \odot_C E_k(m_2), \forall m_1; m_2 \in M$$

C. Commitment Schemes

Commitments are the important part of the cryptographic protocols. Damgard explains the commitment schemes clearly in [3]. Informally speaking, a commitment scheme consists of commit and reveal phases between two parties, called the sender and the receiver. In many cases, the protocols commit and reveal can be done in terms of a single algorithm, requiring no interaction between the sender and receiver at all.

Pedersen Commitment Scheme: Pedersen introduced a commitment scheme in 1992 [6]. Receiver chooses large primes p and q such that q divides p^{n-1}, generator g of the order-q subgroup of $Z^*_{p^n}$, he selects a random secret x from Z_q, $h = g^x \ mod \ p$. p, q, g, h are public, x is secret.

- Commit: To commit to some $m \in Z_q$, sender chooses random $r \in_R Z_q$ and sends $c = g^m h^r \ mod \ p^n$ to the receiver.
- Reveal: To open the commitment, sender reveals m and r, receiver verifies that $c = g^m h^r \ mod \ p^n$.

D. Threshold Cryptography

In cryptography, threshold is meant to distribute basic cryptographic schemes between a group of participant [9]. Each distributed part can be an algorithm or a key. Secret sharing is the main basis of threshold cryptography. In a *secret sharing scheme* there exist a dealer D and participants $P_1, \cdots, P_n$.

In a $(t, n)-$threshold cryptosystem there are n participants in total. If in order to decrypt an encrypted message a number of parties exceeding a threshold is required to cooperate in the decryption protocol. The message is encrypted using a public

key and the corresponding private key is shared among the participating parties. Let n be the number of parties. Such a system is called $(t, n)-$threshold, if at least t of these parties can efficiently decrypt the ciphertext, while less than t have no useful information.

E. Zero knowledge Proofs

Zero knowledge proofs are a type of proof systems which is first introduced by Goldwasser, Micali and Rackoff [11]. In the zero knowledge proofs there are two players, a prover (P) and a verifier (V). The aim is that a prover convinces a verifier of the truth of the argument without revealing any information about the statement.

III. Models

In this section, we first give the components used in our protocol. In the next step, how the private key of the election and the other secret values are generated will be explained. Finally, the main infrastructures of our protocol will be described.

A. Components

The parameters used in this paper are presented in the Table I.

TABLE I
COMPONENTS

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

B. Key Generation

Distributed key generation is the main component of the threshold cryptosystems. There are many solutions to the distributed generation of private keys. Here in our solution, private key will be created by the shares of parties that may be a governmental institution, university or political party which are independent from each other. In our solution we have t of them and any number of institutions less than t cooperating together cannot calculate the private key. The threshold scheme is perfect if knowledge of $t-1$ or fewer shares give no information regarding the private key. In our protocol, the key x is the private key of election whose shares is only known by the trusted parties. the corresponding key $h = g^x mod\ p$ is the public key of election.

Moreover, in our protocol, the secret parameters a_1 belongs to AB, a_2 belongs to CB and a_3 belongs to C are generated by CA which satisfy $a_1 + a_2 = a_3^{-1}\ mod\ p-1$ (a_3 is invertible in $mod\ p-1$). The aim of using this parameters in our protocol is to increase the privacy and security of the system. Moreover, these parameters make the link among the components AB, CB and C. It should be noted that each voter V has a secret number n. This number will be required by the system while casting a vote. The purpose of using this number is to prevent coercion. In our protocol, all secret values are committed by using *Pedersen commitment scheme*.

C. Main infrastructures of our protocol

Before we describe our protocol we would like to highlight the main infrastructures of our system.

- **Voter's Secret Number:** As mentioned before, coercion is one of the most important problem for Internet voting. To overcome this problem, we consider that a voter who is under coercion should use a secret number which is specific for each voter. When voter wants to cast her vote, the system requires her secret number. If voter enters the correct secret number, the vote will be tallied correctly. However, if the entered secret number is not correct, the system will ignore her vote to be tallied without giving a warning to the voter. In that case, the coercer will not notice whether her vote will be tallied.
  Each voter's secret number will be sent to her mobile phone via SMS. Let's assume that coercer have voter's mobile phone for a short time. In this case, there exist solution to satisfy secrecy of her secret number. This number will be sent to voters at an unpredictable time (e.g., anytime in a month) before the election. Coercer has to keep the mobile phones for a long time which is not realistic. On the other hand, the coercer may want from the voter to show her secret number from the phone. To show a fake secret number to the coercer, the voter can demand a new secret number after she receives the first correct secret number. The system will send a new fake secret number. Without legal recourse, always the first secret number will be valid and used in the tallying process; however, fake ones will ignore votes from tallying.
- **Receipt Code:** In the Norwegian system, if the ballot box B and the receipt generator R cooperate, they can obtain private key of election. We consider that, this assumption is strong and should be excluded from an Internet voting system. In our protocol, even if any of the components in the system cooperate, the privacy of the voters will still be ensured. In order to prevent this problem in the Norwegian system, we use the hash values of the encrypted vote instead of their receipt code. That is, a receipt code is the list of hash values of all possible cast-votes. We also note that since every voter has different random number in ElGamal scheme, their encrypted votes and their hash values will be different. Therefore, nobody can understand the others vote knowing the hash value since any two voter casting the same candidate have different receipt values.
  Before the election, High Election Board (HEB) produces a receipt code paper for each voter. Receipt code paper consists of hash values of all possible encrypted votes for a random number. The random number is generated uniquely for each voter, and is sent to the voters inside the receipt code paper. However, it should be noted that the link between voter and her random number have to be unknown in order to satisfy privacy of the voter. Therefore, these papers will be in a sealed envelope and should be anonymously sent to the voters. Before the election period, these papers are delivered to the election

precincts in uniformly random order. During the election period, voters can get randomly one of them from their election precinct or postal services. Thus, nobody knows the link between voter and her random number. For example, let's assume that HEB sends the predetermined random number in receipt code paper to the specific election precinct. Suppose that there exists about 100 voters at a region and therefore about 100 receipt code papers will exist. A voter can select randomly one of the papers. The probability of identifying the link between specific voter and specific random number will be about $\frac{1}{100}$ which is ignorable probability.
We highlight that the verification of the correctness of the system using the receipt code papers is not compulsory, about 5% of the voters will be sufficient in order to guarantee the correctness of the system. For example, voter who trusts the system can cast her vote without verification using the receipt code paper. On the other hand, one who does not trust the system may want to do this checking operation.

- **Bulletin Board (BB):** BB is online and is part of the verification mechanism. After the vote is arrived to the tallying process, the hash value is sent to the voter with SMS by BB. In order to protect coercion, voters are informed about their votes even if it is not counted. There may be some delay here since this process is done periodically. Moreover, voters can verify whether casted votes are passed all the encryption and masking processes over the system without alteration from the BB. That is, voter can check whether her vote is arrived to the tallying procedure. In the BB, the query is done by the hash value and ID number of the voter. When voter makes a query in the system and if any record is found, the record include hash value of the encrypted vote and a commitment of the voter's secret number. That is, the voter can see her ID number, the commitment of the secret number and hash value of the encrypted vote in the BB. Wrong information in the BB guarantees to the voter that at least one of the system component is compromised.
  Let's assume that there exists a coercer who can see the casted-votes in the BB. Here, there might be a possible attack scenario that any coercer can check the receipt codes of the voter from the BB one by one. To solve this problem, the voter casts all the candidates from the list. Therefore, if coercer wants to check from the BB, he will see all the possible voting cases, but he never learns which is the correct choice of the voter.

IV. The main protocol

A. General Structure

Voting system is based on web which can be in one of the portals of government. It is assumed that this portal is authenticating citizens by their electronic ID cards or by user ID and passwords. To sign the vote, every voter needs her ID card during the i-voting process since the private key is identified to the citizen's ID-Card. The corresponding public key with owner ID are sent to AB module in the system in order to verify the voter. Before the election, private and public key of the election is generated (one can use distributed key generation protocol where each party receives a share of the private key and all parties learn public key of the election [1]). Receipt code paper which consists of receipt codes (hash values of encrypted votes) of the all possible votes and used random number will be sent to the voter or to the election precinct in a independent safe channel at the same time. Voters who wish can do the offline checking operation by using receipt code paper. In order to protect voters from coercion we not only give opportunity to voters to cast their votes multiple times, but also we protect them by giving each voter a secret number n. This secret number and the commitment of it are sent to the voter with SMS. The commitment of the secret number is transported with other values while casting the vote. On the other hand, both commitment of secret number and ID number of each voter are given to C so that secret number is checked in the tallying process by C. If the entered secret number is correct the vote is tallied, if it is wrong, it is canceled.

B. Vote Submission

The voter casts her vote as follows: We are now ready to describe our vote submission and counting processes. The voter casts a vote v over a PC or a TC where she is authenticated with her electronic ID (e-ID) card. During the following procedures, you can derive benefit from diagram of our protocol in Figure 1 in Appendix.

- *Online process:*

1) Voter authenticates herself to the voting system using her ID card. She selects her candidate and inputs her secret number n.
2) First, computer encrypts vote $E(v,r) = (g^r, g^v h^r) = c$ where r is the random number, $h = g^x$ public key of the election and x is the private key of the election. In addition, it calculates hash of encrypted vote, $h := H(c)$, and signs this hash using the private key of voter, $Sign := Sign(h)$. Computer sends the sign, encrypted vote, users ID number, zero knowledge proof (ZK_{PC}) of it's own computation and commitment of secret number $(\alpha = commit(r,n))$ to AB.
   $PC \xrightarrow{Sign,c,ID,ZK_{PC},\alpha} AB.$
3) AB verifies $Sign \rightarrow c$ using the public key of the voter and check the proof ZK_{PC}. AB gives a unique number u to each vote so that the next module A can match vote pairs coming from AB and CB. It also adds a sequence number s to each voter's vote in order to determine the final vote in C. (Actually, this sequence number shows how many times does voter cast a vote).
4) AB sends the zero knowledge of it's own computations (ZK_{AB}), u, s and all the data received from PC except *Sign* to CB.
   $AB \xrightarrow{c,ID,ZK_{PC},ZK_{AB},\alpha,s,u} CB.$
5) CB checks the proofs ZK_{PC}, ZK_{AB}. It computes hash of encrypted vote and sends it to the V by using a safe

channel (like SMS). CB also sends this hash data to PC.
$CB \xrightarrow{h} V$ and PC.

6) The received hash value should be checked by PC. Moreover, voter has opportunity to check the received hash from her receipt code paper. By doing those controls the voter will be sure whether her vote is received by CB without any alteration or not. If the received hash value is true, it will be sure that the vote is transmitted by AB truly and the vote is valid; otherwise, she can see that whether her computer is compromised or any alteration is occurred in the system and the vote will be canceled.
7) If the vote is valid, AB and CB are masking the encrypted vote c with a_1, c^{a_1}, and a_2, c^{a_2}, respectively. These keys check that none of AB, CB and A produces any vote using the voters information. These masking operations also ensure that every vote pass over AB and CB. AB send the masked encrypted vote, bullet's unique number, bullet's sequence number, ZK_{AB} and the received data from PC except *Sign* and c to A.
$AB \xrightarrow{c^{a_1},ID,ZK_{PC},ZK_{AB},\alpha,s,u} A$
CB also sends the masked encrypted vote, zero knowledge proof of it's own computation (ZK_{CB}) and the received data from AB except c to A.
$CB \xrightarrow{c^{a_2},ID,ZK_{PC},ZK_{AB},ZK_{CB},\alpha,s,u} A$.
8) A matches vote pairs by the unique number u and multiplies the pair of masked encrypted votes coming from AB and CB.

$$c^{a_1} \cdot c^{a_2} = E(v^{a_1}; r \cdot a_1) \cdot E(v^{a_2}; r \cdot a_2) = E(v^{a_1+a_2}; r \cdot (a_1 + a_2)) = (g^r, g^v h^r)^{a_1+a_2} = c^{a_1+a_2}$$

9) A verifies the proofs ZK_{PC}, ZK_{AB} and ZK_{CB}. It stores all the received data in a database.

- *Offline process:*

10) The masked data $c^{a_1+a_2}$, ZK_A and received data except c^{a_1}, c^{a_2} and u are periodically offline exported to C.
$A \xrightarrow{c^{a_1+a_2},ID,ZK_{PC},ZK_{AB},ZK_{CB},ZK_A,\alpha,s} C$
11) C verifies the proofs ZK_{PC}, ZK_{AB}, ZK_{CB} and ZK_A and decrypts $c^{a_1+a_2}$ by using a_3.

$$(c^{a_1+a_2})^{a_3} mod\ p \rightarrow (c^{a_3^{-1}})^{a_3} = c\ mod\ p.$$

12) Next, C checks the commitment of the secret number, α; If it is correct, the vote is valid; otherwise, invalid.
13) For each vote, the value h is calculated by C and sent to the BB with voter's ID and α.
$C \xrightarrow{h,ID,\alpha} BB$
The data transfer from C to BB is done by using an external disc.

- *Online process:*

14) BB sends h to the voter by SMS in order to guarantee her vote comes to the counter process. Each voter can also check her vote whether her vote is in the BB. BB is listing all the votes (It is not matter whether if voter inputs right or wrong secret number to the system) cast by the voter with the commitment of the secret number. In BB, the query is done by the hash of the encrypted vote and ID number of the voter. Each voter expects hash of her encrypted vote in BB.

C. Counting Process

15) When the election is over, C permutes the list of valid final votes (votes received which have the greatest sequence number and correct secret number). By using the homomorphic property of ElGamal encryption, it gets l of them randomly (This limit is done because of the discrete logarithm problem) and multiply them:

$$E(v_1; r_1) * E(v_2; r_2) * \cdots * E(v_l; r_l) = E(v_1 + v_2 + \cdots + v_l; r_1 + r_2 + \cdots r_l)$$

16) The bunch votes are sent to DS which is far away from C by using a one way filter which does not let any traffic from DS to C. Thus, private key of the election is kept away from the encrypted votes.
$C \xrightarrow{E(v_1+v_2+\cdots+v_l;r_1+r_2+\cdots r_l)} DS$
17) At least t trusted parties in DS gather and decrypt the encrypted results:
DS has $D_x(E(v_1 + v_2 + \cdots + v_l; r_1 + r_2 + \cdots r_l)) = g^{v_1+v_2+\cdots+v_l}$.
18) Finally, election results will be announced.

V. Security Analysis

We now show that our protocol is secure by considering the correctness our system and privacy of voters in for each corrupted party.

Theorem V.1. *Our protocol is correct in the case that PC is corrupted.*

Proof: If PC is corrupted, it can either change the vote or the secret number. If the vote is changed, hash value of the encrypted vote will also be changed. However, the receipt code sent by CB can help the voter to detect whether her vote has been changed by PC. Furthermore, if secret number is changed, the voter can also detect this case during the search from BB using her ID. Namely, during the BB search, the voter can see her vote in two columns. First column will be the hash of her vote and the second one will be commitment of her secret number. Any alteration in this information shows that her vote has been compromised. ■

Theorem V.2. *Our protocol is correct in the case that AB is corrupted.*

Proof: A compromised AB can generate a vote on behalf of a voter. In that case, this fake vote is sent to CB. CB then sends a notification (e.g., SMS) to the voter. If a voter receives suddenly a SMS although she do not vote, the voter notice that there is an attack to the system and a fake vote is casted. Therefore, the notification sent by CB prevents a possible attack by AB. ■

Theorem V.3. *Our protocol is correct in the case that CB is corrupted.*

Proof: Assume that a corrupted CB generates a fake vote and masks it to send to A. A checks the unique number of generated-vote coming from AB. If there is no match, the generated-vote will be canceled. Masking process is performed to protect the system from a corrupted CB. Furthermore, ZK_{CB} proofs are also verified until the counting process. ■

Theorem V.4. *Our protocol is correct in the case that AB and CB are corrupted and cooperate.*

Proof: If AB and CB are compromised and cooperate, they generate a fake vote. Note that their fake generated-vote will be listed in BB. However, the voter notices the malicious cooperation since BB also sends the hash of encrypted vote via SMS to the voter. If a voter receives an SMS from BB where she do not vote, it shows that AB and CB are compromised. In our system, they can not obtain the private key of election and therefore, the attack will not successful. Furthermore, note that a_1 and a_2 are masking keys in the system. These keys verify that none of AB, CB and A produces any vote using the voters information. These masking operations also ensure that every votes pass over AB and CB. ■

Theorem V.5. *Our protocol is correct in the case that PC and BB are corrupted and cooperate.*

Proof: When PC and BB are compromised and cooperate they can agree on canceling the valid votes by changing secret numbers. Assume that PC changes the correct secret number, in this case the valid-vote will not be tallied, and BB shows the commitment of the correct secret number. Therefore, the voter cannot understand whether her vote is canceled. However, ZK-proofs protect the voter from these types of attack. Receipt freeness and privacy is also satisfied by the underlying additive homomorphic encryption method. In our system the votes are encrypted at the PC or TC and transported in encrypted form. After vote submission, voter can check hash of encrypted vote from the bulletin board. After votes have been tallied, in order to announce the results, only the result votes (homomorphically added version of l votes) are decrypted. Therefore, at each step there is no authority who can read the voter's will(privacy) and voter can not have chance to show his will to anybody(receipt freeness). ■

VI. COMPLEXITY

Our i-voting protocol contains sub-protocols between the components. To analyse the complexity of the system, all the sub-protocol steps should be examined separately. Assume that the order of cyclic group G is p, the number of submitted votes is n and the number of votes in a bunch is l. Also assume that all submitted votes are valid.

The complexity of each component: The computational cost of our protocol can be described as follows. Note that we only count the expensive asymmetric operations since symmetric encryptions and hash functions can be ignored.

- In PC; Vote is encrypted by using ElGamal encryption which costs 3 modular exponentiations and 1 modular multiplication for each submission. Each encrypted vote is hashed by a hash function which is a linear function. Hash value of the encrypted vote is signed with ElGamal which costs 1 modular exponentiation and 2 modular multiplications for each submission.
- In AB; Signature is verified, which costs 3 modular exponentiations. The encrypted vote is masked with a_1, which costs 1 modular exponentiation.
- In CB; Encrypted vote is hashed and masked with a_2, which costs 1 modular exponentiation.
- In A; For all vote pairs, 1 multiplication occurs since they are multiplied.
- In C; The masking process is repeated with a_3, which costs 1 modular exponentiation. Until this point, every multiplication and exponentiation occurs for each vote submission. After this point, l votes are grouped and multiplied which costs $(l-1)$ modular multiplications. This process is repeated $\frac{n}{l}$ times. It costs approximately n modular multiplications.
- In DS; The vote bunches are decrypted. In this part, there exists $\frac{n}{l}$ public key decryption. Each decryption process costs 1 modular exponentiation, 1 modular inversion and 1 modular multiplication.

As a result, for each submitted vote, 10 modular exponentiations and 4 modular multiplications are used in the encryption process. Also in counter process, n modular multiplications occur. Additionally, in decryption process, $\frac{n}{l}$ modular exponentiations, $\frac{n}{l}$ modular inversions and $\frac{n}{l}$ modular multiplications occur. We note that the complexities of the modular multiplication, inversion and exponentiation in Z_p are $O(log^2 p)$, $O(log^3 p)$ and $O(log^3 p)$, respectively. Therefore, time complexity of the system is approximately;

$$O((10n + \frac{2n}{l})log^3 p) + O((5n + \frac{n}{l})log^2 p) \approx O(n \cdot log^3 p)$$

Also in the system, 4 zero-knowledge proofs and 15 zero-knowledge proof verifications are used.

VII. CONCLUSION

Internet voting is a growing trend for the countries which are interested in increased voter participation both in the country and overseas. Estonia, Norway and France are a few practical examples that used Internet voting in real elections. In this paper, we construct a new and efficient end-2-end secure Internet voting system. In our protocol, by using the homomorphic property of ElGamal, we gathered, transported and tallied all votes in encrypted form so that no attacker has a chance to see the wish of people. Voters sign their votes by their private key and that made possible for the system to check whether if vote comes from the real voter or not. We prevent voters from coercion with the help of secret number generated before the elections. It is crucial to construct a voting system in which security, transparency, privacy and receipt freeness is satisfied.

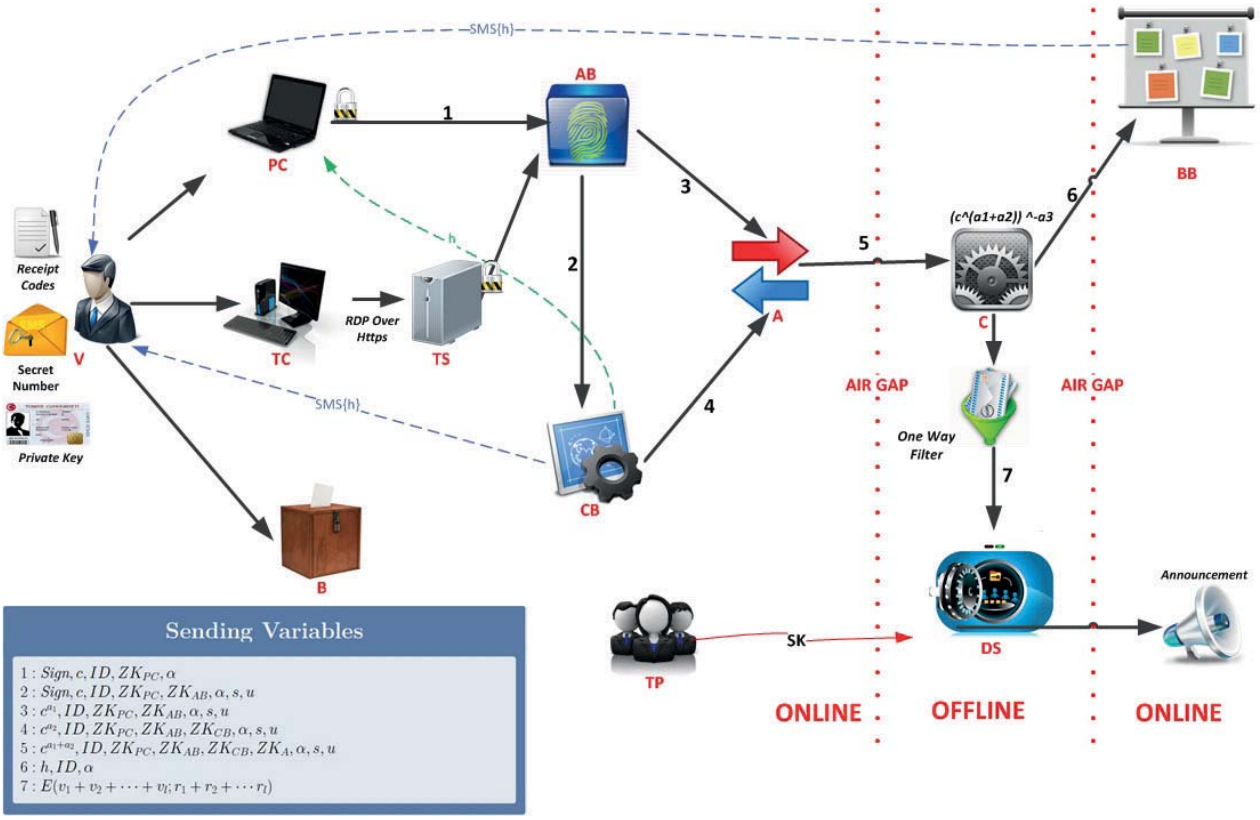

Fig. 1. Our Internet Voting Protocol.

ACKNOWLEDGMENT

Preliminary version of this study is presented as poster at CryptoDays conference, Tubitak, Turkey (June 2013).

APPENDIX

Our protocol is depicted in Figure 1.

REFERENCES

[1] Berry Schoenmakers, *Lecture Notes over Cryptographic Protocols*, Febr. 2013, http://www.win.tue.nl/ berry/2WC13/LectureNotes.pdf

[2] Felix Brandt, *Efficient Cryptographic Protocol Design Based on Distributed ElGamal Encryption*, Springer-Verlag, LNCS, Vol.3935, pp.32-47, 2006.

[3] Ivan Damgard, *Commitment Schemes and Zero-Knowledge Protocols*, Springer-Verlag, LNCS, Vol.1561, pp.63-86, 1999.

[4] Kristian Gjosteen, *Analysis of an Internet Voting Protocol*, March 9, 2010, http://eprint.iacr.org/2010/380.pdf

[5] Martin Hirt and Kazue Sako, *Efficient Receipt-Free Voting Based on Homomorphic Encryption*, Springer-Verlag, LNCS, Vol.1807, pp.539-556, 2000.

[6] Marianne Wiik Oberg, *Improving the Norvegian Internet Voting Protocol*, June 2011, http://daim.idi.ntnu.no/masteroppgaver/005/5823/

[7] Muhammed Ali Bingol and Fatih Birinci and Suleyman Kardas and Mehmet Sabir Kiraz, *Norwegian Internet Voting Protocol Revisited: Security and Privacy Enhancements*, International Conference BulCrypt, Sofia, Bulgaria, September 2012.

[8] P.L.Sven Heiberg and J. Willemson, *On Applying i-voting for Estonian Parliamentary elections*, Springer-Verlag, LNCS, Vol.7187, pp.208-223, 2012.

[9] Ran Canetti , Shafi Goldwasser, *An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack*, Springer-Verlag, LNCS, Vol.1592, pp.90-106, 1999.

[10] Shai Halevi,*Efficient Commitment Schemes with Bounded Sender and Unbounded Receiver*, Springer-Verlag, LNCS, Vol.12, No.2, pp.77-89, 1999.

[11] S. Goldwasser, S. Micali and C. Rackoff, *The Knowledge Complexity of Interactive Proof Systems*, SIAM Journal on Computing, Vol.18, No.1, pp.186 - 208, 1989.

[12] T. ElGamal,*A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory, Vol.31, No.4, pp.469–472, 1985.