

A Non-interactive Zero Knowledge Proof Protocol in an Internet Voting Scheme

Md. Abdul Based and Stig Fr. Mjølsnes

Department of Telematics
Norwegian University of Science and Technology (NTNU)
Email: {based, sfm}@item.ntnu.no

Abstract

An Internet voting scheme allows voters to cast their votes or ballots over Internet. Several information security issues to deploy an Internet voting system are becoming a popular research topic in recent years. An Internet voting system should satisfy information security requirements which include confidentiality, integrity, anonymity, reliability, verifiability, and unreusability of ballots like paper based traditional election system. Satisfying all these requirements in a system is really a very challenging issue. In this paper, an Internet voting scheme is presented that aims to satisfy authentication of voter, confidentiality and integrity of the ballot, validity of ballot, and counting of valid ballots. Here, a novel and efficient Non-interactive Zero Knowledge Proof (NZKP) protocol is presented to prove the validity of anonymous ballots cast by authenticated voter. Hence, authentication of voter is equally considered in this voting scheme.

1 Introduction

An Internet voting system can be either polling-station based or remote for transmitting ballots to election officials. A polling-station based Internet voting scheme provides physical coercion-free elections whereas remote Internet voting lacks this property. This paper considers remote Internet voting that allows voter to vote wherever he/she is (either at home or office).

In cryptography, a Zero Knowledge Proof (ZKP) technique allows a prover to prove his/her statement to a verifier without revealing the insight of the statement. This ZKP technique is a very useful technique in Internet voting system to prove the validity of a ballot without revealing the value of that ballot. This technique ensures the privacy of the voter for voting over Internet. This research work applies the ZKP ideas to a proposed model of Internet voting. Here, the validity of ballots using zero knowledge proof techniques in non-interactive manner (without interaction between the prover and the verifier) are described.

This paper was presented at the NISK-2009 conference.

The term Internet voting is sometimes used as electronic voting. Though Internet voting is also an electronic voting (e-voting), but an e-voting does not necessarily use Internet for transmitting ballots to the election officials. Some background on Internet Voting or e-voting schemes, and ZKP are presented in Section 2. The term ZKP protocol is briefly introduced in Section 3. This section mainly focuses on the basics of different ZKP techniques. An Internet voting scheme using ZKP technique is presented in Section 4. This section includes voter identification and authentication using smartcard (with fingerprint identification) technology, and non-interactive ZKP technique to prove the validity of the ballot in this Internet voting scheme. This section briefly introduces the counting process of the valid ballots, and also presents the completeness, soundness, and zero-knowledgeness of this non-interactive protocol. This paper concludes with Section 5 where the summary of the work and future plans are discussed.

2 Background and Related Work

Internet voting is an ongoing research topic [1, 2, 3, 4, 6]. The United Kingdom, Netherlands, France, USA have partially implemented e-voting. Estonia implemented Internet voting in 2007. Norway plans to test an Internet voting scheme in 2011, targeting a comprehensive Internet voting system in 2017 for national election. Some research works have been done on zero knowledge proofs and homomorphic cryptosystems in [5, 7, 8, 11, 17, 18, 19, 20]. In [5], the author proposed different protocols for voter initialization and voting employing interactive ZKP techniques in his work. In [7], the author presented a simple and efficient publicly verifiable secret sharing scheme and its application to e-voting based on homomorphic secret sharing. In [11], the author presented a proof technique called Universal Composability (UC). He argued on how UC could be used to evaluate zero knowledge and homomorphic cryptosystem based election systems. A multi-candidate election system based on Paillier's cryptosystem and related ZKP techniques are presented in [17]. They showed that their system is efficient enough for practical election systems.

In [27], an efficient receipt-free voting based on homomorphic encryption is presented. Cramer et al. [18] worked on a secure and optimally efficient multi-authority election scheme based on ElGamal Cryptosystem. Mainly they used homomorphic based encryption and decryption techniques in an ElGamal based cryptosystems. Damgard et al. [19] worked on the theory and implementation of an electronic voting system. They also proposed ElGamal based encryption and ZKP in their voting system. Damgard, and Jurik published a paper [20] on Paillier's public key system with application to e-voting. Their work is also based on homomorphic encryption and ZKP techniques, and they proved the efficiency of the system in their paper. Almost in all of these papers [5, 17, 18, 19, 20, 27], the authors worked on interactive ZKP protocols for e-voting systems.

In this paper we present non-interactive zero knowledge proof technique in our proposed remote Internet voting scheme (described in Section 4) that is based on group signature and digital signature. This non-interactive protocol is a modification of an efficient interactive protocol presented in [5]. Variations of ZKP techniques are published in [9, 10, 21, 22, 23, 24, 25, 26].

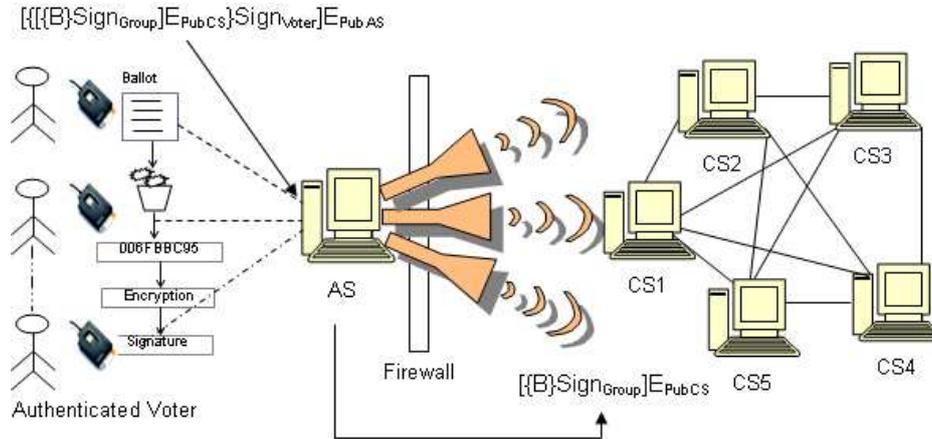


Figure 1: An Internet voting scheme. The voter signs the ballot B using the private key for group signature, and encrypts the ballot using public key of the counting server (CS). Then the voter signs the ballot using the voters own private key, and encrypts the ballot again using the public key of the authentication server (AS). The voter sends this ballot to AS. AS verifies the outer signature of the voter and forwards the encrypted (using the public key of the CS) and signed ballot (using group signature) to the counting servers.

3 A Zero Knowledge Proof (ZKP) Protocol

A Zero Knowledge Proof (ZKP) protocol is a cryptographic technique to prove a statement without revealing the actual proof of that statement. Generally two parties are involved in ZKP protocols. One party, called prover, proves a statement to another party, known as verifier. The Internet voting scheme presented in this paper considers the voter as the prover and the servers (election officials or candidates) as the verifier, where the voter proves the validity of the ballot to the servers.

A ZKP protocol can be classified as either interactive or non-interactive. An interactive ZKP (IZKP) protocol implies that the parties (prover and verifier) must be online to prove the statement. On the other hand, a non-interactive ZKP (NZKP) protocol allows the prover to prove the statement regardless whether the verifier is online or offline. That is, in NZKP protocols, the verifier can verify the statement without online interaction with the prover. Therefore, a NZKP protocol is usually faster and efficient, because this requires no online communication or interaction between the prover and the verifier for proving the statement or claim. On the other hand, a NZKP protocol requires that both the verifier and prover share a random string, usually provided by a trusted third party. Also, a pre-arranged use of this random string is required.

The zero knowledge protocols that are used in this paper for ballot verification and counting are non-interactive. That is, there will be no online interaction between the voter and the ballot Counting Servers (CS), as shown in Figure 1, to prove the validity of the ballot. The Authentication Server (AS) verifies the authenticity of the voter and forwards the encrypted ballots to the counting servers after removing the digital signature of the voter. This process is described in the following section.

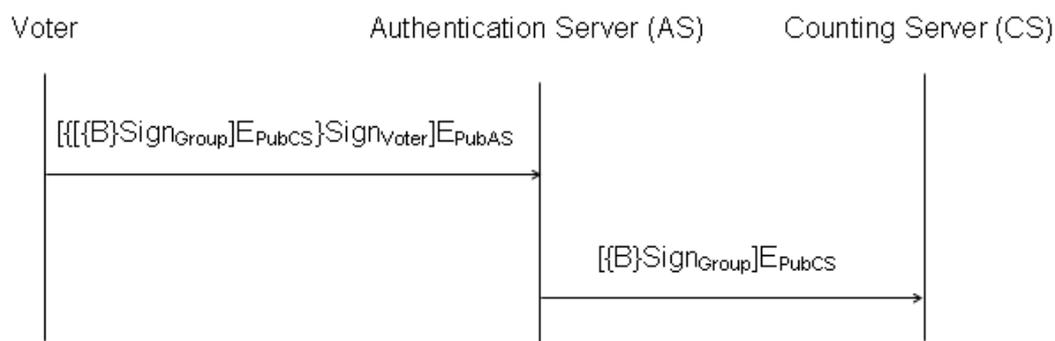


Figure 2: The message sequence diagram between Voter, AS, and CS.

4 A NZKP Protocol in Internet Voting An Internet Voting Scheme

The system structure is shown in Figure 1 and the corresponding message sequence diagram is shown in Figure 2. We assume that the channel between the AS and CS is secured. The scheme assumes that the smartcard is distributed to the voter before the Election Day, and voter list and candidate list are also prepared before that day. The smartcard contains the private key of the voter, the private key of the voter for the group signature (group signature is described in [12, 13, 14, 15, 16]), the public key of the authentication server, the public key of the counting servers, and the fingerprint of the voter. We are assuming a Public Key Infrastructure (PKI) here, without describing the key distribution and key management issues.

Voter authentication will be done by using a smartcard with biometric identification technology, as it is proposed in [6]. The combination of these technologies (smartcard with fingerprint) offers a high degree of user control over personal data. First, the voter will identify himself or herself. After inserting the smartcard into a smartcard reader the voter will supply his/her fingerprint by the fingerprint scanner. The smartcard will verify the fingerprint and if it is ok the voter will get access to the voting web page. The voter and smartcard can communicate using the PKCS#11 libraries provided by the vendor, we will not enter into description how this communication takes place.

The voter will choose a vote/ballot for a particular candidate and will sign the ballot using the private key for the group signature. The voter encrypts the ballot using the public key of the counting server (CS). The voter will then sign the ballot using the private key and encrypt the ballot again using the public key of the server (AS). Then the voter sends this doubly encrypted and signed ballot to the authentication server. This server verifies the signature of the voter and removes this outer signature, and forwards the inner signed and encrypted ballot (encrypted with public key of CS and signed by the private key for group signature) to all the counting servers. The authentication server also sends a notification message to the voter that his/her vote has been sent to the counting servers. The AS also maintains a list of the voters so that the same voter cannot vote again. Each ballot contains a nonce such that no duplicate ballots can be sent by the AS to the CSs. An auditor will observe the activities of the AS so that the AS cannot drop or alter any ballot cast by a voter. Now, for simplicity, we assume that AS is trusted, so it will not

drop, replay or alter any ballot.

In this procedure, two encryptions and two signatures are used to ensure anonymity and integrity of the ballot. As shown in Figure 1, the authentication server (AS) just verifies the signature of the voter and forwards the encrypted and signed ballot to the counting servers, and informs the voter that his/her vote has been accepted. This AS cannot read the ballot, because the ballot is encrypted with the public key of the counting server. The counting servers verify the group signature by using the Group Public Key to see whether the ballot is from a valid voter or not. The CSs can only verify the signature of the group of voters, but, there is no way for these counting servers to know who the voter is (group signature proves the identity of the group, but does not provide any identity of the voter). So this model of Internet voting satisfies both ballot anonymity and integrity.

Ballot Verification using a NZKP Protocol

The interaction in any zero-knowledge proof can be replaced by sharing a common and short random string [25]. We use this idea and modify the interactive ZKP protocol [5] as a non-interactive ZKP protocol to prove the validity of the ballot in our voting scheme. As shown in Figure 1, the voting scheme uses smartcard (with fingerprint) technology for voter identification and authentication. We need some more parameters in this smartcard for ballot verification using this NZKP protocol.

Smartcard. We assume that the smartcard contains $((e, g_1, \text{and } n_1), (e, g_2, \text{and } n_2), \dots, (e, g_j, \text{and } n_j))$ values ($j = \text{number of counting servers or candidates}$) for zero knowledge proof techniques. Here, e is a prime agreed by all the counting servers (each counting server represents a candidate), g_j is an element in $\mathbf{Z}_{n_j}^*$ such that e divides the order of g_j , and $n_j = p_j \cdot q_j$, where p_j and q_j values are chosen by server j such that e divides $(p_j - 1)$, but does not divide $(q_j - 1)$. These p_j and q_j values are obviously private to server j . The value of e should be larger than the total number of eligible voters.

Computations at Voters' Side. The voter randomly picks the values of the string $R = (r_1, r_2, \dots, r_j)$, where

$$\begin{aligned} r_1 &= \sum_{i=1}^j r_{i1} \pmod{e} \\ r_2 &= \sum_{i=1}^j r_{i2} \pmod{e} \\ &\dots \\ &\dots \\ &\dots \\ r_j &= \sum_{i=1}^j r_{ij} \pmod{e} \end{aligned}$$

Here, the random values of r_1, r_2, \dots, r_j are elements of $\mathbf{Z}_{n_1}^*, \mathbf{Z}_{n_2}^*, \dots, \mathbf{Z}_{n_j}^*$ respectively, where j is the total number of candidates or counting servers.

These random values are used to compute the ballot $B = (b_1, b_2, \dots, b_j)$, such that

$$\begin{aligned} b_1 &= (g_1^{r_{11}} \pmod{n_1}, g_2^{r_{12}} \pmod{n_2}, \dots, g_j^{r_{1j}} \pmod{n_j}) \\ b_2 &= (g_1^{r_{21}} \pmod{n_1}, g_2^{r_{22}} \pmod{n_2}, \dots, g_j^{r_{2j}} \pmod{n_j}) \\ &\dots \\ &\dots \end{aligned}$$

$$b_j = (g_1^{r_{j1}} \pmod{n_1}, g_2^{r_{j2}} \pmod{n_2}, \dots, g_j^{r_{jj}} \pmod{n_j})$$

That is, a ballot is sent as shares to j servers such that a vote $V = (v_1, v_2, \dots, v_j)$, where

$$\begin{aligned} v_1 &= \sum_{i=1}^j r_{1i} \pmod{e} \\ v_2 &= \sum_{i=1}^j r_{2i} \pmod{e} \\ &\dots \\ &\dots \\ v_j &= \sum_{i=1}^j r_{ji} \pmod{e} \end{aligned}$$

Each of the ballot shares b_1, b_2, \dots, b_j is valid if all of v_1, v_2, \dots, v_j are either 0 or 1 (mod e). A ballot $B = (b_1, b_2, \dots, b_j)$ is valid if all of b_1, b_2, \dots, b_j are valid, and if $\sum_{i=1}^j v_i = \sum (\sum_{i=1}^j r_{1i}, \sum_{i=1}^j r_{2i}, \dots, \sum_{i=1}^j r_{ji}) = 1 \pmod{e}$. The vote for candidate 1 is v_1 , for candidate 2 is v_2 , and so on. In our voting scheme, we assume that a voter can vote only for one candidate and the candidate with most votes will be elected. So, if a voter wants to vote for candidate 1, the value of v_1 must be 1. If the voter wants to vote for candidate 2, the value of v_2 must be 1, and so on.

For ballot verification, the voter picks $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ and $\beta = (\beta_1, \beta_2, \dots, \beta_k)$, where

$$\begin{aligned} \alpha_1 &= (\alpha_{11}, \alpha_{12}, \dots, \alpha_{1j}) \\ \alpha_2 &= (\alpha_{21}, \alpha_{22}, \dots, \alpha_{2j}) \\ &\dots \\ &\dots \\ \alpha_k &= (\alpha_{k1}, \alpha_{k2}, \dots, \alpha_{kj}) \end{aligned}$$

and

$$\begin{aligned} \beta_1 &= (\beta_{11}, \beta_{12}, \dots, \beta_{1j}) \\ \beta_2 &= (\beta_{21}, \beta_{22}, \dots, \beta_{2j}) \\ &\dots \\ &\dots \\ \beta_k &= (\beta_{k1}, \beta_{k2}, \dots, \beta_{kj}) \end{aligned}$$

Here, k is a security parameter of the system, and $((\alpha_{11}, \alpha_{21}, \dots, \alpha_{k1}), (\beta_{11}, \beta_{21}, \dots, \beta_{k1}))$ are elements of $\mathbf{Z}_{n_1}^*$, $((\alpha_{12}, \alpha_{22}, \dots, \alpha_{k2}), (\beta_{12}, \beta_{22}, \dots, \beta_{k2}))$ are elements of $\mathbf{Z}_{n_2}^*$, and so on. It is important to notice here that the value of each $\alpha_1, \alpha_2, \dots, \alpha_j$ is 0 (mod e), and the value of each $\beta_1, \beta_2, \dots, \beta_j$ is 1 (mod e).

Now, the voter picks a random *bit*, if the $bit_i = 0$ then the voter computes $pair_i = (x_i, y_i)$ else $pair_i = (y_i, x_i)$, where

$$x_i = (g_1^{\alpha_{i1}} \pmod{n_1}, g_2^{\alpha_{i2}} \pmod{n_2}, \dots, g_j^{\alpha_{ij}} \pmod{n_j})$$

and

$$y_i = (g_1^{\beta_{i1}} \pmod{n_1}, g_2^{\beta_{i2}} \pmod{n_2}, \dots, g_j^{\beta_{ij}} \pmod{n_j})$$

Here, $i = 1$ to k . The voter sends the ballot B and these k -pairs to the counting servers. We assume that for non-interactive zero knowledge proof, a trusted third party will provide the k -bit random challenge string c that will be used by both voter and counting server to prove the validity of each ballot [28]. We also consider the case that $[B, \text{ and } pair]$ values have been sent by the voter before the random string is presented to him/her. This implies that the claim is chosen before the random string is presented to the prover [29]. In other words, this indicates that we consider our NZKP protocol as NZK (Non-interactive Zero Knowledge) with Preprocessing according to Feige et al. [30]. That is, in the preliminary stage (before the random string is provided), the voter will send B and $pair$ values to the counting servers, and then, for each ballot the voter will follow the following procedure.

For ballot share b_1 , the voter checks the challenge bit c_i ($i = 1$ to k) of the random string, and answers with d_i ($i = 1$ to k). If $c_i = 0$ then the voter computes $d_i = ((\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ij}), (\beta_{i1}, \beta_{i2}, \dots, \beta_{ij}))$. If $c_i = 1$ then the voter computes $d_i = (r_{11} + \alpha_{i1}, r_{12} + \alpha_{i2}, \dots, r_{1j} + \alpha_{ij})$ if $v_1 = \sum_{i=1}^j r_{1i} = 1 \pmod{e}$. If $v_1 = 0 \pmod{e}$ the voter computes $d_i = (r_{11} + \beta_{i1}, r_{12} + \beta_{i2}, \dots, r_{1j} + \beta_{ij})$. The voter sends this d_i value to the counting server.

Similarly, for ballot share b_2 , the voter checks the challenge bit c_i ($i = 1$ to k) of the random string, and answers with d_i ($i = 1$ to k). If $c_i = 0$ then the voter computes $d_i = ((\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ij}), (\beta_{i1}, \beta_{i2}, \dots, \beta_{ij}))$. If $c_i = 1$ then the voter computes $d_i = (r_{21} + \alpha_{i1}, r_{22} + \alpha_{i2}, \dots, r_{2j} + \alpha_{ij})$ if $v_2 = \sum_{i=1}^j r_{2i} = 1 \pmod{e}$. If $v_2 = 0 \pmod{e}$ the voter computes $d_i = (r_{21} + \beta_{i1}, r_{22} + \beta_{i2}, \dots, r_{2j} + \beta_{ij})$. The voter sends this d_i value to the counting server, and follows the same procedure for each ballot.

Finally, the voter sends the randomly generated string R to the counting servers.

Computations at Servers' Side. The counting servers receive $[B, pair, d, \text{ and } R]$ values from the voter. These servers check the challenge bit c_i ($i = 1$ to k) of the random string. If $c_i = 0$, the servers first check that $d_i[1, 1] + d_i[1, 2] + \dots + d_i[1, j] = 0 \pmod{e}$, and $d_i[2, 1] + d_i[2, 2] + \dots + d_i[2, j] = 1 \pmod{e}$, and then check that $pair_i = ((g_1^{d_i[1,1]}, g_2^{d_i[1,2]}, \dots, g_j^{d_i[1,j]}), (g_1^{d_i[2,1]}, g_2^{d_i[2,2]}, \dots, g_j^{d_i[2,j]}))$ or $((g_1^{d_i[2,1]}, g_2^{d_i[2,2]}, \dots, g_j^{d_i[2,j]}), (g_1^{d_i[1,1]}, g_2^{d_i[1,2]}, \dots, g_j^{d_i[1,j]}))$. If both of these calculations are correct, the servers become sure that $pairs$ were computed correctly.

If $c_i = 1$, the counting servers first check that $(\sum_{m=1}^j d_i[1, m] = 1 \pmod{e}, \sum_{m=1}^j d_i[2, m] = 1 \pmod{e}, \dots, \sum_{m=1}^j d_i[j, m] = 1 \pmod{e})$, and then check that $(g_1^{d_i[1,1]}, g_2^{d_i[2,2]}, \dots, g_j^{d_i[j,j]}) = b_m.pair_i[1]$ or $b_m.pair_i[2]$ ($m = 1$ to j , $i = 1$ to k). Here,

$$\begin{aligned} d_i[1] &= r_{11} + \alpha_{i1}, r_{12} + \alpha_{i2}, \dots, r_{1j} + \alpha_{ij} \text{ OR } r_{11} + \beta_{i1}, r_{12} + \beta_{i2}, \dots, r_{1j} + \beta_{ij} \\ d_i[2] &= r_{21} + \alpha_{i1}, r_{22} + \alpha_{i2}, \dots, r_{2j} + \alpha_{ij} \text{ OR } r_{21} + \beta_{i1}, r_{22} + \beta_{i2}, \dots, r_{2j} + \beta_{ij} \\ &\dots \\ &\dots \\ &\dots \\ d_i[j] &= r_{j1} + \alpha_{i1}, r_{j2} + \alpha_{i2}, \dots, r_{jj} + \alpha_{ij} \text{ OR } r_{j1} + \beta_{i1}, r_{j2} + \beta_{i2}, \dots, r_{jj} + \beta_{ij} \end{aligned}$$

If these calculations are correct then the servers become sure that each of the ballot shares b_1, b_2, \dots, b_j was computed correctly.

Finally, the counting servers check (from the R value) that $\sum_{i=1}^j r_i = 1 \pmod{e}$, and $g_i^{r_i} = \prod_{i=1}^j b_j[i]$. This calculation ensures that only one of b_1, b_2, \dots, b_j is 1, and all others are 0. Thus, the ballot B is validated.

Throughout these computations both at voters' side and servers' side, the validity of a ballot has been proved without revealing anything about a particular choice of a voter in that ballot.

Ballot Counting. After the verification of ballots, the counting servers count the valid ballots only. The counting servers receive ballots (B_1, B_2, \dots, B_p) from p voters where $B_1 = (b_{11}, b_{12}, \dots, b_{1j})$ sent by voter 1, $B_2 = (b_{21}, b_{22}, \dots, b_{2j})$ sent by voter 2, and so on. Thus the net ballot will be $B_p = (\sum_{i=1}^p b_{i1}, \sum_{i=1}^p b_{i2}, \dots, \sum_{i=1}^p b_{ij} \pmod{e})$. The total vote for candidate 1 is $\sum_{l=1}^j (\sum_{i=1}^p r_{1i}[l]) \pmod{e}$, total vote for candidate 2 is $\sum_{l=1}^j (\sum_{i=1}^p r_{2i}[l]) \pmod{e}$, and so on. To compute this tally, the counting servers should publish the sub-tallies. For example, to compute the value of $\sum_{i=1}^j r_{1i}$, r_{11} should be published by the first counting server (CS1), r_{12} should be published by second counting server (CS2), and so on. So, the sub tallies that should be published by server 1 is $(\sum_{i=1}^p b_{i1}[1], \sum_{i=1}^p b_{i2}[1], \dots, \sum_{i=1}^p b_{ij}[1])$, by server 2 is $(\sum_{i=1}^p b_{i1}[2], \sum_{i=1}^p b_{i2}[2], \dots, \sum_{i=1}^p b_{ij}[2])$, and similarly, by server j is $(\sum_{i=1}^p b_{i1}[j], \sum_{i=1}^p b_{i2}[j], \dots, \sum_{i=1}^p b_{ij}[j])$. The counting servers can compute these sub-tallies because they know the factorization of n_i ($i = 1$ to j). After all the servers publish their own sub-tallies, the total tally will be computed. In a threshold homomorphic scheme, when at least the threshold number of servers publish the sub-tallies, then the total tally can be computed [5, 7, 17, 18, 19, 20, 27]. Thus, the system would be fault-tolerant. The counting process is not elaborated further in this paper. A variation of homomorphic cryptosystems for encryption and decryption of ballots have been described in [5, 7, 17, 18, 19, 20]. Using the properties of homomorphic cryptosystems the counting servers can add all encrypted ballots before decryption.

In this NZKP protocol, the computations at the voters' side are dominated by the modular exponentiations, which is of complexity $\mathcal{O}(\log n)$ [5]. So, time complexity for the voter is $\mathcal{O}(kj^2 \log n) = \mathcal{O}(\log n)$, and time complexity for the servers is also $\mathcal{O}(kj^2 \log n) = \mathcal{O}(\log n)$ [5]. This protocol is very efficient in that time-consuming computations can be done offline, and voters' all computations can be done without any interaction with the servers.

Zero-knowledgeness of the NZKP Protocol

The completeness, soundness and zero-knowledgeness of this non-interactive protocol are briefly described in this sub-section.

Completeness. With this property of zero knowledge protocol we assume that both the prover and the verifier will follow the normal protocol. If both parties (in this case the voter and the counting servers) follow the protocol properly, the valid ballot cast by authenticated voter can be accepted by the counting servers with probability one.

Soundness. If at least one of the counting servers is honest (follow the protocol), then with overwhelming probability, a dishonest voter will not be able to cheat by sending an invalid ballot. Overwhelming probability means the probability is $1 - \text{negligible probability}$ [5].

As described earlier, the voter sends a signed and encrypted ballot to the counting servers, the counting servers check the validity of the voter by checking the group signature. The voter also sends d values to the counting servers to prove the validity

of the ballot. All the counting servers are able to test this validity. From the d values, the counting servers check the validity of the pair sent by the voter, and the validity of each ballot b_1, b_2, \dots, b_j . Now, each of b_1, b_2, \dots, b_j will be valid (valid means either 0 or 1) if the voters can answer with d_i for both $c_i = 0$ and $c_i = 1$. If $c_i = 0$, the servers accept the first checks if and only if $d_i[1, 1] + d_i[1, 2] + \dots + d_i[1, j] = 0 \pmod{e}$, and $d_i[2, 1] + d_i[2, 2] + \dots + d_i[2, j] = 1 \pmod{e}$. Similarly, if $c_i = 1$, the servers accept the first check if and only if $(\sum_{m=1}^j d_i[1, m] = 1 \pmod{e}, \sum_{m=1}^j d_i[2, m] = 1 \pmod{e}, \dots, \sum_{m=1}^j d_i[j, m] = 1 \pmod{e})$.

For ballot share b_1 , if the servers accept the second checks for both $c_i = 0$ and $c_i = 1$, then this implies that

$$\begin{aligned} d_i[1, 1] (c_i = 1) &\equiv r_{11} + d_i[1, 1] \pmod{\psi(n)}, \text{ and } c_i = 0 \text{ or } r_{11} + d_i[2, 1] \pmod{\psi(n)}, \\ &\quad \text{and } c_i = 0 \\ d_i[1, 2] (c_i = 1) &\equiv r_{12} + d_i[1, 2] \pmod{\psi(n)}, \text{ and } c_i = 0 \text{ or } r_{12} + d_i[2, 2] \pmod{\psi(n)}, \\ &\quad \text{and } c_i = 0 \\ &\quad \dots \\ &\quad \dots \\ &\quad \dots \\ d_i[1, j] (c_i = 1) &\equiv r_{1j} + d_i[1, j] \pmod{\psi(n)}, \text{ and } c_i = 0 \text{ or } r_{1j} + d_i[2, j] \pmod{\psi(n)}, \\ &\quad \text{and } c_i = 0 \end{aligned}$$

This holds since for two elements f_1 and f_2 in $\mathbf{Z}_{\psi(n)}^*$, if $g^{f_1} = g^{f_2}$, then $f_1 = f_2$ [5]. Here, $\psi(n) = (p - 1)(q - 1)$, when n is a product of two primes p and q .

For $c_i = 1$, $\sum_{m=1}^j d_i[1, m] = 1 \pmod{e}$. This means that for $c_i = 0$,

$$\begin{aligned} (r_{11} + d_i[1, 1], r_{12} + d_i[1, 2], \dots, r_{1j} + d_i[1, j]) &= 1 \pmod{e} \\ \text{or} \\ (r_{11} + d_i[2, 1], r_{12} + d_i[2, 2], \dots, r_{1j} + d_i[2, j]) &= 1 \pmod{e}. \end{aligned}$$

Since $d_i[1, 1] + d_i[1, 2] + \dots + d_i[1, j] = 0 \pmod{e}$, and $d_i[2, 1] + d_i[2, 2] + \dots + d_i[2, j] = 1 \pmod{e}$, this shows that the value of $b_1 = v_1 = 0$ or 1 .

Thus, the servers can check the validity of each ballot. To cheat the servers with this proof the voter has to guess the challenge bit c_i from the random string. The voter can do no better guessing than with probability $1/2$. For k -bit random string the probability is reduced to $\frac{1}{2^k}$.

The validity of the ballot B clearly depends on the validity of each of b_1, b_2, \dots, b_j and the value of R . If each of b_1, b_2, \dots, b_j is valid and $R = \sum_{i=1}^j r_i = 1 \pmod{e}$, then the ballot B is valid. Thus, if at least one server is honest, the server will be able to easily detect the invalid ballot (if there is any invalid ballot cast by a voter). So, there is no way for a dishonest voter to succeed with invalid ballot.

Zero-knowledgeness. Zero-knowledgeness refers that no information whatsoever except the validity of the prover's claim flows to the verifier, in that the verifier's view can be well simulated. In this NZKP protocol, the verifiers view1 = $\{pair_i, c_i, d_i\}$ ($i = 1$ to k) can be simulated by the probabilistic Turing machine M constructed in [5], that on input (B, k) runs in expected polynomial time to construct view2 which is computationally indistinguishable from view1. This NZKP is simpler in that there is no interaction between the prover and the verifier. So, we do not have

to worry about the possible cheating by the verifier to obtain a "more interesting view" [25]. The voters send $[B, pair, R, \text{ and } d]$ values to the counting servers. The ballot B is sent as (b_1, b_2, \dots, b_j) , where j is the number of counting servers. Each of these ballot shares is further divided into shares, for example, $b_1 = (g_1^{r_{11}} \pmod{n_1}, g_2^{r_{12}} \pmod{n_2}, \dots, g_j^{r_{1j}} \pmod{n_j})$, and sent to j counting servers. There is no way for a single counting server to discover the value of the ballot unless the counting servers cooperate. On the other hand, d values can be observed by each and every counting server without cooperation with other counting servers and the counting servers can verify the validity of the ballots individually. After ballot verification, the counting servers jointly count the results. That is, the counting servers receive only the necessary values from the voter to individually prove the validity of the ballot, but the voters do not send any value to these counting servers such that the servers individually can reveal the value of a ballot. This fulfills zero-knowledgeness property of the protocol.

From the above discussion we see that, the NZKP protocol for Internet voting presented in this paper satisfies completeness, soundness, and zero-knowledgeness properties of a zero knowledge protocol.

5 Conclusions and Future Work

Most of the researches on e-voting (based on ZKP and homomorphic cryptosystem) focus on interactive zero knowledge proof protocols. An efficient non-interactive zero knowledge proof based Internet voting scheme is presented in this paper. We have shown that the verifier (server) can verify the validity of the ballot cast by the prover (the voter) without online communication or interaction with the voter. Besides proving the validity of the ballot, this Internet voting scheme provides voter identification and authentication (using smartcard with biometric technology), and anonymity and integrity of the ballot by means of digital signatures (including group signatures). This voting scheme also satisfies all the basic properties of a secure and practical Internet voting system [1]. Thus this Internet voting scheme is a novel scheme for verifying the ballot (using NZKP protocol) and the voter (using digital signatures) without compromising the privacy of the voter and the ballot itself.

The implementation, and evaluation of this Internet voting scheme will be done in the future. One possible theoretical option of analysis is the Universal Composability (UC) technique.

References

- [1] Bo Meng: *Analyzing and Improving Internet Voting Protocol*, Proceedings of the IEEE International Conference on e-Business Engineering, pp. 351-354, ISBN 0-7695-3003-6, IEEE Computer Society, (2007)
- [2] T. Tjøstheim, T. Peacock, and P.Y. A. Ryan: *A model for system-based analysis of voting systems*. Fifteenth International Workshop on Security Protocols, (2007)
- [3] European University Institute, Robert Schuman center for Advanced Studies, Report for the Council of Europe: *Internet Voting in the March 2007 Parliamentary Elections in Estonia*. July 31, (2007)

- [4] Norwegian Ministry of Local Government and regional Development: *Report: Electronic voting- challenges and opportunities*. February, (2006)
- [5] Kenneth R. Iversen: *The Application of Cryptographic Zero-Knowledge Techniques in Computerized Secret Ballot Election Schemes*. Ph.D. dissertation, IDT-report, 1991:3, Norwegian Institute of Technology, February, (1991)
- [6] Md. Abdul Based: *Security Aspects of Internet based Voting*, Proceedings of the International Conference on Telecommunications and Networking (TeNe 08), CISSE 2008, December 5-13, Volume 1: Novel Algorithms and Techniques in Telecommunications and Networking ; Sobh et. al. ISBN: 978-90-481-3661-2, Springer, (2009)
- [7] Berry Schoenmakers: *A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting*. In Advances in Cryptology-CRYPTO 99, Vol. 1966 of Lecture Notes in Computer Science, Springer-Verlag, pp. 148-164, (1999)
- [8] G. Anjan Babau, and Dr. M. Padmavathamma: *OPTIMALLY EFFICIENT MULTI AUTHORITY SECRET BALLOT E-ELECTION SCHEME*. Journal of Theoretical and Applied Information Technology, JATIT (2006)
- [9] Steven G. Krantz: *Zero Knowledge Proofs*, AIM Preprint Series, 2007-46, July 25, (2007)
- [10] H. Shin: *A brief survey of zero-knowledge proofs*, CS-R9232, ISSN 0169-118X, (1992)
- [11] Jens Groth: *Evaluating Security of Voting Schemes in the Universal Composability Framework*. ISBN 978-3-540-22217-0, Springer Berlin/Heidelberg, (2004)
- [12] Xavier Boyen, and Brent Waters: *Compact Group Signatures Without Random Oracles*. EUROCRYPT 2006, LNCS 4004, pp. 427-444, (2006)
- [13] Jan Camenisch, and Markus Michels: *A Group Signature Scheme Based on an RSA-Variant*. BRICS Report Series RS-98-27. ISSN 0909-0878 November, (1998)
- [14] Song Han, Jie Wang, and Wanquan: *An Efficient Identity-Based Group Signature Scheme over Elliptic Curves*. ISBN: 978-3-540-23551-4, Springer Berlin / Heidelberg, pp. 417-429, September 27, (2004)
- [15] D. Chaum, and E. Van Heyst: *Group signatures*. Advances in Cryptography, EUROCRYPT91, Springer-Verlag, Lecture Notes in Computer Science, vol. 547, pp. 257-265, (1991)
- [16] L. Chen, and T.P. Pedersen: *New group signature schemes* . EUROCRYPT95, Springer-Verlag, Lecture Notes in Computer Science, vol. 950, pp. 171-181, (1995)

- [17] Oliver Baudron, Pierre-Alain Fouque, David Pointcheval, Guillaume Poupard, and Jacques Stern: *Practical multi-candidate election scheme*. In PODC 01, pp. 274-283, (2001)
- [18] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers: *A secure and optimally efficient multi-authority election scheme*. In proceedings of EUROCRYPT 97, LNCS series, volume 1233, pp. 103-118, (1997)
- [19] Ivan Damgard, Jens Groth, and Gorm Salomonsen: *The theory and implementation of an electronic voting system*. In D. Gritzalis, editor, Secure Electronic Voting, pp. 77-100. Kluwer Academic Publishers, (2003)
- [20] Ivan Damgard, and Mads J. Jurik: *A generalisation, a simplification and some applications of pailliers probabilistic public-key system*. In 4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001, LNCS series, volume 1992, (2001)
- [21] Jens Groth, Rafail Ostrovsky, and Amit Sahai: *Perfect non-interactive zeroknowledge for NP*. ECCC Report TR05-097, <http://eccc.uni-trier.de/eccc-reports/2005/TR05-097/index.html>, (2005)
- [22] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai: *Robust non-interactive zero knowledge*. In proceedings of CRYPTO 01, LNCS series, volume 2139, pp. 566-598, (2002)
- [23] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano: *Non-interactive zero-knowledge*. A low-randomness characterization of NP. In proceedings of ICALP 99, LNCS series, volume 1644, pp. 271-280, (1999)
- [24] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano: *Noninteractive zero-knowledge*. SIAM Journal of Computation, 20(6):1084-1118, (1991)
- [25] Manuel Blum, Paul Feldman, and Silvio Micali: *Non-interactive zero-knowledge and its applications*. In proceedings of STOC 88, pp. 103-112, (1988)
- [26] Andre Chailloux, Dragos Florin Ciocan, Iordanis Kerenidis and Salil Vadhan: *Interactive and Noninteractive Zero Knowledge are Equivalent in the Help Model*. ISSN: 0302-9743, ISBN: 978-3-540-78523-1, Springer Berlin / Heidelberg, February 26, (2008)
- [27] Martin Hirt, and Kazue Sako: *Efficient Receipt-Free Voting Based on Homomorphic Encryption*. EUROCRYPT 2000, LNCS 1807, pp. 539-556, 2000. Springer-Verlag Berlin Heidelberg (2000)
- [28] Dragos Florin Ciocan, and Salil Vadhan: *Interactive and Noninteractive Zero Knowledge Coincide in the Help Model*. Cryptology ePrint Archive, <http://eprint.iacr.org/2007/389.pdf>, (2007)
- [29] Alfredo De Santis, and Giuseppe Persiano: *Zero-Knowledge Proofs of Knowledge Without Interaction*. In Proceedings of the 33rd Symposium on Foundations of Computer Science 1992, (FOCS '92), Pittsburgh, PA, 24-27 October 1992, pages 427-437, (1992)

- [30] U. Feige, D. Lapidot, and A. Shamir: *Multiple Non-Interactive Zero-Knowledge Proofs Based on a Single Random String*. In proceedings of the 22th Annual Symposium on the Theory of Computing, pp. 308-317, (1990)