Publ. Math. Debrecen
79/3-4 (2011), 479–496
DOI: 10.5486/PMD.2011.5142

A homomorphic encryption-based secure electronic voting scheme

By ANDREA HUSZTI (Debrecen)

Dedicated to Professor Attila Pethő and Professor Kálmán Győry

Abstract. In this paper we propose a homomorphic encryption-based secure electronic voting scheme that is based on [5]. It guarantees eligibility, unreusability, privacy, verifiability and also receipt-freeness and uncoercibility. The scheme can be implemented in a practical environment, since it does not use voting booth or untappable channel, only anonymous channels are applied.

1. Introduction

There is a need for research on secure cryptographic electronic election schemes. Electronic voting systems, compare to traditional paper-based elections, promise that election results will be calculated quickly with less chance of human error and also will reduce costs in a long-term period. CHAUM presented the first e-voting scheme in [4]. Currently three election models are used: the mix-net model, the blind signatures model and the homomorphic encryption model. We briefly describe these.

The mix-net model. CHAUM [4] introduces the concept of a mix-net that is built up from several linked servers called mixes. Each mix randomizes input messages and outputs the permutation of them, such that the input and output

Mathematics Subject Classification: 94A60, 68P25.
Key words and phrases: electronic voting, cryptographic protocols, receipt-freeness, homomorphic encryption.

messages are not linkable to each other. Several schemes based on mix-nets are proposed in the literature ([18], [21], [12]).

The blind signatures model. The concept of blind signatures was introduced by CHAUM [3]. A voting authority authenticates a message, usually an encrypted vote, without knowing the contents. Even if later the (un-blinded) signature is made public, it is impossible to connect the signature to the signing process, *i.e.* to the voter. Schemes based on blind signatures usually use anonymous channels in order to send the un-blinded signature and the encryption of the ballot to a voting authority, assuring the anonymity of the sender. For further schemes see [7], [11], [15], [16], [19].

The homomorphic encryption model. Schemes based on homomorphic encryptions posses property of universal verifiability, while preserve voters' privacy. Let $\mathcal{PT}$ be the plaintext space and $\mathcal{CT}$ the ciphertext space such that $\mathcal{PT}$ is a group under the operation $\oplus$ and $\mathcal{CT}$ is a group under the operation $\otimes$. Let $E_r(m)$ denote encryption of the message m using parameter r. An encryption scheme is $(\otimes, \oplus)$-*homomorphic*, if for given $c_1 = E_{r_1}(m_1)$ and $c_2 = E_{r_2}(m_2)$, there exists an r such that a

$$c_1 \otimes c_2 = E_r(m_1 \oplus m_2).$$

In the election model, proposed by CRAMER *et. al.* [5], a variant of the ELGamal encryption algorithm is applied. Let p, q be large primes such that $q|p-1$, and let G_q a subgroup of $\mathbb{Z}_p^*$ with order q. For this scheme the votes are $m_1 = G$ and $m_0 = 1/G$ (yes/no), where G is a fixed generator of G_q. The secret encryption key is s, randomly chosen by the receiver and the corresponding public key is $h \equiv g^s \mod p$, where g is a generator of G_q. A voter posts a ballot of the form $(x_i, y_i) = (g^\alpha, h^\alpha \cdot G^b)$, where $b \in \{1, -1\}$ and a non-interactive proof of validity. After the deadline the authorities calculate

$$(X, Y) = \left(\prod_{i=1}^{n} x_i \mod p, \prod_{i=1}^{n} y_i \mod p \right)$$

for all valid ballots. Finally, the authorities jointly calculate $W \equiv \frac{Y}{X^s} \mod p$ and get $W \equiv G^T \mod p$, where T is the difference of the yes-votes and no-votes. Since in practice T is not big brute force, Baby step giant step or Pollard rho method might be used to calculate it. Models based on [5] are [13] and [9].

Alternative homomorphic encryption schemes based on PALLIER cryptosystem [17] are proposed cf. [2], [6].

You find a nice, self contained overview about the methods above in [20].

The notions of *receipt-freeness* and *uncoercibility* were introduced by [1]. With receipt-freeness the voter should not be able to prove how he/she has voted even if the voter wants to (e.g. for a reward). In this case the voter colludes with the adversary. With uncoercibility, the coercer should not be able to learn the vote from the voter even if the voter is forced to. Many receipt-free and uncoercible election schemes apply a *voting booth* [1] or an *untappable channel* [15], [16], [21]. An *untappable channel* is a one-way physical apparatus providing perfect secrecy in an information-theoretic sense. It might be achieved either by being physically untappable or by implementing information-theoretic encryption, e.g. a one-time pad. *Voting-booths* besides supplying perfect secrecy allow a voter interactively communicate with an authority. Authors in the literature have pointed out the difficulty of their implementation [14].

The proposed scheme is a homomorphic encryption model based on [5] that is not possessing the property of receipt-freeness or uncoercibility. Lee and Kim in [13] gave a solution for receipt-freeness applying an honest verifier. Hirt and Sako in [9] use an untappable channel to achieve it. Our scheme does not employ a voting booth or an untappable channel, it requires an anonymous return channel [8], which is based on a mix-net approach, hence it can be implemented in practice. It has acceptable performance, four times the computational cost of a basic re-encryption mix-net. We do not suppose the existence of an honest verifier, either.

During the Authorizing stage each voter generates a pseudonym in a way that even the Registry is not able to connect the person to the identification number used during the Vote Cast phase. Voters know before the deadline whether they have casted a valid vote, if a problem occurs the voter can make a claim. Since their encrypted ballot appear on the Bulletin Board and all tallying calculations and results are shown, each voter can verify if his/her vote is considered.

2. Preliminaries

2.1. Requirements. Electronic surveys or elections should possess all the requirements that paper-based elections have, moreover our aim is to achieve more security that traditional ones are able to.

Eligibility. Only eligible voters are allowed to cast votes.

Privacy. All votes remain secret, no one is able to link a vote to the voter, who has casted it. No considerably large coalition of participants not containing the voter himself can gain any information about a voter's vote.

Unreusability. Every eligible voter can cast at most one vote. No one can vote for anyone else.

Fairness. No participants can gain any knowledge about the partial tally during the voting stage, since knowledge of any intermediate result about the election can influence the voters.

Robustness. No participant can disrupt the election. Once a voter cast a vote, no alternation to this vote is permitted. Moreover all valid votes will be counted, whereas all invalid ones will be detected and not counted in the final tally.

Individual verifiability. Each eligible voter is able to verify that his vote was committed as intended and made into the final tally as cast.

Universal verifiability. Any participant or passive observer can check that the election is fair, the final result is exactly the sum of the valid votes.

Receipt-freeness, Uncoercibility. Before the election an adversary may bribe the voter with a demand of casting his favorite vote. This scenario is called vote-buying and receipt-freeness avoids vote-buying. An adversary can also force the voter to cast a particular vote by threatening him. Uncoercibility means coercers cannot menace voters. These requirements should be achieved in a way, that during the election a coercer can observe all public information and communication between the voter and the authorities and can even order the voter how he should behave during the voting process, even supplying him the random bits.

The exact definition of receipt-freeness is quoted from [16]:

Given published information $\mathbb{X}$ (public parameters and information on the bulletin board), adversary $\mathbb{C}$ interactively communicates with a voter V in order to force V to cast $\mathbb{C}$'s favorite vote c^* to an authority $\mathcal{A}$, and finally $\mathbb{C}$ decides whether to accept $View(\mathbb{X} : V)$ or not, and $\mathcal{A}$ decides whether to accept c^* or not. The coercer gets any message from the bulletin board immediately after it is put on the board. $View(\mathbb{X} : V)$ means published information $\mathbb{X}$, c^* and messages that $\mathbb{C}$ receives and sends communicating with V including random bits employed during the voting process.

Definition 2.1. A voting system is receipt-free, if there exists a voter V, such that for any adversary $\mathbb{C}$, voter V can cast $c \quad (c \neq c^*)$ which is accepted by the authority $\mathcal{A}$ under the condition that $View(\mathbb{X} : V)$ is accepted by $\mathbb{C}$.

We suppose that a coercer knows public parameters appearing on the bulletin board, vote c^*, random bits predefined by him and encrypted messages sent by the voter on public channels. Receipt-freeness means $View(\mathbb{X} : V)$ should be prepared in a way that, if a coercer makes all calculations with all the data that

he possesses, then no inaccurate count should turn up. A coercer is not able to monitor each communication channel being used during the voting process, hence encrypted data sent through an anonymous channel is not revealed to him. At the same time a ballot is accepted, if the authority has confirmed all necessary information and validity of ballots.

There are two real-word attacks in [12] enumerated below:

Randomization attack. An attacker coerces a voter to submit randomly formed ballot. In this attack it is not possible to learn what candidate the voter casts a ballot for. The effect of this attack is to cancel the voter's vote with large probability.

Forced-abstention attack. An attacker forces a voter to abstain from voting. This attack happens if an adversary is able to follow who is eligible for voting and who has already voted. Being aware of this knowledge he threatens voters and effectively excludes them from the voting process.

2.2. Participants.

Voters. Let denote voters by $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$. This scheme is designed for small scale elections, hence about few thousands voters participate. Right after the voter has casted his vote he is able to verify whether his vote has been processed or not. We assume that the voter is not observed while casting his vote. Attacks, where a coercer is present or the voter is being recorded by a camera (e.g. cell phone camera) in the moment of voting is outside the scope of this paper.

Candidates. Let define a candidate slate to be an ordered set of n distinct identifiers $\{C_1, C_2, \ldots, C_n\}$, each of which corresponds to a voter choice, typically a candidate or party name.

Registry. Registry denoted by $\mathcal{R}$ is responsible for managing the authorizing stage. It checks voters' eligibility in person, supervising private and public key-generation for voting authorities participating in the election. Besides Registry supervises key-generation, reveals public keys to participants, also sets the necessary parameters for the whole election. We do not suppose that $\mathcal{R}$ is honest, $\mathcal{R}$ might collude with adversaries and divulge information calculated with.

Voting Authorities. Denoted by $\mathcal{A} = \{A_1, A_2, \ldots, A_s\}$. One of the authorities called Verifier Authority (*VA*) manages zero-knowledge proofs of the ballots. *VA* is not expected to be honest. After the voting session has completed, voting authorities tally valid votes. Employees of the voting authorities may also participate

as voters. We suppose, there is at least one authority among them that is honest concerning key generation and message decryption.

Adversaries. Any participant or group of them might be malicious and try to distract the elections or to achieve a favorable voting result even in an illegal way. Voters or even members of the voting authorities may become attackers. An attacker can also be an observer who would threaten or even pay participants to vote in a way he demands it.

2.3. Channels.

Public channel. Participants can send their information via public channels. Attackers are able to tap this information, and the identity of the sender can be traced back. All the messages to the bulletin board are sent through public channels.

Anonymous channel. This channel guarantees the anonymity of the sender. Receiver of the message that has been sent through an anonymous channel does not have any information about the identity of the sender. Especially, anonymous return channels allow two parties even to have a complete conversation, the receiver may reply to the sender. Realization of this channel is described in [8] based on a mix-net approach.

Bulletin board. Bulletin board ($\mathcal{BB}$) is publicly readable. Voters, authorities can write into their section and nobody can modify the content of it.

3. The voting scheme

3.1. Protocol description. The proposed election procedure consists of three distinctive stages: *Authorizing*, *Voting* and *Tallying*.

During the Authorizing stage voters authenticate themselves in person and receive their credentials. All system parameters, sufficient private and public keys are generated. The voter gets his credential in a way that he generates his random reference number, and $\mathcal{R}$ signs it blindly, hence $\mathcal{R}$ cannot connect the credential to the voter. During key-generation $\mathcal{R}$ does not learn anything about other participants' private keys either.

During the Voting stage voters create their ballots. Verifier Authority checks eligibility of the voters and if they have already voted before, following it is verified through a non-interactive zero-knowledge proof whether the encrypted ballots sent by the voters are valid or not. This non-interactive zero-knowledge proof is run for a randomized ballot, hence *VA* does not have any information

about the form of the encrypted ballot. Voters send their ballots and randomized components authorized by the Verifier Authority to the Bulletin Board. If the ballot appearing on $\mathcal{BB}$ is different or missing, then the voter makes a claim and he can cast his vote again.

During the Tallying stage Voting Authorities calculate the multiplication of valid, encrypted ballots on the bulletin board and divide it with the product of randomized components. The final results are decrypted and listed.

3.1.1. *Building Blocks.* The proposed election scheme uses distributed ElGamal public-key cryptosystem. Authorities $(A_1, A_2, \ldots, A_s)$ together, generate public and private keys from key shares and at the end of the voting process they decrypt the encrypted voting result. The following two algorithms describe distributed key generation and the distributed decryption methods. Let P and Q large primes, such that $Q \mid P-1$ and $g \in G_Q$, where G_Q is a subgroup of $\mathbb{Z}_P^*$ with order Q.

Distributed ElGamal Key Generation

Input: P, Q, g

Output: Public key: $h \mod P$, public key shares $h_i \mod P$, private key shares: $K_i \mod Q$

(1) A_i: $K_i \in \mathbb{Z}_Q$, $h_i \equiv g^{K_i} \mod P$

(2) A_i publish $h_i \mod P$ and zero-knowledge proof of knowing $K_i \mod Q$

(3) $\mathcal{R}$ wait until all $h_i \mod P$ are on $\mathcal{BB}$

(4) $\mathcal{R}$ verifies all proofs

(5) $h \equiv \prod_{i=1}^{s} h_i \mod P$ is the public key.

Distributed ElGamal Decryption

Input: P, Q, g, encrypted message: $(a \mod P, b \mod P)$, public key shares: $h_i \mod P$, private key shares: $K_i \mod Q$

Output: message: m

(1) A_i: publish decryption share: $c_i \equiv a^{K_i} \mod P$ and the ZK-proof of equality of DL of $h_i \mod P$ and $c_i \mod P$

(2) $\mathcal{R}$ verifies all proofs

(3) $A \equiv \prod_{i=1}^{s} a_i \mod P$

(4) $m \equiv \frac{b}{A} \mod P$.

During Authorizing Stage and Vote Validation Phase voter V_k generates an identification number that is blindly signed by the corresponding authority $\mathcal{P} \in \{\mathcal{R}, VA\}$, hence the authority is not able to connect the identification number to the voter. Adversary does not learn anything even if he colludes with the

authority $\mathcal{P} \in \{\mathcal{R}, VA\}$. The following algorithm blindly generates a signature for voter V_k's reference number $id_k^{\mathcal{P}}$, where $\mathcal{P} \in \{\mathcal{R}, VA\}$. We assume, that $\mathcal{R}$ and *VA* possess RSA public and secret keys, that might be used for generating and verifying signatures in general.

BlindSigRSA

Input: reference number: $id_k^{\mathcal{P}}$, $(RPK_{\mathcal{P}}, N_{\mathcal{P}})$ RSA public key and modulus of participant $\mathcal{P}$

Output: $(M(id_k^{\mathcal{P}}))^{RSK_{\mathcal{P}}} \mod N_{\mathcal{P}}$, where $RSK_{\mathcal{P}}$ denotes RSA secret key of participant $\mathcal{P}$

(1) V_k: chooses random number: $\varrho_k \in \mathbb{Z}_{N_{\mathcal{P}}}$

(2) $V_k \Rightarrow \mathcal{P}$: $CR_k \equiv M(id_k^{\mathcal{P}}) \cdot \varrho_k^{RPK_{\mathcal{P}}} \mod N_{\mathcal{P}}$

(3) $\mathcal{P} \Rightarrow V_k$:$CR_k^{RSK_{\mathcal{P}}} \mod N_{\mathcal{P}}$

(4) V_k: $(M(id_k^{\mathcal{P}}))^{RSK_{\mathcal{P}}} \equiv \frac{CR_k^{RSK_{\mathcal{P}}}}{\varrho_k} \mod N_{\mathcal{P}}$.

At the end of Vote Validation Phase *VA* authorizes the valid ballots using Meta-ElGamal signature scheme [10] with running

SigGenEG

Input: P, Q, g, message: $m \in G_Q$

Output: signature: $s_m \in \mathbb{Z}_Q$, $R \in \mathbb{Z}_Q$

(1) Chooses random number: $\tilde{k} \in \mathbb{Z}_Q$

(2) $R \equiv g^{\tilde{k}} \mod P$

(3) $R' \equiv (R \mod P) \mod Q$

(4) $m' \equiv (m \mod P) \mod Q$

(5) $s_m \equiv ESK_{VA}^{-1}(m' - \tilde{k} \cdot R') \mod Q$.

SigVerEG

Input: P, Q, g, signature: $s_m \in \mathbb{Z}_Q$, $R \in \mathbb{Z}_Q$, message: m

Output: true, false

(1) $R' \equiv (R \mod P) \mod Q$

(2) $m' \equiv (m \mod P) \mod Q$

(3) Verifies: $(EPK_{VA})^{s_m} \cdot R^{R'} \equiv g^{m'} \mod P$.

During Vote Validation Phase *VA* authorizes a randomized ballot, this way *VA* cannot connect the ballots being processed during Tallying Stage to ballots that he authorized to voters. Voter V_k generates a proof with *ProofGenEG* for his 'pure' ballots from the randomized ballot signatures sent by *VA*. During Vote Cast Phase V_k sends this proof with his ballots to $\mathcal{BB}$ and anyone is able to verify

validity of the ballots with *ProofVerEG* algorithm. *VA* does not learn anything from the values sent to $\mathcal{BB}$: $(\overline{s_m}, \overline{R_m}, \overline{R})$.

ProofGenEG

Input: P, Q, g, signature: $s_m \in \mathbb{Z}_Q, R \in \mathbb{Z}_Q, \tilde{l} \in \mathbb{Z}_Q$
Output: $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$

(1) Chooses random number: $\tilde{v} \in \mathbb{Z}_Q$

(2) $R' \equiv (R \mod P) \mod Q$

(3) $\overline{s_m} \equiv \frac{s_m}{\tilde{l}} \mod Q$

(4) $\overline{R} \equiv R^{\frac{\tilde{v}}{\tilde{l}}} \mod P$

(5) $\overline{T} \equiv \frac{R'}{\tilde{v}} \mod Q$.

ProofVerEG

Input: P, Q, g, $m \in \mathbb{Z}_P$, $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$
Output: true, false

(1) $m' \equiv (m \mod P) \mod Q$

(2) Verifies: $EPK_{VA}^{\overline{s_m}} \cdot \overline{R}^{\overline{T}} \equiv g^{m'} \mod P$.

In the following we discuss each step in more details.

3.1.2. *Authorizing stage.*

(1) Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g a generator element of G_Q. Voting Authorities generate jointly the public and private keys using distributed ElGamal key generation method in a way, that the private key is not divulged, and the public key is output on $\mathcal{BB}$. Public keys are g and $h \equiv g^K \mod P$, where $K \in \mathbb{Z}_Q$ is the corresponding private key.

(2) Registry randomly chooses $v_i \in \mathbb{Z}_Q$, $i = 1, \ldots, n$ elements

$$C_i \equiv g^{v_i} \mod P$$

where C_i represents candidate i from the voter roll and a one-way hash function $M()$ is chosen, v_i, C_i and $M()$ are made public.

(3) Registry sends its RSA public key $(RPK_{\mathcal{R}}, N_{\mathcal{R}})$ to $\mathcal{BB}$.

(4) Verifier Authority generates RSA private (RSK_{VA}) and public keys (RPK_{VA}, N_{VA}) that are being authorized by the Registry, sends the public key to $\mathcal{BB}$.

(5) Verifier Authority calculates ElGamal public and private keys, chooses a random $ESK_{VA} \in \mathbb{Z}_Q$ and

$$EPK_{VA} \equiv g^{ESK_{VA}} \mod P.$$

The private key is ESK_{VA} and the corresponding public key is EPK_{VA}.

(6) Voters show their identification material to the Registry in person, so the adversary cannot simulate the voter during registration. If a voter has the right to vote, a reference number denoted by $id_k^{\mathcal{R}}$ for the voter V_k is generated by V_k and $\mathcal{R}$ as a join random value. Voter V_k and $\mathcal{R}$ runs *BlindSigRSA* algorithm in order to authorize V_k's identification number.

By the end of authorizing stage V_k possesses $id_k^{\mathcal{R}}$ and $(M(id_k^{\mathcal{R}}))^{RSK_R} \mod N_{\mathcal{R}}$. All public keys and parameters are on $\mathcal{BB}$:

$$P, Q, g, h, M(), v_i, C_i, RPK_{VA}, N_{VA}, EPK_{VA}, RPK_R, N_{\mathcal{R}}.$$

However the adversary may observe the signing process or collude with $\mathcal{R}$, still cannot learn anything about V_k's reference number or secret key.

3.1.3. *Voting stage.* The voting stage consists of Vote Validation and Vote Cast phases. Vote Validation phase is a non-interactive zero-knowledge proof based on the idea applied in [5] and [13]. During Vote Validation phase the form of the ballot is proved, i.e. the ElGamal encrypted ballot consists of g^{ϑ} and $h^{\vartheta} \cdot C_i^{(k)}$, where $C_i^{(k)}$ represents candidate i elected by V_k. We note that $C_i^{(k)}$ equals to C_i, that is described before. We use this notation to denote V_k's choice. During the Vote Cast phase the encrypted ballot and the randomized component are sent, that is important for achieving receipt-freeness.

Vote Validation phase

(1) The voter V_k first sends

$$id_k^{\mathcal{R}} \mod N_{\mathcal{R}} \,||\, (M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \mod N_{\mathcal{R}}$$

to *VA*. The Verifier Authority checks if the received credential is authorized by the Registry with $\mathcal{R}$'s public key and whether V_k has voted before. If V_k is eligible for voting *VA* and V_k generates a random value $id_k^{VA} \mod N_{VA}$ that is an identification value used only in vote validation phase, in order to follow if a voter has already run the zero-knowledge proof. Voter V_k initiate *BlindSigRSA* algorithm in order to authorize his identification number and possess $id_k^{VA} \mod N_{VA} || (M(id_k^{VA}))^{RSK_{VA}} \mod N_{VA}$. Since during the authorizing stage, due to the randomization, $id_k^{\mathcal{R}}$ and $(M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \mod N_{\mathcal{R}}$ values are not divulged, no one can connect $id_k^{\mathcal{R}}$ to voter V_k.

(2) V_k sends $id_k^{VA} \mod N_{VA} || (M(id_k^{VA}))^{RSK_{VA}} \mod N_{VA}$ through an anonymous return channel to *VA*. *VA* verifies the signature and if the corresponding voter has not been processed before, sends z_k back through the same channel, where $z_k \in \mathbb{Z}_Q$ random. Since id_k^{VA} signed blindly and anonymous return channel is used, *VA* cannot learn the sender.

(3) V_k chooses a candidate i and the corresponding $C_i^{(k)}$ ($C_i^{(k)} = C_i$) from $\mathcal{BB}$. In order to create his ballot randomly chooses $\alpha_k, \beta_k, \gamma_k \in \mathbb{Z}_Q$ and computes $(G_k, H_k \cdot C_i^{(k)})$ and Y_k where

$$G_k \equiv g^{\alpha_k + \beta_k} \mod P$$

$$H_k \equiv h^{\alpha_k + \beta_k} \mod P$$

$$Y_k \equiv g^{z_k \cdot \gamma_k} \mod P.$$

By randomizing the ballot with β_k, an adversary cannot learn anything from it even if he colludes with *VA*. Y_k plays important role in achieving receipt-freeness.

(4) Following V_k runs a non-interactive zero-knowledge proof to prove that he has constructed the ballot correctly, such that he has chosen the value $C_i^{(k)}$ from the voter roll listed on $\mathcal{BB}$. He chooses $r_j, d_j, w_k \in \mathbb{Z}_Q$ random numbers, where $1 \leq j \leq n$ and $j \neq i$, then calculates

$$(A, B) = (a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n),$$

where

$$a_i \equiv g^{w_k} \mod P,$$

$$b_i \equiv h^{w_k} \mod P,$$

for the elected candidate i and

$$a_j \equiv g^{r_j} \cdot G_k^{d_j} \mod P,$$

$$b_j \equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}} \right)^{d_j} \mod P$$

for all candidates $j \neq i$. We review that $C_i^{(k)} = C_i$.

(5) Further, the voter calculates

$$c_k = M(a_1 || .. || a_n || b_1 || .. || b_n || G_k || H_k \cdot C_i^{(k)} || g || h || id_k^{VA} || (M(id_k^{VA}))^{RSK_{VA}})$$

challenge and

$$(D, R) = (d_1, r_1), (d_2, r_2), \ldots, (d_n, r_n)$$

where for candidate i

$$d_i = c_k - \sum_{j=1, i \neq j}^{n} d_j$$

$$r_i = w_k - (\alpha_k + \beta_k) \cdot d_i.$$

(6) After calculating all the necessary parameters, V_k chooses a random $\tilde{r} \in \mathbb{Z}_P$ and computes

$$\tilde{r} \cdot Y_k \mod P.$$

Hence V_k hides Y_k from *VA* and the adversary.

(7) V_k sends the following encrypted randomized ballot and parameters to *VA* through an anonymous return channel:

$$(A, B)||G_k||H_k \cdot C_i^{(k)}||c_k||(D, R)||id_k^{VA}||(M(id_k^{VA}))^{RSK_{VA}}||\tilde{r} \cdot Y_k.$$

Since an anonymous return channel is used, *VA* does not know the identity of the sender, *i.e.* *VA* cannot connect the data received through the channel to V_k.

(8) After receiving all necessary information *VA* checks whether the voter with id_k^{VA} has already run the zero-knowledge proof, whether id_k^{VA} is signed correctly and calculates the following congruences.

$$c_k \equiv \sum_{j=1}^{n} d_j \mod Q,$$

$$a_j \equiv g^{r_j} \cdot G_k^{d_j} \mod P,\ j = 1, \ldots, n$$

$$b_j \equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}} \right)^{d_j} \mod P,\ j = 1, \ldots, n$$

If id_k^{VA} is correctly signed and not applied before, then the corresponding voter is eligible for voting and this is his first time to run zero-knowledge proof. If a voter was able to run the zero-knowledge proof several times, then he or she would possess more authorized ballots.

(9) If the verification congruences hold, then *VA* signs all the randomized components applying *SigGenEG*. *VA* calculates and sends

$$SigGenEG(G_k) = (s_{m_1}, R_1)$$

$$SigGenEG(H_k \cdot C_i^{(k)} \cdot Y_k \cdot \tilde{r}) = (s_{m_2}, R_2)$$

$$SigGenEG(Y_k \cdot \tilde{r}) = (s_{m_3}, R_3)$$

back to the sender through the anonymous return channel.

(10) Voter after verifies the three signatures of *VA* with

$$SigVerEG(s_{m_1}, R_1, G_k)$$

$$SigVerEG(s_{m_2}, R_2, H_k \cdot C_i^{(k)} \cdot Y_k \cdot \tilde{r})$$

$$SigVerEG(s_{m_3}, R_3, Y_k \cdot \tilde{r})$$

runs *ProofGenEG* algorithms in order to get authorization of the actual ballots being processed during the Tallying Stage. V_k chooses $\tilde{l_1}, \tilde{l_2}, \tilde{l_3}$ in the following way:

$$\tilde{l_1} \equiv (g^{\beta_k} \mod P) \mod Q$$

$$\tilde{l_2} \equiv (h^{\beta_k} \cdot \tilde{r} \mod P) \mod Q$$

$$\tilde{l_3} \equiv (\tilde{r} \mod P) \mod Q$$

and computes

$$ProofGenEG(s_{m_1}, R_1, \tilde{l_1}) = (\overline{s_{m_1}}, \overline{R_1}, \overline{T_1})$$

$$ProofGenEG(s_{m_2}, R_2, \tilde{l_2}) = (\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$$

$$ProofGenEG(s_{m_3}, R_3, \tilde{l_3}) = (\overline{s_{m_3}}, \overline{R_3}, \overline{T_3})$$

Vote Cast phase

(1) Voters send the following information to $\mathcal{BB}$

$$id_k^{\mathcal{R}} || g^{\alpha_k} || (\overline{s_{m_1}}, \overline{R_1}, \overline{T_1}) || h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k || (\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$$

through a public channel and

$$Y_k || (\overline{s_{m_3}}, \overline{R_3}, \overline{T_3})$$

to *VA* through anonymous channel. The form of the ballot is the ElGamal encryption of $C_i^{(k)} \cdot Y_k = g^{v_i + z_k \cdot \gamma_k}$, where z_k is sent by *VA* through an anonymous channel, hence z_k is not known by the adversary.

(2) Voters might check whether their ballots appear on $\mathcal{BB}$. If their ballot is missing or not correct, they can make a claim.

3.1.4. *Tallying stage.* After the voting stage is over the following computations are made:

(1) Verifier Authority runs *ProofVerEG* algorithm for each Y_k and calculates

$$Y \equiv \prod_{k=1}^{m} Y_k \mod P,$$

where only valid randomized component is considered and sends Y to $\mathcal{BB}$.

(2) After verifying validity of encrypted ballots with *ProofVerEG*

$$\Gamma \equiv \prod_{k=1}^{m} g^{\alpha_k} \mod P$$

$$\Lambda \equiv \prod_{k=1}^{m} h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k \mod P$$

appear on $\mathcal{BB}$, where only valid ballots are considered.

(3) After dividing Λ by Y we get the ElGamal encrypted voting result on $\mathcal{BB}$.

(4) Voting Authorities $A_1, A_2, \ldots, A_s$ together calculate the result $C_1^{t_1} \cdot C_2^{t_2} \cdots C_n^{t_n}$ with distributed ElGamal decryption method.

(5) Shanks baby step giant step or Pollard rho method might be applied for calculating t_i, $i = 1, \ldots, n$, which gives the election result for candidate i.

3.2. Security analysis.

Theorem 3.1. *The proposed e-voting scheme is secure, i.e. it satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability and protects against randomization and forced-abstention attack assuming, that at least one of the authorities is reliable.*

Proof. **Eligibility.** Verifier Authority checks validity of voters' credentials $id_k^{\mathcal{R}} || (M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \mod N_{\mathcal{R}}$ with the corresponding $RPK_{\mathcal{R}}$. If the credential is valid, his $id_k^{\mathcal{R}}$ had been authorized, then the voter's identity material showed in person to Registry was accepted.

Privacy. For encrypting the votes randomized, homomorphic ElGamal public-key cryptosystem is employed, that can be decrypted only, if all authorities collaborate. According to the scheme the voter's vote itself is never decrypted.

With the assumption that there is at least one reliable authority, votes remain secret. The vote C_i cannot be derived without knowledge of Y_k. Since during Vote Validation phase all ballots are randomized and cannot be connected to a voter, Verifier Authority does not know how a voter has voted even if *VA* has all information from $\mathcal{BB}$ and zero knowledge proof.

Unreusability. Verifier Authority follows according to the given $id_k^{\mathcal{R}}$ if a voter has casted his valid vote before or not.

Fairness. Determining the tally of the election starts after all the eligible voters have casted their ballots and the votes have been checked if they are valid or not. During the voting stage only the number of eligible voters can be found out.

Robustness. It is detected during the voting phase, if a voter's vote is not valid and only valid votes are considered during the Tallying phase, hence invalid votes cannot distract the elections and it can be also checked if all valid votes are counted. Since all votes are encrypted and they are on $\mathcal{BB}$, authorities or any participant except the voter himself cannot alter votes.

Universal verifiability. After the valid randomized ballots are authorized voters send their encrypted votes on the Bulletin Board. All calculations made on $\mathcal{BB}$, any participant or passive observer can check whether these calculations are correct.

Individual verifiability. The voter himself can check on $\mathcal{BB}$, if his vote has been processed or not. If all public calculations are correct, the result of elections is valid and a voter's vote was made into the final tally as he cast.

Receipt-freeness, Uncoercibility. The proof of receipt-freeness and uncoercibility is based on the fact that there is no enough proof for an adversary how a voter has really voted. An adversary might know a voter's $id_k^{\mathcal{R}}$, $(M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \mod N_{\mathcal{R}}$ and set α_k, γ_k and C_i, v_i, too. During the voting process a voter receives a value z_k and an encrypted ballot

$$Enc_{\alpha_k}(v_i) = (g^{\alpha_k} \mod P, h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k \mod P),$$

where $C_i^{(k)} \cdot Y_k = g^{v_i + z_k \cdot \gamma_k}$. Let suppose a coercer has a demand of vote $v_i^* \neq v_i$ and coercer does not know z_k, then the voter is able to cast his vote v_i in a way, that the coercer will accept encrypted ballot on $\mathcal{BB}$. The voter can say the value received form *VA* is

$$z_k^* \equiv \frac{(v_i + z_k \cdot \gamma_k) - v_i^*}{\gamma_k} \mod Q.$$

Value Y_k never appears on $\mathcal{BB}$ and it is sent during the voting stage through an anonymous channel to *VA* without any identification number or value. *VA* can check its validity, but cannot connect it to a voter. During the Vote Validation phase all data is transported encrypted through an anonymous return channel and no information put on $\mathcal{BB}$.

Randomization attack. If a voter generates randomly formed ballot, it won't be authorized by *VA* during the Vote Validation phase. Only authorized ballots will be considered during the Tallying stage.

Forced-abstention attack. Even Registry does not possess a list of $id_k^{\mathcal{R}}$, since identification numbers are generated by voters and Registry, then they are blindly signed by $\mathcal{R}$, hence an adversary is not able to follow if an eligible voter has voted or not. □

4. Conclusions

The proposed scheme provides basic environments including eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability, recept-freeness and uncoercibility. It is protected against randomization and forced-abstention attacks. The protocol might be implemented in a practical environment since only anonymous channels are applied.

Author is grateful to professors László Csirmaz and Attila Pethő for their valuable remarks and comments.

Acknowledgement. The author is supported by TÁMOP 4.2.1-08/1-2008-003 project. The project is implemented through the New Hungary Development Plan co-financed by the European Social Fund, and the European Regional Development Fund. The author is partially supported by the project GOP-1.1.2-07/1-2008-0001 and also by the Hungarian National Foundation for Scientific Research Grant No. K75566.

References

[1] J. Benaloh and D. Tuinstra, Receipt-free secret-ballot elections, Proceedings of the 26th ACM Symposium on the Theory of Computing, *ACM*, 1994, 544–553.

[2] O. Baudron, P. Fouque, D. Pointcheval, G. Poupard and J. Stern, Practical Multi–Candidate Election System, 20th ACM Symposium on Principles of Distributed Computing, *ACM*, 2001, 274–283.

[3] D. Chaum, Blind Signatures for Untraceable Payments, CRYPTO '82, *Plenum Press*, 1982, 199–203.

[4] D. Chaum, Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, Communications of the ACM 24(2), 1981, 84–88.

[5] R. Cramer, R. Gennaro and B. Schoenmakers, A secure and optimally efficient multi-authority election scheme, Proceedings of EUROCRYPT '97, LNCS, *Springer-Verlag*, 1997; 1233, 103–118.

[6] I. Damgard and M. Juric, A Generalization, a Simplification and Some Applications of Pallier's Probabilistic Public-Key System, Public Key Cryptography'01, LNCS 1992, *Springer-Verlag*, 2001, 119–136.

[7] A. Fujioka, T. Okamoto and K. Ohta, A practical secret voting scheme for large scale elections, In Advances in Cryptology - ASIACRYPT '92, LNCS, *Springer-Verlag*, 1992; 718, 244–251.

[8] P. Golle and M. Jakobsson, Reusable anonymous return channels, Proceedings of the 2003 ACM workshop on Privacy in the electronic society, *ACM Press*, 2003, 94–100.

[9] M. Hirt and K. Sako, Efficient receipt-free voting based on homomorphic encryption, Proceedings of EUROCRYPT 2000, LNCS, *Springer-Verlag*, 2000; 1807, 539–556.

[10] P. Horster, H. Petersen and M. Michels, Meta-ElGAmal signature schemes, Proceedings of the 2nd ACM Conference on Computer and communications security, *ACM*, 1994, 96–107.

[11] A. Huszti, A secure electronic voting scheme, *Periodica Polytechnica Electrical Engineering* **51**/3–4 (2007), 141–146.

[12] A. Juels, D. Catalano and M. Jakobsson, Coercion-Resistant Electronic Elections, Proceedings of the 2005 ACM workshop on Privacy in the electronic society, 2005, 61–70.

[13] B. Lee and K. Kim, Receipt-free electronic voting through collaboration of voter and honest verifier, Proceeding of JW-ISC2000, 2000, 101–108.

[14] E. Magkos, M. Burmester and V. Chrissikopoulos, Receipt-freeness in large-scale elections without untappable channels, In B. Schmid et al., editor, First IFIP Conference on E-Commerce, E-Business, E-Government (I3E), 2001, 683–694.

[15] T. Okamoto, An electronic voting scheme, Proceedings of IFIP '96, Advanced IT Tools, *Chapman & Hall*, 1996, 21–30.

[16] T. Okamoto, Receipt-Free Electronic Voting Schemes for Large Scale Elections, Proceedings of Workshop of Security Protocols '97, LNCS, *Springer-Verlag*, 1996; 1163, 125–132.

[17] P. Pallier, Public-Key Cryptosystems Based on Discrete Logarithm Residues, EUROCRYPT'99, LNCS 1592, *Springer-Verlag*, 1999, 223–238.

[18] C. Park, K. Itoh and K. Kurosawa, Efficient anonymous channel and all/nothing election scheme, In Advances in Cryptology - EUROCRYPT '93, LNCS, *Springer-Verlag*, 1993, 248–259.

[19] I. Ray, I. Ray and N. Narasimhamurthi, An anonymous electronic voting protocol for voting over the Internet, Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS '01), 2001, 188.

[20] Zuzana Rjaskova, Electronic Voting Schemes, Master Thesis, *Comenius University, Bratislava*, 2002.

[21] K. Sako and J. Kilian, Receipt-free mix-type voting schemes - a practical solution to the implementation of voting booth, Proceedings of EUROCRYPT '95, LNCS, *Springer-Verlag*, 1995; 921, 393–403.

ANDREA HUSZTI
FACULTY OF INFORMATICS
UNIVERSITY OF DEBRECEN
H-4010 DEBRECEN, P.O. BOX 12
HUNGARIAN ACADEMY OF SCIENCES
AND UNIVERSITY OF DEBRECEN
HUNGARY

E-mail: huszti.andrea@inf.unideb.hu

(Received February 9, 2011; revised September 20, 2011)