

# An electronic scheme for the Farnel paper-based voting protocol

R. Araújo<sup>1</sup>, R. Custódio<sup>2</sup>, A. Wiesmaier<sup>1</sup>, and T. Takagi<sup>3</sup>

<sup>1</sup> Technische Universität Darmstadt, Germany

<sup>2</sup> George Washington University, USA

<sup>3</sup> Future University Hakodate, Japan

**Abstract.** In this paper we present eFarnel, a new electronic voting scheme based on the paper-based Farnel protocol. The new scheme corrects problems observed in previous proposals to simulate Farnel electronically. Moreover, we also define a new kind of mix net, called F-Mix-Net, which is required by our protocol.

**Key words:** Cryptographic Protocols, Electronic Voting, F-Mix-Net

## 1 Introduction

Electronic voting schemes have been proposed in the last two decades [1,2,3,4] as an alternative to paper-based voting systems. The schemes intend to, not only assure the voting security, but also offer efficiency and flexibility. In this way, they give convenience to the voters, while guaranteeing the accuracy of the result.

In general an electronic voting protocol is based on a set of security requirements that should, at least, offer the same security as obtained by the traditional (paper-based) models. Although a standard is still not proposed, there is a consensus among the researchers that the following requirements are acceptable (see [5,6,7,4] for more information).

### **Exactness :**

- A valid ballot cannot be altered;
- All valid ballots are counted correctly;
- Invalid ballots cannot be counted;

### **Democracy :**

- Only authorized voters are able to vote;
- Each voter votes at most once;

### **Privacy :**

- It is not possible to associate the ballot to the voter who issued it (anonymity);
- No voter can prove that a certain ballot was her (receipt-freeness);
- A coercer cannot force a voter to cast a ballot in a specific way (incoercibility);

- All ballots remain in secret until the end of the voting;

**Universal Verifiability** Anyone is able to verify the correctness of the voting process and its result;

These requirements (or parts of them) have been used by the researchers to achieve voting schemes. The proposed models are based on blind signatures [8], mix nets [1], and homomorphic encryption primitives. Naturally, some schemes also combine these primitives.

In schemes based on blind signatures, such as [9,5,7,10], usually the ballots are first validated (using the blinding mechanism) and then posted through an anonymous channel, before they are decrypted. Normally, these schemes are fast and simple, but not receipt-free. Tatsuaki Okamoto [3] proposed a receipt-free blind signature protocol, but it relies on the use of the strong assumption of an anonymous untappable channel.

Mix net based schemes can be found in [1,11,12,13,14]. Usually, the encrypted ballot (signed by a voter) is published. Before the ballots are decrypted and counted, the signatures are checked and the valid ballots are sent to the mix net. As the mix net is fundamental for the security, some protocols use cascaded mix nets and require them to prove their work. This hampers the efficiency of the protocol. Typically, mix net based schemes are not receipt-free, but a generic model to achieve this requirement was proposed by Byoungcheon Lee et al. [15].

The schemes based on homomorphic encryption, such as [6,2,16,17,4], are simple regarding the ballot issuing, but not efficient in computing the voting result. The main idea is to publish: the encrypted ballot (signed by a voter) and its proof of validity. To obtain the voting result, the signatures and the proofs are verified. Then, the homomorphic property of the employed cryptosystem is used to obtain an encryption containing the sum of all ballots. Homomorphic based schemes are usually receipt-free. However, problems in some schemes as [6] and [18] were discovered, in a way that receipts can be made (see [16] for more details about the problems).

## 1.1 Contribution

In this paper we present eFarnel, a new electronic voting protocol based on the paper scheme introduced by Devegili [19]. The protocol was conceived taking into account all security requirements described in Section 1. Moreover, it solves the problems found in the previous protocols from Devegili and Araújo et al. [20].

Devegili's paper scheme has interesting properties not present in the traditional paper-based model. One of them is that it allows the voters to sign ballots without establishing an association to her own ballot. Due to the properties of this scheme, we establish a new kind of mix net called a F-Mix-Net to be used in our protocol. This feature allows it to use our mix net in the voting phase, in contrast to regular mix nets, which have to be used in the tally phase. This is realized preserving the voters' freedom to vote-and-go.

## 1.2 Related Work

Besides a new paper-based model, Devegili [19] also proposed an electronic voting protocol. Basically, the protocol simulates the new model using two authorities. One is pre-initialized with ballots and works as a ballot box. The other validates ballots for voters through a blind signature mechanism. During the voting, the voter gets her ballot validated, sends it to the "ballot box authority" and receives a receipt (see [19] for more details). This protocol has some drawbacks, such as the voters can make proofs revealing their ballots, it does not simulate the paper model correctly, and collusion among authorities can attest invalid ballots.

Aiming to make a protocol as close as possible to the new paper-based model, Araújo et al. [20] introduced a modified version of Devegili's protocol. The new protocol works like the original one, but during the voting phase each voter receives a ballot from the "ballot box authority", instead of the receipt. The voter must sign this ballot and send it to a new authority, whose work is to receive signed ballots.

As the original electronic protocol, the Araújo et al.' protocol has drawbacks. The voters can still make receipts. Furthermore, they can compromise the voting result by not following the protocol. A voter could, for example, receive a ballot from the "ballot box authority" and abstain to send it to the other authority or, receive a ballot and replace it with a copy of her own ballot.

In eFarnel these drawbacks are surpassed. It is partly related to Aditya et al. [21] and Lee et al. [15]. These schemes apply the idea of a trusted randomizer to provide receipt-freeness in mix net based schemes.

In Aditya et al. [21], the voter makes her ballot, encrypts it, and sends it to a trusted authority. After the voting period, the authority randomizes all received ballots and posts them to the public channel. After that, the authority proves (in a designated verifier way) the re-encryption to each voter. Then, the voter signs a pre-defined message to approve the re-encryption. In the tally period, all approved ballots are sent to the mix net before they are decrypted.

Lee et al. [15] employ a tamper-resistant hardware as the randomizer, instead of a remote authority. This device is given to the voters before the voting period. In order to vote, the voter encrypts her ballot and sends it to her (private) randomizer. Then, the randomizer re-encrypts it, signs the re-encryption, issues a (designated) proof of re-encryption, and sends this information to the voter. Upon accepting the signature and the proof, the voter signs the re-encryption and posts it to the public channel. To make the ballots anonymous, they are sent through a mix net, before they are decrypted.

Our scheme applies a trusted authority that re-encrypts the ballots, like [21] and [15]. However, the authority also acts as a mix net with an additional property. Moreover, the voter does not approve the re-encryption by signing it or a pre-defined message. Instead of she signs a randomly chosen encrypted ballot. Another difference is the fact that eFarnel makes the ballots anonymous during the voting phase and not (as usual) in the tally phase.

Section 2 describes the paper-based voting scheme proposed by Devegili contrasting it with the traditional paper-based model. Section 3 presents the crypto-

graphic primitives used in the electronic protocol as well as the definition of the new feature required by the mix net. Section 4 presents the electronic protocol that represents (as close as possible) that paper-based scheme. Finally, Section 6 summarizes our contributions and presents the future challenges.

## 2 The Farnel paper-based voting protocol

In the traditional paper-based voting system, the voter, after being authenticated by a trusted authority, receives a blank ballot, makes her choice, and casts it into a ballot box. The anonymity is achieved by using a polling-booth and the ballot box. After the voting period, the ballot box is opened and the ballots are counted.

This scheme has several well-known drawbacks which could be used to modify the result of a voting (see [22] for description of attacks in paper-based votings). One of them is the possibility to modify, insert, or even delete ballots after the voting period. Another, is the impossibility to verify who voted using only the information stored in the ballot box.

Aiming to bypass these problems, Devegili [19] introduced a new paper-ballot scheme called Farnel. The solution requires the voters to sign ballots. This way, it is possible to know who the voters were, and any attempt to insert, modify, or delete votes, after the voting period, can be detected.

To avoid an association between the voter and her ballot, the voter does not sign her own ballot, but one from another voter. This is done in the course of the voting process. Thus, Farnel gives new and different warranties to the voter that her ballot will be counted, and the exclusion or inclusion of new votes will not be possible, after the voting phase has finished. The voter can, for example, verify that all ballots are signed either by the voters or by the authority. Moreover, everybody can check who voted without needing the list of voters.

The scheme works as follows: differently from the traditional model, it has two ballot boxes. In a previous stage, before the voting period, the first ballot box is publicly initialized with ballots signed by a trusted authority. This set of ballots must represent, with an equal probability, all possible ballots. The second one starts the voting period empty. As in the traditional model, to vote the voter gets a blank valid ballot from the trusted authority, makes her choice and casts the ballot into the ballot box. In Farnel, this ballot box is the one initialized with signed ballots, i.e., the first ballot box. Upon receiving the voter's ballot, the first ballot box is shuffled and a random ballot is output. After receiving this ballot, the voter signs it and casts it into the second box. This ends the voting process for that voter.

After the voting period has finished, the remaining ballots of the first box are signed by the trusted authority and are inserted into the second box. Then, the second box is opened and all ballots are counted. From this result the ballots, which were pre-inserted into the first box before the voting period, are discounted.

### 3 Communication Channel and Cryptographic Elements

In this Section we introduce the elements which are used to assemble the electronic voting scheme proposed in Section 4.

#### 3.1 Communication Channel

The eFarnel protocol assumes the existence of an authenticated public channel, also called Bulletin Board ( $BB$ ). This model of communication is widely accepted and used several times in the design of voting schemes. The  $BB$  can be read by anyone, but only authorized parties can write on it. Furthermore, nobody can erase or overwrite messages once they have been written.

Normally, the voter uses the  $BB$  to post information directly, e.g., in the scheme presented by Cramer et al. in [2]. Since anyone can verify the posted data, the board contributes to achieve the verifiability requirement. Regarding the proposed scheme, the voter does not post her ballot directly. Instead of that, she sends it to a trusted authority that then publishes it.

In addition to the  $BB$ , the proposed protocol also requires a two-way untappable communication channel between the voter and the trusted authority. As defined by Okamoto in [3], an untappable channel is a physical device where the voter sends a message to an authority and any other parties learn nothing (information theoretically) about the message. In a two-way untappable channel, the authority is also able to send messages to the voter in the same way.

An untappable channel is a fundamental requirement for receipt-free protocols. As stated by Hirt et al. [16], an one-way untappable channel is the weakest physical assumption for a receipt-free protocol.

#### 3.2 Public-Key Cryptosystem

The ElGamal cryptosystem [23] over  $Z_p^*$  is used as basis for our voting scheme. This cryptosystem is applied to make encryptions and signatures.

Take into account a multiplicative subgroup of  $Z_p^*$  with order  $q$ , the public and private keys are generated in the following way:

- choose two (larger) primes  $p, q$  (with  $q \mid p - 1$ );
- the private key is a random number  $x$ , where  $x < p$ ;
- the public key is the triple  $h = g^x \bmod p$ ,  $g$  and  $p$ , where  $g$  is the generator of the subgroup;

To encrypt a plain text  $m$ , a secret random number  $r$  is chosen and the pair  $a, b$  is computed by  $a = g^r$  and  $b = mh^r$ . The pair  $a, b$  is the cipher text.  $m$  is obtained back by computing:  $b/a^x$ .

The ElGamal signature on  $m$  is realized in the following way. Choose a secret random number  $k$  (relative prime to  $p - 1$ ) and compute  $a = g^k \bmod p$ . Then find an  $s$ , such that  $m \equiv xa + ks \bmod p - 1$ .  $(a, s)$  is the signature. The signature is verified checking if:  $h^a a^s \bmod p = g^m \bmod p$ . Naturally, another suitable signature scheme can also be used.

As presented in Section 4,  $n$  talliers (or at least a subset of them) are required to decrypt the ballots. Thus, a threshold version of ElGamal is also employed.

A threshold based ElGamal cryptosystem was proposed by Pedersen in [24]. This protocol does not require a trusted party to distribute the secret key. Instead of this the secret key is jointly generated and verified by all involved parties.

To generate the public key for a  $(k, n)$ -threshold shared private key, where  $n$  is the total number of parties and  $k$  the minimal subset required to decrypt the message, each party  $T_i$  calculates  $h_i = g^{x_i}$ . The result is then committed (see [25] for more information about commitment schemes) and sent to the  $BB$ . After all parties published their commitments, each of them opens the committed value. The public key  $h$  is then computed by the product of all  $h_i$ .

Aiming to permit the private key reconstruction by any minimal subset  $\wedge$  of  $k$  parties, the Lagrange coefficients are used such that:

$$x = \sum_{i \in \wedge} x_i \lambda_{i, \wedge}, \quad \lambda_{i, \wedge} = \prod_{l \in \wedge \setminus \{i\}} \frac{l}{l - i}$$

However, as described by Cramer et al. [2], the cipher text  $(g^r, mh^r)$  can be decrypted, without reconstructing the key, in the following way:

1. Each party sends  $(g^r)^{x_i}$  to the  $BB$  and proves in a zero-knowledge way that  $\log_g h_i = \log_x g^{r x_i}$ ;
2. Let  $\wedge$  be a subset of  $n$  parties that produced the correct proofs, the plain text is obtained by computing:

$$m = y / \prod_{i \in \wedge} x_i^{\lambda_{i, \wedge}}$$

**Awareness of the encryptions** If the voters are allowed to make their encryption freely, a coercer can take advantage of this. One problem is, as stated by Delaune et al. [26], that a coercer could generate an encrypted message and force the voter to use it.

Another problem is, as stated by Jakobsson et al. [14], the fact that a voter could fetch an encrypted message from the  $BB$ , re-encrypt it and then re-post it. This could be used, for example, by a coercer to point out a vote on the  $BB$  and to force the voter to use it. Jakobsson proposes to solve this by requiring the encryptions to be non-malleable (see [27] for more details).

Using non-malleable encryptions also solves the Delaune problem previously stated, since the voter has no way to use an encryption not made by her. According to [12], an ElGamal encryption can be turned non-malleable by proving the knowledge of the random number used to encrypt the message, such that:

Let  $(a = g^r, b = mh^r)$  be the ElGamal encryption of a message  $m$ , where  $r$  is the secret key and  $a$  is the public key. Then  $(a, b)$  plus an additional information<sup>4</sup>

<sup>4</sup> [12] proposed the number of batches decrypted previously. We use the number of votes posted on the  $BB$  previously.

are signed using  $r$  and the Schnorr signature protocol [28]. Note that this is an interactive protocol. Checking the signature and its relation to the  $(a, b)$  assures the knowledge of  $r$ .

Thus, in order to avoid the two problems described above, eFarnel utilizes non-malleable encryption.

### 3.3 Mix net

The mix net is an important primitive to afford anonymity. It is used by many voting schemes, such as [10] and [15]. The main idea is to provide a cryptographic mechanism that receives a set of messages and outputs the same set, but in a random order. This is realized in way that nobody, except the mix net, can tell which incoming message corresponds to which outgoing one.

One or more servers can form a mix net. Usually, multiple servers are used in order to improve the security. Even if some of the servers reveal their secrets, for example the utilized permutation, the anonymity is still kept.

There are two models of mix nets: the Chaum model and the re-encryption model.

In the Chaum model [1], also known as decryption mix net, the message is encrypted with the public key of each individual mix. Thus, the most external encryption is made using the first server's public key and the most internal, the last server's key. Upon receiving the encrypted message each mix permutes the message and removes the respectively outer encryption.

In the re-encryption mix net (see Park [11]) the messages are encrypted with just one public key. As the encrypted messages transverse the servers, each server re-encrypts the messages, using the same public key, but a different randomization and permutes the encryptions. Note that the mix does not know the plain text of the message. This model has received more attention by the researches since it is more efficient.

In both models, some trust in the mix net servers is required. Otherwise, malicious servers can contribute, for instance, to relate incoming to outgoing messages, as presented in [29]. Protocols like [14,13] require the servers to prove their work.

Both models can be used in our scheme. However, as the scheme requires the mix net to hold some additional messages and use them in the mixing process, the mix net has to be an F-Mix-Net, which is defined below.

**Definition 1** *An F-Mix-Net is a mix net with the following additional attribute. Let  $\alpha$  be the set of incoming messages. Let  $\pi$  be an additional set of pre-defined messages. While mixing  $\alpha$  the F-Mix-Net secretly replaces some elements of  $\alpha$  with elements from  $\pi$ .*

### 3.4 Designated-verifier re-encryption proofs

As described in the next Section, the mix net employed in our protocol receives votes and re-encrypts them. To allow the voters to make sure their votes were

correctly re-encrypted, we require the mix net to prove the re-encryption via designated-verifier proofs (see Jakobson [30]).

The general idea is to have an authority providing a re-encryption proof to the voter. As the voter can also make the proof by herself, it become useless if she transfers it. The following designated verifier re-encryption proof was proposed in by Lee and Kim [4] and is used in the eFarnel protocol.

Let  $(a, b) = (g^{r_1}, mh^{r_1})$  be an ElGamal encrypted message  $m$  generated by the voter with the talliers' public key  $h = g^x$ . The randomizer re-encrypts  $(a, b)$  using  $h$  and a random number  $r_2$  to obtain:  $(c, d) = (g^{r_2}, mh^{r_2})$ . The voter's public key is  $h_v = g^{x_v}$  and the corresponding private key is  $x_v$ .

The authority (in our case the mix net) will prove to the voter that  $\log_g(a/c)$  and  $\log_h(b/d)$  have the same value. In order to accomplish this, the authority does:

1. choose the random numbers  $i, k, l \in Z_q$ ;
2. computes  $(e, f) = (g^i, h^i)$  and  $d = g^k h_v^l$ ;
3. computes  $o = H(e, f, c, d)$  and  $u = i - r_2(o + k)$ ;
4. send the proof  $(o, i, k, u)$  to the voter;

The voter verifies if:  $o \stackrel{?}{=} H(g^u(a/c)^{o+k}, h^u(b/d), g^r h_v^l, c, d)$ .

## 4 The Electronic Voting Scheme

This Section presents the eFarnel electronic voting protocol based on the paper-based Farnel scheme described in Section 2.

The electronic protocol uses a mix net as the first ballot box. However, in order to work as proposed the mix net must be an F-Mix-Net as defined in Definition 1. In addition, the mix net also operates as a randomizer, i.e., re-encrypting messages and presenting proofs of the re-encryption.

Unlike the paper-based protocol, the electronic one does not require a trusted authority to validate the votes as will be described below. Thus, each voter makes her vote and sends it directly to the mix net.

In the following description, the mix net is called Trusted Authority ( $TA$ ). The  $TA$  is supposed to work as required and will not act maliciously, for example, deleting votes or colluding with another party. Additionally, the  $TA$  is running on exactly one server.

The eFarnel protocol is composed of four phases: configuration, registration, voting and tally. Each phase has specific tasks that must be accomplished within a certain period of time. The phases are realized in sequence, i.e., to begin a subsequent phase the respective precedent phase must be finished.

Besides the voter and the  $TA$ , the following players also participate in the protocol:

- the registration authority ( $RA$ ), that is responsible to register the voters;
- the  $BBs$  as described in Section 3.1;
- the talliers ( $T$ ), responsible to decrypt the votes and count them;

Each public key has a corresponding digital certificate issued by a trusted certification authority. Moreover, there is a pre-defined list ( $L1$ ) of eligible voters. Both the certificates and  $L1$  are published on the  $BB$  to allow anyone to check them.

Founded on these assumptions and on the primitives described in Section 3, the protocol is now presented.

#### 4.1 Configuration Phase

In this phase, the general parameters of the voting are established. The options are defined as well as the encoding for the ballots. The keys of all authorities ( $RA$ ,  $TA$ , and  $T$ ) are generated and validated by digital certificates. Especially, all talliers ( $T$ ) cooperate to generate a public key  $(h_T, g_T)$  and to share the corresponding private key according to Section 3.2. All digital certificates, available options, and the ballot encoding are published. Also, an error message  $z$ , that will be used to warn that a certain proof was not accepted, is created and signed.

Additionally, the set of encrypted ballots (as described in Section 2) is publicly generated and published on  $BB1$ . This set is also used to publicly initialize  $TA$ . Any modifications made by  $TA$  in this set afterward are kept in secret. Let  $L_{(b)}$  be the set of encryptions kept and subsequently modified by  $TA$ .

#### 4.2 Registration Phase

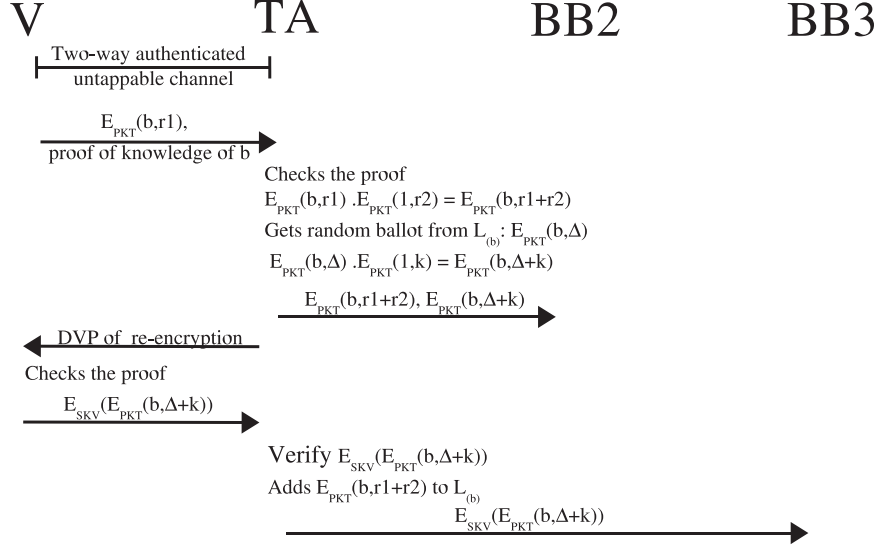
The voter proves her eligibility, for instance by signing a pre-agreed message, to the  $RA$ . At end of this phase, the  $RA$  posts the digital certificates of valid voters on the  $BB$ .

#### 4.3 Voting Phase

This phase consists of several stages. Figure 1 illustrates their interrelation. The stages are explained in detail below.

*Stage 1.* The voter makes her vote  $b$  based on the given ballot encoding and the available options. After that, she generates a secret random number  $r_1$  and uses it to encrypt  $b$  with the  $T$ 's public key  $(h_T, g_T)$ . In addition, the voter turns the encryption non-malleable (see Section 3.2) by providing a proof that she knows  $b$ . Let  $(g_T^{r_1}, bh_T^{r_1})$  be the resulting ElGamal encryption and  $p$  the proof. The voter sends them to  $TA$  via a two-way authenticated untappable channel.

*Stage 2.* After receiving  $p$  and  $(g_T^{r_1}, bh_T^{r_1})$  the  $TA$  verifies  $p$ . If the proof is correct, the  $TA$  generates a secret random number  $r_2$ , uses it to re-encrypt  $(g_T^{r_1}, bh_T^{r_1})$  and obtains  $(g_T^{r_1+r_2}, bh_T^{r_1+r_2})$ . After that, she shuffles her private list  $L_{(b)}$  and randomly gets an encrypted vote. The randomly chosen encrypted vote contains a valid vote, but the  $TA$  cannot see its contents. Let  $(g_T^\Delta, bh_T^\Delta)$  be the randomly chosen encrypted vote. The  $TA$ , creates a random number  $k$  and



**Fig. 1.** This figure presents the electronic scheme's voting phase.  $b$  = the vote.  $r1, r2, k, \Delta$  = random numbers,  $PK_T$  = the tallier public key.  $SK_V$  = the voter private key.  $L_{(b)}$  = the private list of encrypted ballots kept by the TA. DVP = designated verifier proof.

uses it to re-encrypt  $(g_T^\Delta, bh_T^\Delta)$  obtaining  $(g_T^{\Delta+k}, bh_T^{\Delta+k})$ . Then, she publishes  $(g_T^{r_1+r_2}, bh_T^{r_1+r_2})$  and  $(g_T^{\Delta+k}, bh_T^{\Delta+k})$  on *BB2*. These two cipher texts form a pair on *BB2*.

*Stage 3.* Via the designated verifier proof (see Section 3.4) the *TA* proves to the voter, that  $(g_T^{r_1+r_2}, bh_T^{r_1+r_2})$  and  $(g_T^{r_1}, bh_T^{r_1})$  have the same plain text.

*Stage 4.* In order to accept the proof, the voter signs  $(g_T^{\Delta+k}, bh_T^{\Delta+k})$  and sends the signature to *TA*. If the proof is not accepted, she sends the signature of the pre-defined message  $z$ .

*Stage 5.* The *TA* verifies the voter signature over  $(g_T^{\Delta+k}, bh_T^{\Delta+k})$ . If this signature is valid, the *TA* adds  $(g_T^{r_1+r_2}, bh_T^{r_1+r_2})$  to its private list and publishes the signature on *BB3*. Otherwise,  $(g_T^{\Delta+k}, b_r h_T^{\Delta+k})$  is added to the list and the signature over  $z$  to *BB3*.

#### 4.4 Tally Phase

The following activities are realized by the talliers (*T*) in cooperation.

*Stage 1.* The  $T$  checks if all ballots random ballots<sup>5</sup> on  $BB2$  have a corresponding signature on  $BB3$ . The ballots with valid signatures are decrypted and published on  $BB4$ .

*Stage 2.* The  $T$  asks  $TA$  to publish all ballots of her private list  $L_{(b)}$ , modified during the voting phase, on  $BB5$ . The  $TA$  re-encrypts the ballots and publishes them.

*Stage 3.* All ballots on  $BB5$  are decrypted by the  $T$  and published on  $BB6$ .

*Stage 4.* In order to obtain the result, all ballots published on  $BB4$  and  $BB6$  are counted. From the result the ballots published on  $BB1$  are discounted.

## 5 Analysis

As mentioned above, an electronic voting scheme is founded on a set of security requirements. In this Section, the new scheme is examined according to the requirements from Section 1.

We assume that the authorities  $RA$ ,  $TA$ , the  $BB$ , and a subset of  $T$  are trustworthy.

### 5.1 Exactness

A voter issues her ballot and sends it to the  $TA$ , which re-encrypts and publishes it on  $BB2$ . This ballot is not valid until the voter signs the random ballot (received from the  $TA$ ) and sends the signature to  $TA$ . As the  $TA$ , the  $BB$  and a subset of  $T$  are trustworthy, the valid votes will not be altered. Moreover, these authorities assure that all valid ballots will be counted.

### 5.2 Democracy

As the  $RA$  and the  $TA$  are trustworthy, they will only authorize valid voters to vote. Moreover, anyone can verify who voted by checking the signatures from  $BB3$ . This also assures that each voter will issue at most one vote.

### 5.3 Privacy

The  $TA$  randomizes the ballots, and replaces each of them by a randomly chosen one. By this, the ballot sent by a voter can either be in the  $TA$  private list or on  $BB2$ . Since the ballots were randomized, no one can link a vote on  $BB2$  to the respective voter. Hence, the anonymity is assured.

The voter sends her ballot to the  $TA$  which re-encrypts it before it is published. As the  $TA$  proves the ballot re-encryption in a designated way, the voter has no way to use the proof as a receipt. Hence, the scheme is receipt-free.

<sup>5</sup> The ballots which the mix net gave to the voter for signing it.

A coercer could see a voter issuing her ballot and force her to cast a specific vote. However, we state that the coercer has no way to see every voter voting, hence the receipt-free requirement is enough to avoid selling of ballots.

The scheme keeps all ballots in secret until the end of the voting. This is true, as at least one tallier is trusted. Thus, the ballots will not be decrypted until the voting phase has ended.

#### 5.4 Universal Verifiability

This requirement is assured since the ballots are published on the *BBs*. However, as the *TA* substitute votes without presenting any proof of this substitution, it must be trustworthy. This requirement is achieved since the *TA* is trusted.

## 6 Conclusion

We presented eFarnel, a new electronic version of the Farnel paper-based voting scheme. Our protocol aims to simulate the paper-based scheme as close as possible, and to solve the problems existing in the previous electronic versions of Farnel. Especially, the possibility to make receipts and to compromise the result.

For the construction of the protocol well known cryptographic primitives were employed. But as there is no primitive that could simulate Farnel's first ballot-box, a special kind of mix net, called F-Mix-Net, was defined. Differently from the regular mix net concept, an F-Mix-Net is initialized with a set of predefined messages which are swapped in during the shuffle process.

The new protocol still has some drawbacks. One of them is the strong trust in the F-Mix-Net. Another one is the high load of the F-Mix-Net server. It has to re-encrypt votes, to authenticate voters and to check their signatures. Hence, eFarnel is inefficient at the moment.

As future work, we intend to study a scalable scheme for this electronic protocol, in a way to reduce the mix net load. Moreover, we also intend to cut down the trust in the mix net, requiring it to present proofs of its correct operation.

## References

1. Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms. *Communications of the ACM* **24(2)** (1981) 84–88
2. Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. In: *EUROCRYPT*. (1997) 103–118
3. Okamoto, T.: Receipt-free electronic voting schemes for large scale elections. In Christianson, B., Crispo, B., Lomas, T.M.A., Roe, M., eds.: *Security Protocols Workshop*. Volume 1361 of *Lecture Notes in Computer Science.*, Springer (1997) 25–35
4. Lee, B., Kim, K.: Receipt-free electronic voting scheme with a tamper-resistant randomizer. In Lee, P.J., Lim, C.H., eds.: *ICISC*. Volume 2587 of *Lecture Notes in Computer Science.*, Springer (2002) 389–406

5. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In Seberry, J., Zheng, Y., eds.: ASIACRYPT. Volume 718 of Lecture Notes in Computer Science., Springer (1992) 244–251
6. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections (extended abstract). In: STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, New York, NY, USA, ACM Press (1994) 544–553
7. Cranor, L.F., Cytron, R.: Sensus: A security-conscious electronic polling system for the internet. In: HICSS (3). (1997) 561–570
8. Chaum, D.: Blind signature system. In: CRYPTO. (1983) 153
9. Chaum, D.: Elections with unconditionally-secret ballots and disruption equivalent to breaking rsa. In: EUROCRYPT. (1988) 177–182
10. Ohkubo, M., Miura, F., Abe, M., Fujioka, A., Okamoto, T.: An improvement on a practical secret voting scheme. In Mambo, M., Zheng, Y., eds.: ISW. Volume 1729 of Lecture Notes in Computer Science., Springer (1999) 225–234
11. Park, C., Itoh, K., Kurosawa, K.: Efficient anonymous channel and all/nothing election scheme. In: EUROCRYPT. (1993) 248–259
12. Jakobsson, M.: A practical mix. In: EUROCRYPT. (1998) 448–461
13. Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: ACM Conference on Computer and Communications Security. (2001) 116–125
14. Jakobsson, M., Juels, A., Rivest, R.L.: Making mix nets robust for electronic voting by randomized partial checking. In Boneh, D., ed.: USENIX Security Symposium, USENIX (2002) 339–353
15. Lee, B., Boyd, C., Dawson, E., Kim, K., Yang, J., Yoo, S.: Providing receipt-freeness in mixnet-based voting protocols. In Lim, J.I., Lee, D.H., eds.: ICISC. Volume 2971 of Lecture Notes in Computer Science., Springer (2003) 245–258
16. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: EUROCRYPT. (2000) 539–556
17. Baudron, O., Fouque, P.A., and Jacques Stern, D.P., Poupard, G.: Practical multi-candidate election system. In: PODC. (2001) 274–283
18. Magkos, E., Burmester, M., Chrissikopoulos, V.: Receipt-freeness in large-scale elections without untappable channels. In Schmid, B., Stanoevska-Slabeva, K., Tschammer, V., eds.: I3E. Volume 202 of IFIP Conference Proceedings., Kluwer (2001) 683–694
19. Devegili, A.J.: Farnel: Uma proposta de protocolo criptográfico para votação digital (in portuguese). Master's thesis, Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, Brasil (2001)
20. Araújo, R., Devegili, A., Custódio, R.: Farnel: Um protocolo criptográfico para votação digital. (in portuguese). In: Proceedings of II Workshop em Segurança de Sistemas Computacionais, Búzios, Rio de Janeiro, Brasil (2002)
21. Aditya, R., Lee, B., Boyd, C., Dawson, E.: An efficient mixnet-based voting scheme providing receipt-freeness. In Katsikas, S.K., Lopez, J., Pernul, G., eds.: TrustBus. Volume 3184 of Lecture Notes in Computer Science., Springer (2004) 152–161
22. Gumbel, A.: Steal This Vote: Dirty Elections and the Rotten History of Democracy in America. Nation Books (2005)
23. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO. (1984) 10–18
24. Pedersen, T.P.: A threshold cryptosystem without a trusted party (extended abstract). In: EUROCRYPT. (1991) 522–526
25. Goldreich, O.: Foundations of Cryptography: Basic Tools. Cambridge University Press, New York, NY, USA (2000)

26. Delaune, S., Kremer, S., Ryan, M.D.: Receipt-freeness: Formal definition and fault attacks (extended abstract). In: Proceedings of the Workshop Frontiers in Electronic Elections (FEE 2005), Milan, Italy (2005)
27. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: STOC, ACM (1991) 542–552
28. Schnorr, C.P.: Efficient signature generation by smart cards. *J. Cryptology* **4**(3) (1991) 161–174
29. Abe, M., Imai, H.: Flaws in some robust optimistic mix-nets. In Safavi-Naini, R., Seberry, J., eds.: ACISP. Volume 2727 of Lecture Notes in Computer Science., Springer (2003) 39–50
30. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: EUROCRYPT. (1996) 143–154