
Fra: Thomas Tjøstheim
Sendt: 25.04.2006 12:30:56
Til: Postmottak KRD
Kopi:
Emne: Saknr 06/532

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Hei,

Jeg er doktorgradstudent ved universitetet i Bergen hvor jeg for tiden jobber med e-voting. Nedenfor følger noen kommentarer til rapporten: Elektronisk stemmegiving -utfordringer og muligheter.

Kort noen skrivefeil jeg kom over:
s 27 3 avsnitt Finnland, en n for mye...
s49 2 avsnitt, reiser elektronisk stemmegiving... ser ut som en oversettelsesfeil (raises)
s56 2 avsnitt forslå istedenfor foreslå
s98 1 avsnitt kjøp og slag av stemmer (salg istedenfor slag)
s106 5 avsnitt "write-once" medium et det (at istedenfor et?)
s107 3 avsnitt bekrefter at stemme kan sendes inn (stemmen istedenfor stemme?)
Neste etterfølgende setn. : signere valget (kan misforstås, foreslår: signere stemmen eller valg av stemme?)
s161 7 avsnitt Et annet problem er at velgeren ikke vil får tilgang til (få istedenfor får)

Kommentarer til innholdet:
s102 Dobbelkonvolutt system foreslås: velger krypterer først med offentlige nøkkel til valgsystemet og signerer deretter med sin private nøkkel. Forutsetning som nevnes for at identiteter og stemmer ikke skal kunne kobles er at ingen skal ha tilgang til både de digitalt signerte e-stemmene og valgets private nøkkel samtidig. Syntes denne forutsentingen er tynn. Med gitte framgangsmåte er det lett å koble sammenhenger mellom stemmer kryptert med offentlige nøkkel og identitet (når de digitale signaturene sjekkes) . Gitt dekryptering av stemmene så kan det kobles hvilken klartekster som passer til hvilken siffertekster som igjen kan kobles til identiterer fra sjekk av digitale signaturfasen. Det bør understrekes at en form for anonymisering er nødvendig, for eksempel bruk av et mix net.

s106 Gjelder bekreftelse av korret avgitt stemme. Det foreslås matching av digitale signaturer og returnering av kryptert stemme som mulige bekreftelsesmetoder. Dette er uheldig da begge disse bekreftelesene gir en form for kvittering på hva stemmeren har stemt. Gitt en kvittering så kan stemmeren bevise hva han har stemt og dette åpner selvfølgelig opp for kjøp og salg av stemmer. Det gis noe beskyttelse ved at stemmeren har mulighet for å stemme om igjen, men hvis stemmeren ønsker det kan han bevise at han ikke stemmer seinere

ved å la f.eks en stemmekjøper logge trafikk eller oppgi akkreditiv (credendials) slik at stemmekjøperen kan kontrollere at stemmen ikke forandres. På dette grunnlag bør det utarbeides andre forslag for bekreftelse av korrekt avgitt stemme som ikke gir kvittering for stemmen.

Kan også nevnes at undertegnete og Geir Røsland har skrevet en artikkel om remote electronic voting (stemming over Internett) som er tilpasset løsningen anbefalt i rapporten med mulighet for å stemme flere ganger i forhåndsstemmefasen (fase1) og som støtter integrering med tradisjonelt valg (fase2). Artikkelen bygger på Røslands hovedfagsoppgave , men systemet har blitt forbedret og er sendt til ESORICS (European Symposium On Research In Computer Security), legger ved en kopi her.

mvh,

Thomas Tjøstheim

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.2.6 (GNU/Linux)
Comment: Using GnuPG with CentOS - http://enigmail.mozdev.org

iD8DBQFETfpfwHhZ7c+ERTARAgyQAJ9QjSCynzKTQ1Bwj9iuzgAvsP+JLQCgoJhA
KHTUkUvg8pjSyMY426q5Yjw=
=Sth3
-----END PGP SIGNATURE-----

Remote Electronic Voting Using Verifiable Chain Encryption

Thomas Tjøstheim and Geir Røsland

Department of Informatics, University of Bergen, Norway
E-mail: thomast@ii.uib.no and geir@rosland.com

Abstract. In this paper, we describe a new remote electronic voting scheme. A probabilistic multiple-key encryption function based on an extension of ElGamal constitutes the cryptographic basis for the scheme. Ballot encryption, mixing, and tallying are carried out through the construction of verifiable chain encryptions. Verifiability is based on the use of publicly available bulletin boards and scrutinizers representing the different candidates (or parties). The proposed scheme is receipt free and applicable for large scale elections.

Keywords: electronic voting, receipt freeness, scrutinizers, mix networks, coercion.

1 Introduction

Elections are the foundation of any democracy. However, quoting Tom Stoppard: "It's not the voting that's democracy, it's the counting". Throughout the history of voting there have been many examples of vote fraud [1, 2]. Initially, it may seem an easy task to design a fair election protocol, how difficult can it be to count the number of votes cast for each candidate? Further research, quickly reveals that the election process is more complex than first thought.

We propose a new remote electronic voting scheme for large scale elections. Over the last 4 decades, voter turnout has gradually decreased in established democracies [3]. Remote voting is convenient for the voters, and seems to be one effective way to increase voter turnout [4]. There is also a demand for better accuracy. In the 2000 USA presidential election, it is estimated [5] that between 4 and 6 million votes were lost, mainly because of "unreadable ballots". Electronic elections have the potential to offer high accuracy, combined with verifiability and error recovery.

By utilizing a public network to transfer ballots from voters to election authorities, we introduce many potential security issues. Similar to financial transactions, we need to assure the confidentiality and integrity of the information passing through the network. Elections have the additional requirement of anonymity. The fact that the voters' choices must be kept secret, complicates verification and error recovery. Errors must be traced while preserving anonymity, and care must be taken not to provide the voters with verification information that could be exploited by vote buyers or coercers.

The absence of neutral third parties complicates the analysis of election protocols. Anyone, be it candidates, election officials, or the voters themselves could have the motivation to cheat. In this paper, we limit the security analysis to only consider aspects that are directly related to the remote electronic voting scenario. Obviously, threats that are similar for other networked applications, like worms, viruses and implementation bugs will need to be addressed before using the scheme in practice. Rubin [6] considers remote voting and the security of the hosts and the Internet itself.

The rest of the paper is organized as follows: Section 2 introduces the notion *receipt freeness* and shows how our scheme is receipt free, Section 3 describes the election protocol, Section 4 analyzes some important parts of the scheme, Section 5 and Section 6 describe the cryptographic building blocks that form the basis for our scheme, and Section 7 concludes the paper.

2 Receipt Freeness

Receipt freeness is a strong form of privacy, where the secrecy of the ballot is maintained even if a voter cooperates with an adversary, that is, a voter cannot prove her choice of vote to another party. A coercer or vote buyer is dependent on controlling voter behavior to confirm votes, and a receipt free scheme will make it harder to verify how a voter has voted.

The notion of receipt freeness first appeared in a paper by Benaloh and Tuinstra [7]. It was shown later by Hirt and Sako [8] that their scheme lacked the postulated receipt freeness property. An extensive summary of previously proposed receipt free schemes is given by Juels, Catalano and Jakobsson [9]. All of the discussed schemes, except the one suggested by Juels et al. [9], assume the existence of an *untappable channel* between the voters and voting authorities. An untappable channel is a physical communication channel with perfect secrecy. Such an assumption makes the schemes unsuitable for Internet based voting, since the Internet is a public channel which cannot be untappable. The receipt free scheme by Juels et al. [9] defends against forced-abstention attacks and simulation attacks[1] even in the event of a collusion between a minority of the tallying authorities. A drawback with their scheme is that it is not suitable for large scale elections, as the overhead for tallying authorities is quadratic in the number of voters.

We propose an efficient and verifiable receipt free scheme for large scale elections. Obtaining universal verifiability and receipt freeness at the same time raises some problems, as these are somewhat conflicting properties. How can we assure voters of correct ballot postings without providing them with a receipt? A Public Bulletin Board (PBB) cannot be used, since voters would be able to show an adversary how their ballot was constructed, and then point to it on the PBB. We suggest using scrutinizers that represent different political candidates (or parties) to jointly monitor valid posting of ballots. The voters post their votes to a Restricted Bulletin Board (RBB) which only the RBB administrator and scrutinizers have read access to. A collaborative effort to cheat by the scrutinizers would be highly unlikely, given opposing political stands. The use of election observers to scrutinize the election is a trust model that is similar to what is used in traditional poll place elections. After the initial casting phase the ballots go through a mix net [10] to anonymize the voters' choices, before decrypting the ballots and tallying the votes. During the mixing and tallying stages of the election, PBBs are used to post verification information to enable universal verifiability.

An Internet-based receipt free scheme without any possibility of coercion is not practically realizable. Our goal has been to build an understandable, scalable, and secure scheme, that gains the voters' trust. Coercion and vote buying are theoretically possible if for example the coercer has continuous control over the voter or is collaborating with a dishonest election authority. However, the extent of coercion and vote buying will be strictly limited (discussed in Section 4.6).

3 A New Remote Electronic Voting Protocol

3.1 Assumptions

A prerequisite for the scheme is the existence of a national public key infrastructure [11]. Many countries already have, or are developing such infrastructures [12]. Each voter goes through a registration procedure to obtain a signed election certificate. The certificate and corresponding private

[1] Attacks where the voter is forced to disclose her secret keys, enabling the coercer to vote on behalf of the voter (simulating to be the voter).

key are used to prove voter eligibility to the administrators of the scheme. The election administrators will similarly use digital certificates to authenticate themselves when communicating with the voters.

Re-voting can provide resistance to vote buying and coercion. However, it must be impossible for the coercer to detect whether the voter is trying to re-vote or not. Note that even if the communication between the voter and election officials is encrypted, the mere occurrence of traffic between the two parties could indicate an attempt of re-voting. We assume that the traffic generated from re-voting can somehow be hidden from the attacker. This could for instance be done by periodically generating dummy traffic, that is, encrypted random messages sent from the voters [13].

3.2 Cryptographic Basis of the Protocol

The main cryptographic building block of the scheme is a new probabilistic multiple-key encryption function (explained in detail in Section 5) based on an extension of ElGamal [14]. We call the developed function the *election encryption function*, and briefly describe how it is used to create a chain of encryptions. Figure 1 shows the construction of an encryption chain when applying the election encryption function to ballot encryption, mixing and decryption of ballots. The voter first encrypts her ballot B and posts the encrypted ballot X to the RBB. The RBB administrator re-encrypts X and posts the result Y to the PBB. Each mixer re-encrypts the ballots at the PBB before the talliers re-encrypt the result Z, which returns the ballots to the initial states B, given correct operation of the election. The chain is *locked* in the sense that an encrypted ballot only can be decrypted if it has been encrypted in the correct sequence by the voter and all of the designated election administrators. The posting to the RBB and transition from RBB to PBB are only verified by the scrutinizers. While the rest of the steps in the encryption chain are universally verifiable, since verification information is posted to the PBB.

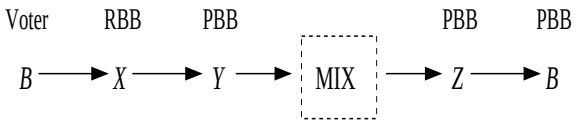

Fig. 1. Building a locked encryption chain.

3.3 The Election Protocol

In the following, we first list the participants and (cryptographic) primitives of the election protocol (see Figure 2), before describing each step of the protocol. Note that the numbering given in the explanation of the protocol matches the numbers in Figure 2.

Participants

- S – Set of scrutinizers, $s \in S$.
- M – Set of mix administrators, $m \in M$.
- T – Set of talliers, $t \in T$.
- V – Set of voters, $v \in V$.
- A – RBB administrator.

Primitives

- Private channels
- Election encryption function
- Public hash function H
- PBB
- RBB
- Verifiable mix net
- Threshold secret sharing scheme

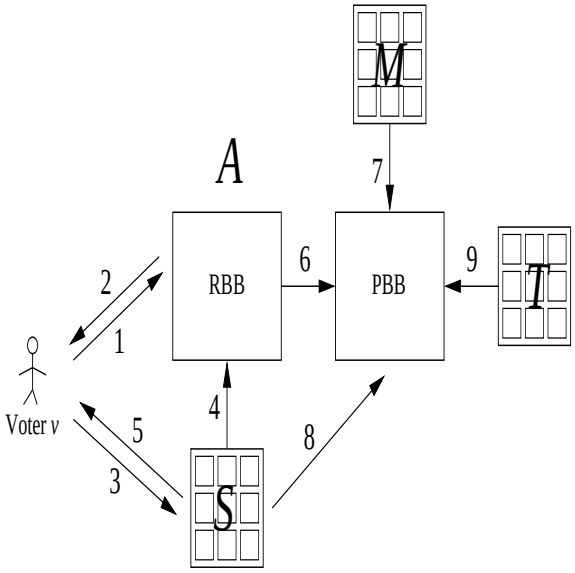

Fig. 2. Protocol overview.

Protocol

1. Each voter v first encrypts her ballot using the election encryption function and then signs a hash of the encrypted ballot with her private key. Voters are authenticated by signing a challenge from the RBB administrator A.
2. If authentication of v was successful, A posts v's encrypted ballot, the hash, and the signed hash to the RBB. An index value i of the posting is returned to v.
3. The voter v authenticates herself to the scrutinizers and states the hash of her encrypted ballot. The motivation for authenticating v, besides double checking that v is an eligible voter, is to prevent a dishonest administrator A from utilizing abstained votes. An additional reason is to avoid unnecessary investigation of requests made by ineligible voters.
4. If authentication of v succeeds, the scrutinizers check the RBB for the received hash.
5. Given that the scrutinizers find a matching hash on the RBB, the associated index value i of the posting is returned to v. Voter v checks if this is the same index value returned from A and submits a complaint to the election officials if they are unequal (not shown in Figure 2).
   If the hash is not found, the scrutinizers investigate the correspondence between v's posted signature and encrypted ballot. A could have modified the posted ballot or v could have asked

for the wrong hash. Only if there is no correspondence between the hash of the posted ballot and the signed hash, can we assume that A has cheated.
During the initial voting phase, the voters are able to update their cast votes. The steps 1–5 can be repeated any number of times, as the RBB only keeps record of each voter's most recently posted ballot.

6. The scrutinizers double check the RBB to make sure that A has not modified any values before transition to the PBB. If the check succeeds, A re-encrypts the ballots at the RBB by using his secret key and the election encryption function. The scrutinizers verify A's encryption and posting to the PBB.
7. A selection of mix administrators constitute the mix net. Each mix administrator uses the election encryption function and his secret key to permute the previously posted batch of ballots at the PBB, and posts the output batch back to the PBB.
8. For each mix, the scrutinizers verify that the ballots are not invalidated or substituted before the next mix can start. The PBB is used to post verification information to enable public verification of the mixing.
9. Given that all verifications hold, a certain number of the talliers jointly compute and publish the decryption key. All ballots can then be decrypted, and a tally can be published. The reason for using a threshold secret sharing scheme is to add robustness, as an attacker could prevent talliers from participating.

4 Analysis

In the following, we study some selected parts of the election protocol that we believe are crucial for the scheme's security and usability. Due to lack of space, we only provide an initial analysis.

4.1 Properties of the Election Protocol

We briefly discuss the security properties [15] of the proposed scheme.

Legitimacy: only registered voters may vote, and only once. The scheme assures legitimacy, since each voter must register for a public-private key pair prior to the election. The private key is used to sign a challenge given by the RBB administrator A, who verifies the voters' signatures with the corresponding public key certificates. We assume that each voter has a dedicated area at the RBB that is overwritten each time the voter re-votes, thus protecting against double-voting.

Ballot secrecy: only the voter should know how she voted. The scheme assures ballot secrecy, since the voters encrypt their own votes. Any device or other party can therefore not learn the voters' choices. A mix net is utilized to anonymize each voter's ballot, since the scrutinizers and the RBB administrator A initially know the correspondence between encrypted ballots and voters.

Individual verifiability: the voter should be able to check that her vote is accurately recorded for tabulation. The scheme assures individual verifiability, since the voters can use the scrutinizers to unambiguously confirm their posting to the RBB. Each voter's signed hash prevents A from cheating, since a posting only is valid if there is a correspondence between the hash of the encrypted ballot and the signature.

Universal verifiability: the final tally should be verifiable by any third party. The scheme assures universal verifiability, since verification information is posted to the PBB, enabling public verification of the mix net and the decryption of the encrypted ballots.

Accuracy: the final tally should reflect the true count of all legitimate cast votes. The scheme assures accuracy, since chain encryptions are utilized, enforcing that a ballot only will be correctly decrypted if it has been processed in correct sequence, and by all the designated participants.

Receipt freeness: a voter cannot by herself prove that she voted in a certain way. The scheme assures receipt freeness, since the voters do not have read access to the RBB, and hence cannot show which vote that was encrypted. Before transition to the PBB the ballots at the RBB are re-encrypted. The purpose of the re-encryption is to cause a permutation of the ballots to prevent v from showing a receipt for her vote, as all participants have read access to the PBB.

4.2 The Role of the Scrutinizers

The scrutinizers play a critical role in the scheme. Their main goal is to increase the voters' trust in the election process by guaranteeing correct operation of the election. However, this raises some practical challenges. We limit our analysis to study how the scrutinizers best can ensure the voters of correct ballot posting to the RBB. How the scrutinizers convince the voters, is essential for the voters' understanding and trust in the correct operation of the election. Ideally, all scrutinizers should be involved in the confirmation of the voters' ballots at the RBB. The voters can then be sure that their ballots are correctly posted, if at least one scrutinizer is honest. We do not think, however, that this is feasible in practice.

A more realizable solution is to let v pick a scrutinizer which she trusts to verify the ballot. A voter v can be quite certain of correct status of her ballot if the index value matches the value received from administrator A. The voter should also have the possibility of asking more than one scrutinizer, to double confirm, and to protect against a possible collusion between a dishonest scrutinizer and A. Each voter should be able to decide how many scrutinizers she needs to ask to be convinced of correct operation. Any irregularities in the answers from the scrutinizers should be reported to the election officials.

4.3 Error Handling

A detailed description of possible errors and recovery strategies must be carried out before the scheme is used. We only focus on how robust the scheme's construction is in terms of error handling. What if the index value returned to v by the scrutinizers is different from the value returned by the administrator A? Clearly, v will not be convinced of correct posting to the RBB. Bits could have been flipped while passing over the network or errors could have been introduced by a careless or cheating administrator. When will most errors occur and be discovered? Note that the scheme has two distinct phases. In the first phase, the voters post their ballots to the RBB and confirm correct ballot posting with help from the scrutinizers. The voters can also change their vote at any time during the first phase. At an official time announced in advance, the second phase starts by moving the ballots from the RBB to the PBB. The ballots go through the mix net, before being decrypted and tallied. A nice feature of this design, is that it is possible to catch errors before the second phase starts. In the event of errors, these can be found and fixed, without having to cancel the election.

This is in contrast to many election schemes that verify correct procedure at the end of the election, making error recovery more troublesome. Note that significant errors introduced in the mixing and tallying phase will be harder to recover from. Errors are, however, more probable in the first phase where more critical information is passed over the public network. In phase two, only information to and from the PBB will be exposed.

4.4 In Combination with a Traditional Election Model

Any remote electronic voting scheme should be integrable with the traditional election model, that is, voting at poll places. Many voters still lack the computer skills or the necessary equipment to participate in an online election. We utilize the fact that our scheme consists of two distinct phases to propose a solution for the combination of both election models. Let the voters cast ballots in the first phase until a pre-announced time. After the first phase ends, a day of traditional poll place election is held in-between phase one and two of the online election. A list of voters that voted in the traditional election is made and delivered to the scrutinizers. The scrutinizers jointly remove ballots at the RBB associated with voters who voted in both elections. The robustness of the election is strengthened through this combined solution. In the case of any serious errors, we have a solid backup solution.

4.5 Mixing in Practice

We mention some practical requirements to prevent information leakage during mixing. Disclosed information could be used for vote buying and coercion purposes. An important restriction is that the mix administrators do not use their own systems for mixing, it would then be easier for an administrator to store pairs of input and output values from the mix. We assume that special purpose mix machines are operated in protected areas, while being supervised by election officials. The calculations inside the machines should not be visible to anyone, but be based on parameters entered by the mix administrators. Correct operation of the mix machines can at any time be verified by running through test batches (for example an Elgamal encrypted input batch that is decrypted in the mix). It is inevitable that the operators of the mix will see some input and output values during work with the mixing and verification. However, the mixers cannot take with them any printouts or any other stored data when they leave the areas for mixing. In practice, it will therefore be very difficult to participate in any form of vote buying or coercion.

Practical organization of electronic elections is a large challenge in itself, and we have only mentioned some rough ideas for the mix phase that especially should have strong security requirements. We conclude that practical security requirements are necessary, regardless of how well a protocol is constructed. There will always be additional reasons to protect the election, like for instance terrorist attacks.

4.6 Coercion

A common criticism against remote electronic voting schemes is the vulnerability to voter coercion. However, this is not a threat specific to Internet-based voting, traditional poll place elections are also exposed. Many poll place election are vulnerable to chain voting [16, p. 373], or the use of cameras to prove what a voter votes. Even so, remote electronic elections are potentially more sensitive, due

to the many ways an adversary can verify voter behavior. The information being sent to the voters can be misused by an adversary to control the voters.

To get an idea of how vulnerable our scheme is to coercion, we consider three scenarios where a coercer threatens a voter. The voter (i) is forced to vote while the coercer watches, the voter (ii) is pressured into revealing her public-private key pair, and the coercer (iii) colludes with a dishonest scrutinizer to improve voter control.

(i) The first scenario represents little danger, since it is difficult to prevent a voter from re-voting at a later time. The coercer could stay with each voter till the mixing and tallying phase, but that would restrict the number of voters the coercer would be able to influence.

(ii) Without her key pair, the voter no longer has the ability to re-vote. The voter could report the loss to the election officials, in order to obtain a new key pair. However, care must be taken to prevent a coercer from finding out. The coercer could find out by trying to re-validate his vote posted to the RBB. A countermeasure could be to always return a positive acknowledgment if a verification request is made from an old key pair.

(iii) A coercer and scrutinizer collusion potentially gives access to all information on the RBB. The coercer could force voters to encrypt a ballot listing a specific candidate and verify that this value is posted at the RBB. This scenario is a real threat, but the probability of large scale coercion will be very small. Note that the dishonest scrutinizer could communicate with the coercer either during the ongoing election or after the election. Since the scrutinizers have conflicting political interests they will be controlling each other. We assume that it is nearly impossible for the dishonest scrutinizer to communicate during the election with the coercer, without being revealed by the other scrutinizers. Operation of a collusion after the election would be harder to detect. We assume that access to the RBB is strictly regulated, such that it is very difficult to copy or remember more than a couple of values on the RBB. The security requirements discussed in 4.5 would also apply here. In Section 4.4 we stressed that an electronic election should be combined with a traditional poll place election. We could utilize the fact that a voter's ballot in the traditional election overwrites the vote given in the electronic election.

5 Election Encryption Function

The election encryption function is a probabilistic multiple-key encryption function for chain encryptions. It forms the basis for the initial ballot encryption, the mix net and the decryption of ballots by the talliers. Our proposal is based on an extension of ElGamal [14] and allows any number of keys.

Select a prime p such that q is a large prime divisor of $p-1$. Select a generator β_0 of G_q, the subgroup of $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$, of order q, and n secret a_i's, such that

$$\begin{aligned} \beta_0^{a_1} &= \beta_1 \pmod p \\ \beta_1^{a_2} &= \beta_2 \pmod p \\ &\vdots \\ \beta_{n-1}^{a_n} &= \beta_n \pmod p \ . \end{aligned}$$

For each step in the encryption function, a random element $k_i \in \mathbb{Z}_q^*$ is chosen. A voter v encrypts her ballot B by

$$E_1(B, k_1) = (y_1, z_1) \ ,$$

where

$$y_1 = \beta_0^{k_1} \pmod p \ \text{ and } \ z_1 = B\beta_n^{k_1} \pmod p \ .$$

The next $n-1$ encryptions are given as

$$E_i(y_{i-1}, z_{i-1}) = (y_i, z_i) \ ,$$

where

$$y_i = y_{i-1}^{a_{n-i+2}}\beta_0^{k_i} \pmod p \ \text{ and } \ z_i = z_{i-1}\beta_{n-i+1}^{k_i} \pmod p \ .$$

The final encrypted pair (y_n, z_n) is given as

$$y_n = y_{n-1}^{a_2}\beta_0^{k_n} \pmod p \ \text{ and } \ z_n = z_{n-1}\beta_1^{k_n} \pmod p \ .$$

Anyone who knows the secret key a_1, can obtain the original ballot B by performing the decryption

$$d_K(y_n, z_n) = z_n(y_n^{a_1})^{-1} = B \pmod p \ .$$

This works since

$$\begin{aligned} z_n(y_n^{a_1})^{-1} &= z_{n-1}\beta_1^{k_n}(y_{n-1}^{a_2}\beta_0^{k_n})^{-a_1} \\ &= z_{n-2}\beta_2^{k_{n-1}}\beta_1^{k_n}((y_{n-2}^{a_3}\beta_0^{k_{n-1}})^{a_2}\beta_0^{k_n})^{-a_1} \\ &= B\beta_n^{k_1}\cdots\beta_2^{k_{n-1}}\beta_1^{k_n}((\beta_0^{-k_1})^{a_1a_2\cdots a_n}\cdots(\beta_0^{-k_{n-1}})^{a_1a_2}(\beta_0^{-k_n})^{a_1}) \pmod p \ . \end{aligned}$$

Note that $\beta_i = \beta_0^{a_1a_2\cdots a_i}$, so

$$\begin{aligned} z_n(y_n^{a_1})^{-1} &= B\beta_n^{k_1}\cdots\beta_2^{k_{n-1}}\beta_1^{k_n}(\beta_n^{-k_1}\cdots\beta_2^{-k_{n-1}}\beta_1^{-k_n}) \\ &= B \pmod p \ . \end{aligned}$$

5.1 Locking the Order of Re-Encryptions

The encryption function described so far has a possible weakness, for $i \geq 2$ one can substitute an encrypted ballot with a new ballot B' and re-encrypt by selecting

$$y_i = \beta_0^{k_i} \ \text{ and } \ z_i = B'\beta_{n-i+1}^{k_i} \pmod p \ .$$

Successive re-encryptions of all E_is of higher order are then required before the substituted ballot can be decrypted. This attack would most likely be detected as each re-encryption is being verified by the scrutinizers. However, we want to avoid the possibility of an attack from a collaboration

of dishonest administrators. A minor modification to the scheme enables us to lock the order of re-encryptions. We add the values (β_0^r, β_n^r), for a random $r \in \mathbb{Z}_q^*$, and require y_1 to be computed as

$$y_1 = \beta_0^r \beta_0^{k_1} \pmod{p} \ .$$

If we include the value β_n^r in the decryption

$$d_K(y_n, z_n) = z_n (y_n^{a_1})^{-1} \beta_n^r \pmod{p} \ ,$$

and this yields a valid ballot, we can be certain that the ballot the voter encrypted has been correctly re-encrypted by all of the key holders.

The public values of the election scheme are $(\beta_0, \beta_1, \cdots, \beta_n, \beta_0^r, \beta_n^r, p, q)$ while the private are $(a_1, a_2, \cdots, a_n, r)$, where each a_i is private to the corresponding administrator of the scheme.

6 Mix Network

The mix net secures the voters' privacy. Each mixer processes the ballots in batches, and shuffles the output according to a random permutation. The purpose of shuffling is to prevent an observer from linking the order of outputs to the order of inputs. A single mixer only knows the local permutation used, so the voters' anonymity is preserved as long as at least one mixer is honest.

Mixing is based on the use of a re-encryption mix net that uses the election encryption function defined in Section 5. A single mixer re-encrypts ballots in batches and permutes the ballots according to a random permutation π (or simply through sorting by decreasing/increasing values). The mixing phase starts by moving ballots from the RBB to the PBB. The mix net is universally verifiable, all information needed to verify the mixes are made available at the PBB. However, the first mix is verified by using the trust model of the scrutinizers. The voters cannot verify the first mix, since they then would be able to not only show a correspondence between their voting intentions and the encrypted ballot at the RBB, but guarantee for a vote being cast since re-voting no longer is possible after the transition to the PBB.

6.1 Mix Net Verification

How can we verify that a mixer performs correctly when re-encrypting ballots? A mixer could cheat by trying to substitute or invalidate ballots. Given

$$y_i = y_{i-1}^{a_{n-i+2}} \beta_0^{k_i} \pmod{p} \ \text{ and } \ z_i = z_{i-1} \beta_{n-i+1}^{k_i} \pmod{p} \ ,$$

we want to verify that y_i and z_i were properly formed. We give the following protocol for the verification of a single (y_i, z_i) pair. The administrator reveals k_i. Note that this does not pose a security risk, as this only gives away the correspondence between z_i and z_{i-1}. An attacker still has to solve the discrete logarithm problem to find the administrator's secret key a_{n-i+2}. When k_i is known, one can verify z_i directly, but, we still need to verify y_i, without revealing a_{n-i+2}. We pick an $\alpha = y_{i-1} \beta_{n-i+1}^t$, for a secret random value $t \in \mathbb{Z}_q^*$, and ask the administrator to encrypt α, and

receive $\sigma = \alpha^{a_{n-i+2}}$. If y_i was rightly formed, then verification of y_i succeeds since

$$\begin{aligned}
\sigma\beta_0^{k_i}\beta_{n-i+2}^{-t} &= \alpha^{a_{n-i+2}}\beta_0^{k_i}\beta_{n-i+2}^{-t} \\
&= (y_{i-1}\beta_{n-i+1}^{t})^{a_{n-i+2}}\beta_0^{k_i}((\beta_{n-i+1})^{a_{n-i+2}})^{-t} \\
&= y_{i-1}^{a_{n-i+2}}\beta_0^{k_i}\beta_{n-i+1}^{a_{n-i+2}t}\beta_{n-i+1}^{-a_{n-i+2}t} \\
&= y_{i-1}^{a_{n-i+2}}\beta_0^{k_i} \\
&= y_i \pmod{p} \ .
\end{aligned}$$

However, instead of validating a single pair (y_i, z_i) we validate the product of y_is and z_is. Note that for all y_is and z_is of a batch we can write

$$\prod y_i = \prod (y_{i-1}^{a_{n-i+2}}\beta_0^{k_i}) = (\prod y_{i-1})^{a_{n-i+2}}\beta_0^{\sum k_i} \pmod{p} \ ,$$

and

$$\prod z_i = \prod (z_{i-1}\beta_{n-i+1}^{k_i}) = \prod (z_{i-1})\beta_{n-i+1}^{\sum k_i} \pmod{p} \ .$$

The mixer publishes $\sum k_i$, such that the encryption of z_is can be verified directly. The scrutinizers pick an $\alpha = (\prod y_{i-1})\beta_{n-i+1}^{t}$, for a random secret value t. The mixer returns $\sigma = \alpha^{a_{n-i+2}}$. Anybody interested can now verify that

$$\begin{aligned}
\sigma\beta_0^{\sum k_i}\beta_{n-i+2}^{-t} &= \alpha^{a_{n-i+2}}\beta_0^{\sum k_i}\beta_{n-i+2}^{-t} \\
&= (\prod (y_{i-1})\beta_{n-i+1}^{t})^{a_{n-i+2}}\beta_0^{\sum k_i}\beta_{n-i+1}^{-a_{n-i+2}t} \\
&= (\prod (y_{i-1}))^{a_{n-i+2}}\beta_0^{\sum k_i} \\
&= \prod (y_i) \pmod{p} \ .
\end{aligned}$$

However, a dishonest administrator can still cheat by

1. Replacing a set of y_{i-1}s or z_{i-1}s with a different set that gives the same product.
2. Calculating $\prod y_i$ and $\prod z_i$ with different k_is, while ensuring that the k_is sum to the same number.

Cheating attempts can be thwarted by using a technique called randomized partial checking [17]. The simple, but effective idea by Jakobsen, Juels and Rivest is to compromise half of the messages in all batches in order to verify mix correctness. Voter privacy is obtained by the mix-net as a whole, by always selecting ballots not compromised in the previous batch. We adapt this technique to our scheme, but verify the output without revealing single ballot correlations.

For each mix, the scrutinizers jointly and randomly pick half of the ballots in the input batch, and require that the mixer can show a set of corresponding ballots in the output batch. The scrutinizers and anyone else interested verify that the product of the chosen ballots in the input batch actually maps to the product of ballots in the output batch. The mixers are prevented from cheating, since they have to guess correctly which ballots the scrutinizers select, and then only change the y_is and z_is that are either all in, or all not in, the selected set. The chance of guessing this for more than just a few ballots is clearly very small.

7 Conclusions and Future Work

In this paper, we propose a new receipt free, remote electronic voting scheme for large scale elections. A new probabilistic multiple-key encryption function is used to carry out chain encryptions. An encryption chain is formed by the voter encrypting her ballot, followed by the mix administrators who each re-encrypt the encrypted ballot to the next step in the chain, while the talliers decrypt the last element in the chain. A ballot will only be correctly decrypted if it has been processed in correct sequence, and by all the designated participants. The integrity of the election is secured through the locked chain of encryptions which prevents insertion of valid ballots after the initial voting phase, where each step of the encryption chain is verifiable. Posting of ballots and RBB administrator A's re-encryption are only verified by the scrutinizers, to prevent a voter from showing how she has voted. The scrutinizers can be trusted to monitor and verify the election process, since they represent different candidates (or parties). Hence, a collaborative effort to cheat would be highly unlikely. Mixing of ballots at the PBB and the tallying phase are universally verifiable.

Vote buying and voter coercion are theoretically possible, but minimized by several countermeasures: the scheme's receipt freeness property, the ability to re-vote in the remote election, the possibility of overwriting the electronic vote in a traditional poll place election, and practical security requirements. We note that the possibilities for vote buying and coercion could nearly be eliminated if the mix net was not publicly verifiable. Even though the trust model of the scrutinizers should be sufficient to verify the mix net, public scrutiny of the mix net is important to gain the voters' trust in the scheme. We believe that the integrity and usefulness of a remote electronic election is most important, and vote buying and coercion on a strictly limited scale can be tolerated, as long as the impact on the election outcome will be negligible.

We have only given an initial analysis of some selected parts of the scheme. Future work will include a more formal and in-depth analysis, in particular we want to consider:

1. **Error handling**: We need to carry out a threat analysis of different errors that could occur during the election phase and describe a detailed scheme for the handling and recovery of these errors. The probability of significant errors that make error recovery impossible should be minimized. One challenge is how to treat errors that occur during initial ballot encryption. These errors will not be detected before the tallying phase and cannot be mapped back to the voter. A possible solution is to let the voters cast dummy votes, to verify that encryption is carried out without errors, before the real vote is cast.
2. **Role of the scrutinizers**: The scrutinizers monitor and verify each step of the election protocol. We need to establish a clearer model for the verification of ballot postings and response to each voter. A method for double checking the RBB before transition to the PBB has not been specified. We cannot require from every voter that she states the hash of the posted ballot to the scrutinizers. A rough idea is to first check for the hashes received from the voters. For the cases in which no matching hashes are found, a database of voter certificates could be utilized to verify the correspondence between posted hashes and signatures at the RBB.
3. **Vote buying and coercion**: A more detailed analysis of scenarios that could occur should be carried out. Note that the RBB administrator A potentially has access to all the correspondences between ballots at the RBB and PBB, while the scrutinizers have a 50 percent chance (given that they randomly pick half of the ballots when verifying A's re-encryption) of showing that a particular ballot at the RBB is not changed before the transition to the PBB. Further requirements to limit access to the RBB should be determined to prevent the scrutinizers and A from storing significant amounts of data from the RBB.

We do not think that the implementation of remote electronic elections is something that should be rushed. There are many uncertainties in the underlying network infrastructure that need to be analyzed and solved before a large scale remote electronic election should be deployed. Currently, there are initiatives to create an alternative to the Internet that is more robust and secure [18].

Acknowledgments

The authors would like to thank Håvard Raddum, Kjell Jørgen Hole and Lars-Helge Netland for many helpful comments.

References

1. A. Gumbel, *Steal This Vote: Dirty Elections and the Rotten History of Democracy in America*, Nation Books Jul. 10 2005.
2. T. Campbell, *Deliver the Vote: A History of Election Fraud, an American Political Tradition–1742-2004*, Carroll & Graf Publishers 2005.
3. R. G. Niemi and H. F. Weisberg, "Controversies in Voting Behaviour," CQ Press, 2001.
4. S. Parker, "Shaking voter apathy up with IT," The Guardian, 11 Dec. 2001. `http://society.guardian.co.uk/modlocalgov/story/0,7999,616636,00.html.`
5. Caltech-MIT Voting Technology Project, "Voting, What Is, What Could Be?," 2001. `http://www.vote.caltech.edu/Reports/2001report.html.`
6. A. Rubin, "Security Considerations for Remote Electronic Voting over the Internet," ;Login: The magazine of Usenix & SAGE, Feb., 2001, vol 26., pp. 20–28. `http://www.usenix.org/publications/login/2001-02/pdfs/rubin.pdf.`
7. J.C. Benaloh and D. Tuinstra, "Receipt-Free Secret-Ballot Elections (extended abstract)," in 26th ACM STOC, pp. 544–553, 1994.
8. M. Hirt and K. Sako, "Efficient Receipt-Free Voting Based on Homomorphic Encryption," Proceedings of Advances in Cryptology – EUROCRYPT'00, pp. 539–556.
9. A. Juels and D. Catalano and M. Jakobsson, "Coercion-Resistant Electronic Elections," 2005, ACM Workshop on Privacy in the Electronic Society.
10. D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," communications of the ACM, 1981.
11. C. Adams and S. Lloyd, *Understanding PKI*, Addison-Wesley 2002.
12. `http://www.e.govt.nz/resources/research/international.html.`
13. R. E. Newman, I. S. Moskovitz and P. Syverson, "Metrics for Traffic Analysis Prevention," 2003, CHACS.
14. T. Elgamal, "A Public Key Cryptosystem and A Signature Scheme Based On Discrete Logarithms," IEEE Transactions and Information Theory, vol 31, no. 4, pp. 469–472, Jul. 1985.
15. P. Y. A. Ryan and T. Peacock, "Prêt à Voter: a Systems Perspective," Technical Report Series, CS-TR: 929, 2005.
16. J. P. Harris, *Election Administration in the United States*, The Brooklings Institution, 1934.
17. M. Jakobsson, A. Juels, and R. L. Rivest, "Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking," 2002, proceedings of the 11th USENIX Security Symposium.
18. D. Talbot, "The Internet is Broken," Technology Review, Dec. 19 2005. `http://www.technologyreview.com/InfoTech/wtr_16051,258,p1.html.`