Coercion-Resistant Receipts in Electronic Elections

Håvard Raddum

Havard.Raddum@ii.uib.no

Department of Informatics, University of Bergen

Abstract

Several suggested Internet-based electronic voting systems provide the voters with receipts to prove that their votes were counted. Unfortunately, these receipts strengthen an adversary's ability to coerce voters. This paper proposes a technique for generating receipts which gives voters a high degree of certainty their votes were counted, but without helping a coercer.

1 Introduction

A secure system for electronic voting over the Internet poses many challenges, and much research on this topic has been done over the last decade. Cryptographic techniques have given good solutions to many of the problems. During the last years some countries have made it possible for voters to cast their votes over the Internet in national elections. Estonia held its first nationwide electronic election in 2005, repeating the effort in 2007 and 2009, with a clear increase in the number of electronic votes cast each time. Norway plans to test electronic voting in some regions in 2011, with a view to host nationwide electronic elections in the following years.

Not surprisingly, there is a lot of skepticism among people about voting over the Internet when it comes to electing president or representatives for the national assembly. Free and fair elections are the basis of any democratic society and traditional paper-based elections work rather well. Therefore, national elections should not be a topic for experimentation. On the other hand, there are several good arguments for allowing voting over the Internet, including better accessibility for the elderly and disabled, as well as citizens traveling during the election. It is also believed that the voter turnout will increase if electronic elections are allowed.

One important advantage electronic elections have over traditional paper-based elections is the ability to provide voters with receipts. In paper-based elections the voters have to assume that procedures and controls for correct tallying are followed after the ballots are put in the ballot box. The receipts in an electronic election should be constructed to give each voter some kind of *proof* that the vote has been counted. One well known election system giving such receipts is Helios [1], but other systems have also been proposed [2, 3, 4, 5].

This paper was presented at the NISK-2010 conference.

The Helios voting system is suitable for many types of elections, nevertheless the author points out that it may not be appropriate to use the system for elections where coercion is considered a non-negligible threat. In a governmental election the problem of coercion cannot be ignored.

The public debate in Norway around the upcoming electronic election has been focused on coercion. The fact that voters do not necessarily choose their ballot in private is enough reason for most skeptics to reject electronic elections over the Internet. We do not know how serious the coercion problem really will be, but for an electronic election system to gain trust and acceptance it is necessary to address the issue.

In this paper we look at problems related to receipts and coercion, and propose a new technique for generating receipts in Internet-based elections vulnerable to coercion.

2 Existing Solutions

In this section we recall well-known solutions to some of the problems facing electronic elections. First, we examine methods for minimizing the problem of coercion, and see how receipts come into conflict with these techniques. Then we look at mix nets, and how they can assure both anonymity for voters and correctness of the result at the same time.

Receipts and Coercion

Coercion in an election means that a voter is pressured into voting in a particular way, or forced to refrain from voting altogether. Coercion-resistant election schemes have been proposed before, see for instance [6], but they do not provide voters with receipts. The problem of coercion is minimized in paper-based elections since the casting of votes takes place in a controlled environment at a polling station. Each voter then has the possibility to vote in secret and no one can prove how anyone voted.

In an electronic election the threat of coercion rises because the voting now can be done from any computer connected to the Internet. Most electronic votes will be cast from the voters' homes where it is not possible to control whether voting is done in private and not under pressure from someone looking over a voter's shoulder. This is an inherent problem of all Internet elections, since part of the reason for having Internet elections in the first place is to increase accessibility by not requiring voters to show up at a polling station.

Unlike many challenges facing Internet elections, the problem of coercion cannot be completely solved by cryptography and protocols. However, there are some useful countermeasures. One of them is to allow voters to re-vote an unlimited number of times as long as the electronic election is open, and only let the most recent vote count. Another countermeasure is to run a normal paper-based election in parallel, and allow voters to vote both electronically and on paper. Only the paper vote will be counted for voters voting both over the Internet and on paper. It is then assumed the coercer will not be able to control a voter for a long enough time to be certain he or she has not re-voted, either electronically or on paper. These countermeasures do not solve the problem of coercion completely, but it seems to be the best we can do. Depending on country and culture, the threat from coercion may be considered small enough to allow voting over the Internet in important nation-wide elections.

As mentioned in the introduction, receipts in electronic elections should be considered an advantage. Voting receipts give a more transparent election process, higher confidence in the accuracy of the election result, and can help lower the risk of election fraud. Receipts in existing electronic election systems also have the desirable property that they cannot be used to show how people voted. On the other hand, receipts can convince a coercer he or she really got the intended votes from the coerced voters. For instance, in Helios you get a receipt (a string of random characters) at the same time you cast your vote. If the coercer is present at the time of voting (s)he may record the receipt and later check whether this particular vote was included in the tally. Hence re-voting at a later time is now detectable by the coercer, and the receipt actually undermines countermeasures against coercion.

ElGamal and Mixing

In order to keep the cast votes secret, we need to encrypt the votes using a public key encryption scheme. A suitable choice is the ElGamal encryption algorithm, which will be used in this paper although other algorithms could also be considered. We briefly recall how ElGamal works [8]:

A large, safe prime q is chosen, and a multiplicative group G_q of order q is constructed with g being a generator for G_q. A secret element, i.e. the secret key, $x \in \mathbf{Z}_q^*$ is chosen at random, and the public key $k = g^x$ is published. The encryption/decryption between plaintext m and ciphertext (a, b), using the random parameter r, is done as follows:

$$(a, b) = (mk^r, g^r) \qquad\qquad m = ab^{-x}$$

ElGamal is a good choice for electronic elections because it incorporates randomness in the encryption function. This means equal votes from different voters will encrypt into different and unrelated ciphertexts, solving the problem of a small plaintext space. In an actual election, the secret key x should only be generated as shares using a secret-sharing scheme. These shares are given to parties with conflicting interests to ensure that no one will be able to decrypt the votes before they have been anonymized and the election is closed.

In systems for electronic elections it is common to use mix nodes in a mix net to anonymize the votes before they are decrypted [7]. An ElGamal ciphertext $(a, b) = (mk^r, g^r)$ can be re-encrypted as $(a', b') = (ak^s, bg^s) = (mk^{r+s}, g^{r+s})$. Note that decryption will still be done in one step and that the party doing the re-encryption does not need to know the plaintext m. The mix net consist of several mix nodes that are chained such that the output of one mix node becomes the input for the next. A mix node takes a set of encrypted votes, re-encrypts them, and outputs the re-encrypted votes in a secret, random order. The final mix node will take the secret ElGamal key and decrypt the anonymized votes so they can be counted. A mix net has two beneficial properties:

- **Untraceability:** Given an encrypted vote in the input to a mix node, it is not possible to find the corresponding re-encryption in the output without knowing the permutation the node has used. So, if an encrypted vote in the input to the first mix node can be tied to a voter, it is only possible to tie a voter to his/her decrypted vote if all the mix nodes are cooperating to defy the secrecy.

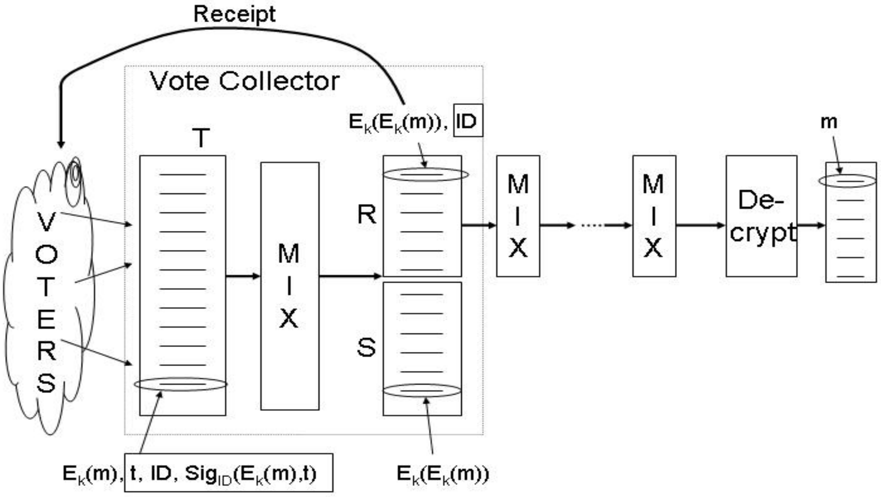

Figure 1: Electronic election system with coercion-resistant receipts. Information enclosed in rectangles is not made public.

- **Universal verifiability:** Despite the fact that one can not determine the correspondence between the input and the output of a mix node, the node can still publish a zero-knowledge proof showing all output ciphertexts are re-encryptions of the input ciphertexts. This proof is universally verifiable, so in principle anyone can verify that the node has been honest and that the set of underlying votes (the plaintexts) are the same in the input and the output. When each mix node publishes such a proof, it becomes universally verifiable that the encrypted votes entering the first mix node have been counted correctly.

We will not go into details on how these two properties are achieved, but rather use them as theorems. Interested readers are referred to [9] or [10] for a detailed description of mixing. Below we explain our new technique for making receipts.

3 Proposal for Coercion-Resistant Receipts

In this section we aim to rectify the outlined problem with receipts in current Internet-based voting systems. We propose a technique for generating receipts which gives voters a high degree of certainty their last electronic votes were counted. At the same time the receipts are of no help to a coercer.

Our threat model encompasses the situation where a coercer is present and monitoring all actions when the voter casts a vote. It seems clear that the receipt cannot be given at the time the vote is cast because it would provide the coercer with useful information. Only after it is no longer possible to cancel the electronic vote will the voter get the receipt. We are then left with the question of why the voter should be convinced the vote was counted. Below we explain the details of the new technique, and answer this question. The whole electronic election system is depicted in Figure 1.

Voting Phase

It is assumed that the voters have the ability to generate digital signatures. The actual voting will typically be done using a Java applet in a web browser. As long as the electronic election is open there will be a server that receives and stores cast votes. We call this server the *vote collector* (VC). We also assume the applet runs its own internal clock, which is synchronized with the time of the VC.

After the voter has authenticated herself to the VC she may cast her vote. She then makes her selection m and the applet sends the following string to the VC:

$$E(m), ID, t, sig_{ID}(E(m)||t),$$

where $E(m)$ is the ElGamal encryption of m, ID is a unique identifier for the voter, t is a timestamp, and $sig_{ID}(E(m)||t)$ is the voter's digital signature on the encrypted vote and the time. The VC keeps a reasonable window which should contain t. If t is acceptable, the VC stores the string $E(m), ID, t, sig_{ID}(E(m)||t)$ in a table T.

Since we are concerned about coercion, we allow the voters to vote multiple times. However, when the VC receives a new vote from someone who has voted before, the old votes are *not* discarded or overwritten, the new vote is just added to T.

Generating Receipts

After the electronic election is closed, the VC will not accept any more electronic votes and T is complete. Now, the VC knows exactly which electronic votes it should count. The VC performs one mix operation (see Figure 1) with all the $E(m)$s from T as input. The output is divided into two lists, R and S. On the list R the VC stores re-encryptions of votes that should be counted. The VC also keeps a list of IDs, linking each vote on R to its voter. On the list S the VC stores re-encryptions of votes that should not be counted.

The VC publishes the set $\{E(m)\}$ from T and the re-encryptions from both R and S. It is publicly known which re-encrypted votes that belong to R and which belong to S. It is now universally verifiable that the VC has not altered the set of plaintext votes in the mix process. The ciphertexts on the list R act as receipts, and the voter with an ID tied to a particular ciphertext in R receives a copy of this ciphertext, digitally signed by the VC. Suitable channels for sending out receipts are SMS or regular email. When the voter receives the receipt she may access R, published by the VC, and check that her receipt is on the list. Section 4 explains why she should be assured her vote was counted when seeing her receipt on R.

Mixing and Counting Electronic Votes

The rest of the process follows standard procedure for counting electronic votes. All (re)encrypted votes in R are entered into a mix net consisting of multiple nodes (see Figure 1). The votes are input into the first mix node and a number of mix operations follow with public proofs of correctness. The output of the last mix node are the actual votes in plaintext, which may easily be counted.

4 Properties of the Proposed Technique

Now we discuss the properties of the proposed technique and determine which problems get solved and which remain.

Coercion-Resistance

The motivation for designing a new kind of receipts in electronic elections is to have receipts that do not undermine any of the countermeasures against coercion. We claim that the receipt the voter gets in our new system does not give a coercer any information about whether the voter has re-voted after the coerced vote was cast.

If the coercer records the encrypted coerced vote, he will see this ciphertext on the list T published by the VC since *all* votes that have been cast are posted here. The question is then whether the coerced vote ended up on R or on S, but the untraceability of the mix operation ensures that the coercer cannot know this. If the coercer looks at the voter's receipt he cannot tell whether it is the re-encryption of the coerced vote or a different one. Hence, the voter may re-vote electronically without the coercer getting any information of this fact from the system.

Does the Receipt Prove the Vote was Counted?

Since the coercer cannot learn from the receipt whether the coerced vote was counted or not, why should the voter believe that his or her last vote was included in the result? In fact, there is no easy explanation for why the voter should be convinced the vote has been correctly counted when given the receipt from the system described here. We believe that it may be impossible to design a coercion-resistant system for voting receipts capable of proving that a particular vote has been counted. This is because the information that could convince the voter of this fact also could be transferred to the coercer, proving to him or her whether the coerced vote was counted or not.

What the voters do know, however, is that each of the receipts they receive corresponds to *some* vote that has been counted on their behalf. This is because there is a universally verifiable link between the list of receipts R and the decrypted votes. It is also universally verifiable that the VC's mix operation is done correctly. The question then comes down to trusting the VC to correctly split the output into countable and non-countable votes.

There are several ways for the VC to cheat, while still obeying the properties of the mix that can be universally verified:

1. The VC may place more than one vote from a voter on R.
2. The VC may put all votes from a voter on S, and none on R.
3. The VC may put a vote that is not the most recent one on R.
4. The VC may fail to record votes from a voter on T altogether.

We suggest to use some trusted auditor to prevent any of the three first threats from happening. The verification of the VC correctly splitting the table T into lists R and S when re-encrypting can be done as follows. The auditor gets the full table T, with IDs, timestamps, and signatures. Note that both timestamps and encrypted votes are signed by the voters, so the VC cannot manipulate these records. With this information the auditor is capable of identifying exactly which encrypted votes from T that should end up on R, and presents this shortened list of encrypted votes, T', to the VC. Next, the VC presents a proof that the partial mix from T' to R is done correctly. If we trust the auditor is honest and not cooperating with the VC except for this protocol, only the fourth item above can allow the VC to send out receipts for votes that do not get counted. We address this remaining issue next.

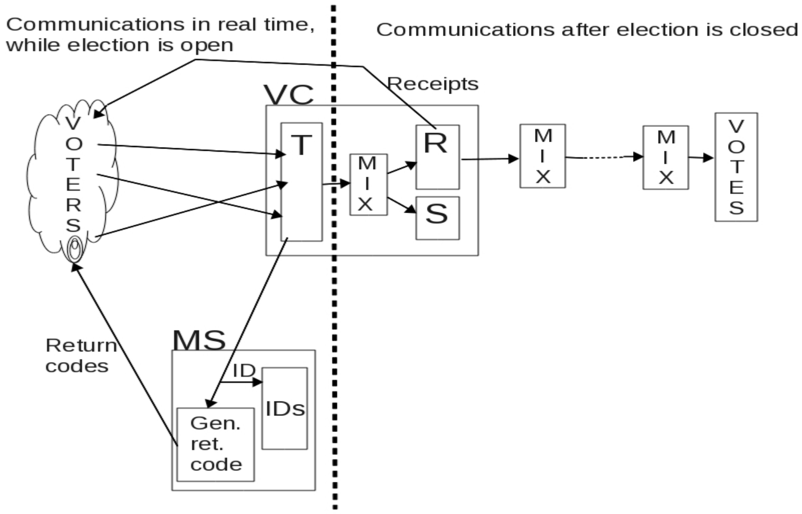
Communications in real time, while election is open
Communications after election is closed
Receipts
VC
VOTERS
T
MIX
R
S
MIX
MIX
VOTES
MS
Return codes
ID
IDs
Gen. ret. code

Figure 2: Complete election system with real-time verification codes.

Message Server as Auditor

In some election systems the voters get a verification code from an election server right after they have cast their vote, see for instance [5]. This code allows the voter to verify that the plaintext content of the vote received by the VC really corresponds to the voter's intent (or coercer's intent in case of coercion), and that no malware on the voter's PC has changed the vote before sending it. In order to preserve voter privacy, and at the same time be able to send the correct code to the voter, this part of the system must use a separate server in addition to the VC. We call this server the *message server* (MS). The MS is responsible for sending verification codes back to the voters, typically by SMS. The system is constructed such that neither the VC nor the MS can learn what the voters have voted, unless they are cooperating outside of the protocol for sending verification codes, see [11].

Therefore, in systems where verification codes are returned to voters we already have an entity that we trust is not conspiring with the VC. Moreover, the MS knows the set of IDs it has sent verification codes to, so it already has its own list of voters that should have a countable electronic vote. Letting the MS act as the trusted auditor, we also solve the fourth issue on the list of threats. The MS will take its list of IDs that have received verification codes and compare it to the IDs that are recorded in T. The two sets of IDs should be the same. If the VC does not store on T the votes it receives from some voter, it cannot contact the MS for sending out verification codes either, otherwise the VC will get caught in the audit. But if some voters do not get their verification codes they may complain, in real time while the election is still open. The complete election system with an MS is shown in Figure 2.

5 Summary

In governmental electronic elections the threat from coercion cannot be ignored. Judging from the public debate in Norway about the upcoming electronic election, this seems to be the issue giving politicians the most concern. The defense against coercion is to allow re-voting.

Electronic elections may gain trust in the population if voters get some kind of receipt that proves their vote was counted. However, such receipts may increase the problem of coercion since a receipt gives a coercer the ability to tell whether re-voting has occurred or not.

We have proposed a technique for making receipts that should not hurt countermeasures against coercion. It seems to be impossible to give voters receipts that prove 100% that their intended vote has been included in the tally, and at the same time not providing a coercer with any information about which vote was counted. The receipts proposed here give the voters a universally verifiable proof that *some* vote has been counted on their behalf. We think this feedback from the election system will be enough reassurance for most voters to believe in the accuracy of the result.

For those who do not fully trust the VC, we have described how a third party auditor can verify correct behavior from the VC. An advantage of the proposed technique is that auditing only needs to be done on one specific part of the system, as the rest is universally verifiable.

References

[1] B. Adida, *Helios: Web-based Open-Audit Voting*, Proceedings of the 17th USENIX Security Symposium, pp. 335–348, 2008.

[2] R. Cramer, R. Gennaro, B. Schoenmakers, *A Secure and Optimally Efficient Multi-Authority Election Scheme*, Proceedings of Eurocrypt'97, Lecture Notes on Computer Science 1233, pp. 103–118, Springer, 1997.

[3] A. Riera, J. Rifa, J. Borrell, *Efficient construction of vote-tags to allow open objection to the tally in electronic elections*, Information Processing Letters 75 (4), pp. 211–215, Elsevier, 2000.

[4] D. Malkhi, O. Margo, E. Pavlov, *E-voting without 'Cryptography'*, Proceedings of Financial Cryptography 2002, Lecture Notes in Computer Science 2357, pp. 1–15, Springer, 2003.

[5] V. Morales, Rocha, M. Soriano, J. Puiggali, *New voter verification scheme using pre-encrypted ballots*, Computer Communications Volume 32, pp. 1219–1227, Elsevier, 2009.

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

[7] D. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Communications of the ACM, vol. 24, no. 2, pp. 84–88, ACM, 1981.

[8] T. ElGamal, *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, Proceedings of CRYPTO'84, Lecture Notes in Computer Science 196, pp. 10–18, Springer, 1985.

[9] J. Furukawa, K. Sako, *An Efficient Scheme for Proving a Shuffle*, Proceedings of Crypto 2001, Lecture Notes in Computer Science 2139, pp. 368–387, Springer, 2001.

[10] M. Abe, *Mix-Networks on Permutation Networks*, Proceedings of Asiacrypt'99, Lecture Notes in Computer Science 1716, pp. 258–273, Springer, 1999.

[11] S. Heiberg, H. Lipmaa, F. van Laenen, *On E-Vote Integrity in the Case of Malicious Voter Computers*, Cryptology ePrint Archive, Report 2010/195, 2010.
`http://eprint.iacr.org/`.