

An uncoercible verifiable electronic voting protocol

A. Riera, J. Borrell, J. Rifà

Dept. d'Informàtica, Universitat Autònoma de Barcelona

Edifici C — 08193 Bellaterra — Catalonia (Spain)

Voice +34 93 581 17 77 — Fax +34 93 581 30 33

E-mail ariera@ccd.uab.es

Abstract

In this paper we present a verifiable voting protocol which removes the possibility of massive coercion or vote buying. In contrast with other previous proposals, in our protocol the mobility of voters is not restricted in any way, keeping the voting scheme as a real network application. This is achieved through the use of tamper-resistant smartcards and the involvement of a Trusted Third Party.

Keywords

Electronic voting schemes, uncoercibility, smartcards

1 INTRODUCTION

The single problem of building secure electronic voting schemes is of great complexity. Many topics are involved in order to guarantee the desired level of security. (Cranor *et al.* 1996) lists four essential core properties that are desirable in almost any election system:

- Accuracy: A system is accurate if (1) it is not possible for a vote to be altered, (2) it is not possible for a validated vote to be eliminated from the final tally, and (3) it is not possible for an invalid vote to be counted in the final tally.
- Democracy: A system is democratic if (1) it permits only eligible voters to vote, and (2) it ensures that each eligible voter can vote only once.
- Privacy: A system is private if (1) neither election authorities nor anyone else can link any ballot to the voter who cast it, and (2) no voter can prove that she voted in a particular way.
- Verifiability: A system is verifiable if voters can independently verify that their votes have been counted correctly.

Recalling the second privacy factor, note that it does not only *allow* the secret of the votes. In fact, votes are also *required* to stay secret. Such property was firstly introduced in (Benaloh *et al.* 1994) and named *uncoercibility*. Voters can be coerced basically in two ways. Either votes may be bought or voters may be intimidated in order to cast a particular vote. Both processes require the voter to prove the coercer in which way she voted. This is straightforward in almost all electronic voting schemes due to the receipts given at voting time by the collecting centre to voters in order to ensure the verifiability of the tally (fourth core property). A voter uses her receipt to prove forgeries on the tally if her vote has not been properly counted. However, the same receipt can also be used by a coercer or vote buyer to obtain proof of the vote cast. Therefore, in some way, uncoercibility and verifiability are opposite properties.

Receipt-free systems are uncoercible. Obviously, if the voter does not obtain any proof of the vote cast, there is no possibility for a third party to coerce the voter with certainty of success. Receipt-freeness of electronic voting schemes has been studied in (Benaloh *et al.* 1994), (Niemi *et al.* 1994), and (Sako *et al.* 1995). The solution offered in (Benaloh *et al.* 1994) is based on the existence of a physically protected voting booth. While the voter is in the voting booth, she is physically separated from outsiders and she cannot write to any channels, public or private. The protocol outlined in (Niemi *et al.* 1994) also requires a voting booth. Even though the booth is used only in a registration phase valid for several votings, it has the added difficulty of requiring private communication lines to each candidate. As a consequence, the authors declare their protocol as clearly impractical. The scheme in (Sako *et al.* 1995) requires the existence of an untappable private channel—a physically untappable channel; cryptographic implementations do not suffice—. In addition some flaws have been described in (Michels *et al.* 1996).

Note that all solutions presented above are based on physical assumptions, not far from those encountered in traditional elections. Voters are required to cast their votes from a physically isolated voting booth or to use untappable communication channels. Note that the latter also forces the voter to be located at some particular place. Such physical assumptions do not fit well with the notion of practical electronic voting schemes to be utilized over computer networks. In particular, the mobility of voters is clearly not considered. In fact, the only advantage gained over traditional methods is the increment of both speed and ease during the ballot counting process. Nonetheless, this can also be reached just by changing the traditional paper-based ballots by magnetic card ballots read by an electronic ballot box.

Ideally, in a network-based election system, voters should be able to vote from any point, thus removing the presence of voting booths. Clearly, if users are able to vote from any computer connected to the network, there is no way to prevent coercers watching the voters at the moment they vote. Total uncoercibility requires indeed the voter to be properly identified and physically isolated from outsiders during the voting phase (or part of it). Our goal is not

to prevent such attacks but to prevent the user from obtaining a receipt of her vote. Note that a receipt in the hands of the voter represents a more powerful threat. Massive coercions or vote buyings can only take place if a huge number of receipts are received through the network and processed off-line.

In this paper we assume a completely distributed voting scenario with no restrictions on the mobility of voters. We propose a voting protocol which removes the existence of receipts in the hands of the voter (and thus the possibility of massive coercion), while at the same time allowing voters to verify the correctness of the tally. If a valid vote has not been counted, the corresponding voter is able to do an open objection (Sako 1994), i.e. she can provide a Trusted Third Party (TTP) with evidence of the forgery with no need to reveal the vote cast. The same TTP is also involved to avoid a kind of attack against our protocol, stopping it at a certain step. In that case, the TTP acts as an arbitrator forcing the voter and the collecting centre to properly finish the protocol.

The strength of our method relies on the use of tamper-resistant smart-cards which are able to keep some information secret even to the owner. Note that the presence of smartcards in security protocols is currently increasing. Actually, it can be assumed that in a near future every citizen will have her own smartcard(s) which will be used indistinctly to digitally sign documents, do electronic commerce, or vote through computer networks.

The rest of the paper is structured as follows. Section 2 outlines the environment in which the protocol is operated and some related general procedures. Section 3 provides a detailed description of the protocol itself, step by step. In section 4 we discuss the security offered, and the problems risen when the protocol is suddenly stopped. Finally, in section 5 some concluding remarks are given.

2 THE ENVIRONMENT

We assume our protocol is operated using the environmental characteristics pointed out in (Borrell *et al.* 1996). In this way, voters establish secret and authentic communication channels with the collecting centre, through an authenticated key exchange (Diffie *et al.* 1992) using a local area network and the certification facilities provided by an independent cryptographic infrastructure. The system does not make use of any anonymous channel. The collecting centre is operated by a full electoral board. The use of a secret sharing scheme removes the possibility of dishonest members of the board affecting the voting.

Our protocol makes use of a Trusted Third Party (TTP). In a local scale ballot, the needed TTP must be newly defined. However, in large scale election systems as defined in (Riera *et al.* 1997), a hierarchy of voting authorities is involved and therefore some of them could easily be used as TTP.

The election process can be divided in the following phases:

1. Registration phase: In this first phase, every voter has to be provided with a smartcard holding her own private key. The correspondent public key must be properly certified and the certificate published in a public directory.
2. Elaboration of the electoral roll: This phase takes no action from voters. The authority on charge of the election elaborates the electoral roll and publishes it. There should be a certain period of time to allow voters to make objections if they so wish.
3. Voting phase: During the voting phase, voters execute the voting protocol as described in section 3.
4. Counting phase: Ballots are counted and the tally is published. Voters can verify that their votes have been properly added to the tally. They are able to prove forgeries on the tally and to do open objections, but it is not possible for them to collude with a vote buyer or to be coerced.

3 OUTLINE OF THE PROTOCOL

3.1 Notation

The following notation is employed in the presentation of the proposed protocol:

- V : Voter.
- SC : Voter's smartcard.
- CC : Collecting centre.
- TTP : Trusted third party.
- PIN : The secret Personal Identification Number required to activate the voter's smartcard.
- STS : Mutual authentication and authenticated key exchange protocol.
- P_{user} and S_{user} : The asymmetric key pair (respectively, the public key and private key) used by $user$.
- $cert(P_{user})$: The certificate of public key P_{user} .
- $M_1 | M_2$: Concatenation of messages M_1 and M_2 .
- $S_{user}(M)$: The digital signature of message M performed with the private key of $user$. For our purposes, $S_{user}(M)$ will comprise both M in clear, and the result of the signature operation over a digest of M .
- K : A key for a symmetric cipher.
- $E_K(M)$: The symmetric encryption of message M using key K .
- $vote$: A data string which can uniquely identify one of the voting options. To avoid guessing attacks $vote$ should be randomly chosen among a large set of valid values for each voting option.
- $ident$: A data string identifying the election.

3.2 Description

The protocol is outlined in Figure 1. Once the voter activates her smartcard by providing the correct PIN, a secret and authentic communication is established between the smartcard (on behalf of the voter) and the collecting centre. This is done by means of the STS protocol (Diffie *et al.* 1992), which ends up with the two parties being authenticated and sharing a common secret session key. Such key will be used to encrypt all the subsequent steps of the protocol (although we have not represented this encryption in Figure 1).

In step one, the smartcard is provided with the string *ident* which will be used in later steps of the protocol. The smartcard receives also the public key generated by the TTP, P_{TTP} . The accompanying certificate assures the authenticity of this key. By receiving the key P_{TTP} , the smartcard is informed that the election has been effectively started. At the end of the voting phase the private key S_{TTP} will be published. As a consequence, the TTP has to generate a new asymmetric key pair before each election takes place. In fact, the TTP used by our protocol does not need to have a permanent nature. It can be constituted just before each election and put out of order after the election ends.

After the smartcard has securely stored the received P_{TTP} , it is ready to accept, in step two, the vote from the voter (in fact, from her user agent or voting program). The vote is then not immediately sent in clear to the collecting centre because this could lead to selective receipt (i.e. the centre could accept only certain ballots and reject others). Instead, in step three, a vote-tag as defined in (Sako 1994) is sent to the centre, together with an encrypted version of the vote. In this way, the vote remains secret to the collecting centre until step five, when it receives the symmetric key K used in the encryption. As in (Sako 1994), we use a public asymmetric key P_{SC} as vote-tag. The collecting centre has to certify (sign) this vote-tag without knowing which has actually been the vote cast. In fact, the collecting centre signs the received vote-tag together with a string identifying the election (*ident*) to avoid possible replay attacks from the voter in further elections. In step four, the collecting centre sends to the smartcard this whole data structure, $S_{CC}(P_{SC} \mid \textit{ident})$. It represents the receipt which will allow the voter to verify that her vote has been properly counted, after the publication of the tally. The collecting centre will not be able to tamper the vote because it is signed with the private key S_{SC} correspondent to the certified vote-tag. Key S_{SC} will always remain unknown to everybody. In fact, even the smartcard will have no knowledge of this key, because it will effectively be erased from its memory after the execution of the protocol. The use of vote-tags allows also to perform open objections to the tally. This means that if the vote is not properly counted, the voter will be able to make a complaint without revealing in which way she voted.

In step five, the smartcard replies with the secret key K . This allows the centre to decrypt the vote. The centre verifies the signature made with S_{SC} ,

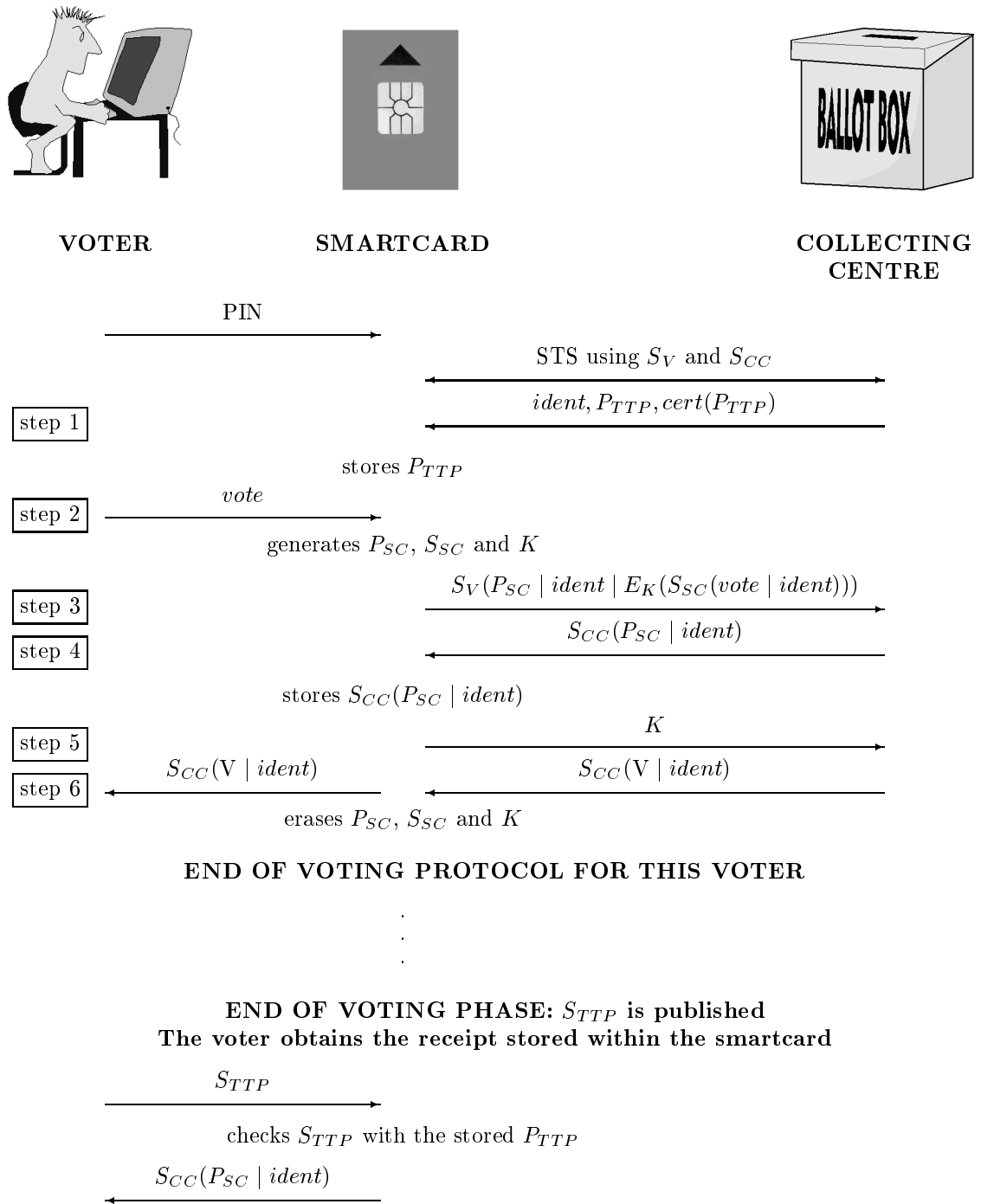


Figure 1 Outline of the voting protocol.

using public verification key P_{SC} . If the signature checking process fails, or the string $ident$ accompanying $vote$ does not match that one present outside the encrypted portion of data, this is reported to the TTP. The collecting centre can prove to the TTP the malfunctioning of the smartcard because P_{SC} , $ident$ and $E_K(S_{SC}(vote | ident))$ are signed together with S_V .

In step six, the collecting centre sends to the smartcard a further receipt which only provides proof that the voter has really voted (it only contains voter's identification and the election's identifier). Since it has no use for a coercer, this second receipt can be directly forwarded to the voter.

At the end of the voting protocol, the smartcard destroys the private key S_{SC} . If not eliminated, such key could be used to prove, after the publication of the tally, which vote corresponds to that particular voter. The smartcard erases also the public key P_{SC} and the symmetric key K .

3.3 Publication of the tally

During the voting phase, the described voting protocol is executed by the collecting centre and every voter. At the end of the voting phase, the collecting centre stops accepting ballots. The counting process is initiated and the final tally is published according to the format shown in Figure 2.

After the publication of the tally, the TTP publishes its private key S_{TTP} . The smartcard receives this key from the user. After verifying that P_{TTP} and S_{TTP} are a correct asymmetric pair, the smartcard is convinced that the election has really ended and the tally is already published. At this point, the smartcard can give the stored receipt to the voter, who can use it to verify the presence of her vote in the tally, as we will see in next section.

4 DISCUSSION

Clearly, given the scheme depicted in previous section, the collecting centre is unable to tamper the tally without being detected. In other words, our voting protocol is verifiable. We will see that the collecting centre is unable to modify or to remove from the tally any of the received ballots.

First, if the collecting centre omitted a ballot in the tally, this can be detected by the affected voter. Every voter just has to look in the tally for the public key P_{SC} included in the receipt obtained from her smartcard (last part of Figure 1). If P_{SC} does not appear in the tally, the voter can do an open objection, showing the receipt $S_{CC}(P_{SC} | ident)$ to the TTP.

The second possibility of fraudulent construction of the tally consists of the modification of a vote, while keeping intact the corresponding original vote-tag in the tally. This could only be possible if the collecting centre was able to break the underlying digital signature mechanism. This possibility is rejected because it can be considered unlikely within a short period of time.

⋮			
P_{SC}	$S_{CC}(P_{SC} ident)$	$vote$	$S_{SC}(vote ident)$
⋮			

Figure 2 Format of the published tally.

After discussing the verifiability of our protocol, we turn now to its uncoercibility. Uncoercibility relies on the fact that the receipt $S_{CC}(P_{SC} | ident)$ is secretly stored within the smartcard during the voting phase. It is therefore essential to provide the voter with the receipt (i.e. to broadcast S_{TTP}) only *after* the tally has already been published. Given the structure defined in Figure 2, to publish the tally actually means also to make public all receipts. In such conditions, the receipt stored in the smartcard has already become useless to a coercer. Actually, it could be retrieved from the tally by anyone.

In previous paragraphs we have assumed that both parties (voter and collecting centre) execute properly the protocol until its end. In fact, the protocol can be attacked just by stopping its execution at a certain step. The discussion focuses now on these cases, going on detail through all the possible points at which the protocol could be stopped:

1. Before step four the voter has not received the receipt $S_{CC}(P_{SC} | ident)$ yet. This means she is not able to prove anything and therefore a dishonest voter does not have any reason to stop the protocol. By the same way, before receiving the key K in step five, the collecting centre does not know the real contents of the data received in step three (i.e. the vote cast). Therefore, a malicious centre is not able to do selective receipt yet.
2. After step four, the voter is in a privileged position since her smartcard has already been provided with the receipt, but the collecting centre still has no knowledge of the key K . If the voter stops the protocol at this moment, she prevents the collecting centre from decrypting the vote. Given such conditions, a dishonest voter could use the receipt to do a fraudulent objection after the publication of the tally.
3. After step five, once the collecting centre has received K , it is able to decrypt the received ballot. Therefore at this point selective receipt could be done by a dishonest centre. However, we have seen previously how this can be effectively detected and proved to the TTP using the receipt kept in the smartcard of the voter. Another chance for a malicious collecting centre trying to do selective receipt is to falsely state that it has not received K (i.e. that the voter has stopped the protocol after step four).

Given this line of argument, it is clear the only possible attacks come either from a dishonest voter who stops the protocol after step four or from the collecting centre who falsely states that fact. Also in these cases the contention will be resolved by the TTP. However, we will see that this time the role played by the TTP must be completely different. (Zhou *et al.* 1996) gives a typification of the distinct roles of TTPs. We are interested in the case of *adjudicators* and *delivery authorities*. An adjudicator is not involved in a protocol unless there is a dispute and a request for arbitration. In such cases the adjudicator is able to state a judgment, based on the evidence provided by disputing parties. In contrast, a delivery authority is always used every time the protocol is executed. The TTP stands between both parties, acting as a forwarding entity. No messages are sent directly from one party to the other, but they are rather always delivered through the TTP.

In our case, we have seen that voters can do objections if the tally has been forged. They are able to provide evidence of the fraud to the TTP, which therefore acts as an adjudicator. However, if the collecting centre states that the voter has stopped the protocol after step four, it is really difficult for an adjudicator to take a decision. It is not clear how the TTP can establish that the collecting centre really obtained K , other than by accepting voter's word that K was sent, and assuming that the network delivered the message. By the same way, to establish that the centre really did not receive K implies to accept centre's word. For this reason, provided that there is this kind of dispute *during* the execution of the protocol, the role of the TTP changes to become a delivery authority. In other words, the TTP stands between the voter and the collecting centre, acting as a forwarding entity. In such configuration, the TTP is able to detect if either the voter is really stopping the protocol or the collecting centre is trying to do selective receipt. In the first case, the voter is considered to cast a null ballot. In the second case, the election is cancelled by the TTP. If the TTP is not able to act as a delivery authority because, when required, one of the parties does not respond, then the TTP takes the same decision as before.

5 CONCLUSIONS

We have adopted a practical approach to the receipt-freeness property of electronic voting schemes, substituting the traditional use of voting booths by tamper-resistant smartcards. We have presented an uncoercible and verifiable voting protocol in which the mobility of voters is not restricted in any way.

Our voting protocol uses a trusted third party to ensure the correctness of the tally. However, the TTP is required only if either the collecting centre performs a fraudulent construction of the tally, or the protocol is suddenly stopped. In both cases, the TTP can easily detect the cheater. This leads to the conclusion that cheating is strongly discouraged.

REFERENCES

- Benaloh, J. and Tuinstra, D. (1994) Receipt-Free Secret-Ballot Elections, in *STOC'94*, 544–553.
- Borrell, J. and Rifà, J. (1996) An Implementable Secure Voting Scheme. *Computers & Security*, **15**, 327–338.
- Cranor, L.F. and Cytron, R.K. (1996) Design and Implementation of a Practical Security-Conscious Electronic Polling System. *Washington University Report WUCS-96-02*.
- Diffie, W. and Van Oorschot, P.C. and Wiener, M.J. (1992) Authentication and Authenticated Key Exchanges. *Designs, Codes and Cryptography*, **2**, 107–125.
- Michels, M. and Horster, P. (1996) Some Remarks on a Receipt-Free and Universally Verifiable Mix-Type Voting Scheme, in *ASIACRYPT'96*, LNCS 1163 (Springer-Verlag), 125–132.
- Niemi, V. and Renvall, A. (1994) How to Prevent Buying of Votes in Computer Elections, in *ASIACRYPT'94*, LNCS 917 (Springer-Verlag), 141–148.
- Riera, A. and Borrell, J. and Rifà, J. (1997) Large Scale Elections by Coordinating Electoral Colleges, in *IFIP SEC'97, Information Security in Research and Business* (Chapman&Hall), 349–362.
- Sako, K. (1994) Electronic Voting Scheme Allowing Open Objection to the Tally. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences Vol.E77-A*, **1**, 24–30.
- Sako, K. and Kilian, J. (1995) Receipt-Free Mix-Type Voting Scheme, in *EUROCRYPT'95*, LNCS 921 (Springer-Verlag), 393–403.
- Zhou, J. and Gollmann, D. (1996) Observations on Non-Repudiation, in *ASIACRYPT'96*, LNCS 1163 (Springer-Verlag), 133–144.

6 ACKNOWLEDGMENTS AND BIOGRAPHY

This work has been partially supported by the Spanish Government Commission CICYT through its project TEL97-0663.

The authors are members of the Combinatorics and Digital Communication Group at the Autonomous University of Barcelona.

Andreu Riera (Manresa, 1970) obtained the graduate degree in Computer Science in 1993 at the AUB. Since then he is working towards the Ph.D. degree in Computer Science in the field of Security Protocols.

Joan Borrell (Girona, 1965) obtained the graduate degree and the Ph.D. degree in Computer Science, in 1989 and 1996, respectively, at the AUB.

Josep Rifà (Manlleu, 1951) obtained the graduate degree in Mathematics at the University of Barcelona in 1973. He obtained the Ph.D. degree in Computer Science in 1987 at the AUB.