The T–Vote Protocol

R. Araújo, A. Wiesmaier, J. Buchmann
TU Darmstadt — Cryptography and Computeralgebra
Hochschulstr. 10, D–64289 Darmstadt

Abstract

We present a secure e–voting protocol using voting booths. A noteworthy attribute of the protocol is that the validator and the tallier are not considered as trustworthy. After defining the protocol we discuss its security within the scope of the commonly agreed security requirements for voting schemes.

1 Introduction

This document presents an electronic voting protocol. It is based on a protocol proposed by Ohkubo et al. [OMA+99] and Kim et al. [KKLA01]. It was implemented and used in the 2002 FIFA World Cup to select the "most valuable players". Our scenario differs in the premises and therefore the protocol was adapted to the new situation.

We also present an analysis of the protocol. It is based on the security requirements for electronic voting schemes that are widely accepted by the scientific community.

The protocol uses blind signatures [Cha83] and mix nets [Cha81] which were both proposed by Chaum. Moreover, it requires the existence of a public channel that displays voting information to everybody.

The next Section presents the security requirements used in the analysis. In Section 3 the protocol is presented. Section 4 presents the protocol analysis. Finally, a conclusion is presented in Section 5.

2 Security Requirements

An electronic voting protocol must satisfy various security requirements. These requirements depend on the application context in which the protocol is used. For example, an opinion poll does not have to guarantee the secrecy of the votes. But in the election of a president of a state, that requirement is crucial.

Although security requirements for electronic voting schemes are not standardized, researchers agree on a set of security requirements. The following requirements are based on the work of Cranor et al. [CC97], Riera [Rie98] and Lee and Kim [LK02].

Exactness:

- A valid vote cannot be altered.
- All valid votes are counted.
- Invalid votes are not counted.

Democracy:

- Only authorized voters are able to vote.
- Each voter issues only one vote.

Privacy:

- Anonymity: It is not possible to associate the vote to the voter who cast it.
- Receipt–freeness: No voter can prove that a certain vote was his.
- Uncoercibility: A coercer cannot force a voter to cast a vote in a specific way.
- All votes remain secret until the end of the voting process.

Verifiability:

- Universal: Anybody can verify that all valid votes were counted.
- Individual: Each voter can verify that his valid vote was counted.

3 A Secure protocol for E-Voting

In this Section we present a protocol that is intended to achieve all requirements described in the last section. Before the protocol is presented, we first introduce the tools and notations used in the description and present the assumptions we made.

3.1 Communication Channel and Cryptographic Primitives

The protocol assumes the existence of an authenticated public channel, also called the *bulletin board*. Everybody can see the messages published on the bulletin board but only authorized parties can write messages to this board. Furthermore, nobody can erase or overwrite messages once they are written.

The protocol uses blind signatures [Cha83] proposed by Chaum. This mechanism prevents a signer from viewing the contents of a message to be signed.

Another anonymity technique is the mix net mechanism [Cha81]. A mix net basically receives a set of messages, decrypts (or reencrypts) them and outputs the new messages in a random order. Thus, it breaks the link among the incoming and outgoing messages.

In order to achieve confidentiality and authentication, public key cryptosystems, for example the RSA cryptosystem [RSA78] proposed by Rivest et al., are used.

3.2 Players

The following parties are the players of the protocol:

Voter (VOT)	The voter votes in a voting booth.
Validator (VAL)	The validator is responsible for validating the ballots.
Bulletin board (BB)	The bulletin board is a public channel.
Mix net (MIX)	The mix net mixes the cast votes.
Tallier (TAL)	The tallier is responsible for the tally.

3.3 Notations

The following notations are used in the description of the protocol.

$E_L(m)$	Encryption of the message m using the public key of entity L.
$S_T(m)$	Signature of the message m using the private key of entity T.
$B(m, r)$	Function to blind the message m with a random number r.
$UB(m, r)$	Function to unblind the message m blinded with a random number r.
v	A filled ballot, also called a vote.

3.4 Assumptions

We assume the following facts are given:

- A trustworthy public key infrastructure (PKI) is available and in use. All public keys used in the protocol are validated and digital certificates are issued by a certification authority. This implies that all encryptions are done with the correct public key. All parties participate in the PKI. The utilized cryptography is strong and considered as practically unbreakable.
- For communication we use a protocol such as TCP/IP [ISI81] which ensures messages delivery. We also assume that the communication is protected by a protocol that guarantees mutual authentication of the parties and confidentiality of the communication, such as TLS [DA81] based on the given PKI certificates.
- The registration phase works correctly.
- There is a trustworthy access control for the voting booth. It ensures that only valid voters are allowed to enter the voting booth and that only one person enters the booth at a time. The booth is constructed in a way that ensures that no third party can observe the voting. This includes eavesdropping via side channels (e.g. via the power consumption).
- The voting booth is trusted in the following way:
  - The booth does not exchange, alter or invalidate votes. It produces exactly the vote the user wants to cast.
  - The booth guides the user in casting the vote. The user cannot produce invalid votes. This includes the verification of the validator's signature on the vote.

  - The voter has a way to actively abstain from voting, e.g. by a special option on the ballot.
  - Neither the voter nor any third party has a way to save or see the blind factor utilized by the booth to blind the vote.
  - The booth presents the vote it published on the BB to the voter.
  - The booth does not collude with other parties.

- The bulletin board is trusted in the following way:
  - It authenticates the subscribers correctly and authorizes the access due to their role.
  - It cannot refuse to publish information from authorized parties.
  - It cannot alter or delete information.
  - It does not collaborate with others parties.

- We trust the mix net in the following way:
  - It mixes correctly.
  - It does not reveal its private key or the permutation.
  - It does not delete, add or exchange votes.
  - It does not collaborate with other parties.

- Thus, the trustworthy parties are:
  - The voting booth.
  - The bulletin board.
  - The mix net.

- Consequently the untrusted parties are:
  - The voter.
  - The validator.
  - The tallier.

- A valid vote is one that:
  - has the correct form;
  - is signed by the validator;
  - is encrypted using the public key of the tallier and the mix net in the correct order;
  - is published on the bulletin board.

3.5 Protocol Description

3.5.1 Registration Phase

The registration phase is out of the scope of the voting protocol. We only demand that at end of this phase, a list of valid voters and their certificates is published on the BB. This list can be verified by everyone.

3.5.2 Voting Phase

Figure 1 illustrates the voting phase of the protocol which is explained in the remainder of this Section.

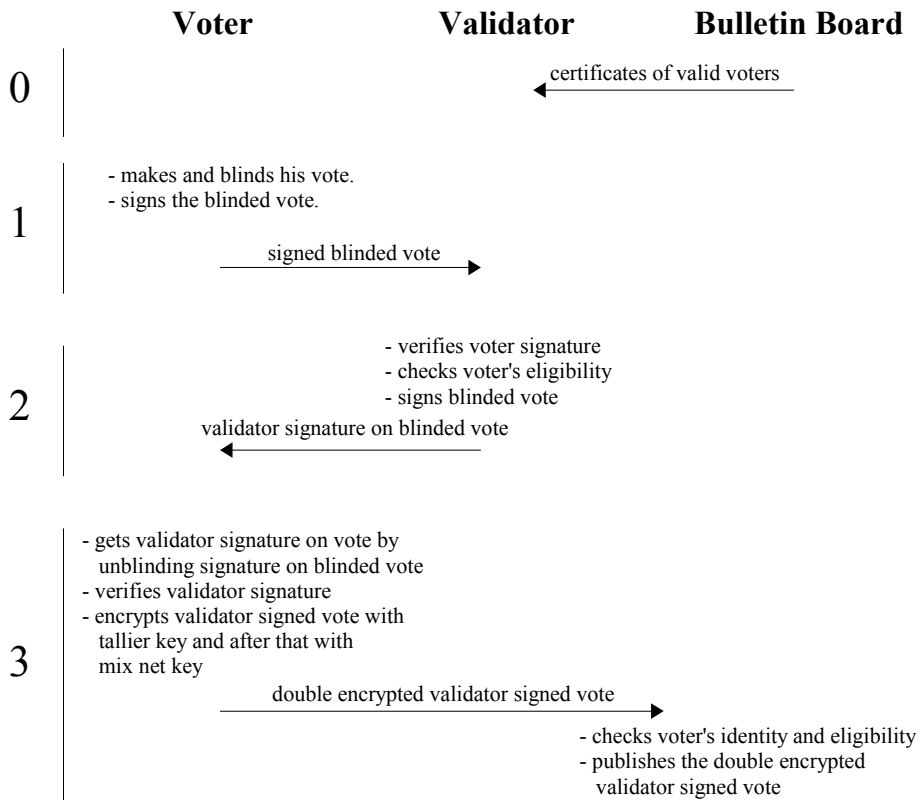

Figure 1: The Voting Phase

Stage 0 The validator fetches the certificates of the valid voters from the BB. This is only done once at the beginning of the voting phase.

The following stages 1-3 are executed for each voter.

Stage 1 The voter generates his vote v by using the voting booth. The booth then chooses a random number r and uses it to blind the vote by computing $x = B(v, r)$. Then, x is signed by the voter and he sends $(x, S_{VOT}(x))$ to the validator.

Stage 2 The validator verifies the voter's signature, checks whether the voter is allowed to vote, and that he has not yet applied for a signature. If this is true, the validator signs x and sends that signature $S_{VAL}(x)$ back to the voter.

Stage 3 Upon receiving $S_{VAL}(x)$, the booth removes the blinding factor r and obtains the validator's signature on the vote $S_{VAL}(v)$. The booth verifies that signature. If it is correct, the vote v together with its signature $S_{VAL}(v)$ is encrypted using the public key of the tallier. This means the booth computes $E_{TAL}(v, S_{VAL}(v))$. Then the booth encrypts $E_{TAL}(v, S_{VAL}(v))$ with the public key of the mix net. This means it calculates $E_{MIX}(E_{TAL}(v, S_{VAL}(v))$. The result is shown to teh voter. If the voter is eligible and has not yet published his vote, the BB allows him to publish $E_{MIX}(E_{TAL}(v, S_{VAL}(v))$.

3.5.3 Tally Phase

Figure 2 illustrates the tally phase of the protocol which is explained in the remainder of this Section.

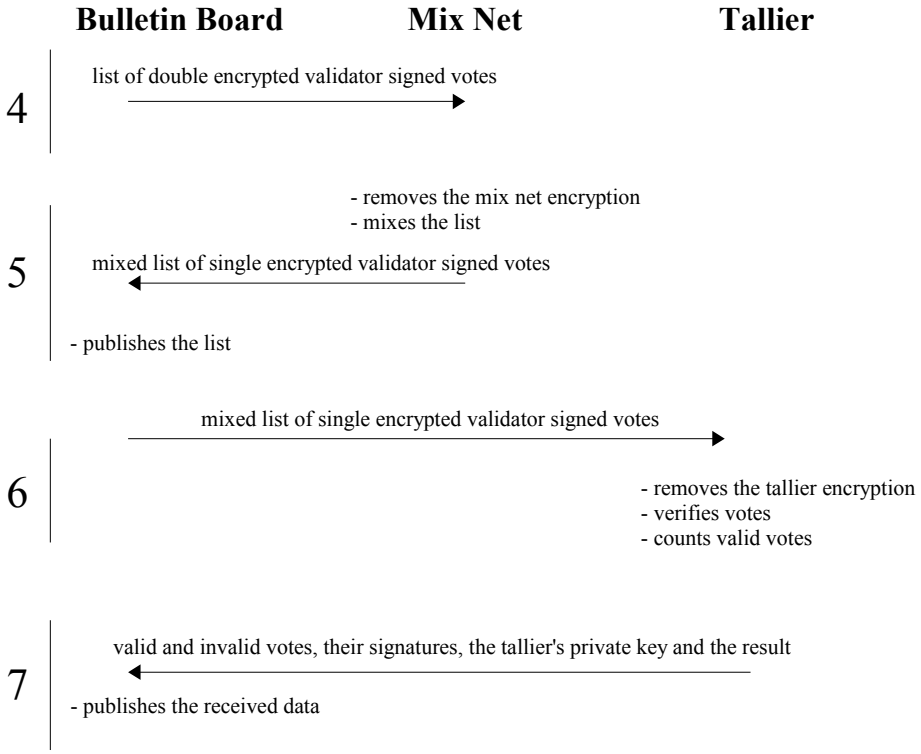

Figure 2: The Tally Phase.

Stage 4 After the voting phase, the mix net fetches the double encrypted votes from the BB.

Stage 5 The mix net removes the outer encryption from the votes using its private key. Then it permutes them and sends the new list back to the BB which publishes it. Note that the votes are still encrypted with the tallier's public key.

Stage 6 Then the tallier fetches the new list from the BB and decrypts the votes. The tallier verifies the validator's signatures on all votes and checks whether the votes are valid. Then he computes the result of the election.

Stage 7 Finally the tallier publishes all valid and invalid votes including their signatures in the respective parts of the bulletin board. The tallier also publishes its private key and the result of the election on the BB.

4 Analysis of the Protocol

4.1 Exactness

A valid vote cannot be altered All valid votes are posted on the bulletin board. Each vote is double encrypted, first using the tallier's key and then the mix net's key. If anyone, except the mix net and the tallier, wishes to alter a vote he needs to compromise the utilized cryptosystem which is impossible.

A malicious tallier could reveal his private key to a colluder which tries to change a vote on the BB. However, since the BB does not permit changes and the mix does not reveal his private key the colluder will not succeed.

After mixing the votes, the mix net will publish them on the BB. These votes are still encrypted with the tallier's key. If a colluder has the tallier's private key, he could try to change votes. However, as the BB does not permit changes, it will refuse this.

Additionally, everybody can see whether an encrypted vote was changed on the BB by tracking the changes on the BB.

All valid votes are counted The voter uses the trusted voting booth to make and verify his vote. This assures that the vote will be made correctly.

The trusted mix net will get the double encrypted votes from the trusted BB, mix them and post the result on the BB. These votes are (now) only encrypted using the tallier's public key. As the tallier is not trusted, he can get the votes from the BB, decrypt them and publish invalid results. However, as the tallier needs to reveal his private key in the tally phase, anyone can verify that all votes were treated correctly. This is accomplished by decrypting the votes (from the BB) and checking the validator's signature.

A malicious mix net could also delete votes. Although this act could be easily detected, there is no way to recover the deleted votes. As the mix net is trusted, it will not tamper the protocol.

Invalid votes are not counted As we saw above the trusted voting booth assures that only valid votes will be posted. Even if an invalid vote (wrong structure, wrong signature, wrong encryption) was published this will be detected after the tallier has decrypted the votes.

A malicious tallier could count invalid votes. This can be detected by anyone because the tallier must reveal his private key. Thus, anyone can decrypt the votes (from the BB) and check the validator's signature.

4.2 Democracy

Only authorized voters are able to vote As long as the validator is honest, he will recognize authorized voters by checking their digital certificates. If the validator is not honest and validates votes for not authorized voters, the trusted BB will refuse to publish the votes. The BB accepts only votes from eligible voters.

Each voter issues only one vote As long as the validator is honest, he will recognize and prevent attempts to vote multiply by recording the voters request for validation.

A malicious validator could validate more than one vote for a voter. However, as the BB is trustworthy, it will recognize this through the authentication of the voters and thus, prevents multiple voting.

4.3 Privacy

It is not possible to associate the vote to the voter who issued it It is not possible to link the vote to the voter by observing the network traffic, as the votes are encrypted and will be mixed.

It is not possible to link the vote to the voter by comparing the voting time and the time when a special vote appeared on the BB. This is because the votes are mixed by the trusted mix net before they are decrypted.

If the validator, the BB, and the tallier collaborate, they cannot associate the vote to the voter, since the validator has no idea which vote he signs, the BB only receives the encrypted votes from the voters and the tallier receives the permuted votes. Moreover, the BB and the Mix net are trustworthy, so they will refuse to cooperate.

The voters apply for the validation of their votes by the validator. As this is realized using blind signature, even if the validator is not trusted, he has no way to know the votes.

The voters publish their votes on the bulletin board and anyone can see the encrypted votes. In order to decrypt the votes before they are mixed the private key of the mix is necessary. As the mix does not cooperate this is not possible. Decrypting the votes after the mixing by using the tallier's key does not reveal the voter because the votes are mixed now.

No voter can prove that a certain vote was his A voter can get the vote validated by the honest validator and use it to prove his vote by showing it to anybody. However, as the voter is forced to use a voting booth which hides the crucial information, he has no way to show the information to anybody to prove his vote.

A voter could cooperate to the malicious validator to prove his vote. The voter could simply reveal the blind factor of the blind signature to the validator. Again as the voter is forced to use the trusted voting booth, he can not cooperate because he lacks the necessary information.

A coercer cannot force a voter to cast a vote in a specific way As the mix net and the voting booth are trusted, the voter has no way to show or

prove his vote to a coercer. The voting booth will assure that a voter cannot be observed while he is voting.

A voter can abstain to vote by casting a special vote using the voting booth. If he does this it cannot be detected whether he voted at all.

All votes remain secret until the end of the voting The votes are encrypted two times using the public keys from the mix net and the tallier. The one way to decrypt votes before the voting's end is having the private keys of these entities.

Even if a malicious entity has access to the tallier's private key, he cannot decrypt the votes. He will need also the mix net's private key. However, as the mix net is trustworthy, it would refuse to cooperate with the malicious entity. Therefore, the votes will remain secret until the end of the election.

4.4 Verifiability

Anybody can verify that all valid votes were counted After the registration phase, a list of eligible voters is published by the registration authority on the BB. The list permits anyone to verify who is eligible to vote and to verify the voter's certificates.

As the voters post their votes and the mix net publishes its results in the BB, anyone can check if the number of initially posted votes is equal to the number of the mixed votes. Moreover, as the mix net and the BB are trusted, they will not change, add or delete votes.

After decrypting the votes, the tallier verifies the validator signatures and publishes the results. The tallier also needs to publish his key on the BB. This permits anyone to decrypt the votes, check the votes and their signatures. Thus, anybody can verify that all valid votes were counted and whether the validator acted correctly.

Each voter can verify that his valid vote was counted As the voter posts his encrypted vote to the BB, he can check whether the published vote is the same as the one made by the voting booth. As we saw above it is guaranteed that all valid votes are counted. It follows directly that each voter's valid vote was counted.

5 Conclusion

Section 2 presented the security requirements for an electronic voting protocol. As seen, the requirements exactness, democracy, privacy and verifiability are fundamental to ensure the security of a voting.

In Section 3, a protocol that achieves all security requirements was presented. We utilized an existing and trustworthy PKI. The communication is guaranteed and secured by using available protocols like TCP/IP and TLS. A "blind signature" and a "mix net" are used by the protocol to assure the anonymity. The voters are not able to receive receipts as the voting booth prevents this. We assumed the voting booth, the bulletin board and the mix net to be trustworthy. In exchange we do not need to trust the voter, the validator and the tallier.

Section 4 showed how the given security requirements are fulfilled by the proposed protocol. We also saw that the untrusted parties are forced to act honestly. If they try to tamper the protocol this will be detected.

The number of communication messages required by the protocol is small. The validator fetches the list of certificates once. Each voter needs only three messages to cast the vote. In the tally phase the amount of the overall communication is basically equal to that of the complete voting phase.

References

[CC97] Lorrie F. Cranor and Ron K. Cytron. Sensus: A Security–Conscious Electronic Polling System for the Internet. In *Proceedings of the Hawaii International Conference on System Sciences*, January 1997. http://lorrie.cranor.org/pubs/hicss/hicss.ps.

[Cha81] David Chaum. Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981. http://web.inf.tu-dresden.de/~hf2/anon/Chaum1981/Chaum1981.pdf.

[Cha83] David Chaum. Blind Signature System. In *Advances in Cryptology: Proceedings of Crypto '83*, pages 153–156. Plenum Publishing, 1983.

[DA81] Tim Dierks and Christopher Allen. The TLS Protocol. IETF RFC 2246, January 1981. http://www.ietf.org/rfc/rfc2246.txt.

[ISI81] Information Sciences Institute. Transmission Control Protocol (TCP). IETF RFC 793, September 1981. http://www.ietf.org/rfc/rfc793.txt.

[KKLA01] Kwangjo Kim, Jinho Kim, Byungcheon Lee, and Gookwhan Ahn. Experimental Design of Worldwide Internet Voting System using PKI. In *Proceedings of SSGRR International Conference on Advances in Infrastructure for Electronic, Business, Science and Education on the Internet*, August 2001. http://citeseer.csail.mit.edu/687206.html.

[LK02] Byoungcheon Lee and Kwangjo Kim. Receipt–Free Electronic Voting Scheme with a Tamper–Resistant Randomizer. In *Information Security and Cryptology — ICISC 2002: 5th International Conference*, pages 389–406, November 2002. http://link.springer.de/link/service/series/0558/bibs/2587/25870389.htm.

[OMA+99] Miyako Ohkubo, Fumiaki Miura, Masayuki Abe, Atsushi Fujioka, and Tatsuaki Okamoto. An Improvement on a Practical Secret Voting Scheme. In *Lecture Notes in Computer Science*, volume 1729, pages 225–234, 1999. http://link.springer.de/link/service/series/0558/bibs/1729/17290225.htm.

[Rie98] Andreu Riera. An Introduction to Electronic Voting Schemes. Technical Report PIRDI 9-98, Universitat Autònoma de Barcelona, September 1998. http://pirdi.uab.es/document/pirdi9.ps.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A Method for Obtaining Digital Signatures and Public–Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. http://theory.lcs.mit.edu/~rivest/rsapaper.pdf.