Equitably Fair Internet Voting

Emmanouil Magkos[1] and Vassilios Chrissikopoulos[2]

1. Department of Informatics
University of Piraeus
80 Karaoli & Dimitriou
Piraeus 18534
Greece
FAX : +30 1 4142264
Phone : +30 1 4142134
email: emagos@unipi.gr (CONTACT AUTHOR)

2. Department of Archiving and Library Studies
Ionian University
Old Palace Corfu, 49100
Greece
email: vchris@ionio.gr

Equitably Fair Internet Voting

Abstract. With the advent of Internet Communications Technologies (ICT), the use of cryptographic protocols is a technical response to the loss of all traditional means that were used so far to establish security in democratic elections. We employ simple cryptographic techniques to address the *"abstaining voters"* problem in electronic elections with central administration. In such elections, voting authorities can cast a bogus vote on behalf of an authorized voter who decides to abstain. Our system is *equitably fair*: while a voter who registers for the election is allowed to abstain from voting (legal abstention), all registered voters who cast an encrypted vote must acknowledge, at some time later, the fact that they have voted. If not (illegal abstention), a cryptographic *time capsule* will be broken and their identity will be disclosed. Our system satisfies most requirements of a secure election and could be used in similar frameworks such as electronic polling and/or surveys over the Web.

Keywords. Electronic voting, equitability, public key cryptography, time capsules, anonymous channels.

1. Introduction

With the advent of Internet Communications Technologies (ICT), electronic voting will become universally accepted in the upcoming years. Although current implementations for Internet elections are rife with security problems, research community promises to address such problems. Furthermore, the use of cryptographic protocols seems to be a technical response to the loss of all traditional means that were used so far to establish security in democratic elections.

In this paper, we employ simple and well-known cryptographic techniques to address the *"abstaining voters"* problem in electronic elections with a simple voting authority. This kind of elections with central administration has been seen as the most promising solution for Internet voting, because it offers efficient administation and demands low complexity of computation. In such elections, during registration, a voter is authenticated in a way that there can be no link between the final vote and the identity of the voter. This is achieved by using *blind signatures* [1] and *anonymous channels* [2]. A drawback for all systems proposed so far is that if a voter is registered for the election but then decides to abstain, the voting authority can cast a bogus vote on behalf of the abstaining voter and get away with it. In such systems, impractical assumptions have to be made, e.g. that all registered voters who wish to abstain submit a blank ballot.

We propose a cryptographically secure election with central administration, and solve the problem of abstaining voters, while preserving vote secrecy and verifiability for the final results. During the election, all registered voters who cast an encrypted vote must also *cast an acknowledgment*, i.e. use their authentic digital signature to acknowledge, at some time later, the fact that they have voted for the election. At the end of the voting phase, there have to be as many encrypted votes as signed acknowledgements, so that everyone is sure that the voting center did not submit any bogus votes. The identity of a faulty voter who does not cast an acknowledgement will be revealed, and a penalty will be imposed.

While it is *fair* for someone who submits an encrypted vote to abstain from the election thereafter, it is not *equitable* towards the "society". With "society" we mean all voters, authorities and independent observers who wish to independently verify the correctness of the election results. Our election is *equitably fair*[1]: voters are allowed to abstain, as long as they have not submitted an encrypted vote (legal abstention). If a voter submits a vote anonymously, then his anonymity is conditionally protected; at some time later, the voter must cast an acknowledgment, otherwise his identity will be disclosed (illegal abstention). At this time, allowing a voter to abstain from the election would be as fair as allowing voters in

[1] Equitable fairness was first introduced in the context of private auctions [3].

traditional elections to vote without signing on the voters' list. For this reason we make use of cryptographic *time capsules*. These are containers that hide the voter's identity; If a voter abstains illegally, an authority has to spend a specific amount of time to reveal the hidden identity. This time should be long enough to prevent the authority from massively breaking the privacy of the voters and little enough to ensure that misbehaving voters will be eventually identified, without obstructing the election process.

Besides equitability, there are several requirements for a secure electronic election:

Privacy:
1: No one can link a vote to the voter.
2: All votes remain secret while voting is not completed.
3: No voter can prove the content of his vote.
Verifiability:
All users can verify the correctness of the results.
Invulnerability:
1: Only eligible voters vote.
2: Each eligible voter can vote only once.
Accuracy:
1: No vote can be altered.
2: No vote can be eliminated
3: An invalid vote cannot be counted.

Our system satisfies most requirements of a secure election and could be used in similar frameworks such as electronic polling and/or surveys over the Web.

The rest of the paper is organized as follows: Section 2 discusses the *"abstaining voters"* problem in elections with central administration and synopsizes most election schemes that suffer from this problem. Section 3 describes the basic building blocks that will be used for the election system presented by Section 4. Section 5 evaluates the system from a security point of view. Conclusions are offered in Section 6.

2. Related Work

Chaum [2] was the first to propose a cryptographic scheme for conducting electronic elections. Since then a few cryptographic schemes for electronic voting have been proposed, especially during the recent years. Among them, much attention has been given to those schemes that employ *blind signatures* and *anonymous channels* [4-12]. In these schemes, the computation overhead is fairly small and administration is very efficient. Furthermore, they naturally realize multiple-value voting and are compatible with other frameworks such as electronic polling and/or electronic surveys. To our knowledge, the only schemes that have been actually implemented are using blind signatures and anonymous channels [4-6]. A voting scheme of the above category works as follows: a voter constructs a commitment for his vote, namely a *vote-tag*. Then, the voter blinds the vote-tag. At high level, this can be seen as sealing the tag on a carbon-paper envelope and submitting the envelope to the voting authority, during a *registration phase*. The authority authenticates the voter, then signs the envelope on the outside and returns it to him. The voter opens the envelope and gets a validated vote-tag. At some time later, during a *voting phase*, the voter uses an anonymous channel to submit the vote and validated vote-tag to the authority. During a *tallying phase*, the authority publishes the list of [votes, vote-tags] on a publicly accessible board.

All voting schemes mentioned above suffer from the *"abstaining voters"* problem: the authority can cast bogus votes on behalf of voters who register but then decide to abstain from subsequent steps. Obviously, both the first invulnerability and the third accuracy requirements are violated with this attack. To deal with the attack, it is often assumed that all abstaining voters actually submit a blank ballot. This is a cumbersome assumption. Below, we show how this attack may undermine the second accuracy requirement too.

Recently, Riera [12] suggested that all voters should submit, at the beginning of the voting phase, a signature to declare that they possess a validated vote-tag. Then, the voting phase takes place, and voters submit their votes using an anonymous channel. The voting authority will publish the list of [votes, vote-tags], together with a shuffled list of signatures. This solution has the following drawback: after the publication of the results, if there are more votes than signatures, it could either mean that the authority has published some bogus votes, or that some voters abstained from submitting a signature prior to submit a vote (we call this a *reverse abstaining* scenario). On the other hand, if there are more signatures than votes, it could either mean that the authority has *eliminated* some votes from the final tally, or that some voters did not submit a vote after submitting a signature. In Section 4 we will present an equitably fair election that solves the "abstaining voters" problem.

3. Building Blocks

Blind Signatures. Blind Signatures are the equivalent of signing carbon-paper-lined envelopes. A user seals a slip of a paper inside such an envelope, which is later signed on the outside. When the envelope is opened, the slip will bear the carbon image of the signature. The notion was invented by Chaum [1], who also was the first to implement blind signatures using the RSA algorithm [13].

In order to establish correctness in a blind signature protocol, a *cut-and-choose* technique [13] can be used: Alice sends *m* blinded messages to Bob, then un-blinds any *m-1* indicated by him. Bob will sign the remaining message. There is a tradeoff between choosing a large *m* (strong correctness), and a small *m* (efficiency).

Cryptographic Time Capsules. A cryptographic time capsule, also known as *timed-released crypto* [14] is a container which one can embed information into and set a time, so that the computational effort of the set time is required by the receiver to recover the information contained in the capsule. On the other hand, the constructor of the capsule possesses a *trapdoor* information that makes it trivial to construct the capsule. A very useful attribute of time capsules is that the computational effort required by the receiver cannot be parallelized. This stands because breaking the capsule usually involves the computation of a number of the form: $X = a^{2^t} \pmod{n}$, where n is a large composite number, a is a random number and t denotes the number of squarings required to compute the capsule. Because each squaring can be performed on the result of the previous squaring, it is not known how to speedup the t squarings via multiple processors.

In [15,16], methods for constructing time capsules for the RSA public key cryptosystem are described. Especially in [15], a very efficient mechanism is presented, where the receiver is convinced in *zero-knowledge* [13] that the capsule contains an RSA encrypted message, which can be recovered after a specific time. Another method for implementing a time capsule can be found on [17], where part of the key that encrypts the message is escrowed to a trusted party.

Anonymous Channels. Anonymous channels cannot be traced (e.g., by using traffic analysis). For example, e-mail anonymity can be established using *Mix* networks [2]. HTTP anonymity can be established using services such as the *Anonymizer* [18], *Crowds* [19], the Lucent Personalized Web Assistant (*LPWA*) [20], and *Onion-Routing* [21]. LPWA and Onion-Routing can handle e-mail in addition to HTTP. Onion-Routing also supports "reply onions" that allow anonymous replies to be sent in response to a previously received anonymous mail.

4. The Voting System

The participants are the voters and the voting center. Voters communicate with the voting center by using both authenticated and anonymous channels. We assume that all entities are

bound by their signatures and that a *Pucblic Key Infrastracture* (PKI) is already in place. The voting center uses a bulletin board for public announcements.

Before the election begins, the voting center announces the parameter t that will be used by all voters for the construction of the time capsule. This parameter reflects the time that will be needed by the voting center to break a capsule. This time should be long enough to prevent the voting center from massively breaking the privacy of the voters and little enough to ensure that identification of illegally abstaining voters will eventually succeed, without obstructing the election process.

There are four distinctive phases that take place in the voting protocol, namely *registration*, *voting*, *claiming* and *tallying* phase. These are depicted in Figure 1. We also present the notation used throughout this section:

- V : the Voter.
- VC : the Voting Center.
- $S_X(m)$: a signature on message m with the secret key (e.g. RSA [13]) of x.
- *blind(m)* : blinding of a message m.
- TC_V : a cryptographic time capsule that contains the identity of V.
- *tag* : a vote-tag, for example this could be a hash[2] (e.g. MD5 [13]) of the vote.
- *elect_id* : an identification number that uniquely identifies the election.

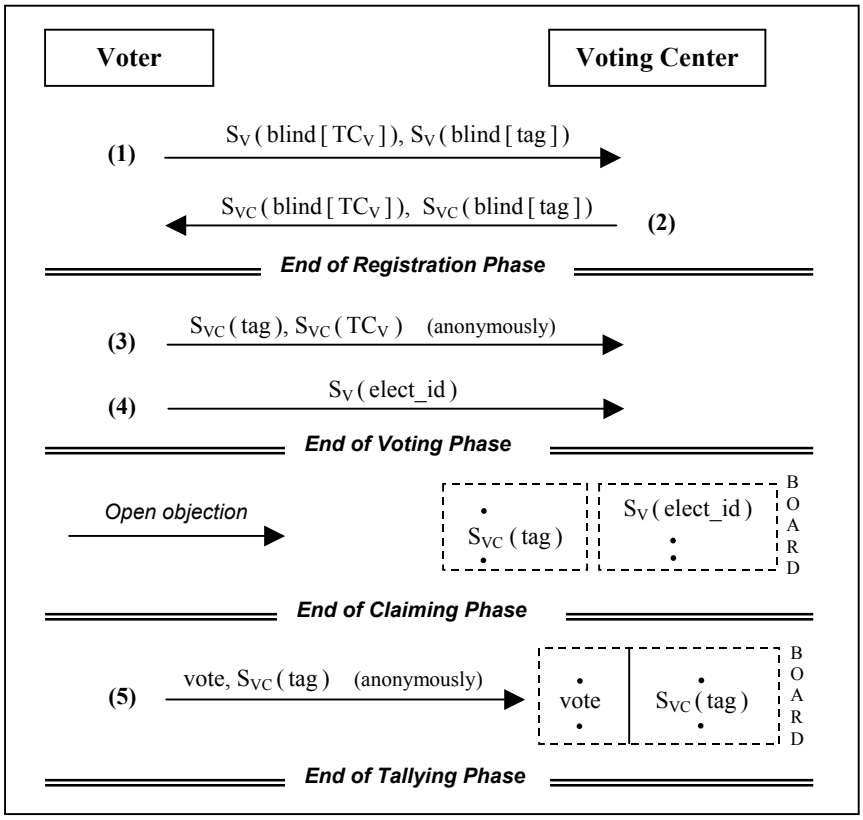

Fig. 1. An Equitably fair Voting Protocol

Registration Phase. A voter, say Victor, uses an authenticated channel and engages on a blind signature protocol with the voting center (VC). Victor constructs and blinds a time capsule of his identity, TC_V, and a commitment of his vote, *tag*. Victor signs both messages with his authentic signature key and then submits them to VC in Step 1. The VC verifies

[2] If hash functions are to be used for commitments, then the vote must be combined with some random padding prior to be given as input to the hash function. Thus, *known-plaintext* attacks [13] will be excluded.

Victor's signature, then validates the blindings and submits them back to Victor in Step 2. For the correctness of the blindings, a cut-and-choose protocol can be used. This guarantees that TC_V can be solved back to Victor in case he misbehaves, while at the same time the VC cannot link the capsule with Victor directly.

Voting Phase. At this point Victor can decide whether he wishes to abstain from the election or not. If Victor wishes to participate, he uses an anonymous channel to submit, in Step 3, the validated tag and capsule to the voting center. This is the "*point of no return*" for Victor. At some time later[3], Victor has to acknowledge his participation to the election by submitting in Step 4 his authentic digital signature on *elect_id*.

At the end of this phase, the number of the signatures must be equal to the number of vote-tags. If not, the time capsule of the abstaining voters will be solved and a penalty will be imposed.

The following procedure takes place in case of abstaining voters: the VC asks for all voters that cast an acknowledgment in Step 4 to submit the trapdoor information for their time capsule that was submitted in Step 3. If a voter cannot identify his validated capsule among the list of all capsules, then he is subject to a penalty (see also the *reverse abstaining* scenario, discussed in Section 2). Otherwise, his name and corresponding vote-tag will be excluded from the list of abstaining voters. All remaining time capsules have to be solved in order to identify the abstaining voters. The penalty should be heavy enough to deter voters from illegally abstaining from the election.

The cost paid for the above solution is that, in case of abstaining voters, the registration and voting phase have to be repeated, for all voters. We believe that this is a minimal cost for solving the "abstaining voters" problem.

Remark 1. So far we assumed that the VC is honest and will not repudiate the receipt of a valid message. Consider a possible scenario where Victor casts an acknowledgment in Step 4 but a faulty VC denies the receipt of the acknowledgment and decides that Victor is an illegally abstaining voter. To deal with this attack, Victor could make use of a *certified delivery* service. Such services are widely used in Internet applications (e.g. [22]). The same service can be used for the submission of the validated tag and capsule in Step 3.

Remark 2. To relieve the VC from the computational burden, there could be a trusted independent authority that will be responsible for breaking the capsules, in case of an illegally abstaining voter. Alternatively, in a distributed scenario voters would give portions of their trapdoor to a set of trusted distributed authorities (e.g. by using secret sharing techniques [23]) that would have to cooperate in order to break the capsule of a specific voter.

Claiming Phase. The VC publishes a list of all validated vote-tags and a shuffled list of the signatures. Anyone can verify that the VC did not submit any votes on behalf of voters who decided not to participate to the election after their registration (legally abstaining voters). If Victor's vote-tag is not published, Victor can broadcast his validated vote-tag and make an *open objection* to the tally (i.e. without giving away his vote). This is the main reason why the claiming phase was separated from the tallying phase: if the results were first published, then the act of complaining would indirectly reveal the exact vote itself, since Victor would not trouble about making an objection if the election results were favourable. The idea of separating the claiming phase from the tallying phase is due to Sako [24].

Tallying Phase. In Step 5, Victor uses an anonymous channel to submit his vote and the validated vote-tag to the VC. Optionally, Victor may use a *certified delivery* service to be sure

[3] If Victor does this immediately after Step 3, then his identity could be indirectly linked to a specific tag.

that the VC will not deny the receipt of the vote. At the end of the Tallying phase, the VC publishes the election results.

Observe that Victor may be allowed to abstain from Step 5. This does not affect the security of the election, since the VC is not able to submit a vote on behalf of Victor at this particular point. To do that, the VC would have to inverse the vote-tag, e.g. break the security of a hash function, which is computationally infeasible.

5. Security Considerations

The protocol presented in Section 4 fulfills most requirements of a secure election. It is *equitably fair*, i.e. it provides protection for both voters and "society" (all honest voters, the voting center, and independent observers) against malicious behavior by any number of participants.

Privacy. The blind signature mechanism (Steps 1-2 in the figure) conceals the link between the vote-tag and the voter who submits it. Note that even if the identity of a faulty voter is revealed at the end of the voting phase, his vote is still protected, since vote-tags are not opened until the tallying phase. Furthermore, the second privacy requirement is always achieved: voters will not submit their cleartext vote until Step 5.

Remark 3. In this paper we do not deal with the third privacy requirement, also known as *uncoercibility*. Uncoercible elections seem to be feasible only under hardware assumptions [25]. Such assumptions are outside the scope of this paper.

Vefifiability. All participants can verify the results of the election. In addition, during the claiming phase the voting center publishes all submitted vote-tags and acknowledgements, so that an independent observer can verify that the voting center did not submit any bogus votes. Additionally, during the tallying phase, the voting center publishes all votes and vote-tags. There can be no inconsistencies between results of the two phases; otherwise, the voting center will be held responsible.

Invulnerability. In Step 1, all voters use their authentic digital signature to engage on the blind signature protocol with the voting center. The voting center checks eligibility of users who apply for a validated vote-tag. Furthermore, the voting center will not issue more than one validated [tag, capsule] pair for a given voter.

Accuracy. We solve the "abstaining voters" problem by requiring that each voter casts an acknowledgement. If a voter has already submitted a vote-tag but abstains from casting an acknowledgment, (illegally abstaining voter), then his identity will be disclosed at a given time, and a penalty will be imposed. At the end of the voting phase, there will be as many vote-tags as signatures, so that anyone can be sure that the voting center did not count a bogus vote.

Voters can make an open objection to the tally in case the voting center has altered or eliminated their votes or vote-tags. Optionally, voters may use a certified delivery service, in Steps 3, 5, to deal with the case of a faulty voting center that denies the receipt of a message.

6. Conclusion

Modern societies more and more rely on computer systems and networks. The development of cryptographic techniques allows us to "computerize" many human activities, such as voting in democratic communities. However there is still a need for the improvement of

electronic voting schemes in order to ensure that we have a secure and practical voting scheme.

In this paper we employed simple and well-known cryptographic techniques to address the "abstaining voters" problem in electronic elections. Our protocol is *equitably fair*. while a voter who registers for the election is allowed to abstain from voting (legal abstention), all registered voters who cast an encrypted vote must also cast an acknowledgment of the fact that they have voted for the election. Each encrypted vote is accompanied with a cryptographic time capsule that hides the identity of the voter. If a voter submits a tag but decides not to cast an acknowledgement (illegal abstention), then an authority (it could be the voting center) will break the capsule and the voter will be subject to a penalty. In orde to break the capsule, the authority will have a computer running for a specific time amount. This time should be long enough to prevent the authority from massively breaking the privacy of the voters and small enough to respond the needs for a practical election system.

Our system satisfies most requirements of a secure election and could be used in similar frameworks such as electronic polling and/or surveys over the Web.

Acknowledgments

This work is partially supported by the Secretariat for Research and Technology of Greece.

References

[1] D., Chaum: Blind Signatures for Untraceable Payments. Advances in Cryptology--CRYPTO '82, Plenum Press, 1982, pp. 199--203.

[2] D., Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM, Vol. 24, No. 2, 1981, pp. 84--88.

[3] E., Magkos, M., Burmester, V., Chrissikopoulos: An Equitably Fair On-Line Auction Scheme. 1st International Conference on Electronic Commerce and Web Technologies, EC-Web 2000, LNCS Vol. 1875, Springer-Verlag, 2000, pp. 72-83.

[4] M., Herschberg: Secure Electronic Voting Using the World Wide Web. Master's Thesis, Massachusetts Institute of Technology, 1997, http://theory.lcs.mit.edu/~cis/theses/ herschberg-masters.pdf

[5] L., Cranor, R., Cytron: Sensus: A Security-Conscious Electronic Polling System for the Internet. Hawaii International Conference on System Sciences, Wailea, Hawaii, 1997, http://www.research.att.com/~lorrie/pubs/hicss/hicss.html.

[6] B. Davenport, A. Newberger, and J. Woodard: Creating a Secure Digital Voting Prococol for Campus Elections. Princeton University, 1996, available at http://www.princeton.edu/bpd/ voting/paper. html.

[7] Q., He, Z., Su: A New Practical Secure e-voting scheme. 14th International Information Security Conference, IFIP/SEC' 98, 1998.

[8] W., Juang, C., Lei: A Secure and Practical Electronic Voting Scheme for Real World Environments. IEICE Transactions Fundamentals, Special Section on Cryptography and Information Security, Vol. E80, No. 1, 1997, pp. 64--71.

[9] W., Juang, C., Lei: A Collision-free Secret Ballot Protocol for Computerized General Elections. Computers & Security, Vol. 15, No. 4, 1996, pp. 339--348.

[10] P., Horster, M., Michels, H., Petersen: Blind Multisignature Schemes and their Relevance to Electronic Voting, 11th Annual Computer Security Applications Conference, IEEE Press, 1995, pp.149--156.

[11] T., Okamoto: Receipt-Free Electronic Voting Schemes for Large Scale Elections. Workshop of Security Protocols'97, LNCS Vol. 1163, Springer-Verlag, 1996, pp. 125-132.

[12] A., Riera: An Introduction to Electronic Voting Schemes. Technical Report PIRDI-9/98, University of Barcelona, October 1998, available at http://pirdi.uab.es/document/pirdi9.ps.

[13] B., Schneier: Applied Cryptography--Protocols, Algorithms and Source Code in C, 2nd Edition, 1996.

[14] R., Rivest, A., Shamir, D. Wagner: Time-Lock Puzzles and Timed-release Crypto. Manuscript, March 1996, available: http://theory.lcs.mit.edu/~rivest/RivestShamirWagner-timelock.ps.

[15] W., Mao: Timed-Release Cryptography. Technical Report HPL-2001-37, Hewlett-Packard Laboratories, United Kingdom, March 7, 2001.

[16] D., Boneh, M., Naor: Timed Commitments. Advances in Cryptology, CRYPTO-'2000, LNCS 1880, Springer-Verlag, 2000, pp. 236--254.

[17] M., Bellare, S., Goldwasser: Encapsulated Key-Escrow. MIT Laboratory for Computer Science Technical Report 688, April 1996, available at http://www.cse.ucsd.edu/ users/mihir.

[18] Community ConneXion, Inc., http://www.anonymizer.com.

[19] M., Reiter A., Rubin: Crowds, Anonymity for Web Transactions. DIMACS Technical Report 97-15, April 1997, http://www.research.att.com/projects/crowds/.

[20] The Lucent Personalized Web Assistant, http://lpwa.com.

[21] D., Goldschlag, M., Reed, P., Syverson: Onion Routing for Anonymous and Private Communications. Communications of the ACM, Vol. 42, No. 2, 1999, pp. 39--41.

[22] www.certifiedmail.com/.

[23] Y., Desmedt: Threshold Cryptography. European Transactions on Telecommunications, Vol. 5(4), 1994, pp. 449-457.

[24] K., Sako: Electronic Voting Scheme Allowing Open Objection to the Tally. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences Vol. E77-A (1), 1994, pp. 24-30.

[25] E., Magkos, M., Burmester, V., Chrissikopoulos: Receipt-freeness in Large-scale Elections without Untappability Assumptions. 1st IFIP conference on e-commerce, e-business, and e-government, Kluwer Academics Publishers, 2001, pp. 683-693.