A Protocol for Anonymous and Accurate E-polling

Danilo Bruschi Igor Nai Fovino
Andrea Lanzi

Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano, via Comelico 39/41,
I-20135 Milano MI, Italy {bruschi, nai}@dico.unimi.it
andrew@security.dico.unimi.it

Abstract. E-polling systems are a fundamental component of any e-democracy system as they represent the most appropriate tool for fostering citizens participation to public debates. Contrarily to e-voting protocols, they are characterized by less stringent security requirements in particular they can tolerate errors affecting a small percentage of votes, without the compromision of the final result. Thus the construction of accurate and privacy respectful e-polling protocols is an objective which should be pursued by the research community as it is more close than the construction of practical e-voting protocols. However so far all the research efforts have been spent on the construction of e-voting protocols and the existing e-polling protocols are not respectful of the most elementary security and privacy rules. In this paper we propose a simple protocol for an accurate and anonymous e-polling system. More precisely our protocol satisfies the following properties: a vote cannot be altered, duplicated, or removed without being detected, votes remain anonymous. Moreover voters will be able to measure the level of trust of the process and its accuracy by verifying that their own votes have been counted correctly.

1 Introduction

The milestone of any democracy is participation. Obviously, such a postulate holds even in the case of e-democracy. In such a case the most appropriate tool for fostering citizens participation to public debates is based on electronic polling systems, i.e. systems which enable people to express their opinion on specific issues. The model of direct democracy, which is hindered by nowadays population sizes and state organizations, would then become possible. Besides being a fundamental component of any e-democracy system, e-polling systems are also quite attractive from the technological point of view since they do not have stringent security requirements as the e-voting protocols, thus they can be implemented and deployed more easily. In particular, we remind that the most crucial properties which an e-voting protocol has to satisfy are: *Democracy* i.e. only eligible voters are permitted to vote, and they can do so only once, *Accuracy* i.e. a voter's vote cannot be altered, duplicated, or removed without being detected, and *Privacy* i.e. votes remain anonymous. A minimum flaw in the complete satisfaction of any of these requirements will result in an unacceptable e-voting protocol, as either the privacy of citizens or the accuracy of the final result will be compromised. It is the absence of any margin of error which makes very difficult the construction of any practical e-voting protocol. Fortunately, such considerations do not hold in the case of e-polling systems. Such systems are mostly interested in capturing general trends and people orientation, thus they can tolerate errors affecting a small percentage of votes, without the compromission of the final result. Obviously, they have to be designed in order to satisfy all the above mentioned properties (democracy, accuracy and privacy), considering however that some level of imprecision or some misbehavior can be tolerated.

Electronic polling systems have recently appeared on the Internet (e.g. www.epoll.com, www.misterpoll.com). However, such systems are not designed to maintain the level of security and privacy that we would expect, for example there is no way to express an opinion anonymously and there is no way for verifying the accuracy of the results.

In this paper we devise a new protocol for implementing a polling system based on the notion of anonymous credential. We briefly recall that an anonymous credential (introduced in 1985 by Chaum [12]) is an authorization token released by a trusted party, which entitles the owner to perform a specific set of activities on a particular system, without revealing his/her real identity.

Roughly speaking, our protocol works as follows. Any legitimate voter is provided by an anonymous voting credential, released by a Trusted Authority, which enables him to anonymously access to a poll system, once and only once, and to participate to a voting session. Anonymous votes are collected by the system and subsequently counted and published. In order to verify the accuracy and the level of trust of the entire process, a mechanism has been devised which enables any voter to verify that its own vote has not been altered and has been correctly counted in the final tally. The protocol is based on the HTTP protocol and it is transparent to both the browser and the web server, which means that it can be easily implemented using any web server and web browser.

It is not difficult to see that such a protocol satisfies the properties of accuracy and privacy. The accuracy property is satisfied by providing any voter with a mechanism for verifying that his own vote has been correctly counted, while the privacy property is guaranteed by the use of anonymous credentials. However, the use of anonymous credential has a well known drawback know as credential sharing. By credential sharing we mean the possibility that the legitimate user of a credential transfers it to another user which will use it. From the point of view of our protocol this would mean that either a non legitimate user will be able to participate to a poll session or a legitimate user will have the opportunity of voting more than once; in both cases the democracy property will be infringed. This is a very serious problem in the case of voting system, however we believe that it is not in the case of a polling system for the following reasons. In order to influence the final tally of a poll a user should collect too many credentials and the effort would not be comparable with the result, people participate in polling session on a voluntary basis in order to provide their own contribution to a decision process, thus there is no reasons for a user to enroll in a polling session for giving up its own credential to another user.

This paper is organized as follows. Section 2 describes the notation used throughout the paper and provides the definitions of peculiar concepts used for designing the protocol. Section 3 contains a detailed description of the credential system adopted. Section 4 is devoted to the protocol description. Section 5 contains a correctness analysis of the protocol. Section 6 is a brief overview of the state of the art in the field. As so far all the research efforts have been dedicated to voting protocol, and no attention has been deserved to polling protocols, we will briefly describe the most important achievements in the field of voting protocols.

2 Preliminary Definitions

In the paper we will frequently refer to the following terms:

- ***Voters****: the subjects interested in participating to a polling . The number of voters is a priori unknown.*
- ***Authentication token****: any mechanism used for verify the identity of a user such as password, Digital Certificates, biometric.*
- ***Vote Certificate*** *a digital certificate which witness the eligibility of a user to participate to a polling session.*
- ***Polling Server****: the unit that collects the votes.*
- ***Trusted Third Party (TTP)****: an entity which guarantees the eligibility of a voter and releases the vote certificates.*

We assume that TTP is provided by a pair of asymmetric keys (public and private) $< K_{pub\_TTP}, K_{priv\_TTP} >$, and K_{pub_TTP} is known to any polling server. By $S^C(m)$ we denote the message m digitally signed by C with RSA algorithm. We also assume that the TTP is able to recognize any of the authentication tokens belonging to the voters and distributes Vote Certificates. In our protocol we will use the blind scheme signature on RSA algorithm, as introduced in [11].

We briefly recall, that through the blind signature operation a party A can obtain a digital signature on a message m from a party B without revealing the content of m. The blind signature of a message m by B will be computed as follows. Let (e, n) and d be respectively the public and private RSA keys of B; initially A chooses a random number $rand$ and sends the following quantity M to B:

$$M = (m * rand^e) \ mod \ n \ . \tag{1}$$

Once B receives M it signs M with its own private key d and sends the result to A, i.e.

$$S^B(M) = (M)^d \ mod \ n \ . \tag{2}$$

Once A receives $S^B(M)$ it performs the following transformations, which will enable him to obtain a "packet" containing the original message m digitally signed by B:

$$S^B(m) = \frac{S^B(M)}{rand} = m^d \ mod \ n \ . \tag{3}$$

3 The Credentials System

As just mentioned we adopted the mechanism of anonymous credential for satisfying the anonymity property. The credential system we devised for our protocol works in the following way.

Any eligible voter , contacts off line, the TTP, which after a face to face authentication, releases to the user a Voting Certificate i.e. a secret random number r, which is stored by the TTP in a database together with a user authentication token (for example the SHA1 of a password or of a X.509 certificate); this random number r is used to unlock the pseudo credential only who knows the secret r (real voter). Once a user has performed such a prescription phase, he is entitled for getting a vote credential. Such a credential is represented by an anonymous certificate signed by the TTP, which enables the bearer to perform a single polling operation.

When a voter needs an anonymous credential for accessing a polling server, he contacts the TTP, and exhibits his Voting Certificate. Subsequently he chooses an integer *cred*, calculates the SHA1(cred) and start the blind signature process on such a quantity equation (1). The TTP, according with the blind signature scheme explained above equation (2), signs the quantity received by the voter and multiply the result by a random number r and sends it to the user, we call such a quantity *pseudo credential*. More formally this is the message exchange among the entities involved in this phase, which we recall is performed on an encrypted channel [10].

User→ TTP : $\{(\mathrm{SHA1}(Voting\_Certificate))\}$
TTP → User : Ack
User → TTP: {M}
TTP→ User: $\{S^{TTP}(M) * r\}$

where:

(e,n) = RSA TTP public key
d = RSA TTP private key
m = SHA1(cred)
rand = random number used to perform the blind signature
M = $(m * rand^e)\ mod\ n$
r = random number used to protect the credential

Once the pseudo credential has been received, the client unlock the pseudo credential dividing by random number **r** and then it performs the last phase of blind signature scheme equation (3) and it obtains:

$$S^{TTP}(m) = \frac{S^{TTP}(M)}{rand} = m^d\ mod\ n\ . \tag{4}$$

Then the user builds the anonymous voting credential, that is equal:

$$(cred, S^{TTP}(m))\ . \tag{5}$$

At this point the client can verify the TTP signature on the *voting credential* equation (5). Such *voting credential* has some important properties, first of all it does not contain any information about the identity of the user, thus guaranteeing the privacy property, secondly it can be used as one time credential.

4 The Application Architecture

In this section we provide an overview of the e-polling protocol we devised. When a voter decides to participate to a poll session he/she connects to the polling server using his own browser. A proxy installed on the server will intercept the request, and replies with a request for a voting credential. Using the vote certificate the client will apply to the TTP for a "voting credential". The TTP, after the validation of the certificate sends to the voter a pseudo credential, which would be used by the voter for accessing the polling server. More precisely the protocol works as follows.

1. Once a client requests a service, the Credential Proxy (previously installed on the voter machine (see Fig. 1)) forwards the request to the server. If the server site has not a Validation Proxy (installed only on the polling servers), the communication follows the path of a standard connection. Otherwise, if it is the case, it returns the request for a credential to the Credential Proxy.

2. The credential proxy, establishes an encrypted and mutually authenticated channel (e.g TLS [10]) with the TTP, builds the credential and sends to the TTP the digest of the credential adopting a blind signature scheme.

3. The TTP checks if the owner of the *authentication token*(see sec. 3) has the right to vote, if it is the case, the TTP retrieves from the database the random number r associated to the client and it builds the pseudo credential

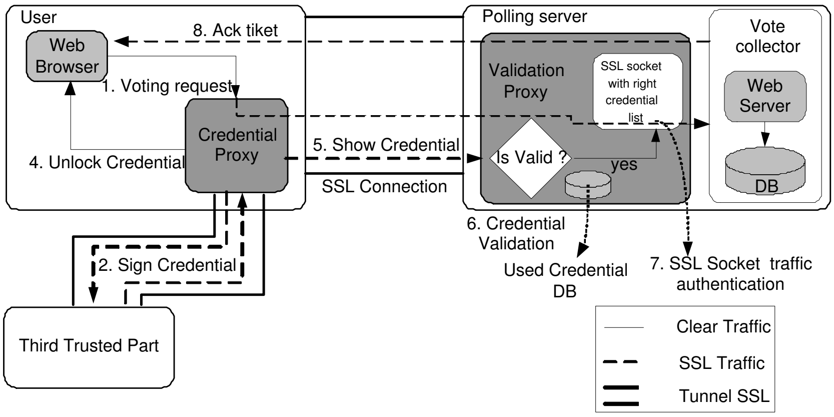

Fig. 1. The Application Logical Scheme

described in the previous section; Finally it sends the pseudo credential to the Credential Proxy.

4. The Credential Proxy unlock the pseudo credential and then it builds the anonymous credential.

5. The Credential Proxy establishes an encrypted communication channel with the polling server (only server side authentication to preserve its anonymity), the poll server sets the state of the connection pending, and then the Credential proxy sends the anonymous credential to the Validation proxy.

6. The Validation proxy (on the Poll server) checks the TTP's sign on the credential, if it is valid it checks if it has been already used, if this is the case the connection will be closed. In the other case, it stores the credential into database and it sets the authorized state into a special memory structure that is used to manage the connections.

7. The voter performs its vote and then the Credential Proxy creates a number (that will be used as *control ticket*) and it sends the digest of this number with the ack ticket request to the Validation proxy, the poll server signs this digest and sends it to the voter. At this point the voter checks the sign on the ack ticket, stores it and closes the connection.

8. At the end of the polling a web page containing all the votes received associated with the relative *control ticket* will be published. The voter in order to control if the vote has not changed during the vote operation, can control if the vote associated to the ticket is right.

5 Correctness Analysis

In this section we show that the protocol we have devised satisfies the properties of accuracy and privacy as defined in Section 2.

Property 1. A voting credential can be used only once

Such a property can be violated only if a vote credential is used more than once, i.e. either when a voter request more vote credentials for the same polling or when a signed credential is used more times.

As explained in Sec. 6 this is not possible as the TTP flags any user which receives a vote credential and the polling server collect all the used credentials in order to discover the multiple use of a credential.

Property 2 (Accuracy violation). The protocol satisfies the accuracy property

Such a property can theoretically be violated in two ways. Either performing a man in the middle attack on the channel connecting the client to the polling server or compromising the polling server itself. However the first attack cannot be performed as we adopted an encrypted channel with mutual authentication. Instead the second attack is avoided by the release of a receipt, that is the proof that a correct vote operation is performed; thus the voter has the possibility to control in an anonymous way if his vote has been correctly counted.

Property 3 (Privacy violation). The protocol satisfies the privacy property

Two are the possible point in which the privacy property can be violated:

1. *The network between client(voter) and polling server*
   In this case the threat is related to the risk of network sniffing. Adopting encrypted channel (e.g. TLS [10]) it is possible to avoid this threat.

2. *The polling server*
   Assuming that the polling server is corrupted, the voter privacy (at logical level) is guaranteed by the use of the anonymous credential scheme. This scheme is not enough in order to protect the network privacy(e.g. ip address); in this case a solution can be the adoption between client and polling server of a Crowd scheme [9].

In the case in which we assume that the TTP is not really trusted, a collaboration between TTP and polling server must be taken in account. In our system the TTP does not know the credential that he signs (a blind signature scheme is adopted), then the collaboration between the two servers cannot product any type of information that can violate the privacy of the voter.

6 Related Results

As just mentioned, we are not aware of specific papers which address the construction of e-polling protocols, as all of the efforts in such a field has been concentrated on the construction of e-voting protocols. Historically, three are the main approaches adopted to solve the challenge of electronic vote:

- Chaum's Mix-nets scheme [2] based on the concept of network permutation in order to preserve the voter privacy obfuscating the link.
- Chaums's Blind signature[12] in which, as in our case, the notion of blind signature is heavily used.
- Homomorphic schemes in which properties of probabilistic cryptosystems are used [16].

We will now analyze such approaches in the context of e-polling protocols, starting from the consideration that for e-polling protocols, the time efficiency and simplicity are two important goals. For these reasons we believe that approaches based on Mixnet (see for example [15] [17] [18]) and homomorphic functions (see for example [19] [13] [20] [21]) are too complex and time consuming for being adopted in polling systems. In the following we report some of the attempts performed in devising voting systems, which constituted a point of reference for our research on e-polling systems. We will first describe the cryptographic protocols proposed so far and then we will briefly describe the implementations we are aware of.

The first electronic voting protocol was published by Chaum in 1981 [2]. It relies upon public key cryptography as all electronic voting protocols, however it does not guarantee voters' privacy. Chaum then proposed a protocol which unconditionally conceals voters' identity [3] but the entire voting procedure could be disrupted by a single voter. A solution to this problem was suggested by Cohen in [4], but the protocol proposed is not simple (college-level mathematics is required for voters to independently verify election results) nor efficient. In 1994, Benaloh and Tuinstra [1] proposed a voting protocol which allows voters to easily verify the results but voting booths must be used, thus violating the mobility property. Nurmi, Salomaa, and Santean [8] designed a protocol, known as Two Agency Protocol, that preserves the ease of verification properties and relax the "booth" constraint, thus yielding a protocol where voters can easily verify the results. However, it lacked voters' privacy. A breakthrough in the design of electronic voting protocols was realised by Fujioka, Okamoto and Ohta [6], who solved the privacy problem of the Two Agency Protocol using the blind signatures technique introduced in 1982 by Chaum. The Fujioka, Okamoto and Ohta protocol is generally indicated in the literature as the first practical electronic voting scheme. It still does not address the problem of preventing administrators from casting votes for abstained voters. The problem could be solved if abstained voters were forced to cast blank votes, which is clearly a hardly practical solution. Moreover, in this scheme the voter preference is encrypted before to be sent to the vote recipient. This implies during the counting phase that the voter anonymously sends the key to decrypt the vote. The presence of this additional phase in the usual voting scheme is really not practical. Horster, Michaels and Petersen in [14] maintaining the original schema of Fujioka, eliminate this phase adopting a blind multi signature scheme. In this work the presence of more than one administrator (that has the task to sign the vote according the blind scheme) is required. The security in this case is based on the concept that at least one of the administrators is honest. The SENSUS protocol [5], which was implemented and tested with simulated elections, overcomes the abstained voters problem. Karro and Wang [7] proved that SENSUS too suffers from some drawbacks such as the lack of accuracy and proposed another protocol that solves the identified problem. However, from our perspective, neither this protocol satisfies the accuracy property since it is possible to impersonate voters. Furthermore, we note that both protocols [5][7] make some impractical assumptions such as the existence of three or four trusted third parties, which must not collude in order to guarantee the correctness of the protocol, and the existence of a trusted third party that generates the cryptographic keys, which enable the voters to vote but not to cheat.

The design of a flawless electronic voting protocol is very difficult, and no protocols have been designed yet that completely satisfy all the requirements. This negatively influenced the realization of working prototypes. A fairly exhaustive overview of the state of the art regarding voting protocols and their implementations can be found at http://lorrie.cranor.org/voting/hotlist.html. To our knowledge, very little software is available that implements such protocols. Two systems are available, both explicitly written for non-government elections: Sensus and Evox. Sensus is an electronic polling system developed by Lorrie Cranor [5], but after the initial implementation, was never deployed nor maintained. The latter An electronic voting system is under development at MIT based on [6] It was realized by Ronald L. Rivest, Mark Herschberg, Ben Adida, and Randy Milbert. Unlike Sensus, the implementation is being maintained and improved.

References

1. Benaloh, J., and Tuinstra, D. "Receipt-free secret-ballot elections." In Proceedings of the Twenty-sixth Annual ACM Symposium on the Theory of Computing, 1994, pp. 544-553.
2. Chaum,D. "Untraceable electronic mail, return addresses, and digital pseudonyms." Communications of the ACM, 24(2), 1981, pp. 84-88.
3. Chaum, D. "Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA." Lecture Notes in Computer Science, N. 330, Springer-Verlag, 1988, pp. 177-182.
4. Cohen, J. D. "Improving privacy in cryptographic elections." Tech. Rep. YALEU/DCS/TR-454, Yale University, February 1986.
5. Cranor, L. F., and Cytron R. K. "Sensus: A security-Conscious Electronic polling system for Internet." In Proceedings of the Hawaii International Conference on System sciences. Hawaii, 1997.

6. Fujioka, A., Okamoto, T., and Ohta, K. "A practical secret voting scheme for large scale elections." In Advances in Cryptology - AUSCRYPT 92, J. Seberry and Y. Zheng, Eds., Lecture Notes in Computer Science, N. 718, Springer-Verlag, 1993, pp. 244-251.
7. Karro J., and Wang J. "Towards a practical, secure and very large scale online Election." In Proceedings of fifteenth Computer Security Applications Conference, 1999, pp. 161-169.
8. Nurmi, H., Salomaa, A., and Santean, L. "Secret ballot elections in computer networks." Computers & Security, 36(10), 1991, pp. 553-560.
9. Michael K. Reiter and Aviel D. Rubin, Anonymous Web transactions with Crowds, Commun. ACM, vol. 42, number 2, 1999,0001-0782, pag. 32-48, ACM Press,
10. rfc 2246 T. Dierks C. Allen Certicom January 1999
11. Blind signatures for untraceable payments Advances in criptology Cripto '82. LCNS pp. 199-203
12. D. Chaum, Security without identification: transaction systems to make big brother obsolete Communication of the ACM, 1985,vol. 28, number 10, pages 1030, month October
13. M. Hirt, K. Sako Efficient Receipt-free voting based on homomorphic encryption pages 539-556, LNCS 1807, 2000, Springer-Verlag
14. P. Horster, M. Michels, H. Petersen Blind multi-signature schemes and their relevance to electronic voting Proc. 11th Annual Computer Security Applications Conference, New Orleans, IEEE Press, (1995), pp. 149 - 155.
15. C. Park, K Itoh, k. Kurusawa. ALL-nothing Election scheme and anonymous channel. Eurocrypt '93, pages 248-259.Springer-Verlag, LNCS 765, 1993
16. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In Crypto '84, pages 10-18. Springer-Verlag, LNCS 196, 1984
17. K. Sako, J. Kilian. Receipt free mix-type voting scheme. Eurocrypt '95, pages 392-403. Springer-Verlag, LNCS 921, 1995
18. M. Jakobson, A. Juels, R. L. Rivest. Making mix nets robust for electronic voting by randomized partial checking. USENIX '02, D. Boneh editor, pages 339-353, 2002
19. K. Sako, J. Kilian. Secure voting using partial compatible homomorphisms. Crypt'94, pages 248-259. Springer-Verlag, LNCS 839, 1994
20. I. Damgrd, M. Jurik, A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System, PKC 2001, Springer-Verlag, LNCS pg. 119-136, 2001
21. O. Baudron, P. Forque, D. Pointcheval, G Poupard, J. Stern. Practical multi-candidate election system. PODC'01, pages 274-283, ACM 2001