Browser Based Agile E-Voting System

Sriperumbuduru Kandala Simhalu and Keiji Takeda

Carnegie Mellon Cylab, Japan
{eskay,tkeiji}@cmu.edu

Abstract. In the recent past, in spite of several real world implementations available for Internet [browser-based] e-voting, there seems to be a pattern emerging, one of apathy towards improving voter convenience / participation. The goal of the proposed system evolves from the premise that, there should be a priority shift towards addressing the needs of the Voter, hence, most of the other requirements of this system, such as Security, Anonymity, Universal Verifiability, Individual Verifiability, Receipt-Freeness and Fairness are a direct by-product of this goal. In order to secure a higher voter participation, the proposed system considers the trade-offs between strict adherence to essential properties and practicality / user-convenience. To further secure the voter confidence/trust, a practical approach to Individual Verifiability has been implemented, without compromising the Receipt-Freeness property. So, with such flexibility and consumer-oriented approach, it is but evident that Agility is the hallmark of this project.

Keywords: E-Voting; Anonymity, Individual Verifiability, Agile Voting.

1 Introduction

Internet-based E-Voting has been a topic of intense discussion worldwide in the recent past [1,2,3]. There have been efforts made, both in the academia and the industry to research ways of implementing secure, practical and scalable e-voting systems.

This paper is yet another effort to provide a practical, voter-friendly e-voting system. The motivation behind this paper is to provide self-organized groups, along the lines of the Agile team model[5], with the ability to conduct frequent and practical browser-based polling on myriad topics. In order to meet the requirements of such a practical system, a browser-based system would seem appropriate, given the potential of such an application, in terms of its reach and ubiquity. Practicality forms the foundation stone for formulation of the requirements and subsequently the implementation of this system. One of the reasons for taking up this approach is to increase the voter participation. It has been found that the voter turnout over the last 40 years has been declining[6], which is definitely not desirable for the free world.

2 Browser-Based E-Voting System

There are currently a considerable number of implementations of Internet based e-voting systems[2,3,4,7,8,9,10,11,12,15,16,17,18]. However, only browser based,

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 62 – 69, 2007.

open-source e-voting systems would be considered in this paper. As this paper's primary focus is voter convenience/participation, so, a browser-based system would be appropriate, as it increases the reach. As for open-source systems, the voters can get them evaluated by an expert. A comprehensive study on e-voting schemes is beyond the scope of this paper, however one can find an informative study on electronic voting schemes done by Poovendran et al.[13].

2.1 Existing Browser-Based Open-Source E-Voting Systems

This section deals with the existing browser-based open-source systems. The respective systems are described briefly in terms of features they exhibit. The below mentioned systems were the high-profile ones, as most of them are products coming out of well-acclaimed universities or governments.

2.1.1 Adder

Adder is an homomorphic-based remote Internet voting system[4]. As per[4], it adheres, "to the following design goals:Transparency, Universal Verifiability, Privacy, Distributed Trust. Each procedure is "supervised" by multiple authorities, and the final sum cannot be revealed without the cooperation of a given number of authorities. Any attempt to undermine the procedure will require the corruption of a large number of authorities. Authorities and voters may overlap arbitrarily. Thus, it is possible for the voters themselves to ensure trustworthiness (or have an active role in it)."

2.1.2 Condorcet Internet Voting Service (CIVS)

As per [14], "CIVS is a web-based free Internet voting service that makes it easy to conduct elections and polls on the Web. Each voter ranks a set of possible choices. Combined, these rankings are used to construct an overall ranking that anonymously summarizes the opinions of all voters. "

2.1.3 KOA

As per [8,19,20],"KOA stands for *Kiezen Op Afstand* and denotes an experiment in voting over the Internet conducted by the Dutch government. The experiment was specifically conducted for the European elections in June 2004."

"The KOA system may well be the first Free Software Internet voting system developed for, used by, and subsequently released by a government in the world."

2.2 An Overview of the Proposed System

The system has two distinct roles, the Poll Initiator and the Poll Invitee. The Poll Initiator is responsible for most of the administrative work that goes into the making of a poll, such as creating groups, creating polls and sending ballots to Poll Invitees via e-mail. The Poll Invitee has to click on the link provided to him/her through e-mail in order to vote / authenticate and verify results. Figure(s) 1 and 2, depict the interactions between the Poll Initiator/Poll Invitee and the various system components. The proposed system has two major focus areas, voter participation and Individual Verifiability. The previous related work [27,28,29] which involves the selection of

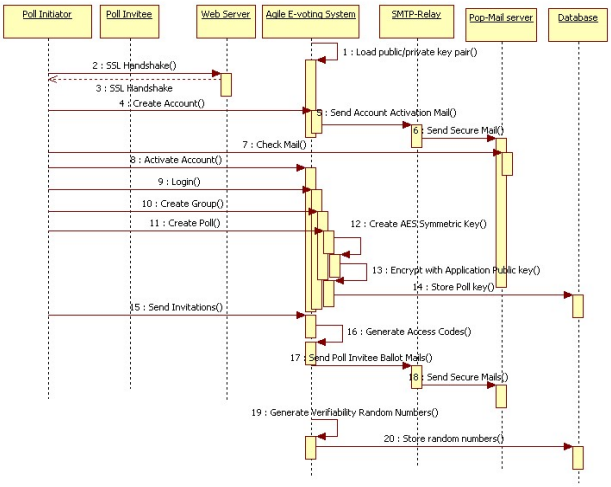

Fig. 1. Poll Initiator Interactions

random numbers apart from the core candidate selection process, either do not provide Individual Verifiability, or they do provide but without maintaining the Receipt-Freeness property. Another related work at MIT[30], has been shown to have serious privacy and security issues[31].

3 *Agile* E-Voting

Below are the properties of the proposed system, and the description of how each of them satisfies the primary requirements of voter participation/convenience.

Security: There are several measures taken to ensure security of the proposed system. First of all, the communication channel between the voter's browser and the server is secured using Secure Socket Layer/Transport Layer Security(SSL/TLS)[21]. Hence, the HTTP traffic from/to the voter to/from the server is now secured. This helps to gain the voters' confidence in the system, as eavesdropping, and to an extent man-in-the middle attack[22,23] can be ruled out. Secondly, the system uses a combination of symmetric key[24] and public key infrastructure[25], to make sure that the vote cast, cannot be tampered with, by adversaries. For each Poll created, an AES[26] key is generated, and the votes cast for that poll are encrypted using the AES key and stored against that poll. This AES key is itself encrypted using the application/owner's public key, and again stored against the poll. So, the vote data stored is secured, as it is now in encrypted form.

Anonymity: In order to overcome the privacy issues, the voter is provided with a unique randomized token for a particular poll that s/he is part of. So, this offers a first level of anonymity, as the voter now is not bound to his/her identity. So, his/her eligibility to vote is only recognized by the random token provided to him/her.

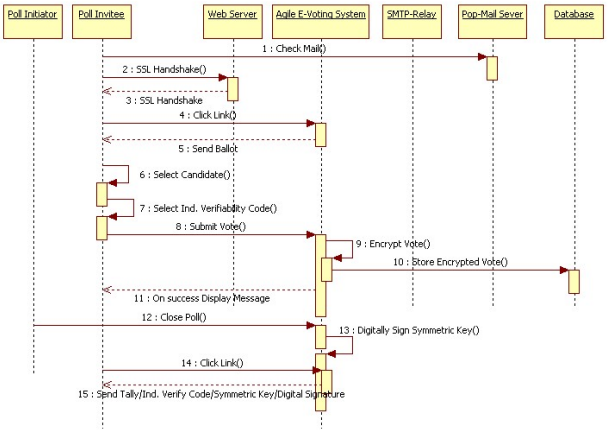

Fig. 2. Poll Invitee Interactions

A Second level of Anonymity is created when the voter uses this token to vote. The final vote is encrypted and stored against the poll, and not against the random token. This provides another layer of anonymity, as now, the encrypted vote cannot be traced back to any particular voter, even if the random token from the e-mail sent to the voter, gets compromised. The design created to implement this requirement would not only allow Anonymity as discussed above, but also would provide the voter with convenient ways to help his/her participation. The token distribution is done using email. The Poll Initiator would send a mail to the voter with a link that would have the token embedded within that link. So, all that the voter has to do, is click that link in order to cast vote. As the vote is cast, it is encrypted and stored against the Poll.

Universal Verifiability: As all of the voters have to check if only valid votes were cast, and no invalid voters were counted, so, during the tally phase, the poll invitee list and the votes cast have to be displayed for verification. Now, this needs to be done while keeping the anonymity property intact. The system design separates the random voter-token storage, and the storage of the vote cast by the voter. Additionally, during the tally phase, the system displays all the encrypted votes, along with the AES symmetric key for that particular Poll with which the votes were encrypted. Also, the digital signature of the symmetric key, signed by the public key of the application/owner is displayed along with the public key itself. Apart from this, the random tokens eligible for the particular poll along with the vote cast status is also displayed. So, this helps the voters to completely verify the polling process for that particular poll.

Individual Verifiability: This property has been the most controversial and difficult one to implement. None of the existing browser-based open-source systems implement this property successfully. However, others such as, Sensus[18] provide Individual Verifiability but at the cost of Receipt-Freeness. Compromising the property of Receipt-freeness would lead to vote buying, as now the voter has a receipt that would identify his/her vote. The proposed system is perhaps the only known browser-based open-source system that provides Individual Verifiability without compromising any

other e-voting property. This is achieved by the system using an additional(optional) set of random numbers displayed alongside the candidates' selection area. The set of random numbers are independent and unrelated to the candidate list. Here, we have two approaches for Individual Verifiability depending on the scale of deployment.

The First approach is only applicable for small scale deployment, say with a voter-to-poll ratio of maximum 20:1. In this approach, during the poll creation, (2 * N) unique random numbers are generated and stored against the poll, where N is the number of voters for that poll. Now, when each voter accesses the system using the token, then along with the candidate list, N random numbers are chosen randomly from the (2 * N) random number set, by the system, and are provided as a single selection input set to the voter. The voter after selecting the candidate, can opt for choosing one random number, out of the set of N. Once s/he submits his/her vote, the selected random number is encrypted and stored alongside the encrypted vote, against that particular poll. Also, the selected random number is deleted from the (2 * N) set, for that particular poll. Hence, for the next voter to poll, the set of N random numbers would come from[(2 * N) – 1], the subsequent voter would get his/her N random numbers from [(2 * N) – 2], and the last voter, would get it from the set of [(2 * N) – N] = N random numbers. This procedure ensures that each of the (2 * N) unique random numbers would appear at least once on any of the voter's N random number list, during the poll.

So, why are we taking such a lot of pain in innovating conditions in this procedure? The reasons are obvious, first, to make sure it is Receipt-Free, so that the voter cannot prove to a candidate at the tally stage, that the random number against the vote is what s/he had chosen. Secondly, to be able to uniquely identify his/her vote during the tally stage.

Now, as it is a remote Internet voting, so the voter has complete control over the machine, hence, the voter has the option of taking a screen shot and using it as a receipt/proof of his/her candidate selection. But, even if the voter selects a random number and takes a screen shot before clicking the submit button, still s/he would not be able to prove to others that the random number chosen was indeed his/her selection.

The reasons being:

1) The voter can change the selection value anytime before voting.
2) As for every voter, a new set of N random values show up on the screen, every time s/he accesses the voting link, before actually submitting the vote.
3) As there is a possibility of the same random number [which was not chosen by previous voters who had cast the vote] appearing on different voter screens, so the screen shot of the random selection loses its authenticity, as a proof of candidate selection.

Hence, the voter cannot prove to others convincingly, about his/her candidate selection, but s/he is completely sure that the random number against the vote displayed during the tally phase is what s/he selected, because, after his/her submission of the vote, that particular random number gets deleted from the (2 * N) list, and would never appear on any subsequent voter's N random number list.

So, now lets move on to the second approach of Individual Verifiability. This approach is applicable for small as well as large scale deployment.

Here, instead of (2 * N) random numbers generated and N random number list provided to the voter, which does not seem to be scalable, as well as user-friendly, now we have N +(P -1) random numbers generated, where N is the number of voters participating for a particular poll, and P is the size of the set of random numbers provided to the voter. The rest of the procedures/reasons/benefits are the same as that of the first approach. The value of N need not be equal to that of the number of voters, we chose N+(P-1), just to make sure with a probability of 1, that any random number shown once on the N-random number list of the voter [which was not selected by any previous/current user], would at least appear once in the list of N-random numbers displayed to the subsequent voters. One can always lower the probability value, and set the value of N and P accordingly in order to make sure that there are not many similar random numbers showing up at the same time on the screen of voters, who might simultaneous access and vote. However, the system takes care of the issue of simultaneous access by voters. First of all, the screen gets refreshed after a certain short time interval, hence a new set of random numbers would get displayed after every refresh. Any random numbers shown on the previous list, if used up by other voters during this time interval, would never show up as part of the N-random number set for this voter as well as subsequent voters.

Secondly, even if two or more users get the same random number and select it as part of their Individual Verifiability, the system would only allow the first request of vote submission to succeed, the subsequent request(s)/voter(s) would be appropriately notified and a new set of random numbers would be displayed for re-submission. Thus, Individual Verifiability with Receipt-Freeness would lead to increased voter participation.

Receipt-Freeness: This property relieves the user from the burden of managing obscure data, thus being more user-friendly. The Individual Verifiability property itself would bring in the necessary confidence/trust needed from the user towards the proposed system. Additionally, the possibility of vote buying is nullified, due to receipt-freeness, as the candidates/adversaries would not be able to convincingly verify the voters' selection. The implementation of this property perhaps needed the least effort.

Fairness: As the poll is underway, no-one should be able to calculate the partial tally. The voter gets to view/verify the result only after the poll is closed by the Poll Initiator. The Poll Initiator never gets to view the results, unless s/he is part of the electorate him/herself, and in the later case, s/he can only view the result after the poll is closed, thus maintaining fairness.

4 Comparison Matrix

The previous section addressed the properties of the proposed system, and did an evaluation of the system based on those properties. Below is a quick comparison between the existing systems and the proposed *agile e-voting* system on various properties.

S.No.	Application	Anonymity	Security	Universal Verifiability	Individual Verifiability	Fairness	Receipt-freeness	Login-free
1	Adder	Yes	Yes	Yes	No	Yes	Yes	No
2	CIVS	No	Yes	No	No	Yes	No	Yes
3	KOA	Yes	Yes	Yes	No	Yes	No	No
4	*Agile e-voting*	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Fig. 3. Comparison Matrix - Between various browser-based open-source e-voting systems

5 Conclusion

An innovative voter-friendly browser-based agile system was proposed in this paper. The e-voting properties such as, Security, Anonymity, Verifiability, Receipt-Freeness and Fairness were implemented by the system. The essential value addition that the proposed system brings to the industry, is the attempt to put the focus back onto the voter, and derive/relate other requirements based on the voter convenience. It was shown that the proposed Agile e-voting system addressed most of the issues in the existing browser-based e-voting systems. Most importantly, a new approach to Individual Verifiability was crafted, which would allow the voter to cross verify his/her vote during the tally phase. The evaluation of the system as per the properties mentioned earlier was detailed out. The tricky In-coercibility property implementation would be the logical next first step towards future enhancements. Thus, with such user-friendly features implemented, the agile e-voting system hopefully would aid in increasing the voter participation.

References

1. Brace Kimball, W.: Overview of Voting Equipment Usage in United States, Direct Recording Electronic(DRE) Voting. Kimball Brace's Statement to United States Election Assistance Commission (May 5, 2004)
2. Kevin, C.: CRS Report for Congress, Internet Voting. Order Code RS20639 (January 31, 2003)
3. California Internet Voting Task Force: A Report on the Feasibility of Internet Voting (January 2000)
4. Kiayias, A., Michael, K., David, W.: An Internet Voting System Supporting User Privacy. In: Jesshope, C., Egan, C. (eds.) ACSAC 2006. LNCS, vol. 4186, Springer, Heidelberg (2006)
5. Rising, L., Janoff, N.S.: (2000) The Scrum Software Development Process for Small Teams. IEEE Software (July/August, 2000)
6. Niemi, R.G., Weisberg, H.F.: Controversies in Voting Behavior. CQ Press, Washington, D.C (2001)
7. VoteHere VHTi: Frequently Asked Questions, `http://www.votehere.com/faq_toc.php`
8. Cochran, D.: Secure internet voting in Ireland using the Open Source: Kiezen op Afstand (KOA) remote voting system. Master's thesis, University College Dublin (2006)
9. Kim, K.: Killer Application of PKI to Internet Voting. In: IWAP 2002. LNCS, vol. 1233, Springer, Heidelberg (2002)

10. GNU.FREE: Heavy-Duty Internet Voting, http://www.j-dom.org/users/re.html
11. An Untraceable, Universally Verifiable Voting Scheme, Professor Philip Klein, Seminar in Cryptology (December 12, 1995)
12. http://www.e-poll-project.net/
13. Krishna, S., Sampigethaya, S.K., Poovendran, R.: A Framework and Taxonomy for Comparison of Electronic Voting Schemes. Elsevier Journal of Computers and Security 25(2), 137–153 (2006)
14. Condorcet Internet Voting Service, http://www.cs.cornell.edu/andru/civs.html
15. PollPub, http://www.pollpub.com/
16. BuzzVote, http://www.buzzvote.com/
17. VoteHereVHTi: Frequently Asked Questions, http://www.votehere.com/faq_toc.php
18. Sensus, http://lorrie.cranor.org/voting/sensus/
19. Kiniry, J., Morkan, A., Cochran, D., Fairmichael, F., Chalin, P., Oostdijk, M., Hubbers, E.: The KOA remote voting system: A summary of work to date. In: Proceedings of Trustworthy Global Computing (2006)
20. Nijmeegs instituut voor informatica en informatiekunde. Security of Systems, http://www.sos.cs.ru.nl/research/koa/ ; KiezenopAfstand 2004
21. IETF RFC 4346: The Transport Layer Security (TLS) Protocol, Version 1.1. URL, http://www.ietf.org/rfc/rfc4346.txt
22. Alberto, O., Marco, V.: Man-In-The-Middle Attacks, http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-ornaghi-valleri.pdf
23. Diffie, W., Hellman, M.E.: Exhaustive Cryptanalysis of the NBS Data Encryption Standard. Computer 10(6), 74–84 (1977)
24. Menezes, A., van Oorschot, P., Van stone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton, USA
25. Hellman, M.E.: An Overview of Public Key Cryptography. IEEE Communications Magazine, 42–49 (May 2002)
26. Advanced Encryption Standard(AES), Federal Information Processing Standards Publication 197 (November 26, 2001)
27. Neff, C.A.: Practical high certainty intent verification for encrypted votes. Draft(2004), http://www.votehere.net/vhti/documentation/vsv2.0.3638.pdf
28. Reynolds, D.J.: A method for electronic voting with coercionfree receipt. FEE 2005 (2005), Presentation, http://www.win.tue.nl/ berry/fee2005/ presentations/reynolds.ppt
29. Moran, T., Naor, M.: Receipt Free Universally Verifiable Voting With Everlasting Privacy. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 373–392. Springer, Heidelberg (2006)
30. Ronald, L.: Rivest.: The ThreeBallot Voting System (October 2006), http://theory.lcs.mit.edu/rivest/Rivest-TheThreeBallotVotingSystem.pdf
31. Jones, H., Juang, J., Belote, G.: ThreeBallot in the Field (December 2006), http://courses.csail.mit.edu/6.857/projects/threeBallotPaper.pdf