Self-Enforcing Electronic Voting

Feng Hao, Brian Randell, Dylan Clarke

School of Computing Science
Newcastle University, UK

Abstract. Verifiable electronic voting has been extensively researched for over twenty years, but few protocols have achieved real-life deployment. A key impediment, we argue, is caused by the existing protocols' universal reliance on the probity of the tallying authorities. This might seem surprising to many people as dependence on tallying authorities has been a de facto standard in the field. However, this dependence is actually a legacy inherited from traditional physical voting, one that has proved problematic in the electronic context. In this paper, we propose a radically new concept called "self-enforcing electronic voting", which refers to voting systems that are free from reliance on any tallying authority. This proposal goes significantly further than all existing or proposed e-voting systems. We explain the feasibility of this new approach, with a theoretical definition of the system properties, a concrete engineering design, a practical implementation, and real-world trial experiments. We also highlight some open issues for further research.

1 Introduction

Self-enforcing security protocols are powerful, as they do not depend on any external authorities, and thus are much more secure and deployable than those that do. Similarly, we define "self-enforcing electronic voting" as a voting system that does not depend on any external tallying authorities. By this definition, existing e-voting protocols such as Helios [1] are not self-enforcing, as they universally rely on tallying authorities.

The idea of depending on tallying authorities is a historic legacy. From the beginning of voting, the perceived trustworthiness of the authorities has been an important factor in persuading the general public to trust the vote tallying process, at least when buttressed by the use of effective independent observers. This tradition also carries over to e-voting, where existing voting protocols all involve tallying authorities.

But, the theoretical assumption of trustworthy tallying authorities in e-voting proves very tricky to implement. For example, in the recent deployment of the Helios e-voting system in the Université catholique de Louvain (UCL), two problems were reported [1]. The first is usability. For fairness and security, the UCL election committee appointed tallying authorities from diversified backgrounds. But, the chosen tallying authorities knew little about cryptography. They were incapable of understanding and performing complex cryptographic operations,

and in reality had to rely on external crypto experts to perform the authorization tasks. The second related problem is security. It was assumed that the authorities were subject to a threshold control: each of them was responsible for safeguarding a private key, and it took a quorum of authorities to be able to tally the votes. However, it is possible that authorities might lose their private keys, which could lead to a catastrophic result (the inability to complete the tallying would in effect act as a Denial of Service attack to the entire election). Hence, the election committee decided to centrally back up all of the tallying authorities' private keys and entrust the copies to a notary public. Obviously, the notary public was able to view all the secret votes, which completely defeats the assumed threshold control.

2 Proposal

Our proposal is to eliminate the reliance on (allegdly) trustworthy tallying authorities altogether. We call the resultant voting system "self-enforcing electronic voting".

Figure 1 portrays the evolution of voting technologies. From the beginning of voting, central authority has been playing a pivotal role in instilling trust in the tallying process (an issue which is neatly captured by a famous saying attributed to Joseph Stalin: "It's not the people who vote that count; it's the people who count the votes"). In the current e-voting deployment around the world (US, India, Brazil, etc.), voters have to trust their e-voting machine completely, and transitively have to trust the authority that certifies the machine. Researchers have been trying to apply cryptography to make these e-voting systems more verifiable. The current state of the art in the field involves distributing trust among several authorities using some cryptographic threshold scheme. This is an improvement over relying on a single authority, but not a clean solution. Voters have to trust the authorities do not collude but they cannot verify that this is indeed the case. The mere fear that the authorities might collude to compromise the election would have deterred many voters in the first place. Against this background, our vision of next generation e-voting is that the election will need to be self-enforcing: free from reliance on any tallying authorities.

We now give a slightly more formal definition of "self-enforcing e-voting". For simplicity of explanation, we discuss just the case of a single-candidate election. However, the definition can be easily generalized to multi-candidate elections.

Consider a machine that pre-computes a table of N electronic ballots before the election. As shown in Table 1, each ballot contains two cryptograms representing: "No" and "Yes" (which correspond to numeric values 0 and 1 respectively in our implementation). We thereby define a (single-candidate) self-enforcing e-voting system as one that satisfies the following properties.

1. *Well-formedness* – For each ballot, given n_i or y_i, it is easy for anyone to verify that the given cryptogram is an encryption of one of the two values: "No" and "Yes".

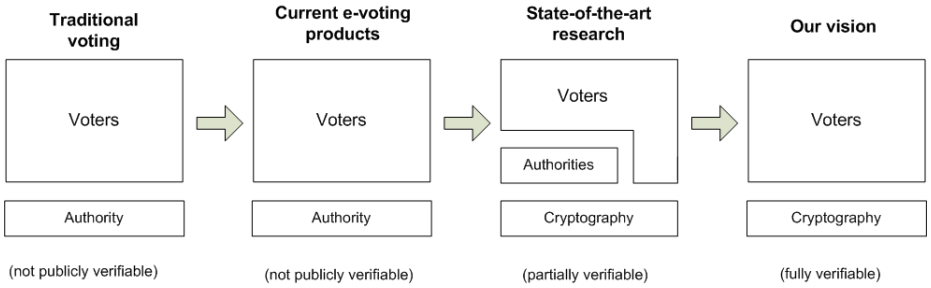

Fig. 1. Evolution of voting technologies

2. *Concealing* – For each ballot, given n_i without y_i, or given y_i without n_i, it is infeasible to compute whether the given cryptogram represents "No" or "Yes".
3. *Revealing* – For each ballot, given both n_i and y_i, it is easy for anyone to compute which cryptogram represents "No" and which represents "Yes".
4. *Self-tallying* – Given a set of choices of one arbitrary cryptogram from the two on each of the N ballots, it is easy for anyone to compute the tally of "Yes".

Ballot no i	No_Cryptogram n_i	Yes_Cryptogram y_i
1	n_1	y_1
...	...	...
N	n_N	y_N

Table 1. Pre-computed electronic ballots

3 Feasibility

In this section, we show the proposed system is feasible by addressing the following three questions: 1) is it possible? 2) how does it work? 3) is it practical?

The first question is whether "self-enforcing e-voting" is theoretically possible. We answer this question affirmatively. In particular, the DRE-i protocol (Hao, Kreeger, 2011) as described in [2] fulfils all of the above properties. However, the DRE-i protocol is only one specific embodiment; there may be other methods to realize "self-enforcing e-voting". Hence, in this paper, we will only discuss the abstraction of system, focusing on the essence of the new proposal.

The second question is how does such a system work in practice. Given a cryptosystem that satisfies the four properties, it is straightforward to apply it to e-voting. As an example, let us consider an on-site voting system that

employs a touch-screen Direct Recording Electronic (DRE) machine to records votes. Before the election, the machine generates a table of N ballots as shown in Table 1 (N should be significantly bigger than the total number of eligible voters).

During the election, each authenticated voter casts her vote in two stages. First, the DRE machine displays an unused ballot with both cryptograms initially hidden. The voter touches one choice from the screen - say she chooses "No". The DRE machine reveals the value of the No_Cryptogram on the screen. In stage two, the voter should either "confirm" or "cancel" the vote. If she chooses "confirm", the ballot is casted. If she selects "cancel", she essentially performs the auditing function. The machine must in this case reveal the other cryptogram. With both cryptograms revealed, the ballot is counted as a dummy vote. Dummy votes will not add to the tally. The voter will be assigned another unused ballot and repeat the same two-stage process to vote.

In the above procedure, all the revealed cryptograms (including valid as well as dummy votes) will be published on a public bulletin board and also printed on the receipt. Based on the *Well-formedness* Property, anyone can verify that the confirmed vote is either "No" or "Yes". Based on the *Concealing* Property, the single revealed cryptogram does not disclose the voter's choice, so the receipt does not show how the voter had voted (this is to prevent coercion). However, the same *Concealing* Property also implies that the DRE machine might cheat – when one selects "No", the machine may display Yes_Cryptogram, and the voter would not be able to tell the difference (i.e., knowing that his vote had been subverted). This is where the *Revealing* Property becomes useful. If the machine cheats by switching the cryptograms, this will be caught when the voter chooses to cancel (i.e., audit) the vote. Finally, the *Self-tallying* Property ensures that when the election finishes, any member of the public is able to tally the "Yes" votes without involving any tallying authorities. (Satisfying the *Self-tallying* Property is the most difficult part in the design of a self-enforcing voting system. The way that is achieved in [2] is based on a technique proposed six years ago at SPW'06 [3].)

The third question – perhaps the most important and most difficult – is whether the proposed system is practical. We answer this question positively by presenting two real-world trial elections based on the self-enforcing e-voting technology, as we detail in the following section.

4 Trial elections

Based on the DRE-i protocol specification in [2], we have developed a fully functional self-enforcing e-voting system, and conducted two (admittedly rather small scale) trial elections among members of the School of Computing Science at the Newcastle University to elect the "favorite chocolate" and "favorite cheese" in October, and November, 2011 respectively. (The two winners were "Quality Street" and "Wensleydale".) In total, 74 people participated our trials.

In both election trials, all participants were the staff and PhD students of the Computing Science department of Newcastle University. The first election involved participants trying three types of chocolate and then voting for their favourite flavour. The second election involved participants trying three types of cheese and then voting for their favourite type.

In both cases, participants were able to try the items on offer and then take a random paper slip from a box placed next to the food. The food and the paper slips were put in a room that was only accessible to the department staff and PhD students. The slip contained a web address for on-line voting and a random passcode. Each participant was free to take a random passcode, so voting was anonymous. When participants had finished voting on-line, they were asked to complete a questionnaire.

4.1 Chocolate Election

This trial was based on an implementation of the DRE-i protocol using a multiplicative cyclic group – the same underlying group as the Digital Signing Algorithm (DSA). The voting trial had 39 participants. All participants casted a confirmed vote. Four of them audited a ballot once (i.e., casted a dummy vote) before confirming their votes. No voter audited more than once.

The feedback questionnaire consisted of 6 statements and respondents were asked to indicate their agreement or disagreement on a Likert scale from 1 to 5 (i.e., "strongly agree", "agree", "neural", "disagree", "strongly disagree"). The statements were as follows:

1. I understood how to vote.
2. Voting was easy.
3. I understood how to check my ballot had been recorded correctly.
4. Checking that my ballot had been correctly recorded was easy.
5. I understood why I was being asked to check ballots.
6. I felt confident that my vote had been recorded correctly.

The feedback questionaire was answered by 23 voters. One respondent did not give an answer for statement 2; other than this, every respondent gave an answer for every statement. The results are shown in Table 2

4.2 Cheese Election

The subsequent cheese election trial was conducted one month after the chocolate election. For this trial, we changed the implementation to using an additive cyclic group – the same underlying group as the Elliptic Curve Digital Signing Algorithm (ECDSA). The basic DRE-i protocol remained the same, but the implementation became more efficient and the size of the receipt significantly shorter.

The voting trial had 35 participants. All participants casted a confirmed vote, and one voter audited a ballot once before confirming the final vote. No voter audited more than once.

Table 2. Feedback Questionaire on the chocolate election

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	*Average score (nearest option)*
1	13	7	3	0	0	*1.57 (Agree)*
2	10	7	3	2	0	*1.86 (Agree)*
3	7	6	5	4	1	*2.39 (Agree)*
4	4	2	7	7	3	*3.13 (Neutral)*
5	5	6	0	9	2	*2.86 (Neutral)*
6	7	7	3	4	2	*2.43 (Agree)*

The feedback questionaire consisted of 7 statements and respondents were asked to indicate their agreement or disagreement on the same Likert scale as before. The statements were as follows:

1. I understood how to vote.
2. Voting was easy.
3. I understood how to check my ballot had been recorded correctly
4. Checking that my ballot had been correctly recorded was easy.
5. I understood why I was being asked to check ballots.
6. I felt confident that my vote had been recorded correctly.
7. I felt confident that my vote was anonymous.

The questionaire was answered by 20 voters. Every respondent gave an answer to every statement.

Table 3. Feedback Questionnaire on the cheese election

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Average score *(nearest option)*
1	15	2	2	1	0	*1.45 (Strongly agree)*
2	12	1	6	1	0	*1.80 (Agree)*
3	4	9	2	5	0	*2.40 (Agree)*
4	2	4	6	7	1	*3.05 (Neutral)*
5	4	6	4	5	1	*2.65 (Neutral)*
6	5	6	6	3	0	*2.35 (Agree)*
7	7	5	6	1	1	*2.20 (Agree)*

5 User feedbacks and open issues

In the trials, we did not provide any manual to explain how to vote. The only thing that was available to the voter is a slip that contains a voting website URL and a random password. We wanted to see how far voters could go. The feedback shows that users generally found the system intuitive to use – overall

86% of users stated that they well understood how to vote even without any manual or prior training; 71% rated the voting procedure as "easy"/"very easy"; 58% expressed their "confidence"/"strong confidence" that their votes had been recorded correctly. Clearly, there is still a lot of room for improvement, but we found the initial user feedbacks broadly encouraging.

So far our pioneering investigation on self-enforcing e-voting has increased our belief that it has great potential. However, we have also identified a number of open issues. (In fact, most of these issues are generally applicable to all e-voting protocols, not specifically to ours.)

1. *Zero Knowledge Proofs* – The *Well-formedness* Property requires using the ZKPs to ensure the correct format of the ciphertext. Existing ZKPs techniques can well support the first-past-the-post type of voting (using the 1-of-n ZKP), but not the ranked choice voting. More research in this aspect is much needed.
2. *Receipts* – Voters need to verify the receipt against the public bulletin board to ensure the data match. However, the cryptographic data printed on the receipt is essentially random. Comparing random data is trivial to a computer, but tedious to a human. In our first trial, about 43% users expressed difficulties in comparing receipts against the data on public bulletin board. In the subsequent trial, we improved the implementation using Elliptic Curve Cryptography to make the receipt significantly shorter. But still 40% users found it difficult to visually compare the receipt with the bulletin board. This highlights a serious usability problem that has been generally ignored in the past. An often-suggested solution is to apply some visual hashing technique to transform the random data to a visual pattern. However, we have not found a really good visual hashing algorithm that is suitable for practical use. An alternative solution, as we propose, is to print the receipt on a transparency, that the user can overlay on the computer screen to verify that the data match identically.
3. *Re-voting* – Re-voting permits a voter to vote multiple times but ensures that only the last vote counts. Although it appears to be a useful feature, re-voting should be implemented with caution. This is because re-voting essentially overwrites the previous vote, and thus invalidates the previous receipt. Disputes may arise if the re-voting requests are not handled in a publicly evident manner. We know it is possible to implement re-voting at a polling station (as explained in [2]), but currently we do not know how to do it securely in an entirely remote setting. (We notice that Helios allows re-voting in an Internet election [1], but we have not seen detailed analysis on how this was securely achieved.)
4. *Anonymity* – In our implementation, anonymity is achieved through physical means: each voter takes a random password to vote. Of course, this solution will not work in an entirely remote setting. The Helios system claims to protect user's anonymity over the Internet by using pseudonyms [1]. But we think that claim is incorrect – the server can still work out the matching between the real user and the pseudonym. The problem of how to achieve

real anonymity in remote e-voting – without any physical means – is still unsolved. In fact, we do not even know if the problem is solvable.

5. *Auditing* – If we allow voters to audit the system (i.e., voter-initiated auditing), how many of them would actually endeavor to do that? This is an important question, but has been generally neglected in the past. For example, we did not find concrete data in the Helios paper [1]. Our own trials indicated that – without any incentive scheme – only a very small percentage of our voters had bothered to audit the system: only 7%. (And these were all computer scientists, many of whom were particularly interested in security and dependability.) This shows the necessity of employing dedicated auditors (possibly from the representatives of different parties) for national-scale elections. But still, we consider it important to set up incentive schemes to encourage voters to audit the system. One specific example of the scheme is suggested here as follows: if the voter chooses to audit the system, he will get a charity token, and he will be free to donate it to one of the charities near the exit of the polling station. We regard this simple solution as an especially ethical form of so-called "ethical bribery" – we term it the "Waitrose scheme" to acknowledge a similar charity program run by Waitrose food stores in the UK.
6. *Understandability* – How does one get the general public to trust a system that they know they do not understand, especially one whose use/reliance on cryptography is all too invisible and mysterious to them? We do not yet have a good answer to that, but the same question applies to all e-voting schemes that involve cryptography.
7. *Dependability* – In conventional e-voting schemes, tallying authorities have some error-correction capability: if some ballots are lost, the authorities may decide to exclude the missing ballots and only tally the rest (of course, this capability may be misused). By contrast, self-enforcing e-voting schemes are free from tallying authorities (and hence free from any potential misuse), but they naturally lose the external error-correction capability. As a result, all electronic data on the public bulletin board must be precise and complete; otherwise, the public verification of the integrity of the tally will fail. In the extreme disastrous situation where electronic data on the bulletin board is corrupted, the system will degenerate to the current e-voting products, where the e-voting machine announces a tally at the end of election, but the public are unable to verify it. In summary, self-enforcing e-voting has bought its property of self-tallying at a cost of much higher dependability requirements. We believe this is achievable, but further research is needed.

6 Conclusion

In this paper, we have outlined a new design called "self-enforcing electronic voting". This addresses a critical problem in e-voting: ensuring the integrity of the election tallying result when the e-voting machine is totally corrupted. And this goal is achieved without any tallying authority involvement, which greatly simplifies the election management and organisation.

Our proposal is only the first step in exploring a new direction in e-voting and there are still a number of open issues to resolve, but we believe it has promising potential of becoming the next-generation e-voting.

References

1. B. Adida, O. de Marneffe, O. Pereira, and J.J. Quisquater, "Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios," Proceedings of the Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, 2009.
2. F. Hao, M.N. Kreeger, "Every vote counts: ensuring integrity in DRE-based voting system," Newcastle University technical report No. 1268, 2011. Available at `http://www.cs.ncl.ac.uk/publications/trs/papers/1268.pdf`.
3. F. Hao, P. Zielinski, "A 2-round anonymous veto protocol," Proceedings of the14th International Workshop on Security Protocols, Cambridge, LNCS 5087, pp. 202-211, 2006.