Helios Verification: To Alleviate, or to Nominate: Is That The Question, Or Shall We Have Both?

Stephan Neumann[1], M. Maina Olembo[1], Karen Renaud[2], and Melanie Volkamer[1]

[1] Security, Usability and Society, CASED, TU Darmstadt
[2] School of Computing Science, University of Glasgow
Email:firstname.lastname@cased.de; firstname.lastname@glasgow.ac.uk

Abstract. Helios is an end-to-end verifiable remote electronic voting system which has been used for elections in academic contexts. It allows voters to verify that their vote was cast, and included in the final tally, as intended. User studies have shown that voters are unlikely to perform this verification, probably due to the effortful and cumbersome manual steps that are required by the system. To address this challenge, we propose, in this paper, two improvements: the first is to ameliorate the cumbersome nature of the verification process as much as possible. We offer two suggestions for doing this. To accommodate voters who have no interest in verifying, we propose a further improvement: delegation. This will allow voters to nominate a trusted third party to perform the verification on their behalf *as and when they cast their vote*. Hence no extra effort is required, and we can exploit existing trust in public institutions to provide voters with the assurance that the voting process is indeed honest and above board. In addition to providing end-to-end verifiability in a less effortful manner, we provide *stored as cast* and *tallied as stored* verifiability as well, for voters who do not wish to verify their own votes.

1 Introduction

Internet voting is a hot topic, continuing to attract interest in real world elections and in academic research. Helios is one Internet-based voting system that has been used in a variety of elections, primarily in academic contexts. For example, Helios was used to elect the university president at the Université catholique de Louvain [4], and since 2010, to elect the Board of Directors of the International Association for Cryptologic Research (IACR), to conduct the Princeton undergraduate student election [22], and for the Board Election of the Institute of Public Works Engineering Australia (IPWEA).

Vote integrity in Helios implicitly relies on two assumptions. The first is that voters *will act* to verify that their vote was indeed "cast as intended" and "stored as cast". This requires them to take deliberate action subsequent to casting their vote. The second assumption is that the voting environment (i.e. the device such as the smartphone or laptop used to cast the vote) is trustworthy and has not been compromised. The latter assumption ensures that vote integrity and secrecy are assured.

The first assumption can only hold if all voters are motivated to verify, and able to do so. Helios verification requires voters to write down a set of characters computed by the voting device and then to compare them manually to another set of characters computed by a verifier. Findings from user studies on verification in Helios suggest that voters tend not to verify their votes [28, 17, 16]. This applies to "cast as intended" verification ("stored as cast" verification has yet to be evaluated). Participants in these studies found the process cumbersome, especially since they were asked to repeat it several times. (Such repetition is considered necessary to achieve adequate security levels.)

Since verification is considered a valuable feature in guaranteeing the integrity of the election process, we suggest two improvements. The first is to attempt to ease the process so that voters no longer find it so arduous. We address this issue by proposing two means for simplifying the "cast as intended" verifiability process: (1) copy, web browser search and paste; and (2) a verification smartphone App (which weakens the second assumption that Helios relies on with respect to vote integrity). Moreover, we also propose a simple way to conduct the "stored as cast" verification process which currently is similarly cumbersome (although it differs from cast as intended verification as it only needs to be conducted once). For both verification options, we have taken an exploratory human-centered design approach following the guidelines given by [15] in designing and integrating these proposals into Helios.

The second is to accept that the interface might never be able to make it trivial enough and that there will still be many voters who will not verify. We therefore further tailor the voting process to acknowledge this reality by allowing voters to nominate one of a number of trusted institutes to verify on their behalf. We hope to achieve more effective security for future elections that use Helios.

The rest of this paper is structured as follows: We first present the Helios voting system in Section 2. In Sections 3 and 4, we present our first improvement, namely two alleviation approaches to ease the verification process. Section 5 proposes our second improvement, namely delegating verification to a trusted party. Section 6 presents related work in the area of usability of electronic voting and Section 7 concludes this paper.

2 Helios Voting System

Helios is an Internet-based, open source, end-to-end verifiable voting system [3]. It provides *cast as intended* verifiability, since voters can check that their actual votes are correctly encrypted. It also supports *stored as cast* verifiability, where voters can check that the encrypted votes are received by, and stored at the voting server, for tallying purposes without modification. Finally *tallied as stored* verifiability is supported, where any interested parties (including voters) can check that all stored votes have been tallied correctly. If it is confirmed that all stored votes were properly tallied, voters can be assured that their votes were properly tallied if they acted previously to conduct the other two verifiability steps. Note that in Helios vote integrity, as well as vote secrecy, is only assured

under the assumption that the voting environment (the device, the browser and the network connection) is trustworthy. As such, the ability to verify only enables voters or observers to detect Helios system malfunction.

In order to provide "cast as intended" verifiability, Helios uses the Benaloh Challenge [6]. "Stored as cast" verifiability is achieved using a public web bulletin board that acts as a voting server to store votes [3, 4]. Either verifiable homomorphic tallying [4] or a verifiable mix net and verifiable decryption [8] are used to provide "tallied as stored" verifiability.

The focus of our research is on the first two verification steps as these have to be conducted by the voters themselves (if not, vote secrecy would be violated) while "tallied as stored" verification steps can be conducted by any interested party. Correspondingly, we also only explain these two steps in more detail. We will start our discussion with the original Helios version and then explain the relevant modifications proposed in [16] which we use as a basis for the research proposed in this paper.

Voters use the Helios interface, referred to as a *ballot preparation system* (BPS), to select the candidate(s) of their choice. The BPS encrypts this selection (i.e. the vote) and displays a hash value of the encrypted vote. Voters are supposed to record the displayed hash value in order to carry out both verification steps. Since the SHA256 hash algorithm and Base64 encoding are used, voters currently have to record 43 characters.[3]

Voters can then decide to cast the encrypted votes or to verify whether the genuine vote was indeed encrypted (which is referred to as the Benaloh Challenge). If voters decide to verify that their vote was "cast as intended", they interact with the independent and trustworthy *ballot verifier system* (BVS). BPS confirms the candidate(s) and the randomness used for encryption by displaying this information, which voters are supposed to select and copy to the clipboard in order to paste it into the BVS. The BVS then encrypts the corresponding vote and generates the hash value of this encryption. This hash value is displayed together with the provided candidate(s). In order to complete the "cast as intended" verification process voters need to visually compare both values to ensure that they are correct (the 43 characters of the hash value and the candidate/s). Note that since both hash values are displayed on the same device it is crucial that this device be trustworthy — if not, it would always display the expected hash value, independent from what BVS computes.

Voters can repeat the "cast as intended" verification step as many times as required and this should actually be repeated to ensure integrity. Votes that have been verified i.e. votes for which the randomness used for encryption has been revealed, can no longer be cast since voters could then prove how they voted if

[3] It should be emphasized that SHA256 provides overwhelming integrity assurance at the current point in time. According to Lenstra's revised recommendations for key length [18], a shorter hash value can be justified for this context. Assuming an election phase of 30 days and the adversary's financial limitations of 3 million dollars, a hash size length of 155 bits is sufficient to provide adequate security. According to the PGP word list [29], 155 bits could be presented by 20 words.

this vote could be cast directly. Therefore, depending on the Helios version, voters are re-issued with an empty ballot, or the vote is automatically re-encrypted. As the BVS is given the content of the encryption, it is recommended that voters verify test votes (that is, not necessarily the same as the one that will be cast) to avoid BVS being able to derive intermediate outcomes. To support this, it helps to use the Helios version in which the voter needs to start with the empty ballot once he/she has verified. While Helios provides verifiability, it does not provide accountability; i.e. if people or the verification tools falsely claim that the "cast as intended" or "stored as cast" step could not be completed, there is no way to distinguish between the two cases: dishonest voters/tools or Helios being untrustworthy.

If voters choose to cast the vote, either directly or after vote encryption, they are prompted to authenticate themselves, and their encrypted votes are then posted to the public web bulletin board together with the hash value of the encrypted vote. To verify that the vote is "stored as cast", voters need to check whether the correct hash value appears on the public web bulletin board next to the voter's name [3], or pseudonym [4] depending on the Helios version used. It is only necessary to check this once as the Helios security model assumes that observers or other trusted institutes continuously make copies of the bulletin board and would detect malicious behaviour on the part of the bulletin board with respect to integrity. For example, detecting removal or replacement of single votes should be trivial.

Various aspects of "cast as intended" verification were improved in [17]. This included instructions and simplifications (in particular reducing the number of required steps from 15 to 7). For this paper it is relevant to point out that (1) the information necessary for the BVS is automatically transferred from the BPS and (2) several BVS systems are provided by different trusted institutes. Voters now only need to trust one particular institute and not rely on only one BVS. In [17], the authors show that this approach is as secure as the original Helios version. Note, both versions rely on the assumption that voters check that they are on the proper voting and verification web pages. The simplified verification procedure is depicted in Figure 1.

These simplified steps and improved interfaces were tested in a lab user study with 34 participants [16]. The outcome was that most of the participants were able to verify[4] (after receiving instructions to do so). However, they still complained about the required steps being too cumbersome.

Therefore we propose less cumbersome alternatives to verify that the vote is "cast as intended" but also propose mechanisms to verify that the vote is "stored as cast". We focus on simple elections (n out of m candidate(s) to be selected in one race) to simplify the explanations in the following sections.

[4] The authors of [16] reported that two participants were not able to as they only went to the web page to select a trusted institute but then went back to the ballot preparation system without actually verifying their vote. This can be fixed by disabling the button back to the voting process until one of the institutes is selected.

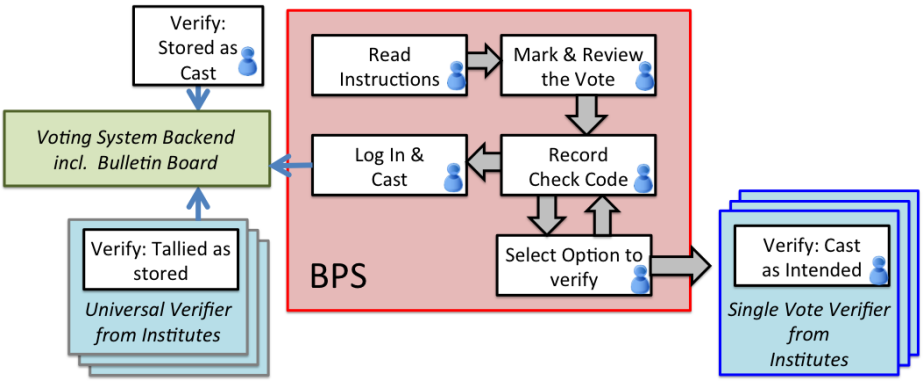

Fig. 1. Improved ballot casting and verification procedure according to [17]

3 Alleviation Option 1: Copy, Search, Paste

In this section we present and discuss the processes that voters would carry out to verify their vote using the first alternative to the previously described verifiability steps, namely the 'copy, search and paste' approach.

3.1 "Cast as Intended" Verification

Using this scheme to verify whether a vote is cast as intended, voters have to:

1. **COPY:** copy the check-code (using Ctrl + C) to the clipboard[5] which is displayed after the vote has been encrypted (see Figure 2, step 2)[6]. Note, on clicking the icon i for more information, voters would see the following instruction: 'Highlight the check-code with the mouse. Simultaneously press the Control and C keys to copy it to the clipboard.'
2. **SEARCH:** Voters can then decide to check the encrypted vote and select a verifiability institute, to which they will be re-directed. The interface provides several options to verify (see Figure 3): Six institutes are represented by their corresponding logos. An option to check using the QR checker App (outlined in the following section), and the Manual Check option, which refers to that provided by the original Helios interfaces also appear. The institutes[7] listed here were identified in a user study for Germany, detailed

[5] Remember, Helios assumes the voting platform, i.e. the voters' device, to be trustworthy. Hence there is an implicit assumption that the clipboard content is indeed equal to the displayed text.

[6] The steps and shortcuts outlined here are suitable for use on a Microsoft Windows platform. This differs from the instructions for other operating systems and platforms, such as Linux, Mac OS X, iOS, and Android. The idea is that the server detects the browser being used and displays the corresponding instructions.

[7] Note that we are not implying that the any of these institutes have agreed to provide such a service.

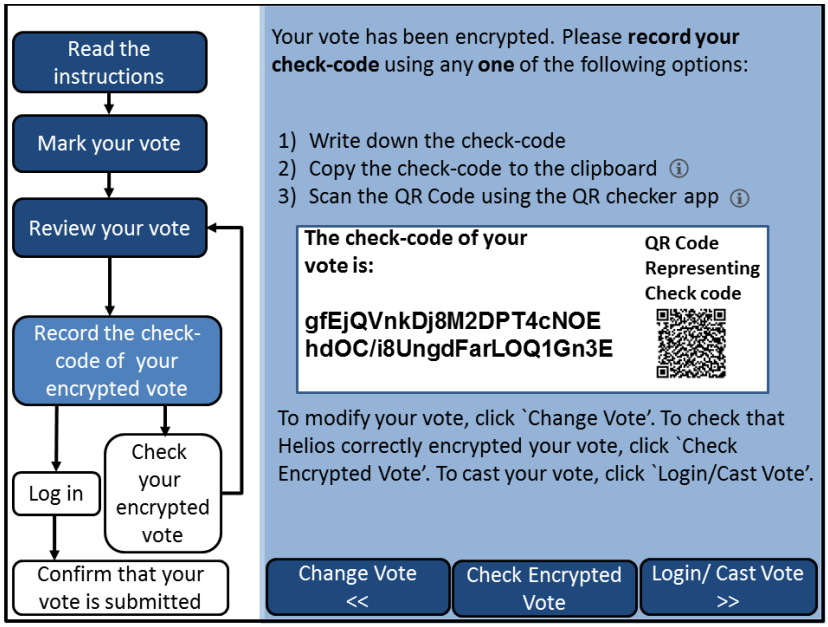

Fig. 2. Options to record the hash value (referred to as 'check-code')

in [20]. On this new web page, the institute displays the check-code and the candidate(s) as in previous versions. In addition, it displays instructions to compare the check-codes (using Ctrl + F and Ctrl + V) and to check the candidate(s). If voters do not know how to proceed, the following instructions are displayed:

(a) To open the web browser search bar, simultaneously press the Control and F keys on the keyboard.
(b) To paste the copied check-code from the clipboard to the search bar, simultaneously press the Control and V keys. Press the Enter button if necessary to compare the value in the clipboard to the value displayed below.

3. **PASTE:** Voters use Ctrl-V to paste. If the displayed check-code is highlighted and the candidate(s) is/are the one(s) selected before, voters can be sure that the vote was properly encrypted.

This proposed amelioration approach removes the need for manual recording and should thus ease the verification process.

3.2 "Stored as Cast" Verification

In order to confirm that the check-code has been stored correctly on the public web bulletin board, voters have to visit the corresponding web page and check this using the same copy, search and paste approach. They have to confirm

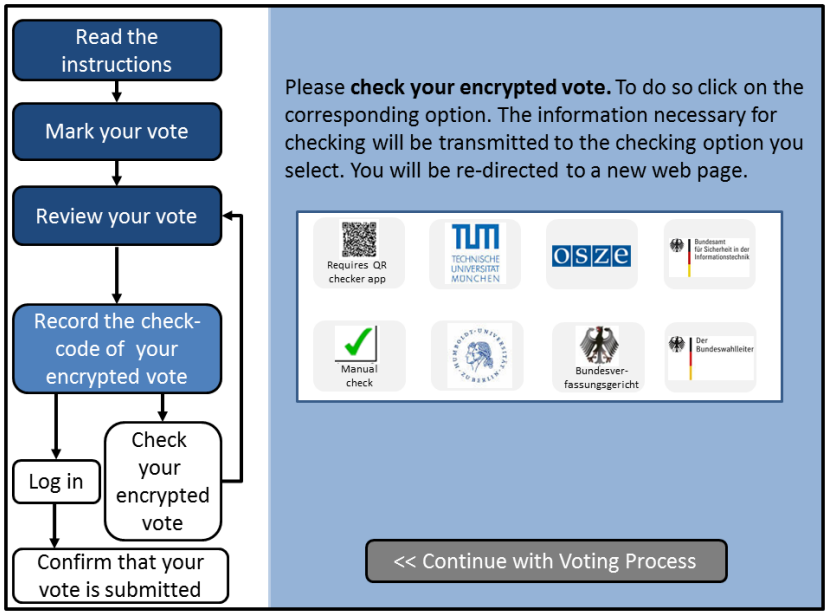

Fig. 3. Checking options available to voters - voters select the QR logo

whether the check-code appearing next to their voter ID or pseudonym is highlighted. Note that access to the public web bulletin board is provided once voters have been successfully authenticated, and their vote has been stored. Since the check-code is already in the clipboard it is possible for voters to carry out the second action independent of the first verification step. Again, as compared to previous versions, no manual comparison of the original hash value and the one displayed next to the voters' ID or pseudonym is necessary.

3.3 Reflection & User Testing

While it is obvious that this approach is less cumbersome, we wanted to determine whether voters would actually be able to conduct the necessary steps. We conducted a small user study including only participants who indicated that they would cast their vote over the Internet to ensure that we had a representative sample in the context of Internet voting. Prior to beginning the study, participants were briefly given context information, explaining how the system operated, and how to verify their votes. Twenty-eight female and 16 male participants took part in the study, with an estimated average age of 26 years. We asked all participants to use the copy, search and paste technique.

Thirty-seven out of the 44 participants were able to copy the hash value into the clipboard on their computer without further instructions. Thirty-five of these also correctly identified the hash value from the list without further instructions.

After further instructions 86.4% of participants were able to carry out the copy, search and paste actions on the web browser. However, the results are admittedly less than perfect and show that we cannot rely solely on this approach. Extra support has to be provided for voters who can then revert to the manual process. Hence we still have to retain and facilitate a manual verification process (see Figure 2, step 1). To assist voters, several fields designed to hold these check-codes can be provided with the election material that is sent to voters, along with their voting credentials.

4 Alleviation Option 2: QR Codes & Smartphone App

In this section, we describe the processes that voters would engage in to verify that their votes were cast as intended and stored as cast by using a smartphone App[8]. A corresponding App would be developed by several trusted institutes. Voters decide which institute to trust and then download the App from that institute. The very sceptical could download Apps from several institutes, and use all of them to verify. Note, voters with a background in Computer Science can optionally develop their own App. The App approach, similar to the copy, search and paste approach, also avoids manual recording and visual comparison, making verification less cumbersome.

4.1 "Cast as Intended" Verification

To provide cast as intended verifiability, the ballot preparation system will display a QR code representing the check-code in addition to the human readable value (see Figure 2). Voters will scan the QR code using the App. The App's interface will display the scanned check-code (see Figure 4 for the corresponding App interface). Voters, on deciding to verify their votes, select the option 'Check Encrypted Vote' on the Helios interface (see Figure 2). Voters then select the QR checker App option (see Figure 3 for a corresponding interface). A second QR code is displayed. Note that while this QR code could be displayed on the same web page as the checking options (Figure 3), we required users to deliberately choose the option. The problem with allowing direct casting was uncovered by [16] who reported that some voters did not know how to proceed when they were on the web page providing the options. Some thought that they had already verified and did not proceed to do so. We thus opted to disable the 'Continue with Voting' button until at least one of the options on the web page had been chosen by voters and a separate window opened. When voters return to this first page, the 'Continue with Voting' button is enabled.

Using the App interface, voters elect to check that their votes are correctly encrypted. They then select the 'Scan' button to scan this second QR code. To avoid confusion, the App also instructs voters on the required steps to be carried

[8] Note that this obviously means that voters use one device to cast their votes and another to verify: perhaps a laptop and smartphone or two different smartphones.

out on the Helios interface. The App then displays the result (see Figure 5): either informing voters of a mismatch or a match in comparing the check-codes. In case of a match, it displays the candidate(s) and prompts voters to confirm whether this is what they marked earlier on the ballot. If voters confirm that the vote is correct, they will be reassured that the Helios voting system acted with integrity. The message will also recommend that they check the vote several times before submitting a final vote. If voters indicate that there is an error they will be directed to contact the election commission for more information on how to proceed.

Two prominent reasons why the check-codes might not match are that either the voter has made a mistake, such as scanning non-matching QR codes, or the Helios system could be untrustworthy. To address the first eventuality, the App would first instruct voters to make sure that they had scanned the correct QR code. If voters are certain of the mismatch, they can vote from another device, or contact the election authorities.

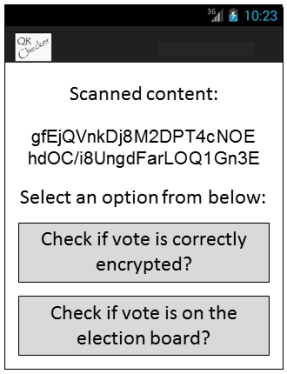

Fig. 4. Scanned check-code and options for the voter

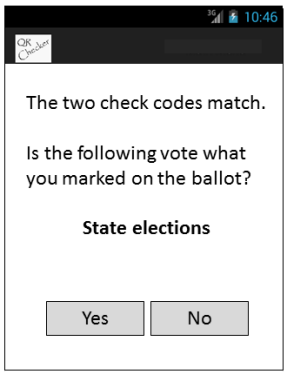

Fig. 5. Results of the verification process

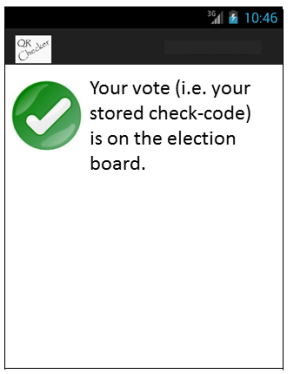

Fig. 6. Positive result that vote is stored on the public web bulletin board

4.2 "Stored as Cast" Verification

In order to verify that votes are stored as cast, after scanning the first QR code (shown in Figure 2), voters can opt to check the public web bulletin board. Voters select the option 'Check if vote is on the election board' (see Figure 4). To prevent voters from proceeding in error, they are prompted to indicate whether they have submitted their vote or not. If they indicate that they have not yet done so, they are instructed to do so before proceeding. Once voters have submitted their vote, (or if they indicate that they have done so), the App checks the public web bulletin board by sending a query containing the received check-code. It then displays a message confirming whether the check-code was

successfully stored on the public web bulletin board (see Figure 6). Given that Helios assumes that the public web bulletin board is continuously monitored by multiple parties, voters only have to check once.

4.3 Reflection

We anticipate that this solution will be acceptable to voters. Smartphones had a 51% penetration rate in Germany[9] in December 2012 [2] and during 2012 over 1.7 million smartphone Apps were downloaded in Germany [1]. These numbers suggest that this verification solution is likely to be accepted by smartphone owners. Initial research suggests that smartphone owners who would be willing to vote over the Internet would verify their votes given an appropriate motivating message [19]. We did not carry out a user study to test the usability of the QR code based App since it is very simple and we already know that people are able to scan and use QR codes very successfully, as reported in various contexts including travel information [9], libraries [5] and consumer communication [12]. Hence we expect this application not to present users with any particular challenges. This will be increasingly true as the younger technically adept generation ages and reaches voting age.

Finally, the App weakens the need for the assumptions that Helios and the previous amelioration option rely on, since voters now no longer need to trust the voting platform or the App, with respect to integrity. The multiple Apps, developed by different trusted institutes, will be bound to reveal any deception that would have been harder to detect using the previous mechanism. We acknowledge that not all voters own smartphones therefore, to ensure inclusivity, other verification mechanisms that do not require extra devices will have to be retained.

5 Delegating Verification to a Nominated Party

The arduous nature of the "write down and manually compare" process in Helios is addressed by the amelioration proposals we presented in the previous two sections. If the first two proposed options are integrated in future elections, it should make the "cast as intended" and the "stored as cast" verification process in Helios more efficient. Yet the element of effort undoubtedly remains, and humans are, unfortunately, effort misers.

5.1 Proposal to Extend the Process

It is difficult to propose further simplifications while retaining the existing process of (1) selecting the candidates, (2) verifying with one or more smartphone Apps or web services from trustworthy institutes, (3) casting a vote and (4) verifying with one or several smartphone Apps or trustworthy web services. Given

[9] Where this research was carried out.

that a significant number of voters will not make the effort to verify, we propose re-thinking the vote casting/verification process.

In the following, we focus on voters who would not verify their votes, while verification measures of the previous sections remain unchanged. We propose changing the voting protocol in the following way: The various institutes will provide web clients to allow voters to cast their votes directly. Hence, voters are provided with several voting URLs in the election invitation letter — one for each of the trustworthy institutes participating in the election. After deciding whom to trust, voters use the vote casting web client provided by that trusted institute. Voters cast a vote over the trusted web client and the institute takes care of the remaining verifiability steps. Throughout the vote casting process, the verification part would be less prominent as indicated in Figure 7. The institutes would verify that all votes cast using their own web service are stored correctly, i.e., they are "stored as cast". In addition, they would still — as in the original Helios as well as in our improved versions — be able to verify that all stored votes were properly tallied. We thus promote the idea of nomination, or indeed delegation. Note that delegation is a well-known and widely-used process [13]. An overview of the Helios variation, with nomination, is shown in Figure 8.

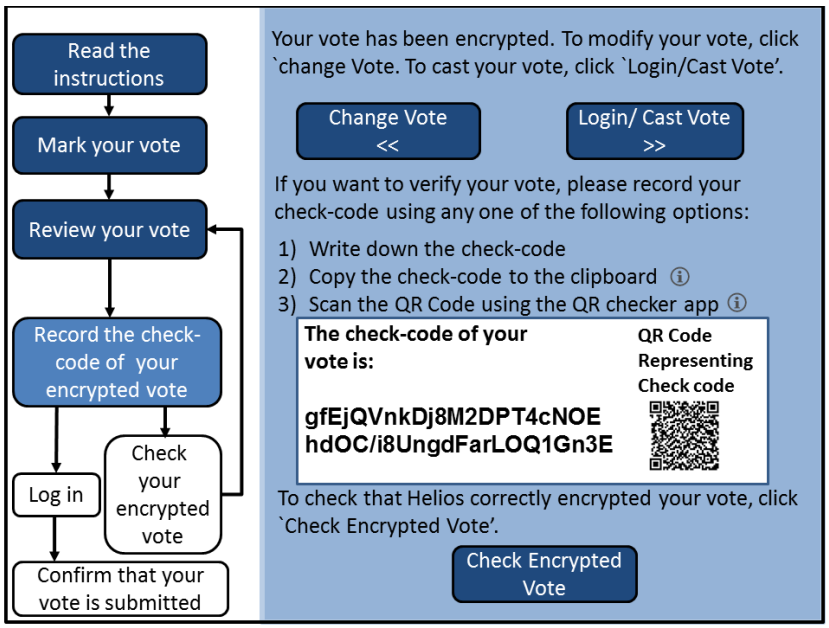

Fig. 7. Revised vote casting interface

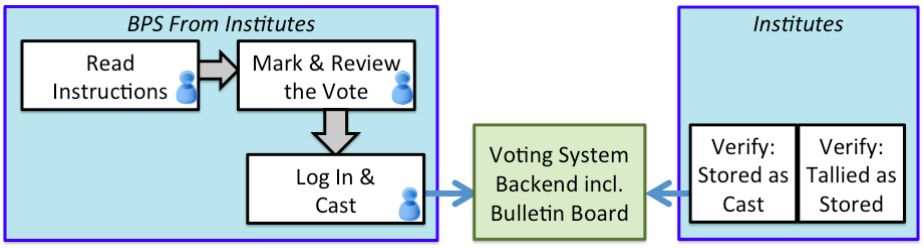

Fig. 8. Helios voting with nomination

5.2 Reflection

For voters benefiting from the amelioration, the fact that someone else is also verifying does not reduce security based on the vote casting / verification processes from [17]. For those voters *not* verifying their votes, integrity assurance is improved with the delegation approach because it is not possible for all votes to be manipulated by a single web client. Rather, one "bad" web client could manipulate only a subset of all the votes. On the other hand, distributing trust among several institutes does not address the issue of malicious voting environments. If the voting environment is compromised, integrity violations cannot be detected by any trusted institute.

6 Related Work

We will review two types of related work: (1) research aiming to improve Helios from a technical perspective and (2) human-centered research on verification.

Since Helios was introduced, a number of technical improvements have been proposed. Only two are of interest as they go on to propose implementations of their work. Cortier *et al.* [11] present an open-source variant of Helios providing distributed key generation. The authors do not focus on improvements regarding voter interaction with the system. Tsoukalas *et al.* [27] present Zeus, a verifiable voting and counting system based on Helios. The verifying process is modified such that the voter can decide how the vote should be handled once it leaves the voting platform. If voters submit so-called audit codes, generated by the voting server and sent to them via a secondary channel, they indicate that these submitted votes should be audited and not counted. The authors however point out that no voters in the elections run with Zeus used the audit feature. In our work we additionally propose a solution for voters who may not take up the verification opportunity.

Verifiability has been studied in various contexts from a human perspective while most often in the context of (plain text) voter verifiable paper audit trails (VV-PATs) in a poll site election setting; e.g. Cohen [10] and later on Selker *et*

al.[10] Selker et al. [25] showed that people are unlikely to detect manipulations in the paper audit trails. Similar conclusions were drawn by Herrnson *et al.* [14] based on the times participants spend examining the printouts. Recently, Budurushi *et al.* [7] reported a promising finding. The authors found that the number of voters verifying plain text VV-PATs increases significantly, if voters are confronted with pre-printed "just-in-time" verification instructions. Besides plain text VV-PATs, verification has also been studied from a human perspective in the context of cryptographic verifiability. More precisely, it has been studied in the context of the Prêt à Voter [24] and Scantegrity [26] electronic voting systems. Both groups show that voters are not very likely to understand the concepts behind cryptographic verifiability. In [24], participants stated that they would be unlikely to verify their votes in an election. These results are not too surprising given the fact that, according to [21], most voters have a trust model rather than a verification model in mind when it comes to elections. Driven by these findings, Olembo *et al.* [19] studied how voters could be motivated to verify by different messages and instructions in different situations. Olembo *et al.* tested the impact of three messages based on risk, norms and analogies on intention to verify. The authors explain that there was no significant impact on intention to verify. This warrants further investigation into the nomination option as it seems much more closely aligned with the voter's electronic voting mental models.

7 Conclusion

This paper presents a number of proposals to improve the usability of the verification process in the Helios voting system. While the first two proposals actually improve the usability of the verification process by advancing two new options to verify, namely the *copy, search, and paste* approach and the *QR code checking*, the third option adds the opportunity to nominate trusted institutions to verify on voters' behalf. This third option is a departure from earlier research on verification. It is motivated by the realization that security researchers and voters have different mental models with respect to the verifiability of electronic voting systems. Researchers consider verifiability of electronic voting systems essential, because, in contrast to traditional voting, electronic voting builds upon insecure technology such as computers and the Internet. In some countries verifiability is required by law, which justifies the attention being paid to this aspect by electronic voting researchers. Voters, however, seem to have a different perspective. According to [21], most voters have a trust model rather than a verification model in mind when it comes to elections. Moreover, verifiability may make very little sense to voters in economic terms: it carries a cost but delivers very little personal benefit. Indeed, if the voters uncover fraud it will cost them even more effort to report it and to follow up to ensure that the case is investigated. At the

[10] We are aware of the problems regarding Selker *et al.*'s study design identified by Quesenbery *et al.* [23]. We mention it here as the trend not to verify printouts remains.

end of the day voters may conclude that they would rather not know, than have to expend effort based on certain knowledge of fraud. We thus concluded that it was worth integrating all three options into Helios. As such, the answer to the question posed in the title "*Helios Verification: To Alleviate, or to Nominate: Is That The Question, Or Shall We Have Both?*" is: *We need both.*

Acknowledgments

This work was developed within the project ModIWa2 - Juristisch-informatische Modellierung von Internetwahlen, which is funded by the Deutsche Forschungsgemeinschaft (DFG, German Science Foundation). Further support was provided by CASED (www.cased.de) and Micromata (www.micromata.de).

References

1. Anzahl der Downloads mobiler Apps in Deutschland in den Jahren 2009 bis 2012 (in Millionen) (November 2012), `http://de.statista.com/statistik/daten/studie/168038/umfrage/anzahl-der-downloads-mobiler-apps-in-deutschland-seit-2009/`, Accessed 30th January, 2013
2. 2013 Future in Focus - Digitales Deutschland (2013), `http://www.comscore.com/Insights/Presentations_and_Whitepapers/2013/2013_Future_in_Focus_Digitales_Deutschland`
3. Adida, B.: Helios: Web-based Open-Audit Voting. In: Proceedings of the 17th Symposium on Security. pp. 335 – 348. Usenix Association, Berkeley, CA, USA (2008)
4. Adida, B., De Marneffe, O., Pereira, O., Quisquater, J.J.: Electing a university president using open-audit voting: Analysis of real-world use of Helios. In: Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections. pp. 10–10. USENIX Association (2009)
5. Ashford, R.: QR codes and academic libraries reaching mobile users. College & Research Libraries News 71(10), 526–530 (2010)
6. Benaloh, J.: Simple verifiable elections. In: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop - EVT'06:. pp. 5–5 (2006)
7. Budurushi, J., Woide, M., Volkamer, M.: Introducing Precautionary Behavior by Temporal Diversion of Voter Attention from Casting to Verifying their Vote. In: Workshop on Usable Security (USEC 2014) (2014)
8. Bulens, P., Giry, D., Pereira, O.: Running Mixnet-based Elections with Helios. In: Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections. pp. 6–6. EVT/WOTE'11, USENIX Association, Berkeley, CA, USA (2011)
9. Canadi, M., Höpken, W., Fuchs, M.: Application of QR codes in online travel distribution. In: Information and Communication Technologies in Tourism 2010, pp. 137–148. Springer (2010)
10. Cohen, S.B.: Auditing Technology for Electronic Voting Machines (2005), master thesis, MIT, Media Lab.

11. Cortier, V., Galindo, D., Glondu, S., Izabachène, M.: Distributed ElGamal á La Pedersen: Application to Helios. In: Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society. pp. 131–142. WPES '13, ACM (2013)
12. Dou, X., Li, H.: Creative use of QR codes in consumer communication. International Journal of Mobile Marketing 3(2) (2008)
13. Hawkins, D.G.: Delegation and Agency in International Organizations. Cambridge University Press (2006)
14. Herrnson, P.S., Niemi, R.G., Hanmer, M.J., Francia, P.L., Bederson, B.B., Conrad, F., Traugott, M.: The Promise and Pitfalls of Electronic Voting: Results from a Usability Field Test (2005), http://www.capc.umd.edu/rpts/Promise_and_Pitfalls_of_Electronic_Voting.pdf (last accessed 29.03.2014)
15. International Committee for Information Technology Standards, ISO: Ergonomics of Human-system Interaction – Part 210: Human-centred Design for Interactive Systems (2011)
16. Karayumak, F., Kauer, M., Olembo, M.M., Volk, T., Volkamer, M.: User Study of the Improved Helios Voting System Interface. In: Proceedings of STAST 2011 (2011)
17. Karayumak, F., Kauer, M., Olembo, M.M., Volkamer, M.: Usability Analysis of Helios - An Open Source Verifiable Remote Electronic Voting System. In: Proceedings of EVT/WOTE '11 (2011)
18. Lenstra, A.K.: Key lengths. In: Handbook of Information Security, chap. 114 (2004)
19. Olembo, M.M., Renaud, K., Bartsch, S., Volkamer, M.: Voter, What Message Will Motivate You to Verify Your Vote? In: Workshop on Usable Security (USEC 2014) (2014)
20. Olembo, M.M., Volkamer, M.: A Study to Identify Trusted Verifying Institutes in Germany (2014), TU Darmstadt. Technical Report
21. Olembo, M.M., Bartsch, S., Volkamer, M.: Mental Models of Verifiability in Voting. In: E-Voting and Identify, pp. 142–155. Springer (2013)
22. Princeton Undergraduate Student Government: The Elections Handbook. http://princetonusg.com/?page_id=975
23. Quesenbery, W., Cugini, J., Chisnell, D., Killam, B., Redish, G.: Comments on: Selker, Rosenzweig, and Pandolfo (2006). A Methodology for Testing Voting Systems. Journal of Usability Studies 2, 7–21
24. Schneider, S., Llewellyn, M., Culnane, C., Heather, J., Srinivasan, S., Xia, Z.: Focus Group Views on Prêt à Voter 1.0. In: International Workshop on Requirements Engineering for Electronic Voting Systems, REVOTE 2011, year = 2011,
25. Selker, T., Pandolfo, A.: A Methodology for Testing Voting Systems. Journal of Usability Studies 2(1), 7–21 (2006)
26. Sherman, A.T., Carback, R., Chaum, D., Clark, J., Essex, A., Herrnson, P.S., Mayberry, T., Stefan, P., L., R.R., Shen, E., Sinha, B., Vora, P.: Scantegrity Mock Election at Takoma Park. Electronic Voting 2010 (EVOTE2010) pp. 45 – 61 (2010)
27. Tsoukalas, G., Papadimitriou, K., Louridas, P., Tsanakas, P.: From Helios to Zeus. USENIX Journal of Election Technology and Systems (JETS) and Electronic Voting Technology Workshop/Workshop on Trustworthy Elections, EVT/WOTE'13 pp. 1–17 (2013)
28. Weber, J., Hengartner, U.: Usability Study of the Open Audit Voting System Helios (2009), http://www.jannaweber.com/wp-content/uploads/2009/09/858Helios.pdf, Accessed 30th January, 2013
29. Zimmermann, P.R.: PGPfone: Pretty Good Privacy Phone Owner's Manual. http://web.mit.edu/network/pgpfone/manual (1995)