Comparing "Challenge-Based" and "Code-Based" Internet Voting Verification Implementations

Oksana Kulyk, Jan Henzel, Karen Renaud, Melanie Volkamer

Comparing "Challenge-Based" and "Code-Based" Internet Voting Verification Implementations

Oksana Kulyk[1,4], Jan Henzel[2], Karen Renaud[3], and Melanie Volkamer[4]

[1] IT University of Copenhagen, Denmark
okku@itu.dk
[2] Technische Universität Darmstadt, Darmstadt, Germany
[3] Abertay University, Scotland
k.renaud@abertay.ac.uk
[4] Karlsruhe University of Technology, Germany
melanie.volkamer@kit.edu

Abstract. Internet-enabled voting introduces an element of invisibility and unfamiliarity into the voting process, which makes it very different from traditional voting. Voters might be concerned about their vote being recorded correctly and included in the final tally. To mitigate mistrust, many Internet-enabled voting systems build verifiability into their systems. This allows voters to verify that their votes have been *cast as intended*, *stored as cast* and *tallied as stored* at the conclusion of the voting period. Verification implementations have not been universally successful, mostly due to voter difficulties using them. Here, we evaluate two *cast as intended* verification approaches in a lab study: (1) "Challenge-Based" and (2) "Code-Based". We assessed cast-as-intended vote verification efficacy, and identified usability issues related to verifying and/or vote casting. We also explored acceptance issues post-verification, to see whether our participants were willing to engage with Internet voting in a real election. Our study revealed the superiority of the code-based approach, in terms of ability to verify effectively. In terms of real-life Internet voting acceptance, convenience encourages acceptance, while security concerns and complexity might lead to rejection.

Keywords: User study · Usable security · Verifiability · Electronic voting

1 Introduction

Internet voting has been the topic of public discussion in many countries. Estonia and Switzerland, for example, conduct legally-binding elections using Internet voting as a voting channel. Internet voting does deliver advantages, such as providing support to voters who are abroad or housebound. On the other hand, new security risks are also undeniably introduced into the process due to the deployment of technology where previously only paper has been used [22]. Unless

effective security measures are implemented, hackers can indeed manipulate the election on a large scale, possibly changing the outcomes of elections [9, 14]. End-to-end verifiability [13] is a way to alleviate concerns related to the threat of illicit manipulations. End-to-end verifiable systems allow voters to verify that their votes have been *cast as intended*, *stored as cast* and *tallied as stored* to lead to a credible and trustworthy election outcome. A voting system that is end-to-end verifiable provides reassurance that the election outcome does indeed reflect the will of the voters.

"*Cast-as-intended*" verification allows the voter to verify that his/her vote has not been manipulated during vote casting. However, verification only detects manipulations if voters are: able to *complete* the verification task effectively. Moreover, the effort that is expended affording verification is only warranted if voters are *willing* to cast their vote using Internet voting technology in elections. It is important to note that cast-as-intended verification *has* to be performed by the voters themselves, to preserve vote secrecy. This makes the usability and understandability of the verification process crucial.

We report on a study that compared the usability of two alternative cast-as-intended verification approaches: *challenge-based* and *code-based*, primarily in terms of efficacy. Both are significant players in the field of Internet voting.

We commence by presenting the background to our study (Section 2), including a discussion of the security of the two tested approaches, and a brief review of other verification studies. Section 3 details our research questions and hypotheses. Section 4 then details our methodology and research questions. Section 5 reports on the outcome of the study and provides the answers to our research questions. Section 6 discusses the results and suggests future directions for research. Section 7 concludes.

2 Background

Challenge-based approach: This approach requires the voter to choose a voting option on the voting website, and encrypt their option. They are now faced with two options: (1) cast the encrypted vote, or (2) confirm that the system has encrypted the vote correctly by using a verifier to reveal the contents of the encrypted vote. A verified vote cannot be cast: it has to be discarded so that the verifier's output cannot be used to facilitate vote selling. The challenge-based approach is most commonly used by the so-called Benaloh challenge [6, 7].

Depending on the implementation, the verifier software can run either on a vote-casting device, on an external device owned by the voter, or via a 3rd party website. Another difference is that the verified vote can either automatically be re-encrypted by the voting system, or discarded, requiring the voter to start again from scratch. The latter is the preferred course of action because it preserves vote secrecy — re-randomisation essentially casts the same vote, and vote secrecy is thereby compromised. If the voter commences making their choice from scratch, vote secrecy is preserved.

A number of Benaloh implementations have been proposed [4, 18, 27]. Of these, the so-called mobile approach [27], which uses an installed Smartphone *verifier app* (Fig. 1), seems the most promising, in terms of both security and usability [25]. The variant of the challenge-based approach that we tested in this study used a Smartphone verifier app, and required voters to discard verified votes.

The individual steps of the verification process are (see Fig. 1):(1) The voter marks her chosen voting option, which is then encrypted by the voting client. (2) The cryptographic hash of the encrypted vote is displayed as a QR-code. (3) The voter scans the QR-code using the Smartphone verifier app and (4) chooses either the "cast" or the "verify" option. (5) If the voter chooses the "verify" option, the voting client displays a QR-code with the chosen voting option and the randomness used for encrypting the vote. (6) The voter scans the QR-code. (7) The verifier app encrypts the voting option with the displayed randomness code. The resulting encrypted vote is compared to the check code scanned in previous step. If they are identical, the voting client has encrypted the vote correctly. If the encryption does not match the code, verification has failed. (8)If encrypted code matches the QR code, the verifier app displays the voting option from the verification data to the voter. (9) The voter compares the voting option output by the verifier app with her intention. If the options match, the vote has been verified. Otherwise, the vote has been manipulated. (10) Once verification is complete, the voter returns to step (1), where she goes through the entire process again.

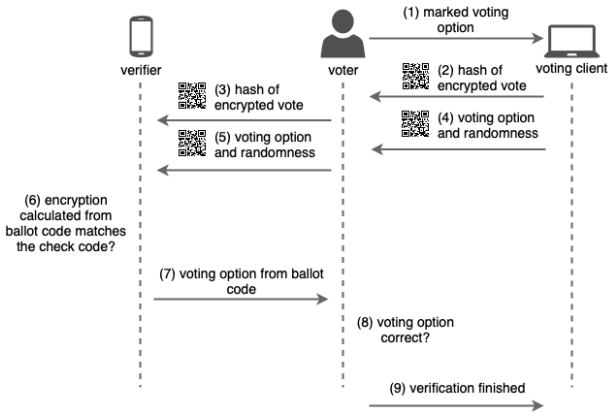

Fig. 1: Verification using the challenge-based approach.

Code-based approach: Each voter is issued with an individualised *code sheet*, which is distributed to voters before the election, via snail mail. The sheet provides a check code for each individual voting option (the sheets differ from

voter when it comes to all the shown codes). When the voter casts a vote, his/her individual check code for that option is displayed by the voting client. To verify, the voter confirms that the displayed output code matches his/her sheet's check code which is next to his/her voting option. Diverse variants of this approach exist, including one used in the actual Swiss elections [34]. Some code sheets contain both confirmation *and* finalisation codes, which, again, are unique to each voter. The voter inputs the confirmation code after he/she has checked the validity of the displayed check code. The finalisation code is displayed to reassure the voter that the voting system has (1) indeed cast the vote as intended, and (2) that the vote has been verified and confirmed by the voter. An example of the code sheet is provided in Fig. 2.

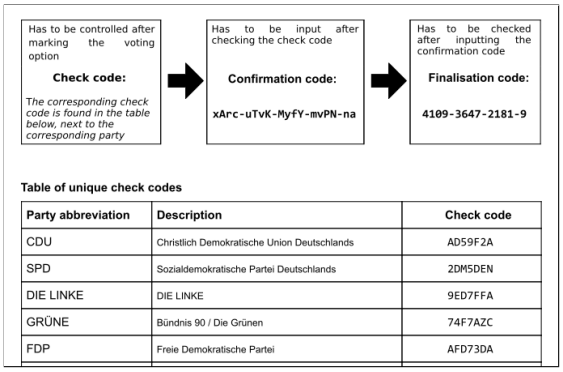

Fig. 2: The code sheet for the code-based approach.

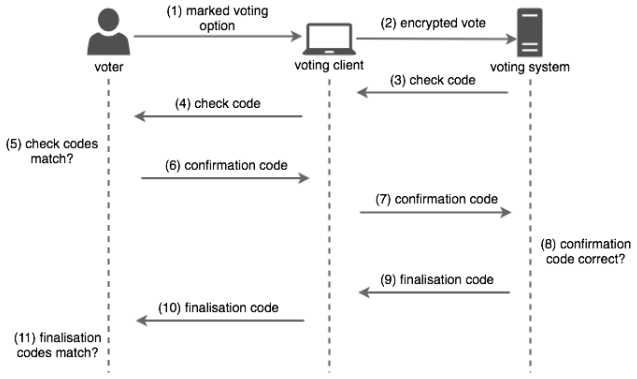

Fig. 3: Verification using the code-based approach.

The code-based verification process is depicted in Fig. 3. The individual verification steps are as follows: (1) The voter marks her chosen voting option via the voting client. (2)The voting client encrypts the chosen option and sends it to the voting system. (3) Upon receiving the cast vote, the voting system searches for the person's individualised return code corresponding to the cast voting option[5] and sends the return code back to the voting client. (4) The voting client displays the return code. (5) The voter checks whether the output return code matches the return code provided on her individualised code sheet. If the codes match, the voter knows that the voting system has received the intended vote. (Otherwise, the vote has been manipulated.) (6) If the voter is happy that the codes match, she inputs the so-called *confirmation code*, which is also found on her code sheet. (7) The voting client sends the input confirmation code to the voting system. (8) The voting system checks whether the provided confirmation code is correct. If the code is correct, the voter has verified that the cast vote reflects her intention. Otherwise, the voter has failed to confirm the correctness of the vote, either due to receiving an invalid return code or due to failing to perform the verification at all. In that case, the vote will not be stored by the voting system. (9) If the confirmation code is correct, the *finalisation code* is sent to the voting client. (10) The voting client displays the finalisation code. (11) The voter compares the displayed finalisation code to the finalisation code on her code sheet. If the codes match, the voter knows that her vote has been permanently stored in the voting system. A failure to receive the correct finalisation code would be evidence of hacker activity.

2.1 Security of the Approaches

In this section, we provide details of the security assumptions required by the two approaches.

Challenge-Based Approach: As shown in Figure 1, the challenge-based approach involves the following entities: the voter, the voting client, the verifier and the voting server. The verifiability assured by the challenge-based approach relies on the assumption that the voting client does not know, in advance, whether the encrypted vote will be cast or verified. It is therefore crucial to encourage voters to perform the verification, and to do so repeatedly. Otherwise, if each voter either only verifies once, or not at all, an adversary controlling the voting client could change votes after a single verification, knowing that their actions are unlikely to be detected. Another assumption is that either the voting client or the verifier is trustworthy. If they collaborate, manipulations will not be detected. That is why independent verifies are usually considered.

Code-Based Approach: As shown in Figure 3, the code-based approach involves the following entities: the voter, the voting client, the registrar, the code generator and the voting server.

[5] Note that this search is performed via cryptographic anonymisation techniques, so that vote secrecy remains intact.

The verifiability assured by the code-based approach relies on the assumption that the voting client can display the correct check code corresponding to the voter's chosen voting option only if the vote has indeed been cast as intended.

Note that we do not elaborate on the other security properties of the corresponding voting systems (e.g. vote privacy), or on other aspects of verifiability (e.g. whether the stored votes have been tallied correctly), see [4, 12] for more information.

2.2 Previous Studies on Human Factors in Verifiable E-Voting Systems

A number of studies have investigated human factors of verifiable e-voting systems. The research has focused on several aspects, including usability of verifiable voting systems, verification-related mental models and empirical studies of voter behaviour in verifiable real-world elections.

Several studies focused on usability at the polls. A number evaluated usability of the investigated systems in terms of satisfaction, e.g. [23, 24, 30, 37]. While the results of some studies revealed high satisfaction scores, they did not measure verification effectiveness in terms of the extent to which the participants were actually able to verify their votes. The evaluation of effectiveness was included in other studies, such as Pret-a-Voter and Scantegrity II [1, 2], BingoVote [5], StarVote [3] and EasyVote [8]. While some of these studies report high rates of success in terms of verifications [3, 8], the results of other studies [1, 2, 5] reveal several issues. These include misconceptions related to the verification process, leading to study participants being unable to complete the verification successfully. Other papers focus on evaluating the usability of verifiable Internet voting systems. In particular, different variants of the Helios voting system, which implements challenge-based verification approach, have been evaluated [1, 18, 25, 36], all revealing verification process issues. Of these, those that specifically measured verification success reported between 43% [1] and 81.25% [25] success rates. The usability of the code-based approach was investigated by [11], revealing usability issues, such as participants being confused about the different kinds of codes used for voting and verification. Marky *et al.* [26] investigated the usability of a variant of a code-based approach, the so-called 'code voting', by comparing different code modalities. The study reported that all of the participants were able to cast their vote successfully using all the modalities, and that the modality of using a QR code for entering the voting codes received the highest SUS score. However, no evaluation of verification has been done. The usability, in terms of satisfaction (without considering effectiveness), and acceptance of code-based verification (and variants) was evaluated in [20], which found that the participants were more willing to use a system with the highest security assurance in a real-world election, even if they also found it less usable due to complexity of entering and comparing different kinds of codes. Distler *et al.* [10] investigated the user experience related to the Selene voting system, which uses a verification approach based on tracking codes. While they did not evaluate the effectiveness of the verification, their results have shown that the participants felt less secure after verifying than before,

and that displaying the security mechanisms, such as mentioning of encryption during vote casting, makes the participants feel more secure, but at the same time perceive the voting app to be less understandable.

Other studies focused on the voters' verification-related mental models [28, 29, 32]. These studies revealed a number of factors that would potentially prevent voters from verifying, such as a lack of knowledge of verification procedures, perceived verification effort or misconceptions about verification itself.

Finally, empirical studies using data from real-world elections have evaluated the extent to which the voters actually verify their votes. As such, only 31.4% of surveyed voters verified their votes using the challenge-based approach in the Helios voting system in the IACR (International Association of Cryptographic Research) elections [19]. In the Estonian Internet voting system, where the verification process is similar to that used in the challenge-based approach[6], only 4% of voters verified their votes [17]. The surveys of the code-based approach report successful verifications between 70% [31] and 90% [35] (self-reported by the survey participants who participated in an election which used a voting system with code-based verification).

While many of the aforementioned studies revealed usability shortcomings and mental models preventing voters from verification in different voting systems, most of them did not attempt to compare different verification approaches directly. As such, only very few empirical comparisons between various cast-as-intended verifiability implementations in terms of effectiveness have been carried out [2, 25], none considering a code-based verification approach. It is furthermore difficult to compare the approaches using the data from the real-world elections, due to the differences in elections scenarios (i.e. elections conducted in different countries and in different settings) and in the methods the data was collected (i.e. log audits vs. self-reporting). When it comes to a comparison between code-based and challenge-based approaches, it therefore remains an open question, which one of them is more suitable to ensure the effectiveness of the verification.

3 Research Questions

We want to compare verification *effectiveness* of the two approaches. We also want to reveal usability issues leading to difficulties for participants during both vote casting and verification, in order to identify the need for further improvements.

Note, that we do not compare other metrics related to the voting itself, such as efficiency of the verification or satisfaction with the process. As such, comparing efficiency is particularly challenging: verification is conducted repeatedly in the challenge-based approach, as described in Section 2, while the voters using the code-based approach only have to verify once. Furthermore, because participants were confronted with manipulations, this could bias their perceptions of trustworthiness, rendering satisfaction measures unreliable.

[6] The main difference is that the verified vote in the Estonian voting system does not have to be discarded and can be cast post-verification. We refer to [15, 17] for more details.

RQ1: Our first research question is: *Are there any differences between the challenge- and code-based approaches, in terms of verification effectiveness?*

As mentioned in Section 2.2, various previous studies investigating the effectiveness of the two approaches report the rate of successful verifications to be between 70% and 90% for the code-based approach, and between 43% and 81.25% for the challenge-based approach. While, as mentioned previously, the results of these studies are not strictly comparable due to different study settings, they do give us an indication that the code-based approach might make it easier for voters to verify. We therefore propose to conduct a one-tailed comparison of the approaches and to test the following hypotheses:

H_0**:** Participants detect the manipulation of their vote equally using the code-based and challenge-based approaches.

H_1**:** Significantly more participants are able to detect the manipulation of their vote using the code-based approach compared to challenge-based approach.

To uncover usability problems that might explain efficacy issues, we also explore:

- Can participants complete verification? (If not, what prevents this?)
- Can participants verify their cast vote effectively i.e. do they detect anomalies that the verification process is designed to reveal? (If not, what prevents this?)

RQ2: Our second question is: *What factors encourage or discourage acceptance of Internet voting in real world elections?*

The next section will explain how we went about answering these questions in our study.

4 Methodology

The main goal of our investigation was to compare the verification effectiveness afforded by the two approaches. To do this, a between-subjects user study was carried out, with participants randomly assigned to one of the two verification approach groups. All participants were instructed to use a mock Internet voting system and requested to verify their votes. We recorded participants' feedback throughout the experiment to help us to identify verification process issues and to garner insights into potential interface and process improvements.

The purpose of the cast-as-intended verification step is to give the voter the opportunity to detect vote manipulations. If anomalies are detected as quickly as possible, during the election, the authorities can be informed and are able to take immediate remediation steps. Hence we engineered our system to deliberately change cast votes. This helped us to test verification efficacy. Participants who detected the manipulation were told to cast and verify a second vote in a so-called *second run*, and this second vote was not manipulated. Participants who did not detect the manipulation completed the experiment as normal, and were not required to cast another vote.

4.1 Ethics

The study adhered to the guidelines of the ethics commission at the first author's institution, where the research was carried out. All participants completed a consent form at the outset. The consent form informed them that their interactions with the website would be recorded via screen recording software, that the data collected from the study would be stored and processed anonymously for research purposes, and that the participants could drop out at any time without negative consequences. They were paid proportionally, according to the amount of time they spent doing the experiment.

4.2 Implementation

For the purpose of the study, a mock variant of each of the two approaches was implemented: one of which implemented the challenge-based approach, and the other the code-based approach. The German parliament elections were used as the election scenario. Correspondingly, the list of candidates included the political parties that were also present on the ballot during the 2017 parliament election. The mock website ballot interface provided in Fig. 4[7]. The participants of the study accessed the mock website via a lab laptop provided to them during the study. The website was accessed via the URL "`https://bundestagswahlen.de`". This actually directed the browser to the local host[8]

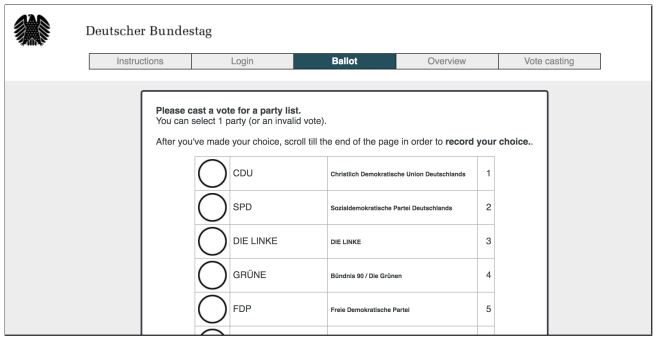

Fig. 4: The voting interface.

Challenge-based: The mock voting system was based on the mobile approach introduced in [27] and evaluated by [25]. For the purposes of this study,

[7] All the screenshots provided in the paper are translated from German. The study itself was conducted in German.

[8] This was achieved by modifying the local hosts on the laptop. In order to communicate website credibility, a SSL certificate was issued by creating a new certificate chain starting with a new certificate authority. The SSL certificate was imported into the browser, so that the participants of the study could see the reassuring green lock in the browser address bar.

a verifier app was developed and installed on a lab Smartphone (see Figure 5). The app collects timestamps of participants' interactions to support analysis.

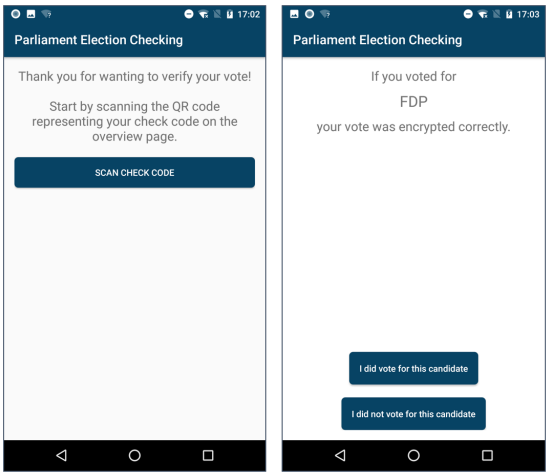

Fig. 5: The verifier app for challenge-based voting.

Code-based: Code sheets were designed (see Figure 2) with the codes consisting of seven alphanumerical characters (mirroring the original Neuchâtel voting system used in the elections in Switzerland[9]).

All interactions with the voting website were logged with time stamps, and screen recordings were made.

4.3 Background Story

The participants were told that the goal of the study was to test the usability of an Internet voting system. They received election information materials, as well as instructions related to their tasks. Participants evaluating the challenge-based approach used a lab Smartphone with the installed verifier app. Participants in the code-based group received a code sheet (similar to the one used in Switzerland).

Participants were given instructions in the form of a role card. For the code-based approach, the voter was instructed to cast a vote for the SPD[10] party and subsequently to verify their vote. For the challenge-based approach, the voter was instructed to verify a vote for SPD, and then to cast a vote for the Green

[9] Note, that even though the population in Switzerland is much smaller than Germany's, the code length allows for the generation of 36^7 unique codes, which makes the system suitable for the population of ca. 62 million. The code sheet resembled the original voting system's code sheets.

[10] German Social-Democratic Party

party[11]. Pre-defined parties were used to preserve vote secrecy because their interactions with the system were observed and recorded during the experiment.

4.4 Manipulation

The first cast vote was surreptitiously manipulated by the system to check that the voters are able to verify effectively. It was changed from SPD to FDP[12]. FDP was chosen due to its similarity to SPD and the proximity of these parties on the ballot, which made the manipulation harder to detect. The challenge-based approach makes this manipulation detectable because the verifier app correctly reflects the manipulated vote (FDP), instead of the intended vote (SPD) (see Fig. 5 on the right). Using the code-based approach, the voting system outputs a check code assigned to FDP on the code sheet (in our study, "AFD73DA" instead of "2DM5DEN", see Fig. 6).

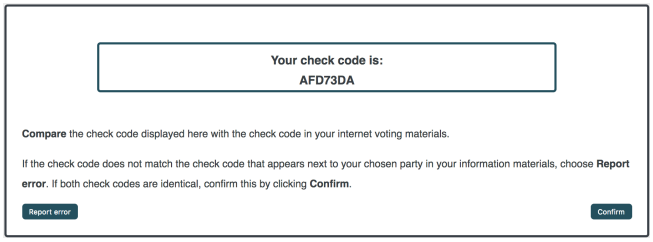

Fig. 6: The voting software's output during code-based verification.

4.5 Procedure

The study consisted of the following phases:

(1) Manipulation phase: Participants were welcomed and signed the consent form. The participants were given a role card outlining their tasks, as well as the voting information materials. After familiarising themselves with the role card, they proceeded to carry out their tasks. For the challenge-based approach group, this meant verifying a vote for SPD and casting a vote for the Green party; and, for the code-based approach group, casting and verifying a vote for SPD.

(2) Run-up phase: If a participant detected a manipulation, she was made aware that the manipulation had been introduced deliberately. The experimenter then installed an uncompromised voting system, and told the participant this. The participant was instructed to verify and cast their vote again. This time, no manipulation occurred, and participants could verify and cast their votes without hindrance.

[11] Note, re-casting is mandated by the challenge-based approach.
[12] German Free Democratic Party.

(3) Conclusion phase: Having used the system and not noticed the manipulation, or after the run-up phase, participants completed questionnaires. They were encouraged to express their opinions of the systems (i.e. whether they experienced any issues verifying), and demographic data was collected. We also asked them the question "*Would you use this Internet voting system in a real parliament election? Please explain your answer.*" We asked this to detect factors that would encourage or discourage acceptance of Internet voting in the future. Participants were debriefed.

5 Results

Participants were recruited via a snowball principle. They were told that the study would take approximately 15 to 30 minutes, based on pre-test experience, and offered remuneration of €10.

5.1 Participants

A total of 61 participants took part in the study, with 31 randomly assigned to the challenge-based approach group and 30 to the code-based approach group. Of these, 35 were male, 25 female and one participant preferred not specify their gender. Participant ages ranged from 19 to 67, with a median age of 33. When asked about their highest level of education, 7 had completed middle school, 10 had graduated from high school, 8 had completed an apprenticeship, 9 had a degree from the university of applied sciences, 25 had a university degree, and 2 had a doctorate.

5.2 RQ1: Verification Effectiveness

All the participants in the code-based approach group were able successfully to verify their vote and detect the manipulation during the manipulation phase. Furthermore, all were able successfully to complete the verification during the run-up stage. In the challenge-based group, seven of the participants were unable to perform verification during the manipulation stage.

A one-tailed Fisher's exact test compared the number of successful verifications during the manipulation phase and revealed a significant difference between the groups ($p < 0.01$). H_1 is thereby supported.

Usability Issues

We observed the participants' behaviours as they used the system. If a participant commented on some issue while completing the tasks, this was recorded. We also examined the screen recordings to see whether the participants made errors while engaging with the vote casting or verification interface. For example, they could have clicked on the wrong button or entered the wrong code. We also analysed the participants' responses to the final questionnaire.

Usability Issues with Challenge-based approach. A number of participants were unable to verify. Furthermore, several participants experienced problems casting their votes during the run-up phase. We describe the issues these participants experienced, as well as issues mentioned by successful participants.

One important issue was the **confusion about the verification process.** Some comments: "*too cumbersome, difficult to understand*" or mentioning that they had to "*develop an understanding of the system and what has to be scanned*". They were also confused due to presence of two QR codes that had to be scanned in the correct order when using the verifier app. They expressed the need for clearer instructions: "*Make it clearer, which QR code should be scanned*". Further improvement suggestions focused on overall better and more understandable instructions for verification: "*Purpose of the verification, and maybe a flowchart: What is assured during each step?*".

Indeed, such confusion also affected participants' ability to verify their vote. Of 31 participants, seven were unable to verify. One did not attempt to verify, admitting that the process seemed too complicated despite the instructions. The other six only scanned the check code (Step (3) in Figure 1), then not knowing how to proceed. Half thought that they had verified successfully, while the other half admitted that they had had problems verifying.

Some participants experienced **confusion about vote re-casting**, as the challenge-based approach requires the voter to discard verified votes: "*Evidently, I only verified, but did not cast my vote. Have not understood the system yet*". '*At the end I forgot, that the vote will be deleted after verification, and almost did not cast a vote*".

Improvement suggestions: "*Maybe the hint that the vote has to be cast again after verifying needs to be more visible and salient on every page*".

As evidenced from the screen recordings and observations, three of the participants in the challenge-based group also failed to recast their vote, and another three only succeeded in casting their vote after several attempts. In all cases, the participants proceeded to verification by clicking on the "Verify" button when they tried to cast their vote. Because the challenge-based approach does not allow the casting of a verified vote, the ballot was discarded and the participants were redirected to a page requiring them to cast their vote again. Some were able to figure out how to cast their vote, but others gave up.

In addition to mentioning the issues related to the non-intuitive process of verifying and vote casting, a number of participants lamented the **complexity of the voting system.** They asked for more information: "*Additional information for people with less technical experience*". Other proposed improvements to clarity of the website interface included: "*Clearer instructions (another formulations, reordering of buttons, another choice of colors*". "*Reflect status with colors and symbols, e.g. red for a failed verification*".

Usability Issues with Code-Based Approach. All participants detected the manipulation, completed the verification and cast their votes during the run-up

phase. Nonetheless, a number of issues emerged during the study, highlighting the potential for improvement.

The **complexity and length of the codes** was raised as an issue by several participants: "*Usability can be better; especially regarding complex codes*". When asked to propose improvements, several participants focused on making it easier to enter and compare codes. As such, use of additional hardware, such as tokens similar to TAN-generators for online banking, or RFID-cards, have been suggested: "*A transponder, such as in [anonymised] or online banking...*". Indeed, the issue of code complexity emerged from the screen recordings. While all participants were able to verify and cast their votes, two did so only after several attempts.

A number of participants expressed their **need for more information** and desire to know more about the technical details of the system: "*As an IT-specialist I would love to know more about the storage of data and encryption*". This was also mentioned as an important predicor of system acceptance: "*The need/comprehensibility of the check code should be explained better (for acceptance). Same for the code of 'encryption*".

A number of participants suggested **improving the instructions** and information materials. Several suggested aligning different instructions more effectively: "*Build in step-by-step screenshots in the information materials*"."*Use symbols for simpler matching between verification steps on paper and on computer screen*".

Several participants required **more reassurance** from the system, or extra options to confirm vote verification: "*As the last step, not just a finalisation text, but a button*". "*I would like to confirm the finalisation code*". "*The confirmation page can be more formal and leave a code with which one could confirm that the voting was successful*".

A few responses included suggestions for maintaining the **security of the verification process**: "*It is good when the system provides the option to report a manipulation or highlights the possiblity – with this, the user will be more careful and attentive*". "*If the codes of different parties are too similar, the comparison by the voter is prone to making errors*".

Finally, a number of **user interface design improvements** were suggested. In particular: ("*Bigger buttons and images, support for the visually impaired*"); ("*For inputting the confirmation code, focus should move to the next form field*).

Provision of a demo system was proposed, so that potential voters can familiarise themselves with the voting and verification processes before the election: "*Set up a test election with fake parties*".

Some participants were **sceptical about the integrity of the verification.** Even a positive verification did not alleviate mistrust: "*Still not clear, how the verification can ensure that at the end the vote that is being transferred and processed and has not been manipulated*".

5.3 RQ2: Acceptance of Internet Voting

We focused our analysis on the code-based approach, as it was better than the challenge-based one in affording effective verification, as demonstrated by

the aforementioned analysis. The responses to all responses were analysed via independent and agreed-upon open coding by at least two authors.

Reasons for acceptance. Overall, of 30 participants who used the code-based approach, 21 said that they would be willing to use the system in real-world elections. The following reasons were cited:

Convenience: This was a common theme: participants were positive about the proposed system: "*It saves me the way to the polling station, and if I want to apply for postal vote, the post traffic*".

Reassurance: The presence of verification procedure positively influenced the perceived security of the system among some of the participants: "*seems secure due to multiple verification steps.*"

Reasons for non-acceptance. 9 of the 30 participants were unwilling to use the system in a real-world election, citing the following reasons:

Security concerns: A general mistrust towards Internet voting were present among several participants: "*Does not matter how secure the Internet voting system can be, there are always ways to manipulate it.*".

Complexity of the system and lack of transparency: This was commonly mentioned as one of the reasons for not using the proposed voting system: "*Too many random strings. Encryption? What is it? How does it work?*".

6 Discussion

Evaluation Results: Our study shows that the code-based approach outperforms the challenge-based approach with respect to effectiveness. Indeed, while all of our participants were able to verify their vote using the code-based approach, only 77.4% (24 out of 31) were able to do so with the challenge-based approach. Furthermore, of the seven participants who could not verify, three were convinced that they had indeed verified successfully. Such shortcomings would leave manipulations undetected.

For these reasons, we conclude that the code-based approach is superior to the challenge-based approach with respect to verification efficacy. However, as described in Section 2, these approaches rely on different security assumptions, such as the need to rely on the trustworthiness of various voting system components in the code-based approach, or the need to rely on the trustworthiness of the verifier (i.e. a smartphone with an installed verifier app) for the challenge-based approach. Therefore, the person deciding on which approach to be used in a particular election scenario has to take these assumptions into consideration.

Our study revealed a number of issues with the investigated approaches. The participants who used the challenge-based approach were particularly confused about the relationship between verification and vote casting, not appreciating that a verified vote cannot be cast. This confusion led to several participants failing to cast their votes. Although the interfaces provided instructions explaining that the vote had to be discarded after verification, these were evidently not understandable or noticeable enough. This issue was also revealed by other studies

into the usability of the challenge-based approach, the most recent being the study reported by [25].

The process of discarding a verified vote, only to cast another *unverified vote*, is counter-intuitive and conflicts with pre-existing paper-based voting mental models. Moreover, asking voters to verify more than once will probably confuse them even more. There does not seem to be an easy way to improve this situation, from a usability perspective, because the problems are caused by the protocol of the underlying system.

Another issue related to the challenge-based approach was reported by [25]. This is the lack of feedback, which led some voters to abort verification prematurely, all the while thinking that they had indeed verified successfully. In particular, participants in our study stopped after scanning the first QR code (the check code). The instructions, both on the website and in the verifier app, asked them to scan a second QR code (the ballot code). These were seemingly overlooked.

For the code-based approach, the most prominent issue mentioned by the participants was the length and complexity of the codes that they had either to input or compare. This issue was also reported by [25]. While their study did not investigate the code-based approach, it included the evaluation of a variant of the challenge-based approach where the check code had to be written down and then compared manually, instead of using the verifier app. Similar to our study, the participants in [25] commented on the complexity of the code and expressed their concerns that the voters might make errors either writing down or comparing the codes. While the complexity of the codes is inherent in the system, to satisfy security requirements, the difficulties that the participants experienced suggest that other modalities for representing the codes should be investigated, such as, for example, passphrases or visual hashes.

The results of our study show that even though all of the participants were able to cast and verify their votes using the code-based approach; one third were still unwilling to use the system in a real-world election. They cite such reasons as security concerns, complexity of the system and lack of transparency. This suggests that while it is crucial to ensure that the Internet voting systems can be used effectively by the voters, this is not enough – a point that was also argued in previous research [21]. Yet, other studies show that voters would be ready to accept Internet voting systems that involve complex steps in both vote casting and verification, if the security of the system, which generates the complexity, is communicated to them [20]. This suggests that ways to inform the voters about the underlying mechanisms of the system, and the extent to which these ensure the security of the election, would be helpful in encouraging greater acceptance of Internet voting.

Finally, we should make the point that verification is explicitly provided in order to allay voter concerns about the integrity of a system that replaces paper-based voting. That being so, it is entirely possible that as people become accustomed to using Internet voting systems they will trust them more, and see even less need to verify. This would be a very interesting topic for a future study.

Limitations and Future Work: Many of the participants of our study were relatively young and highly educated. Because these demographics generally tend to be early adopters of new technologies, several studies on the demographics of Internet voting in different countries show that young and highly educated voters do tend to be over-represented among the voters who chose to cast their vote online [16, 33, 34]. Furthermore, people from these demographics generally have more advanced computer skills, as compared to the rest of the population. The usability issues they experiences are very likely affect other demographic groups, perhaps even more severely. It is worth noting that Internet voting is almost never the only available voting channel in practice, so that voters who do not feel confident using the Internet voting system still can cast their ballots in the traditional way. However, in order to study the effect of verification procedures amongst more general population, further studies might be needed.

The focus of our study was on the effectiveness of the verification process. As such, we did not measure the user experience and the satisfaction with the investigated systems. The reason for this was the design of our study, which included the manipulations of the vote. While these manipulations were necessary in order to measure the effectiveness with which they carried out the verification task, they also biased the user experience, as the voting system was intentionally designed to behave anomalously. The investigation of user experience in the cast-as-intended approaches, and the role it plays on the acceptance of the Internet voting systems, remains a topic for future research.

The participants were explicitly requested to verify their vote. However, empirical studies have shown that voters are often not motivated enough to verify [17]. This becomes a particular problem when verification is optional and not incentivised. Moreover, while the verification step is mandatory in the code-based approach, it is entirely possible for a participant simply to click through the verification of the check code without comparing it to their code sheets. Further studies are necessary in order better to model the voter's behaviour with respect to cast-as-intended verification in real-world elections.

Finally, our study only considered the attack scenario where an adversary manipulates a cast vote. We did not consider cases where the adversary might also attempt to manipulate the interface of the voting website (see [21] for examples). Such an attempt might be detected by attentive voters but voters could misunderstand verification instructions or overlook important information – as we observed during our study. Investigating the extent of such manipulations, and developing countermeasures, is an important direction for future investigations.

7 Conclusion

We carried out a study to compare the implementations of two verification approaches, one being challenge-based and the other code-based. We wanted to see whether people could verify their cast votes successfully, which entailed spotting a deliberately-introduced vote manipulation. We observed them carrying out the tasks, and recorded any errors they made, as well as their reservations,

difficulties they experienced during the process, and suggestions for improvement. We found that participants in the code-based group could verify more successfully, and that this group also detected our introduced manipulation more reliably than those in the challenge-based group. In other words verification effectiveness in the code-based group was superior to that of the challenge-based group. We also revealed a worrying unwillingness to engage with Internet voting, even amongst our young and educated participants. If they do not accept voting it is even less likely that older demographics will do so.

Our analysis revealed that the idea of verification, being a fairly alien concept, is problematic. More needs to be done to familiarise voters with the differences between paper and Internet voting, to prepare them for the requirements of the new digital voting era.

Acknowledgements

This work was partially conducted within the Center of Information Security and Trust at the IT University of Copenhagen (ITU CIST) and also supported by the German Federal Ministry of Education and Research within the Competence Center for Applied Security Technology (KASTEL).

References

1. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II. The USENIX Journal of Election Technology and Systems **2**(3), 26–56 (2014)
2. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: From error to error: Why voters could not cast a ballot and verify their vote with Helios, Prêt à Voter, and Scantegrity II. USENIX Journal of Election Technology and Systems (JETS) **3**(2), 1–19 (2015)
3. Acemyan, C.Z., Kortum, P., Byrne, M.D., Wallach, D.S.: Summative usability assessments of star-vote: A cryptographically secure e2e voting system that has been empirically proven to be easy to use. Human factors p. 0018720818812586 (2018)
4. Adida, B.: Helios: Web-based Open-Audit Voting. In: USENIX Security Symposium. vol. 17, pp. 335–348. USENIX Association, Berkeley, CA, USA (2008)
5. Bär, M., Henrich, C., Müller-Quade, J., Röhrich, S., Stüber, C.: Real world experiences with bingo voting and a comparison of usability. In: IAVoSS Workshop On Trustworthy Elections (WOTE 2008) (2008)
6. Benaloh, J.: Simple Verifiable Elections. Electronic Voting Technology Workshop **EVT '06** (2006)
7. Benaloh, J.: Ballot casting assurance via voter-initiated poll station auditing. Electronic Voting Technology Workshop **EVT '07** (2007)
8. Budurushi, J., Renaud, K., Volkamer, M., Woide, M.: An investigation into the usability of electronic voting systems for complex elections. Annals of Telecommunications **71**(7-8), 309–322 (2016)
9. Chang-Fong, N., Essex, A.: The cloudier side of cryptographic end-to-end verifiable voting: A security analysis of Helios. In: Proceedings of the 32nd Annual Conference on Computer Security Applications. pp. 324–335. ACM (2016)

10. Distler, V., Zollinger, M.L., Lallemand, C., Roenne, P., Ryan, P., Koenig, V.: Security–visible, yet unseen? how displaying security mechanisms impacts user experience and perceived security. In: Proceedings of ACM CHI Conference on Human Factors in Computing Systems (CHI2019) (2019)
11. Fuglerud, K.S., Røssvoll, T.H.: An evaluation of web-based voting usability and accessibility. Universal Access in the Information Society **11**(4), 359–373 (2012)
12. Galindo, D., Guasch, S., Puiggali, J.: Neuchâtel's Cast-as-Intended Verification Mechanism. In: International Conference on E-Voting and Identity. pp. 3–18. Springer, Bern, Switzerland (2015)
13. Gharadaghy, R., Volkamer, M.: Verifiability in Electronic Voting-Explanations for Non Security Experts. In: Electronic Voting. pp. 151–162 (2010)
14. Halderman, J.A., Teague, V.: The New South Wales iVote system: Security failures and verification flaws in a live online election. In: International Conference on E-voting and Identity. pp. 35–53. Springer, September 2-4, 2015 Bern, Switzerland (2015)
15. Heiberg, S., Martens, T., Vinkel, P., Willemson, J.: Improving the verifiability of the Estonian Internet Voting scheme. In: International Joint Conference on Electronic Voting. pp. 92–107. Springer (2016)
16. Heiberg, S., Parsovs, A., Willemson, J.: Log analysis of Estonian internet voting 2013–2014. In: International Conference on E-Voting and Identity. pp. 19–34. Springer (2015)
17. Heiberg, S., Willemson, J.: Verifiable Internet Voting in Estonia. In: 6th International Conference on Electronic Voting, Verifying the Vote (EVOTE). pp. 1–8. IEEE (Oct 2014)
18. Karayumak, F., Olembo, M.M., Kauer, M., Volkamer, M.: Usability Analysis of Helios-An Open Source Verifiable Remote Electronic Voting System. In: Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections. EVT/WOTE'11, USENIX Association, Berkeley, CA, USA (2011)
19. Kiayias, A., Zacharias, T., Zhang, B.: Ceremonies for end-to-end verifiable elections. In: IACR International Workshop on Public Key Cryptography. pp. 305–334. Springer (2017)
20. Kulyk, O., Neumann, S., Budurushi, J., Volkamer, M.: Nothing comes for free: How much usability can you sacrifice for security? IEEE Security & Privacy **15**(3), 24–29 (2017)
21. Kulyk, O., Volkamer, M.: Usability is not enough: Lessons learned fromhuman factors in securityresearch for verifiability. E-Vote-ID 2018 p. 66 (2018)
22. Langone, A.: An 11-Year-Old Hacked Into a U.S. Voting System (2018), `http://time.com/5366171/11-year-old-hacked-into-us-voting-system-10-minutes/` August 14
23. Mac Namara, D., Gibson, P., Oakley, K.: A preliminary study on a dualvote and prêt à voter hybrid system. In: CeDEM 12 Conference for E-Democracy and Open Government 3-4 May 2012 Danube-University Krems, Austria. p. 77. Edition-Donau-Univ. Krems (2012)
24. Mac Namara, D., Scully, T., Gibson, P.: Dualvote addressing usability and verifiability issues in electronic voting systems (2011), `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.399.7284`
25. Marky, K., Kulyk, O., Renaud, K., Volkamer, M.: What Did I Really Vote For? In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. p. 176. ACM (2018)

26. Marky, K., Schmitz, M., Lange, F., Mhlhuser, M.: Usability of Code Voting Modalities. In: CHI Conference on Human Factors in Computing Systems, Glasgow, Scotland UK. CHI Conference on Human Factors in Computing Systems Late Breaking Work. ACM (2019)
27. Neumann, S., Olembo, M.M., Renaud, K., Volkamer, M.: Helios Verification: To Alleviate, or to Nominate: Is That the Question, or Shall we Have Both? In: International Conference on Electronic Government and the Information Systems Perspective. pp. 246–260. Springer (2014)
28. 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)
29. Olembo, M.M., Bartsch, S., Volkamer, M.: Mental Models of Verifiability in Voting. In: International Conference on E-Voting and Identity. pp. 142–155. Springer (2013)
30. Oostveen, A.M., Van den Besselaar, P.: Users' experiences with e-voting: A comparative case study. Journal of Electronic Governance **2**(4) (2009)
31. Puiggalí, J., Cucurull, J., Guasch, S., Krimmer, R.: Verifiability experiences in government online voting systems. In: International Joint Conference on Electronic Voting. pp. 248–263. Springer (2017)
32. Schneider, S., Llewellyn, M., Culnane, C., Heather, J., Srinivasan, S., Xia, Z.: Focus group views on pret a voter 1.0. In: Requirements Engineering for Electronic Voting Systems (REVOTE), 2011 International Workshop on. pp. 56–65. IEEE (2011)
33. Serdült, U., Germann, M., Harris, M., Mendez, F., Portenier, A.: Who are the internet voters? Innovation and the Public Sector **27**, 27–41 (2015)
34. Serdult, U., Germann, M., Mendez, F., Portenier, A., Wellig, C.: Fifteen Years of Internet Voting in Switzerland [History, Governance and Use]. In: ICEDEG 2015: 2nd International Conference on eDemocracy & eGovernment. pp. 126–132. IEEE (Apr 2015)
35. Stenerud, I.S.G., Bull, C.: When reality comes knocking. Norwegian experiences with verifiable electronic voting. Electronic Voting **205**, 21–33 (2012)
36. Weber, J.L., Hengartner, U.: Usability study of the open audit voting system helios. `http://www.jannaweber.com/wpcontent/uploads/2009/09/858Helios.pdf` (2009), [Online; accessed: 22-December-2017]
37. Winckler, M., Bernhaupt, R., Palanque, P., Lundin, D., Leach, K., Ryan, P., Alberdi, E., Strigini, L.: Assessing the usability of open verifiable e-voting systems: a trial with the system prêt à voter. Proceedings of ICE-GOV pp. 281–296 (2009)