Increasing security without decreasing usability: A comparison of various verifiable voting systems

Melanie Volkamer, *Karlsruhe Institute of Technology;* Oksana Kulyk, *IT University Copenhagen;* Jonas Ludwig and Niklas Fuhrberg, *Karlsruhe Institute of Technology*

https://www.usenix.org/conference/soups2022/presentation/volkamer

This paper is included in the Proceedings of the Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022).

August 8–9, 2022 • Boston, MA, USA

978-1-939133-30-4

Open access to the Proceedings of the Eighteenth Symposium on Usable Privacy and Security is sponsored by USENIX.

Increasing security without decreasing usability: A comparison of various verifiable voting systems

Melanie Volkamer
Karlsruhe Institute of Technology
melanie.volkamer@kit.edu

Oksana Kulyk
IT University of Copenhagen
okku@itu.dk

Jonas Ludwig
Karlsruhe Institute of Technology
jonas.ludwig@student.kit.edu

Niklas Fuhrberg
Karlsruhe Institute of Technology
niklas.fuhrberg@student.kit.edu

Abstract

Electronic voting researchers advocate for verifiable voting schemes to maximise election integrity. In order to maximise vote secrecy, so-called code-voting approaches were proposed. Both verifiability and code voting require voters to expend additional effort during vote casting. Verifiability has been used in actual elections, but this is not the case for code voting due to usability concerns. There is little evidence from empirical studies attesting to its usability. Our main contribution is to extend an existing verifiable voting system (used for real world elections) with a code-voting approach to improve the system's security properties. We minimise voter effort as corresponding QR codes are scanned instead of requiring manual code entry. We conducted a user study to evaluate the general usability of this proposal as well as its manipulation-detection efficacy. In particular, we found that extending the considered verifiable voting systems with code-voting approaches to enhance vote secrecy is feasible because we could not observe a significant decrease in general usability while manipulation detection improved significantly.

1 Introduction

The pandemic caused an increasing number of organisations to contemplate vote casting over the Internet for secret polls, and governments are also considering online elections. However, this requires deployment of various cryptographic techniques to ensure vote secrecy and election integrity. One of these techniques is *individual verifiability*. It allows voters to verify that their vote is cast correctly and also that their vote has been recorded as cast. Another one is universal verifiabilty which provides strong cryptographic proofs, that enable independent third parties to verify that the final tally correctly reflects all recorded votes. Together, they facilitated detection of attacks on the election's integrity and, in the absence of detected manipulations, permits strong guarantees of election integrity. The level of achieved vote secrecy depends on the verifiable voting scheme being in place. However, most verifiable voting schemes do not defend against a compromised voting client (i.e., the voter's laptop, smartphone or vote-casting application) that can violate vote secrecy despite the presence of verifiability.

From a usability perspective, individual verifiability is particularly challenging, as this requires voters themselves to undertake extra steps in addition to casting their vote. Therefore, it is not surprising that a range of usability and manipulation-detection efficacy studies have been carried out, e.g. [1–17]. Yet, the studied verifiable voting schemes rely on the trustworthiness of voting clients with respect to vote secrecy.

Our paper focuses on the verifiable voting system used in Switzerland[1] for elections and referenda. The usability of this system and corresponding voting materials -– in general, as well as with respect to its efficacy to enable voters to detect manipulations — was evaluated and improved by [16, 17]. The Swiss system assumes that attackers cannot manipulate voters' devices nor the vote-casting application. If this assumption does not hold, vote secrecy could be violated.

One way to address the question of voting client trustworthiness is *code voting*. This means that voters cast their vote by entering so-called voting codes, which are uniquely assigned and delivered to each individual voter: one per voting option. Because the vote casting device cannot map the voting code entered to any of the options, the assumption of trustworthy voting clients is no longer required to ensure vote secrecy. Code-voting schemes have already enjoyed attention from

USENIX Symposium on Usable Privacy and Security (SOUPS) 2022.
August 7–9, 2022, Boston, MA, United States.

[1] Switzerland is one of the few countries to allow Internet Voting for political elections and referenda. As a direct democracy, it holds several elections/referenda per year. Most of their elections and referenda are based on simple ballots (*m* out of a small number of *n* options).

security researchers, e.g. by [18–22]. However, these schemes have not been used for real elections. Usability concerns are often voiced by election officials due to the additional complexity of handling these codes. To the best of our knowledge, only two user studies were conducted in which code-voting approaches were included. In [23], three different code-voting approaches were evaluated in a within-subjects study. None of these approaches provided any means of verifiability. In [24], the authors compared three vote casting approaches (each with a different security level), including one approach with voting codes and a confirmation code to enable individual verifiability. The within-subject study mainly evaluated acceptance of the three approaches. While the authors also report on System Usability Scales (SUS) for each approach, the code-voting and verifying approach responses might have been influenced by the fact that participants used two less secure and also less complicated approaches beforehand. Note, the efficacy in enabling voters to detect manipulations for code-voting based verifiable schemes has not been evaluated by any user studies, yet.

In this paper, we make two proposals to improve the security of one concrete implementation of a verifiable voting scheme: The Swiss voting system. In essence, first, we extend the Swiss verifiable system by code-voting, thus improving it towards vote secrecy (*proposal-code-voting-with-QR-codes*). To address potential usability shortcomings, we propose that voters are issued with QR codes, so that they can use a camera-equipped device, most likely their smartphone, to cast their vote. These QR codes also contain the so called initialisation code. This allows us to use longer initialisation codes compared to those voters need to manually entered in the original system. Thereby, we also improve the security of voter authentication compared to the original Swiss voting system. Second, we propose a variant of the Swiss system (*proposal-standard-voting-with-QR-codes*) that does not rely on code voting but still uses QR codes for the so called initialisation code. This second proposal is equally vulnerable to vote secrecy violations from malicious voting clients as the original system. However, it provides more evidence with respect to voter authentication as compared to the original system. Thus, from a security perspective, our first proposal out-performs the second, and both out-perform the original system and thereby also its improved versions from [16, 17]. Note, from a security perspective, our first proposal also out-performs the approaches evaluated in [23] (which uses code voting, but does not provide any means to verify).

Besides proposing these two improvements to the Swiss verifiable voting system, the goal of this paper is to report on the evaluation of both systems in terms of general usability as well as in terms of manipulation-detection efficacy. We compared our results with those of the original system reported in [16] and to relevant related work. We developed a study protocol which facilitated remote participation, i.e., study materials incl. the voting materials was sent through the post, because Covid-regulations did not permit face-to-face user studies. Using this protocol, we conducted a user study – consisting of two between-subjects experiments – with 139 participants. The first experiment evaluated the general usability of both proposals. The second experiment evaluated their manipulation-detection efficacy.

In summary, our contributions are as follows[2]: (1) We make *two proposals* to improve security of the Swiss voting system. (2) We evaluate the *general usability* of these two proposals. We compare our results in comparison to those reported by Kulyk et al. in [16] for the original system.We did not detect a decrease in general usability of our proposals as compared to the original system. The average SUS scores are compared to those reported in relevant related work. (3) We evaluated the *efficacy* of our two proposals with respect to manipulation detection. Both performed significantly better in this as compared to the original system (using the data from [16]). We also compare the manipulation-detection efficacy of our proposals with those reported in relevant related work. (4) We propose a study protocol which allows user studies to be conducted *remotely* in the context of electronic voting. We also discuss lessons learned.

With our research, we are in particular the first to study the efficacy in enabling voters to detect manipulations for a code-voting based verifiable schemes. While there is also room for further improvements to enable even more voters to detect manipulations, our research shows that the Swiss election officials should consider extending their system as proposed in this paper. In general, our research indicates that it is worth considering more QR-code-enabled code-voting for verifiable voting systems, because they can be implemented in such a way that their usability is comparable to the usability of extant systems while it is not required anymore to trust that voting clients are trustworthy.

2 Related Work

Several security analyses of electronic voting systems have found serious vulnerabilities, e.g., [26–28]. These results show the importance of verifiability. Verifiable voting schemes enable voters, candidates, and election officials to check whether or not the voting system has been manipulated. Unsurprisingly, the research community focuses on such verifiable voting schemes. Several were already being used for real elections and secret polls.

Several studies have evaluated the usability of verifiable electronic voting systems, e.g. [1–16, 24]. While most of these studies focus on one of five approaches that enable voters to verify, Marky et al. compared approaches with each other [15]. Most of the user studies reported good results for the general usability. The manipulation-detection efficacy results were mixed. In particular those studying actual systems in use have shown less optimistic results with regards to manipulation

[2]One of the two proposals (i.e. the proposal-code-voting-with-QR-codes was presented in a work-in-progress paper together with its general usability evaluation [25] – but without the evaluation on its manipulation detection efficacy.

detection. This is the case for the Norwegian Internet voting system in [11], and for the Swiss voting system in [16, 17].

Some researchers studied user perceptions and mental models of verifiable voting systems. Distler et al. [14], for example, found that users felt less secure after having verified. Other works, e.g. [1, 9, 10, 13, 29], identified misconceptions which prevented study participants from verifying their vote. These studies have concluded not only that improvements need to be made in the usability of verifiable voting systems needs to be improved. They also stressed the importance of properly communicating the 'extra' steps needed to verify.

While verifiability is used to maximise the election integrity guarantees, an important building block for maximising vote secrecy is the use of voting codes. Here, voters receive an individual code for each voting option. These codes are usually sent via the national postal service. Instead of selecting their option on the screen, they enter the corresponding voting code. The security of code-voting schemes have already been studied, e.g., in [18–22]. The usability of code-voting schemes, however, has been the subject of only two studies, i. e. [23, 24].

The usability of three different approaches to implement a non-verifiable code-voting scheme have been evaluated in [23]. In a between-subject study with 18 participants, they used the SUS items to report on the usability of three approaches to enter voting codes: (1) manually entering the code, (2) scanning a corresponding QR code from a booklet of QR codes, and (3) tangible objects. The focus of the study was on the process of casting a vote, while other steps, such as voter authentication, were not part of the participants' task. Furthermore, there was also no verifiability in place. The mean SUS performance for the manual approach was 61.25, for the QR-code approach 84.02, and for the tangible objects 78.61. While the study has several limitations, the authors found that code voting as such can be usable. In particular they report that the QR-code approach significantly out-performs the manual approach in terms of usability. We were inspired by their work to further consider code voting based on QR codes. However, we wanted to propose an entire system that provides both voter authentication and verifiability – and not just consider vote casting alone. We thereby explored using QR codes instead of having voters manually entering codes, because QR codes have advantages not only in terms of usability but also in terms of security: QR codes enable voters to enter large codes much more efficiently and effectively. Larger codes can significantly improve the security of verifiable voting schemes both with code-voting (see our proposal-code-voting-with-QR-codes) and without code-voting (see our proposal-standard-voting-with-QR-codes).

Three voting schemes on different security levels were evaluated using a between-subjects study in [24]. One of their schemes employed a verifiable code-voting. The authors mainly studied the impact of explaining the need for the various additional security-related steps (besides clicking on the voter's preferred candidate) on user acceptance. Because of the need to explain each scheme, all participants interacted with the three schemes in the same order. The authors conclude that although the verifiable code-voting scheme obtained a SUS value of only 67, the participants tended to prefer this less usable system to a more usable but less secure system. The authors also report that the SUS performance for the code-voting approach was significantly lower than the none-code-voting approaches. However, participants may have been biased when judging the verifiable code-voting scheme, having already interacted with two more usable approaches. The research focus of this study differed from ours (acceptance versus usability/manipulation-detection efficacy). In particular, the authors did not study manipulation-detection efficacy. Furthermore, the verifiable code-voting approach in [24] requires voters to manually enter voting codes while in our proposal, codes are entered by scanning corresponding QR codes.

3 Background

We begin by first explaining the Swiss online voting system, as we propose security improvements for it (see Section 4). We then summarise the usability improvements that have been proposed for this system by Kulyk et al. [16] for this system.

3.1 The Swiss Electronic Voting System

In the Swiss voting system, the process of casting a vote proceeds as follows (see also Figure 4 in the Appendix, which shows the underlying voting scheme): Voters receive an individual code sheet (usually called a polling sheet) via the postal service, containing one initialisation code, check codes for each voting option, one confirmation code, and one finalisation code – all being unique for each voter. It should be noted that Switzerland has no electronic ID system by which voters could be authenticated online. Therefore, the system generates an election-specific key pair for all voters. The private key is derived from the initialisation code.

To start the vote casting process, voters manually enter their initialisation code (which is provided to them on their polling sheet) by typing the corresponding characters in the corresponding field of the election webpage and then selecting their voting option using the election webpage. The election webpage then displays a check code with which voters are supposed to compare the code next to their voting option on their polling sheet. If the check codes match, the voter confirms the correct code by (again manually) entering the confirmation code. If the check code is incorrect (or no check code is displayed), the voter is supposed to complain. Finally, voters receive a finalisation code that should match the code on their polling sheet, as a confirmation, that their vote has been recorded. If this is not the case, again they are supposed to submit a complain. It should be noted here that the check code would be sufficient to verify. From from an organisational perspective, however, it is recommended that there are two additional steps and codes, respectively, in order to allow the

system to be able to react to complaining voters; in this way, voters reporting issues ca be offered an alternative voting channel (postal or in person). Furthermore, the communication between the voters' devices and the election infrastructure is secured on the transport layer using TLS.

Usability and Security Considerations. According to the requirements in [30], the initialisation must have at least 20 characters, each check codes must have at least four digits, the confirmation code must have at least nine digits, and the finalisation code eight digits. Thus, in terms of general usability, the most error-prone task is to manually enter the initialisation code as well as the confirmation code. More information on the underlying scheme, e.g. how the election infrastructure[3] computes the codes to be sent back in distributed manner and how the votes are tallied in a verifiable way, is provided at [31]. The Swiss voting system relies on the trustworthiness of the voting client (i.e., the voters' laptops or smartphones, and the vote-casting application) for ensuring vote secrecy. The voting system also assumes that the printers in charge of printing the polling sheets do not maliciously cooperate with the voting client to violate election integrity. For the second assumption to be realistic, printing presses are operated offline. We refer to this system including the voting materials and user interfaces as the **'original system'**.

3.2 Improvements and Study Reported in [16]

Two papers have proposed and evaluated usability improvements for the Swiss system [16, 17][4]. Both approaches propose changes in the design of the polling sheet to a more step-by-step instruction and a reduction in the information provided on the election webpage. We base our research on the improvement from Kulyk et al. from [16] for the following reasons. First, Kulyk et al. studied manipulation-detection efficacy with respect to two different manipulation approaches, while Marky et al. studied only one. Second, the study design in [17] is less reliable with respect to manipulation-detection efficacy. The low reliability is due to: (a) In [17], manipulation had to be reported using a corresponding button on the election webpage, which is contrary to the adversary model that assumes a potentially malicious voting client; and (b) participants used the system twice, once without manipulation and then with manipulation, potentially making it much easier to detect a difference from the previous election, than if the last election is some months or even years ago. Third, the authors of [17] did not specify the details of their manipulations. Most importantly, they failed to describe the changes introduced to the user interfaces; thus it is not clear from the paper, how easy it was for participants to detect the manipulation. Fourth, the data from Kulyk et al. [16] is available on the Internet[5].

In [16], Kulyk et al. analysed the voting materials and the election webpage of the Swiss system through several brainstorming and feedback sessions with lay persons and various experts. Based on the issued raised in these discussions – particularly those that may prevent voters from detecting manipulations – Kulyk et al. proposed a revision of both the voting materials (see Fig. 2 in [16]) and the election webpage (see Fig. 3 in [16]). The voting scheme is as described in Figure 4 in the Appendix. It should be noted that, while their focus was on usability improvements to the Swiss system[6], our proposals focus on security improvements and their potential implications for both general usability and the manipulation-detection efficacy.

Both the original system and the improved system were evaluated on two parameters: (1) the general usability based on the System Usability Scale (SUS) and (2) their manipulation-detection efficacy, i.e., whether voters could detect manipulations of their cast votes. The evaluation was based on a lab study of a total of 128 participants. Their study evaluated the general usability of the original system compared to the improved system ('general usabilitygroups'). In addition, the study tested the manipulation-detection efficacy ('manipulation groups') by manipulating the votes cast during the study. Their research included two types of manipulations, both of which would enable attackers to change the intended vote to a vote preferred by the attacker, if undetected. The difference between the manipulations lay in the changes to the user interface: In the first manipulation type (called '*replace-manipulation*'), the attacker would show the check code which the attacker would obtain from the election infrastructure after having cast their own vote (which is different from the vote cast by the actual voter). This check code differs from the code that voters would expect. Thus, the attacker would need to hope that voters do not check whether the displayed check code is correct, because the interface says 'Continue by scanning the confirmation code'. In the second type of manipulation ('*remove-manipulation*'), no check code would be shown; instead, the voter would see a message confirming that their cast vote had been accepted by the voting system. Afterwards, the vote casting would continue – for both manipulation types – as described in the polling sheet.

Thus, Kulyk et al.'s user study consisted of six groups – three groups using the original system and three using their improved system. Participants were randomly assigned to one of the groups. All groups were told that the study goal was to evaluate the system's usability, and all participants were given the task of casting a vote. All participants carried out the following steps: First, they were given an information packet containing an informed consent form, general information about the study, role card (including which option to select) and actual voting materials (including the election letter and the polling sheet). After the participants had read this information, they

[3] The election infrastructure is a composition of several services conducted by independent parties. The details of their interaction are not relevant for the usability of the cast as intended verifiability functionality.

[4] The authors of [15] also studied an improved version of the Swiss system. However, the improved version is identical with the version studied in [17].

[5] `https://secuso.aifb.kit.edu/downloads/voting_manipulations_2020.xlsx`

[6] This is also the case for [17].

could use the study laptop to proceed with vote casting. The election letter explained to participants that they should contact the (study) support in case of any problems. After these initial orientation steps, the following steps differed depending on the situation. Those participants assigned to the 'general usability-groups' and those from the other groups who did not report the manipulation were asked to fill out a questionnaire. Those in the 'manipulation groups' were debriefed afterwards, i.e. they were informed about the manipulation and that the actual goal of the experiment was to study the manipulation-detection efficacy. Those who reported the manipulation were first debriefed and then asked whether they would be willing to continue the study and to complete the survey questionnaire. The study found that the detection rates for both types of manipulation were significantly higher for their improved system compared to the original system (76% versus 100% and 10% versus 43%). The SUS scores were very similar for both schemes with 79.9 for the original system and 80.9 for the improved scheme.

We refer to this system including the voting materials and user interfaces as **'system from [16]'**.

4 Proposed Voting Systems

In this section, we describe the two extensions which we propose for improving the security of the Swiss voting system. Both extensions have been evaluated (see Sections 5 and 6).

4.1 Security Improvements with Voting Codes (proposal-code-voting-with-QR-codes)

The first proposed improvement is to extend the Swiss voting system using code voting. With this extension, the individual polling sheet from the Swiss system also contains one individual voting code per voting option on the ballot. The voting codes are different for each voter. Thus, voters must enter the voting code corresponding to their chosen option. The different independent parties building the election infrastructure together deduce the actual option from each cast voting code during the tallying. As the voting client cannot map the voting code to any of the options, there is no need to operate with the assumption of a trustworthy voting client[7].

While this proposal increases the security level compared to the original Swiss system and to the improved system studied in [16, 17], the actual usage of voting codes made the voting process more complicated (and potentially less intuitive for voters). Furthermore, in order to achieve an adequate level of security, the codes need to be complex enough to prevent the adversary from guessing valid voting codes. However, the longer the voting codes are the more error prone and less usable it is to enter these codes manually. To address this shortcoming,

[7]Note, however, that one needs to ensure that the mapping of the voting codes to options for each voter remains secret to the adversary. Therefore it is important that the printers are operated offline.

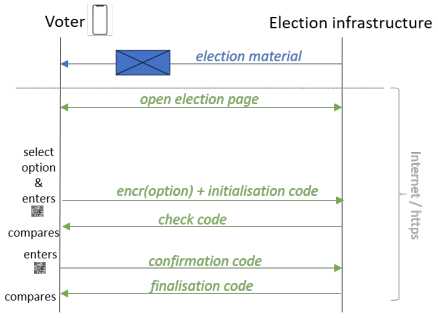


Figure 1: Vote casting with the proposal-code-voting-with-QR-codes

we propose that voters use their camera-equipped computer device, i.e. most likely their smartphones, to cast a vote by scanning a corresponding QR code (containing the complex voting code), as smartphones are now capable of scanning QR codes. While the use of QR codes as voting codes was already proposed by Marky et al. [23], the authors did not consider any means of voter authentication and verifiability.

With this proposal, entering the initialisation code is not needed as a separate step anymore: Given that QR codes can encode a lot more characters than needed for the voting code, they can contain both the initialisation code as well as the corresponding voting code. The QR code can actually contain an even longer initialisation code than the one used in the Swiss voting system as it does not need to be entered manually anymore. Moreover, the initialisation QR code can potentially encode the voters' actual cryptographic private key[8]. Hence, the security level for voter authentication is also increased. We also propose that QR codes are used to provide confirmation codes on the polling sheets. With this improvement, voters avoid having to enter any codes manually. The corresponding (simplified) scheme is depicted in Figure 1.

To integrate these ideas we revised the voting materials from [16] accordingly. The polling sheet is not one sheet anymore. It is a leaflet (see Figure 10 in the Appendix) with one voting card per each voting option (see Figure 5 in the Appendix). As we do not trust the voting client, scanning the QR code for the selected option does not require having the check code(s), nor the actual voting option present, nor the confirmation code, nor another option's QR Code. For similar reasons, the finalisation code must not be visible when scanning the confirmation code. Therefore, voting codes are presented on voting cards, the confirmation code is on a different page. The finalisation code is covered by a scratch field. Furthermore, the voting card should be placed on the inner page so as to ensure that the remaining voting cards are not too close by when scanning. The election webpage was revised as well (see

[8]The Swiss system is design for contexts in which voters do not possess any electronic ID. Instead an election specific key pair is generated. In the original system, the voter receives the initialisation code from which the actual private key is derived. With the QR code, the actual private key can be sent to voters.

Figure 6 in the Appendix). The entire voting system was developed and improved through feedback from participants. We refer to this first proposal, including the voting materials and user interfaces as **'proposal-code-voting-with-QR-codes'**.

4.2 Second Security Improvement (proposal-standard-voting-with-QR-codes)

The security of the original Swiss voting system as well as the system studied in [16, 17] can also benefit from using QR codes and using the camera-equipped smartphone in the vote casting process even in cases where there is no switching to a code-voting scheme. We propose to use QR codes for both the initialisation code and the confirmation code. In the Swiss voting system, both codes need to be entered manually. By using QR codes, more information can be transferred without decreasing the usability. As explained in the previous subsection using QR codes containing the initialisation code would increase the security for voter authentication as the actual private key can be included. To illustrated the changes from the original Swiss voting scheme, we provide a description of the corresponding scheme in Figure 7 in the Appendix. We revised the polling sheet (see Figure 14 in the Appendix) and the election webpage (see Figure 8 in the Appendix), accordingly. We refer to this system, including the voting materials and user interfaces as **'proposal-standard-voting-with-QR-codes'**.

4.3 Considered Manipulation-Types

As mentioned in Section 3.2, Kulyk et. al [16] examined two different types of manipulations in [16], both of which simulated an attack where the adversary attempts to cast a different vote on behalf of the voter. However, such an attack would not be possible in our proposal-code-voting-with-QR-codes, as adversaries would need to know the voting code for the option for which they want to cast a vote for – which is not the case by design of code-voting schemes.

Nonetheless, adversaries can still attempt to nullify votes by blocking the transmission of the voting code to the election infrastructure and manipulating the voting client so that the voter believes that their vote has been cast successfully. This type of attack is less attractive than replacing the vote, as the attacker would need to know how the voter intended to vote, in order to block only those votes considered "undesirable". Otherwise the attacker might accidentally nullify the votes in favour of their preferred candidate. While this might be the case with high degree of certainty if the attacker knew enough about the voter (e.g. geography, age), removing too many votes would trigger suspicion if there was, for example, an unexpectedly low turnout of a certain demographic (e.g. voters living in an area historically known to support a particular political party). This is another, albeit small, advantage of our proposal-code-voting-with-QR-codes.

From a voters point of view such an attack would resemble the *remove-manipulation* investigated in [16] (see Section 3.2): after entering the voting-code, the election webpage would confirm that the check code entered was correct. Figure 11 (a) and (b) in the Appendix shows how the manipulated interfaces could appear given such a manipulation. As such, Step 4 confirms that the vote was cast. Furthermore, it indicates that the check code is correct and that the voter can continue their vote casting process. For proposal-code-voting-with-QR-codes, it is not possible for the adversary to show the finalisation code, as they are unable to send a valid voting code to the election infrastructure. Therefore, in order to avoid alerting the voter, the adversary would need to change these steps as well: Instead of asking voters to compare the displayed finalisation code with the code listed on the polling sheet, the manipulated voting client could ask voters to enter the finalisation code.

For the proposal-standard-voting-with-QR-codes, we assume –similar to the remove-manipulation described in [16] (see Section 3.2) – that the adversary forwards their altered vote to the election infrastructure i.e. the adversary would try to change the vote. Figure 11 in the Appendix (c) shows how the Step 4 interface would appear if such a manipulation were carried out. The remaining steps would be the same. If voters did not notice that no check code had been displayed and as such continued with the process, their altered vote would actually be stored and tallied. Manipulation of voters under the proposal-code-voting-with-QR-codes is potentially less obvious than for the proposal-standard-voting-with-QR-codes, as only one step is changed instead of two. As mentioned above, this type of attack would be more preferable for an attacker, but it cannot be applied under a code-voting-based scheme.

For the voter who experiences the voting process, their comparative perceptions for the proposal-code-voting-with-QR-codes and the proposal-standard-voting-with-QR-codes is shown in Figures 2 and 9 (see Appendix), respectively.

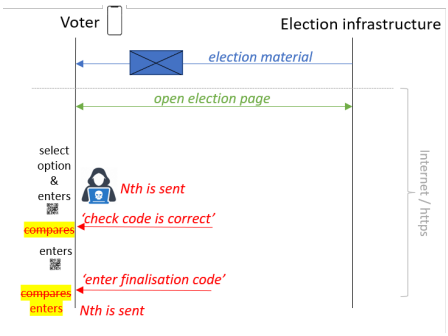


Figure 2: Manipulation for the proposal-code-voting-with-QR-codes.

5 Methodology

We first introduce our research questions and corresponding hypotheses. This is followed by a description of our study procedure. We then discuss ethical issues, how we meet data protection regulations, and how we recruited our participants.

5.1 Research Questions, Hypotheses

Our proposals improve the security level of the original scheme. We aim to answer the question, how these proposals perform with respect to the general usability as well as in terms of the manipulation-detection efficacy. Correspondingly, we define the following research questions:

RQ1 *How does each of our proposals perform in terms of general usability (measured as the System Usability Scale)?*

The study comparing the system from [16] and the original system did not find any significant difference between these two systems with respect to their SUS scores. However, both our new proposals include steps that might be less familiar to the users (i.e. scanning QR codes as compared to manual input of codes) and less comparable with traditional paper-based voting (i.e. using codes to enter a vote instead of choosing among the voting options presented in plain text on the screen). These new steps may have a negative effect on the general usability. We therefore define the following hypotheses:

$H_{1,1}$**:** The proposal-standard-voting-with-QR-codes has a significantly lower general usability than the one from [16].

$H_{1,2}$**:** The proposal-code-voting-with-QR-codes has a significantly lower general usability than the system from [16].

$H_{2,1}$**:** The proposal-standard-voting-with-QR-codes has a significantly lower general usability than the original one.

$H_{2,2}$**:** The proposal-code-voting-with-QR-codes has a significantly lower general usability than the original system.

Note, we decided against conducting statistical tests comparing to other relevant related work. As these studies have not made their data publicly available, the validity of such tests performed using only reported aggregate data (i.e. average SUS value) would be limited. Instead, we compare the descriptive data from related work in Section 7.

RQ2 *How do both proposals perform in terms of manipulation-detection efficacy (measured as the rate of participants detecting and reporting the manipulation)?*

We base our voting materials and election webpage for both of our proposals on the revision from [16], which had a significantly higher manipulation detection rate than the original system. Therefore, we expect that our proposals will also outperform the original system with respect to manipulation-detection efficacy. We define the following hypotheses:

$H_{3,1}$**:** The proposal-standard-voting-with-QR-codes has a significantly higher manipulation-detection efficacy than the original system.

$H_{3,2}$**:** The proposal-code-voting-with-QR-codes has a significantly higher manipulation-detection efficacy than the original system.

5.2 Study Procedure

In this subsection, we describe how we conducted the study[9], i.e. the two experiments consisting of two groups each. The study was conducted in German, as were the voting materials and the election webpage. The text has been translated into English for the purpose of this paper.

First, in order to address RQ1, we conducted an experiment with two groups (i.e. for the proposal-standard-voting-with-QR-codes and the proposal-code-voting-with-QR-codes) to evaluate the general usability of the corresponding schemes. We then conducted the second experiment in order to evaluate the manipulation-detection efficacy – addressing RQ2. The study protocol was very similar for these two experiments. In the following paragraph, we describe the overall study procedure and explain were it differs for the two experiments.

Both experiments were announced as a remote study to evaluate the usability of an online voting system. The ballot of the simulated election contained four options. After having agreed to participate in the study, participants received the study materials in an envelope, either via postal service or from someone whom they knew. They received the study materials some days prior to the start of the experiment. Both experiments ran for two weeks each. The envelope had the following content:

- An information letter describing the study, the time frame, the other materials included, the procedures, and information that they can withdraw their participation at any time. In a footnote of the information letter, the link to the post-survey was included.
- A role card explaining who they were suppose to be in terms of the experiment and which option they were to vote for as part of the user study.
- An inner envelope with the actual voting materials, i.e.: (i) the official election letter from the election officials recommending them to first read the polling sheet before casting the vote. Furthermore, it mentions that in case of problems or questions they should call the (study) support; (ii) the polling sheet (and for the proposal-code-voting-with-QR-codes group, the cards with the voting-code); see in the Appendix Figures 10 and 14 for the polling sheets of both groups and Figure 5 for the voting cards with the voting codes.

[9] Figure 15 in the Appendix provides an overview of the study procedure. For a description of the different groups, see Figure 12 in the Appendix.

Participants were instructed to open the envelope and to read the information letter and the role card. Afterwards, they were supposed to open the inner envelope with the voting materials, then, to read the polling sheet before commencing their actual vote casting. So far, the process was the sane for all four groups. The following steps differ and are therefore explained in separate paragraphs:

The **two groups studied to assess the general usability** could cast their vote according to the polling sheet. After having completed the vote-casting process, the election webpage displayed the link to the post-survey. This survey begins with information about the study and data collection. It contained the informed consent. Then, this survey asked the questions from the System Usability Scale (SUS) questionnaire and collected feedback on the system. The survey also included demographic questions. Finally, we asked participants to refrain from speaking with each other about this study until after they had completed their participation tasks.

The **two groups to study the manipulation-detection efficacy** received the manipulated interfaces as described in Section 4.3. *In the case that participants did not notice the manipulation or had noticed it but did not call the (study) support*, they could just finish casting their vote. After they had completed the voting process, the election page displayed the link to the post-survey. This survey, first, provided information about the study and data collection. It included an informed consent form. Participants then received a debriefing. If they decided to continue with the survey, they were asked whether they detected the manipulation that they had read about in the debriefing text. Afterwards, feedback on the scheme was collected, and the survey included demographic questions. The question regarding detection of manipulation offered the participant three options: (1) I noticed it and I called on the phone the (study) support; (2) I noticed it but I did not call the (study) support; and (3) I did not notice the manipulation. In case the first option was selected in the survey, participants had to confirm this option by entering the number 22. Those who called the (study) support received this number on the phone after they had reported the manipulation they had observed. In case the second option was selected, participants were asked an additional open text question on why they did not call the (study) support. *In the case that study participants did notice the manipulation and called the (study) support*, the support person first asked about details of the problems they observed. The goal was to first ensure they actually observed the manipulation and to determine whether the person calling was in the proposal-code-voting-with-QR-codes group or in the proposal-standard-voting-with-QR-codes group. Afterwards participants were debriefed. If they decided to continue participating in the study, they were provided with instructions on where to find the link to the post-survey[10] and with the number 22. The support person thanked the participant for taking part in the study and took note about the group membership and the time. In case the support person could not answer the call, this participants was called back as soon as possible. All telephone numbers were subsequently deleted.

[10]The post-survey was the same as for those who did not call the (study) support.

5.3 Ethics, Data Protection, Recruitment

The study protocol was approved by the ethics committee of our university. Their requirements also include various legal issues such as compliance with data protection laws. The study materials given to the participants contained a telephone number and an email address allowing them to get in touch with us in case of general questions regarding the study or if they had any other unresolved issues. As we could not guarantee a 24/7 service, the two participants who were unable to not reach us were subsequently called back.

The study was announced as a user study intended to evaluate the usability of an online voting system. Participants in the experiment to test the manipulation-detection efficacy who called the (study) support in order to report the manipulation were debriefed on the phone, and it was explained to them that they could withdraw from the study if they desired and that if they withdrew their data would be deleted. Furthermore, every participant in this experiment who filled out the survey received the debriefing text (see Appendix 16 for the corresponding text) regardless of whether or not they had contacted the (study) support beforehand. The telephone numbers from study participants who called were deleted after the call.

Participants received a role card describing who they are for the study and which option on the ballot they should select. This approach was used to ensure that we did not gather information about participants' actual vote. Furthermore, all participants received the same credentials (per group), i.e. the same voting materials, and the study's election server did not store participants' IP addresses.

Participants were recruited in two different ways: Through public channels announcing the study as well as through a snowball method, asking those who agreed to participate to announce it to their friends and family. In the first case, we usually sent the study materials via postal mail to those who had agreed to participate. In the second case, we usually send the first contact person the study materials to distribute it further.

The study announcement included information about the study, an explanation why we need their postal address if they want to participate, how we treat their postal address in terms of confidentiality, as well as confirmation that by sending us their postal address they agreed that we send them the study materials. In particular, the information related to the participant's postal address was important in fulfilling the data protection requirements. Potential participants were also informed that even after having received the materials that they could withdraw from the study at any time without any negative consequences.

We first prepared the study materials. When all the envelopes had been sealed, we mixed them up so as not to know which person would be assigned to which group.

Experiments	Age	Gender
Data from [16]	34.34/15.54	66F, 62M
Our proposals, no manip.	40.45/16.43	40F, 40M
Our proposals, manip.	42/15.85	29F, 26M

Table 1: Demographic data for participants for age Mean/SD. In their study, Kulyk et al. [16], authors report demographic data for both groups together.

Afterwards, the addresses were added. The addresses were deleted once they were put on the envelopes. For the survey, we used SocSciSurvey which is GDPR compliant[11].

Following discussions with our data protection officer, we decided not to offer financial reimbursement to participants. This was also mentioned in the recruiting brochure. As we did not know our participants personally, and since they came from all over the country, the only way to collect payment details would have been via email or postal mail. The first option would have been questionable in terms of data protection (e.g., many people do not know how to encrypt emails), while the second option would have imposed extra burdens on the participants. We also regarded that the participants' total time and effort in participating in our study to be much less compared to Kulyk et al. [16], where the participants had to come to the lab on a particular day. In our case, participants could participate from home and where flexible, could participate anytime within a two-week period. Participants in our pre-study used an average of 20 minutes, including phoning the (study) support to report the manipulation.

6 Results

While recruiting the participants for our study, we also sent out the voting instructions to 200 participants, 100 of which for the first study (which involved no manipulations) and 100 for the second study (involving manipulations).

Eventually, a total of 135 people completed their respective study, 80 of which in the non-manipulation study (evaluating RQ1) and 55 in the study where their vote has been manipulated (evaluating RQ2). Table 1 shows the demographics of the participants of both of our studies alongside a comparison of the participants in Kulyk et al. [16]. The results for the two research questions are explained in the following two subsections[12] before briefly summarising the feedback we received. All statistical calculations for our hypotheses are performed using *R* packages "stats" and "rstatix".

[11] For their data protection policy see `https://www.soscisurvey.de/en/data-protection`.

[12] Figure 13 in the Appendix shows the overview of the results.

6.1 RQ1 - general usability

The mean values of the SUS score were 84.1 for proposal-standard-voting-with-QR-codes and 82.2 for proposal-code-voting-with-QR-codes, which corresponds to the grade between 'good' and 'excellent' according to Bangor et al. [32]. This is comparable to the scores of the original system and the system from [16] (mean values of 79.1 and 80.9 respectively). Figure 3 furthermore depicts the distribution of the SUS scores (as boxplot) for all the four systems. The Mann-Whitney tests[13] failed to confirm the hypotheses $H_{1,1}$, $H_{1,2}$, $H_{2,1}$, $H_{2,2}$ (p-values of .658, .739, .932 and .975 respectively)[14]. We also calculated the effect size of the differences between the systems studied by Kulyk et al. [16] and our proposals, showing small effect sizes of these differences (see Appendix). Thus, for the general usability, we were not able to detect any difference between our proposals and those evaluated by Kulyk et al. [16]. While we acknowledge that this finding by itself is not a proof that there is no such decrease, we can conclude that it is at least unlikely that such a decrease, if at all present, is significant.

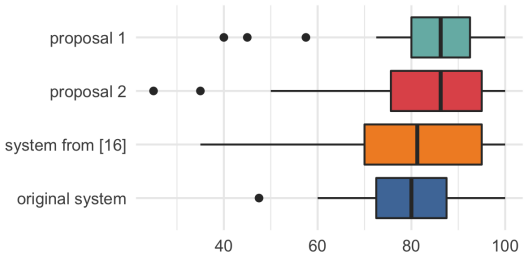


Figure 3: Boxplots of SUS scores. Proposal 1 is proposal-code-voting-with-QR-codes. Proposal 2 is proposal-standard-voting-with-QR-codes.

6.2 RQ2 - manipulation-detection efficacy

In our study, 22 participants called the (study) support to report the manipulation (11 in the proposal-standard-voting-with-QR-codes group and 11 in the proposal-code-voting-with-QR-codes group). This number is deduced from the survey. Table 2 shows the distribution of participants who reported the manipulation. While only 10% of the participants noticed the manipulation using the original system, the detection rate was at the same level for the system from [16], the proposal-standard-voting-with-QR-codes and the proposal-code-voting-with-QR-codes. Fisher's test shows a significant difference between the original system and the proposal-standard-voting-with-QR-codes ($p = .049$[15], $OR = 0.178$, 95% CI [0, 0.807]) as well as between the original system

[13] We chose to use a non-parametric test because the distribution of scores did not resemble a normal distribution.

[14] For a complete statistical overview, see Table 4 in the Appendix.

[15] The p-values are reported after a Bonferroni adjustment.

and the proposal-code-voting-with-QR-codes ($p = .0404$, $OR = 0.168, 95\%$ CI $[0, 0.765]$), confirming both $H_{3,1}$ and $H_{3,2}$.

	not detected	detected
Original System	18 (90%)	2 (10%)
System from [16]	12 (57%)	9 (43%)
Proposal 1	16 (59%)	11 (41%)
Proposal 2	17 (61%)	11 (39%)

Table 2: Manipulation detection rates. Proposal 1 is proposal-code-voting-with-QR-codes. Proposal 2 is proposal-standard-voting-with-QR-codes.

In the survey, 17 participants (8 in the proposal-standard-voting-with-QR-codes group and 9 in the proposal-code-voting-with-QR-codes group) reported that they had detected the manipulation but decided not to contact the (study) support. In an open text question field, they were asked to explain why they had decided not to call the (study) support. Their responses were analysed via open coding that was done by two of the paper authors. The answers were first coded independently and then discussed between the two authors. We identified three different types of answers (i.e. three different codes): Plausible reason, not critical, and mistake. The mapping of quotes to answers is provided in Table 5 in the Appendix[16]. Our findings can be summarised as follows: Five of the participants named a *plausible reason* for not contacting the (study) support, such as not wanting to disturb the support person by calling them at a late hour, or being outside of the country with high fees for international calls. Eight of the participants reported feeling that the error they noticed was *not critical*, e.g. they believed that their vote had been cast successfully despite of a check code because the voting website told them so. Three of the participants believed that the error was due to either their own *mistake* or a *mistake* made by those running the study (e.g. error in the voting materials).

We concluded that the five participants with *plausible reason* did not contact the study examiner because of the study setting but that they would probably have done so in a real election. As such, we assume that voters are more motivated to report discrepancies if the integrity of their real vote depended on it (hence, they would be prepared to exert more effort than in the study setting), and that ideally they would be aware about the availability of a reporting hotline (thus avoiding the situations where the voters are reluctant to call because of the late hour). Hence, if we count these 'plausible reason' participants as having detected the manipulation, the numbers would be higher with 13 out of 27 (48%) for the proposal-code-voting-with-QR-codes and 14 out of 28 (50%) for the proposal-standard-voting-with-QR-codes. As we are aware that our interpretations are subjective and in so far as these higher numbers are based on self-reported data, we only considered those participants who actually called in when conducting the above hypothesis test.

[16] We translated them using forward-backward translation

6.3 Feedback on the proposed schemes

We received a lot of positive feedback, including feedback from participants in the manipulation-experiment. In particular the participants stated that the instructions they received were clear. There were several suggestions for small improvements: i.e. removing the check-list icons on the right side, as this was not necessary to complete vote casting; using a larger piece of paper to have everything on a single page / a larger font size; and rendering the election URL as a QR code. In particular, several participants who did not detect the manipulation recommended that the instructions state more clearly the need to follow each step precisely, some also recommending that more explanation be provided as to why this was important.

7 Discussion

Our study shows that it is possible to improve existing (verifiable) voting systems (in particular the system used in Swiss elections) to provide enhanced security guarantees while we did not detect a decrease with respect to the general usability. The security advantages of the proposal-code-voting-with-QR-codes compared to the original system are three-fold: (1) security improvements on the scheme level (i.e., better guarantees with regards to vote secrecy), (2) fewer incentives for the adversary to attempt vote manipulation by targeting the voting client (as even if the attack were successful and were not detected by the voter, the adversary would have only managed to block votes as opposed to replacing them with a vote for another option – as it would be possible using the original system [17], and (3) significant better manipulation-detection efficacy.

The proposal-code-voting-with-QR-codes also shows a high average SUS score (with 84.1 being considered between 'good' and 'excellent' usability according to [32, 33]), which is inline with the results reported by Kulyk et al. [16, 17] (although their schemes did not use voting codes). Furthermore, these findings are inline with the results of [23]. The authors of [23] evaluated *non-verifiable* code-voting schemes. They report a mean SUS score of 84 when using QR codes as voting codes. The SUS score drops to 61 when entering voting codes manually. Similarly, in [24], the authors report for their code-voting verifiable system – in which voters had to enter voting codes manually – a SUS score of 67. Thus, it appears that code voting decreases the usability when codes need to be manually entered while the usability level is not affected when QR codes are used instead. This is also supported by the fact that our second proposal (proposal-standard-voting-with-QR-codes) in which QR codes were used to enter the initialisation code and the confirmation code obtained a SUS score of 82.2. Another possibility is that the pandemic indirectly enabled such high SUS scores, as there were many instances where scanning QR codes proved convenient. Our results for manipulation-detection efficacy confirms that the proposed improvements described by Kulyk

[17] (1) and (2) also hold for [16, 17].

		Code Voting	Verifiable	SUS (mean)	Efficacy	Voter Authentication	Study Type
Our paper	proposal 1	QR Codes	yes	82	41%	included	between
	proposal 2	no	yes	82	39%	included	between
[24]	approach 1	no	no	88	no	not included	within
	approach 2	no	yes	83	no	not included	within
	approach 3	manual	yes	67	no	not included	within
[23]	approach 1	manual	no	61	no	not included	within
	approach 2	QR Codes	no	84	no	not included	within
	approach 3	tangibles	no	79	no	not included	within
[17]	original scheme	no	yes	81	33%*	included	within
	improvement	no	yes	85	100%*	included	within
[16]	original scheme	no	yes	80	76%/10%**	included	between
	improvement	no	yes	81	100%/43%**	included	between
[15]	code-sheet***	no	yes	85	100%*	not included	between

Table 3: Overview of the properties of our own proposals and approaches from related work. Remarks: * Results are based on a unrealistic setting (see Section 3.2). ** Two different types of manipulations were evaluated, the second one is similar to the one tested in our paper; the other one is easier to detect. *** The authors evaluated five different types of verifiability techniques tested, only their code-sheet has similar properties to the verifiability techniques in place for our proposals.

et al. in [16] actually enabled voters to detect manipulations significantly more often compared to the original system – as we based our proposed systems on the improvements from [16]. For a security and usability comparison between our proposals and the most relevant related work, see Table 3.

While our findings indicate that our proposals perform significantly better than the original system, an attacker controlling the voting client or the vote-casting application still has a high success rate of manipulating the vote without being detected. While the detection rate is comparable to the rate reported for the corresponding manipulation type in [16], manipulation effort for the proposal-code-voting-with-QR-codes is less attractive for adversaries, as they can only delete votes but not replace/alter them. Whether the detection rate is sufficiently high is a question to be decided upon by the election officials on a case-by-case basis. This decision depends – besides other facts like the importance of the election – on how complaints are treated. This kind of risk assessment, in particular deciding whether the risks are acceptable as compared to paper-based voting, is necessary for any kind of technology used in elections [34].

In order to increase the detection rate as well as the usability, as future work, the received feedback should be applied. One improvement would be to ensure that the voting materials have clearer and more salient statements, alerting the voter about the importance of following the process from the polling sheet in detail and to and to stop their voting if they receive a response that is not as described in the polling sheet. Such statements could be supplemented with videos demonstrating the process of vote casting. On top of that, further measures might be needed to both increase awareness of the importance of verifiability and to explain why the polling sheet can be trusted but not necessarily the information displayed on the screen. The development and evaluation of such measures, and their effect on voters' trust in the election system (using, e.g., a questionnaire developed in [35]) is also an important direction of future work.

Study limitations: Our study has limitations similar to other user studies evaluating the general usability of electronic voting systems and the manipulation-detection efficacy in verifiable electronic voting (e.g., the user studies in [1–14, 24]). First, in these studies, participants cast a vote they were asked to cast in a mock election. Compared to actual elections, this vote is not as personally engaging for them. Second, participating in a study and, thus, agreeing to take time for it, may thus lead the participants to spend more time in reading the instructions compared to casting a vote in an actual election. Both of these aspects may have an effect – in particular on the manipulation-detection efficacy. However, introducing vote manipulations in an actual election in order to measure manipulation-detection efficacy would pose critical ethical and legal issues. Hence, some kind of mock election process will remain. Another limitations of all these studies testing manipulation-detection efficacy (including ours) is that we need to trust that those few participants who know each other have not informed others about the manipulation element. This is in particular important for our setting (using the snowball

method for the recruitment and conducting a remote study).

We studied one implementation of adversaries' attempt to make voters believe that their vote was cast as intended while their vote will not be considered in the tally. The details could vary, i.e., the text displayed to convince voters that their vote was submitted correctly, although the steps did not conform to the steps in the polling sheet. As future work, one could study the attack using a different text. Adapting existing formal methods for modelling security-critical processes in human-computer interactions [36, 37] could help towards developing a more systematic approach to identifying different implementations of such attacks.

We compared our results with those from [16]. They conducted a lab study while our study was conducted. The study from [16] has therefore a higher internal validity and a lower external validity than our remote study. We could not control for various factors (incl. whether and how long they spend on reading the material, whether they were alone, and whether they tried several times before calling the (study) support). One may argue that this is actually the same when enabling enabling online voting.

Limitations of the proposed systems: Despite its security advantages, code voting is limited with regards to the type of elections for which it can be used. As such, it is most suitable for approval voting, that is, elections with 1 out of n options. Even then, however, it is yet to be studied whether the system remains usable when the number of available voting options increases. Applying code voting to other voting rules, such as m out of n or ranked voting, is much less feasible. The decision on whether to apply our proposal-code-voting-with-QR-codes or any other system based on code voting, can only be made on the basis of a particular election. A further issue that needs to be addressed is adapting our proposals, as well as the original system, to meet the needs of visually impaired voters, which might be particularly challenging due to the reliance of these systems on paper-based materials that have to be distributed and read by the voters. Finally, aside from the risks addressed by our proposal-code-voting-with-QR-codes (namely, threats towards vote secrecy and vote integrity resulting from compromised voting clients), several other issues need to be addressed in order to make Internet voting feasible in practice. Such threats, including but not limited to voter coercion or vote buying, or general social engineering attacks, are well-known and acknowledged by both the academic community and election practitioners. Identifying ways to mitigate these threats is an important research topic. As such, one important recommendation (see also [38]) is to offer Internet voting as a secondary voting channel, encouraging voters to cast their vote on paper if they experience problems with the Internet voting system or if Internet voting is not easily accessible to them.

Lessons learned for remote studies: Although this is the first time that we have conducted a remote study, we had good experiences, particularly with respect to recruiting participants. However, there are also a few lessons learned that we believe should be shared with the community: (1) Make it more clear to participants when the (study) support is available and that the contact person is affiliated with the research team. (2) Explain what to do, if the (study) support cannot be reached, e.g., providing an alternative channel. (3) Test the prototypes on a variety of devices, operating systems and web browsers. Participation should be restricted to settings in which such tests have been thoroughly conducted. (4) Carefully select the sending out of the materials and the study period. For the second experiment, which was to commence on January 2nd, we sent out the materials just before the Christmas holidays. We believe that predictable delays in the postal system may have caused lower turnout in our second experiment.

8 Conclusion

Verifiable voting schemes are the de-facto standard when considering online voting for political elections. At the same time, the verifiable voting systems in place can provide adequate vote secrecy only where the voting client is trustworthy. While this shortcoming can be addressed with code voting, such approaches are currently not considered, as the community and election officials are concerned about the usability implications. Prior to undertaking our own study, there was little evidence from empirical studies that could demonstrate general usability or a lack thereof. The effect of code voting on the manipulation-detection efficacy was also not known. Our study has shown that code-voting verifiable voting schemes are worth considering, as the cumbersome steps of entering voting codes manually can be replaced by easy-enough steps – i.e., scanning QR codes – without significantly reducing the usability, while enabling systems with higher security guarantees. In the concrete instance of the Swiss verifiable system, our first proposal (the proposal-code-voting-with-QR-codes) has the following advantages compared to the original system: the trust assumption regarding the voting client is not needed anymore, manipulating the election outcome is less attractive as votes can only be removed but not replaced/changed, and the tested manipulation was detected significantly more often. We also used the QR code scanning solution for the second proposal, the proposal-standard-voting-with-QR-codes. While this proposal is less attractive than our first proposal from a security point of view, the fact that this systems did also not significantly reduce the usability underscores the value of QR code scanning as a useful element to be integrated in a vote casting process – in particular if it increases the overall security of the voting system. Thus, our findings should encourage further research on combining QR-code-enabled code voting with verifiable schemes.

Acknowledgements

We would like to thank Reto Koenig and Philipp Locher for their participation in the discussion on technical aspects of

how code voting can fit into the cryptographic protocol of the Swiss system as well as on how to design the polling sheet of the proposal-code-voting-with-QR-codes in a way that the security properties of the scheme are not violated. This research was further supported by funding from the topic Engineering Secure Systems, subtopic 46.23.01 Methods for Engineering Secure Systems, of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

References

[1] M. Bär, C. Henrich, J. Müller-Quade, S. Röhrich, and C. Stüber, "Real world experiences with bingo voting and a comparison of usability," in *EVT/WOTE*, 2008.

[2] J.-L. Weber and U. Hengartner, "Usability Study of the Open Audit Voting System Helios." `https://www.jannaweber.com/wp-content/uploads/2009/09/858Helios.pdf`, 2009. [Online, February 16th 2022].

[3] A.-M. Oostveen and P. Van den Besselaar, "Users' experiences with e-voting: A comparative case study," *Journal of Electronic Governance*, vol. 2, no. 4, 2009.

[4] M. Winckler, R. Bernhaupt, P. Palanque, D. Lundin, K. Leach, P. Ryan, E. Alberdi, and L. Strigini, "Assessing the Usability of Open Verifiable E-Voting Systems: a Trial with the System Prêt à Voter," in *ICE-GOV*, pp. 281–296, 2009.

[5] F. Karayumak, M. M. Olembo, M. Kauer, and M. Volkamer, "Usability Analysis of Helios-An Open Source Verifiable Remote Electronic Voting System," in *EVT/WOTE*, USENIX, 2011.

[6] D. MacNamara, T. Scully, and P. Gibson, "Dualvote addressing usability and verifiability issues in electronic voting systems," 2011. `http://www-public.it-sudparis.eu/~gibson/Research/Publications/E-Copies/MacNamaraSGCOQ11.pdf`, [Online, February 16th 2022].

[7] D. MacNamara, P. Gibson, and K. Oakley, "A preliminary study on a DualVote and Prêt à Voter hybrid system," in *CeDEM*, p. 77, 2012.

[8] K. S. Fuglerud and T. H. Røssvoll, "An evaluation of web-based voting usability and accessibility," *Universal Access in the Information Society*, vol. 11, no. 4, pp. 359–373, 2012.

[9] C. Z. Acemyan, P. Kortum, M. D. Byrne, and D. S. Wallach, "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*, vol. 2, no. 3, pp. 26–56, 2014.

[10] C. Z. Acemyan, P. Kortum, M. D. Byrne, and D. S. Wallach, "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*, vol. 3, no. 2, pp. 1–19, 2015.

[11] K. Gjøsteen and A. S. Lund, "An experiment on the security of the Norwegian electronic voting protocol," *Annals of Telecommunications*, vol. 71, no. 7-8, pp. 299–307, 2016.

[12] C. Z. Acemyan, P. Kortum, M. D. Byrne, and D. S. Wallach, "Summative Usability Assessments of STAR-Vote: A Cryptographically Secure e2e Voting System That Has Been Empirically Proven to Be Easy to Use," *Human Factors*, pp. 1–24, 2018.

[13] K. Marky, O. Kulyk, K. Renaud, and M. Volkamer, "What Did I Really Vote For?," in *ACM CHI*, p. 176, 2018.

[14] V. Distler, M.-L. Zollinger, C. Lallemand, P. Roenne, P. Ryan, and V. Koenig, "Security–visible, yet unseen? how displaying security mechanisms impacts user experience and perceived security," in *ACM CHI*, pp. 605:1–605:13, 2019.

[15] K. Marky, M.-L. Zollinger, P. Roenne, P. Y. Ryan, T. Grube, and K. Kunze, "Investigating usability and user experience of individually verifiable internet voting schemes," *ACM Trans. Comput.-Hum. Interact*, vol. 28, no. 5, 2021.

[16] O. Kulyk, M. Volkamer, M. Müller, and K. Renaud, "Towards improving the efficacy of code-based verification in internet voting," in *VOTING Workshop at Financial Crypto*, Springer, 2020.

[17] K. Marky, V. Zimmermann, M. Funk, J. Daubert, K. Bleck, and M. Mühlhäuser, "Improving the Usability and UX of the Swiss Internet Voting Interface," in *ACM CHI*, 2020.

[18] D. Chaum, "Surevote: technical overview," in *Proceedings of the workshop on trustworthy elections (WOTE'01)*, 2001.

[19] J. Helbach and J. Schwenk, "Secure internet voting with code sheets," in *E-Voting and Identity*, pp. 166–177, Springer, 2007.

[20] R. Joaquim, C. Ribeiro, and P. Ferreira, "Veryvote: A voter verifiable code voting system," in *E-Voting and Identity*, pp. 106–121, Springer, 2009.

[21] P. Y. Ryan and V. Teague, "Pretty good democracy," in *Security Protocols Workshop*, vol. 17, pp. 111–130, Springer, 2009.

[22] J. Budurushi, S. Neumann, M. M. Olembo, and M. Volkamer, "Pretty Understandable Democracy - A Secure and Understandable Internet Voting Scheme," in *ARES*, pp. 198–207, 2013.

[23] K. Marky, M. Schmitz, F. Lange, and M. Mühlhäuser, "Usability of Code Voting Modalities," in *ACM CHI*, 2019.

[24] O. Kulyk, S. Neumann, J. Budurushi, and M. Volkamer, "Nothing comes for free: How much usability can you sacrifice for security?," *IEEE Security & Privacy*, vol. 15, no. 3, pp. 24–29, 2017.

[25] O. Kulyk, J. Ludwig, M. Volkamer, R. E. Koenig, and P. Locher, "Usable verifiable secrecy-preserving e-voting," in *6th Joint International Conference on Electronic Voting*, pp. 337 – 353, University of Tartu Press, 2021.

[26] A. Aviv, P. Černy, S. Clark, E. Cronin, G. Shah, M. Sherr, and M. Blaze, "Security Evaluation of ES&S Voting Machines and Election Management System," in *Proceedings of the Conference on Electronic Voting Technology*, EVT'08, (USA), USENIX Association, 2008.

[27] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman, "Attacking the Washington, D.C. Internet Voting System," in *Financial Cryptography and Data Security*, pp. 114–128, 2012.

[28] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman, "Security Analysis of the Estonian Internet Voting System," in *CCS*, p. 703–715, ACM, 2014.

[29] M.-L. Zollinger, E. Estaji, P. Y. Ryan, and K. Marky, "'Just for the Sake of Transparency': Exploring Voter Mental Models of Verifiability," in *International Joint Conference on Electronic Voting*, pp. 155–170, Springer, 2021.

[30] *Verordnung der Bundeskanzlei über die elektronische Stimmabgabe (VEleS) (July 1st 2018)*. Die Schweizerische Bundeskanzlei , 2018. `https://www.fedlex.admin.ch/eli/cc/2013/859/de`, [Online, February 16th 2022].

[31] "Specification of the Swiss voting system," `https://gitlab.com/swisspost-evoting`, [Online, February 16th 2022].

[32] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," *Journal of Usability Studies*, vol. 4, no. 3, pp. 114–123, 2009.

[33] A. Bangor, P. T. Kortum, and J. T. Miller, "An Empirical Evaluation of the System Usability Scale," *International Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

[34] L. F. Cranor, "In search of the perfect voting technology: No easy answers," in *Secure Electronic Voting*, pp. 17–30, Springer, 2003.

[35] C. Z. Acemyan, P. Kortum, and F. L. Oswald, "The Trust in Voting Systems (TVS) Measure," *International Journal of Technology and Human Interaction (IJTHI)*, vol. 18, no. 1, pp. 1–23, 2022.

[36] L. J. Osterweil, M. Bishop, H. M. Conboy, H. Phan, B. I. Simidchieva, G. S. Avrunin, L. A. Clarke, and S. Peisert, "A Comprehensive Framework for Using Iterative Analysis to Improve Human-Intensive Process Security: An Election Example," 2017.

[37] M. Bishop, M. Doroud, C. Gates, and J. Hunker, "Attribution in the future internet: The second summer of the sisterhood," *The Institute Ecole Supérieure en Informatique Electronique et Automatique, Laval, France 5-6 July 2012 Edited by*, p. 63, 2012.

[38] C. of Europe, "Recommendation cm/rec(2017)5[1] of the committee of ministers to member states on standards for e-voting." `https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=0900001680726f6f#globalcontainer`, last visited 24.05.2022.

Appendix

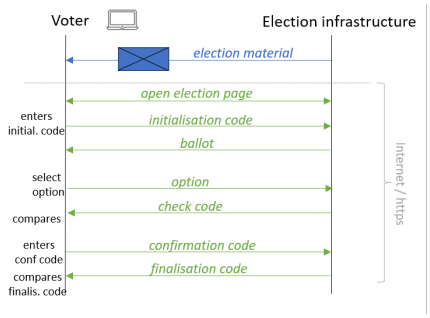


Figure 4: Vote casting with the Swiss voting scheme.

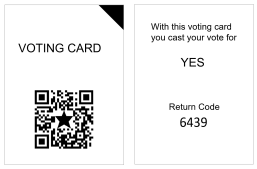
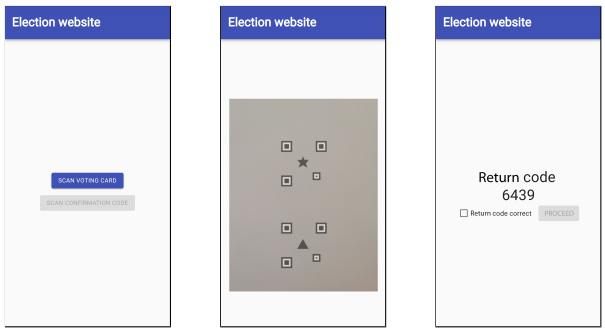


Figure 5: Voting Card (front and back side).



(a) Step 2 (b) Step 3 (c) Step 4

(d) Step 5(a) (e) Step 5(b) (f) Step 6

Figure 6: Voting webpage for proposal-code-voting-with-QR-codes.

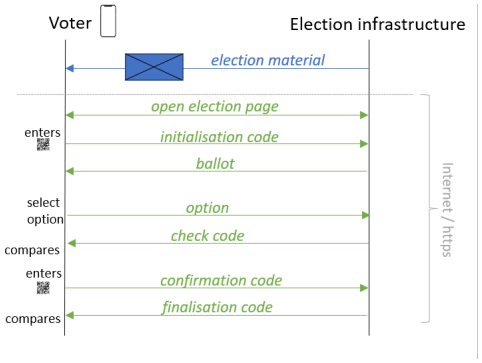
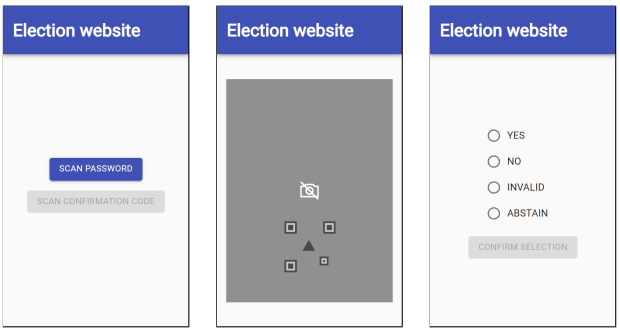


Figure 7: Vote casting with the proposal-standard-voting-with-QR-codes.



(a) Step 2(a) (b) Step 2(b) (c) Step 3

Figure 8: Voting webpage for the proposal-standard-voting-with-QR-codes, only displaying steps that are different from Figure 6.

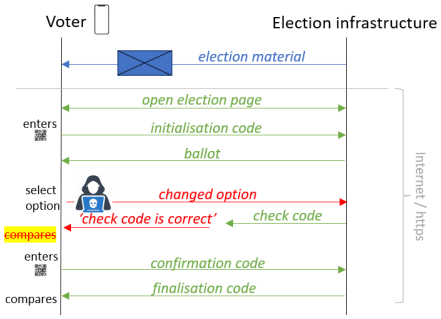
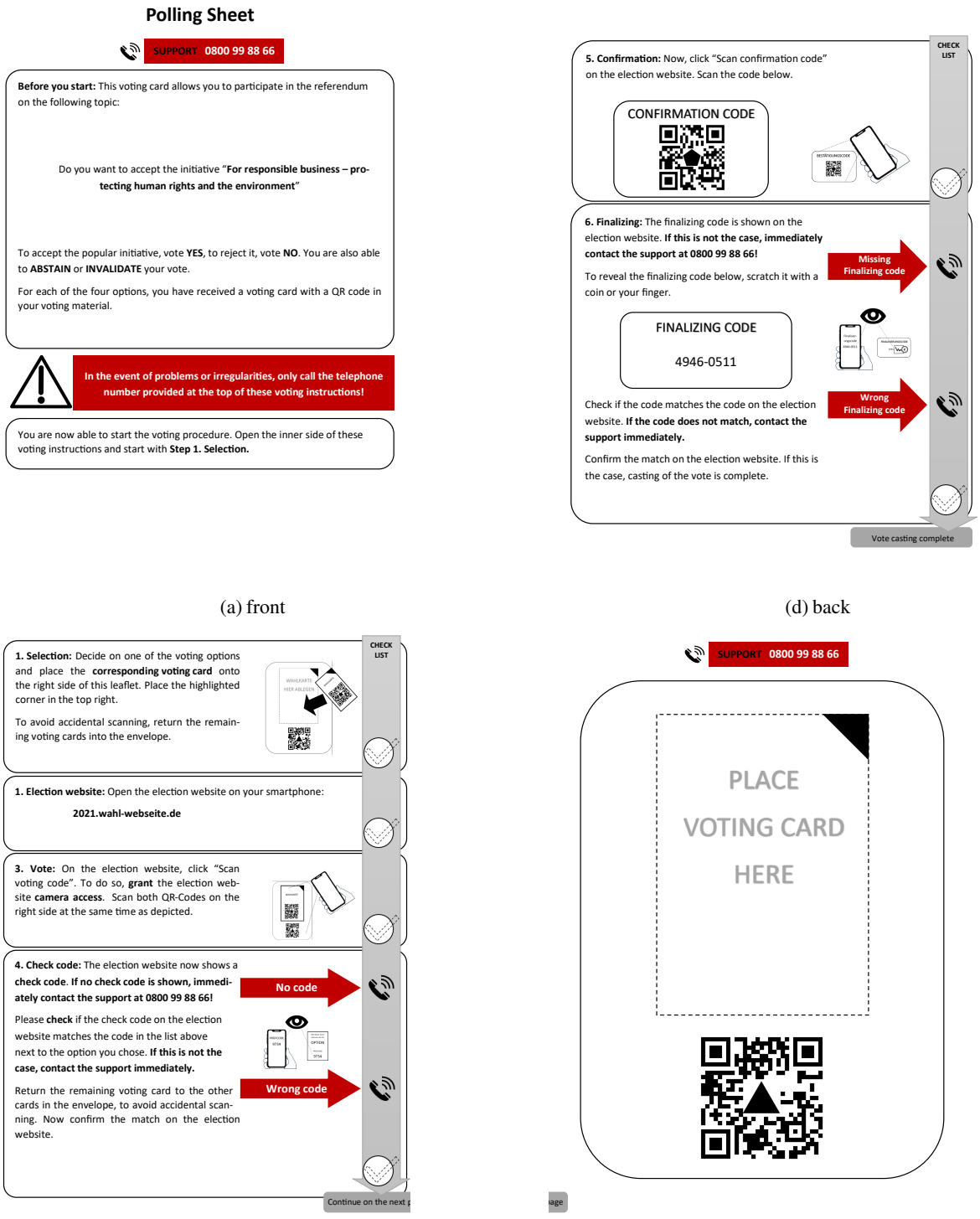


Figure 9: Manipulation for the proposal-standard-voting-with-QR-codes.



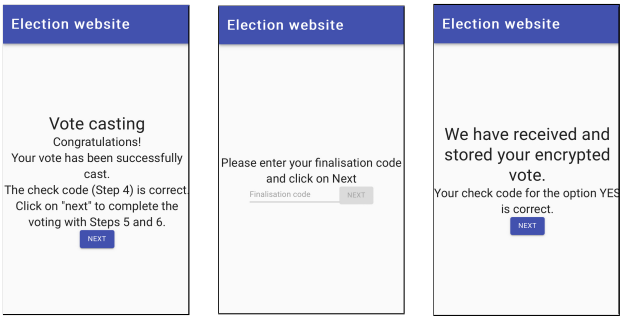
(a) front

(d) back

(b) inner - left

(c) inner - right

Figure 10: Polling sheet for the proposal-code-voting-with-QR-codes system.



(a) proposal-code-voting-with-QR-codes, Step 4

(b) proposal-code-voting-with-QR-codes, Step 6

(c) proposal-standard-voting-with-QR-codes, Step 4

Figure 11: Manipulation of the website for both proposal-standard-voting-with-QR-codes and proposal-code-voting-with-QR-codes.

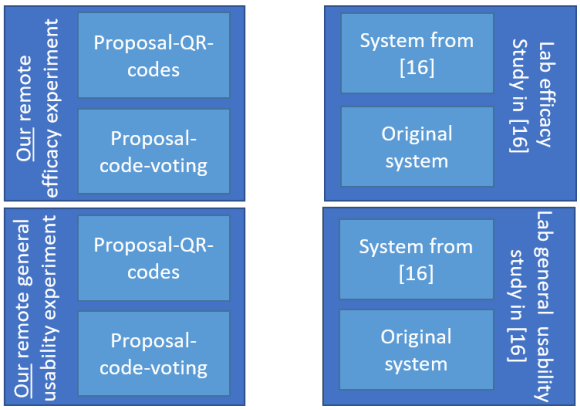


Figure 12: Overview of the considered groups.

Hypothesis	Estimate	Statistic	p	Effect size
$H_{1,1}$	-0.00	394	0.658	0.0508
$H_{1,2}$	-2.50	342	0.739	0.0830
$H_{2,1}$	-5.00	375	0.932	0.184
$H_{2,2}$	-5.00	306	0.975	0.25

Table 4: Comparison of general usability (evaluating RQ1) - p-values without adjustments for multiple comparisons.

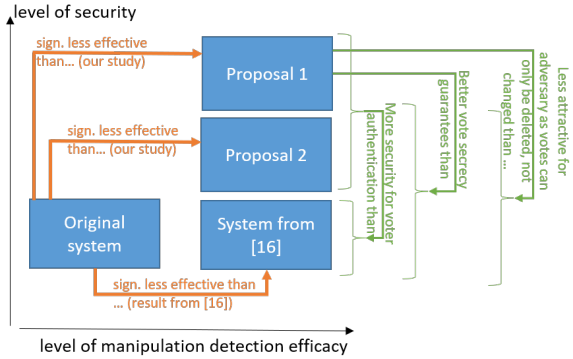
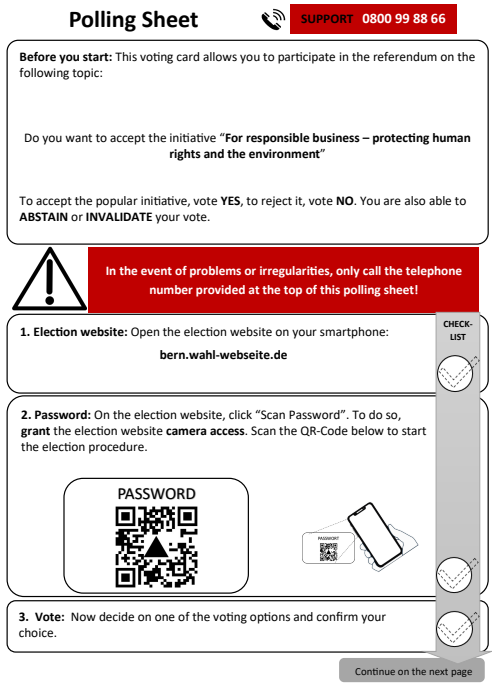
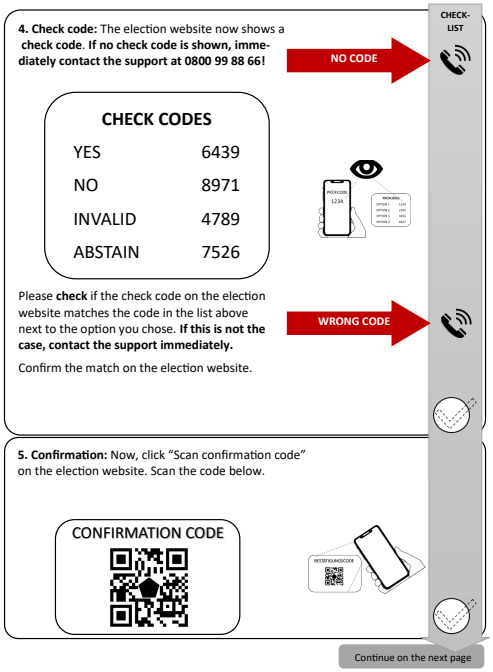


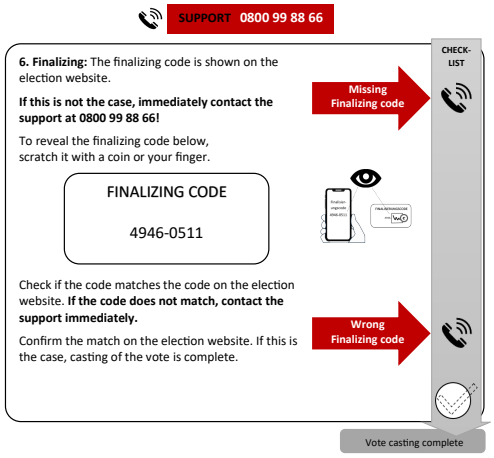
Figure 13: Overall result (the text on the arrows should be read in the following way: system A [–] <text on arrow > [–>] system B means system A is <text on arrow> system B. e.g. the original system is sign. less effective than the proposal-code-voting-with-QR-codes). Proposal 1 is proposal-code-voting-with-QR-codes. Proposal 2 is proposal-standard-voting-with-QR-codes.



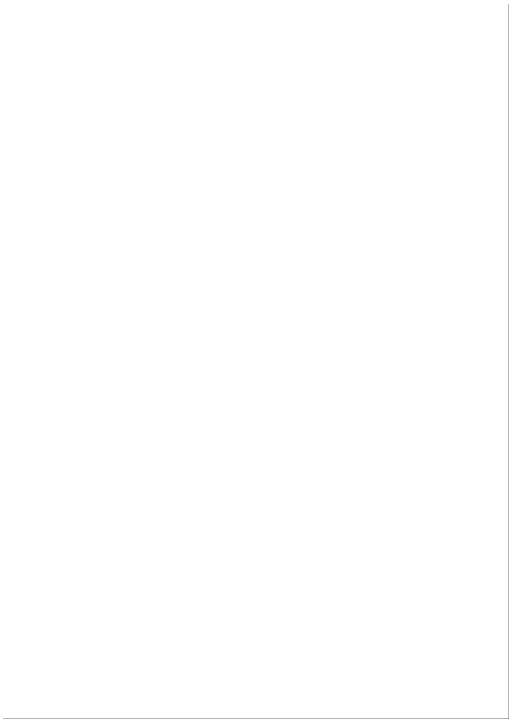
(a) front



(b) inner - left



(c) inner - right



(d) back (blank)

Figure 14: Polling sheet for the proposal-standard-voting-with-QR-codes.

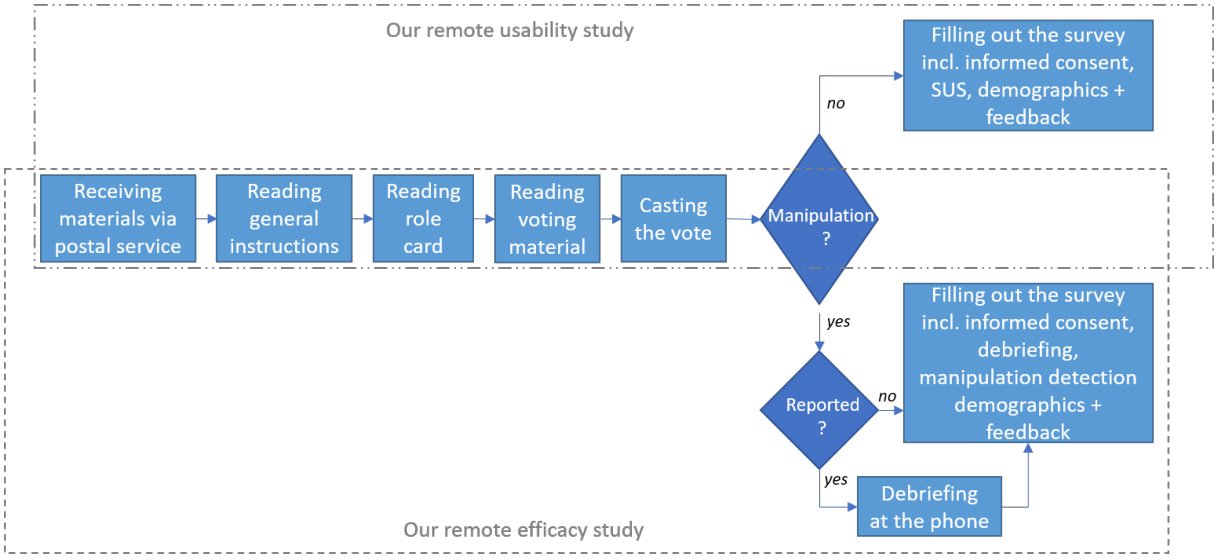


Figure 15: Study procedure for both the (general) usability and the efficacy study.

2. Debriefing

At this point, we would like to inform you that the vote casting contained a deviation. It was included intentionally by us. We simulated an attempt at vote manipulation.

An important aspect of the usability of online voting systems is to allow voters to reliably check if the voting is verifiably correct. In other words: the systems shall allow voters to reliably recognize manipulations. Not recognizing the deviation indicates poor usability of the evaluated online voting system.

Please consider that the study would not be valid if we had informed you about the discrepancy at the beginning of the study. This would have likely influenced you to further look for discrepancies than would be the case for a usual vote. Please also consider that the objective of this study was not to test you, but to test the system. Not recognizing the discrepancy is a symptom of insufficient usability of the online voting system, not inability the participants.

We hope you understand this approach. We would however understand if you decide to withdraw your participation after you received this information. If you continue with the study you will help to improve the usability of online voting systems. If you decide withdraw your participation, your information on this platform will not be evaluated.

If you would like to receive a personal debriefing, please contact the principal investigator ([redacted for review]).

○ I read the debriefing and was sufficiently debriefed.

Next

Figure 16: Debriefing of participants at the beginning of the questionnaire

Code	Recognised, but did not call, because:
not critical	I did recognise the manipulation as such. The website informed me that the check code was correct (without it being shown). I was satisfied with that.
not critical	I assumed the voting was correct even without another check code.
not critical	It was suggested that everything was in order. I would have wished that the mandatory adherence to the provided steps was indicated even more clearly. Such technical measures should be arranged.
not critical	I did not take it seriously enough.
not critical	I did not know that it was a manipulation. In the respective step, I received feedback from the platform that the last step was successful. Hence I did not call.
not critical	Too much effort.
not critical	Because it is just before 23 o'clock and I did not want to wake the study examiner.
not critical	I assumed there was a reason for it.
other	Unsafe.
mistake by user or examiner	I recognised that step 4 could not be performed as described and initially tried to correct a user error on my end by redoing the previous steps. By pressing the next-option everything proceeded until the end and, because I was attested a successful voting, I did not further question this – I'm just a sheep in such matters [smirking Emoji]. I wish you to gain lots of knowledge with this study.
mistake by user or examiner	I find the effort to call someone too much. Especially in the case of a fictional study. Moreover, I did find it very peculiar that both the role card and cover letter were printed double-sided with different salutations on each side. I initially did not see the second page and the cover letter and role card did not match, so I assumed that the study was flawed.
mistake by user or examiner	I was late with the test and assumed a flaw in the creation of the material.
plausible reason	I am currently abroad and a call would have been costly. I had planned to contact the support via E-Mail after completion.
plausible reason	Answering machine.
plausible reason	It was late at night, I did not want to call anyone at that time.
plausible reason	Recklessness. Time (nearly 23 o'clock). Assumption, that it was right anyhow.
plausible reason	I did call, but no one answered.

Table 5: Stated reasons participants recognised the manipulation but did not call the support