Design and Evaluation of Verifiable Voting Systems Based on Tracking Code Verification

Christina Nissen[1], Oksana Kulyk[1], Melanie Volkamer[2], Lara Elisabeth Fredrich[3] und Helena Hermansen[3]

Abstract: To mitigate security risks of Internet voting, techniques for verifiability have been developed, allowing the voters to verify that their cast vote has not been manipulated. One such technique is the use of tracking codes, which does not rely on complex cryptographic mechanisms, and therefore is often assumed to be more intuitive for the voters. However, no systematic evaluation of the usability, verification efficacy, and perceived trustworthiness of these systems has yet been conducted. Our contribution evaluates two variants of a tracking code-based system in a user study ($N = 306$), testing both of the variants in the absence of vote manipulations as well as using different simulated tactics of vote manipulation. We conclude that both of our proposed variants are perceived as easy to use, transparent, and trustworthy in the absence of vote manipulation. However, we found varying rates of verification efficacy based on the vote manipulation tactic, manipulation with detection rates ranging from 0% to 76%. We conclude that using tracking codes can be a viable approach. At the same time, more awareness of potential manipulations needs to be raised to achieve the required security level, irrespective of the verification technique in place.

Keywords: tracking code-based verification, usability, manipulation detection, end-to-end verifiability, user study

1 Introduction

The concept of end-to-end verifiability has been introduced to reduce risks of large-scale vote manipulations in internet voting and has been accepted as the gold standard in designing secure voting systems. As such, a number of techniques for *cast-as-intended*, *recorded-as-cast* and *tallied-as-stored* verifiability has been proposed, with *cast-as-intended* techniques presenting a particular challenge from the usability point of view. As such, since many of them rely on complex cryptographic techniques and non-intuitive procedures, this can lead to voters being unable to properly apply the verification steps [KV18].

One of these cast-as-intended techniques relies on the use of so called *tracking codes* [Kü16a; RRI15; RRR21]. During vote casting, a tracking code is either automatically assigned to voters by the voting system or (fully or partially) self-chosen by the voter is assigned. The tracking code is then stored next to the voter's cast vote. At the end of the election, the cast (plain text) votes are published together with their associated tracking codes, so that the voter can verify that the vote next to their code is indeed their intended choice.

[1] IT University of Copenhagen, Denmark, chfn@itu.dk; okku@itu.dk

[2] Karlsruhe Instutute for Technology, Germany, melanie.volkamer@kit.edu

[3] No affiliation, elisabeth@elisabethfredrich.eu; helena.hermansen@outlook.com

Due to the simplicity of the verification procedure, verification using tracking codes has been previously suggested as a promising way to implement voter verification in a usable way [Kü16b; Ma22]. It has furthermore been used in real-world elections, in particular, in the internal party elections of the Christian Democratic Union (CDU) party in Germany in 2020 to decide whom the party should nominate as chancellor for the next general parliamentary elections [Be21]. However, no systematic evaluation of these systems and effectiveness of their verification techniques has been conducted yet. Our contribution in this work is therefore the answer to the following research questions:

RQ1 How effectively can voters detect various types of simulated manipulations in tracking code-based systems?

RQ2 How usable and trusted are the tracking code-based systems perceived?

Answering these research questions, we conduct an online study ($N = 306$). In particular, to study **RQ1**, we simulate three different types of vote manipulations, relying on the approach from previous research [Vo22a]. Our study reveals that the detection rate of manipulated votes differs depending on the manipulation tactic chosen by the adversary, with 76% of voters being able to detect manipulation in the simplest tactic, yet none of the voters detecting the manipulation with the most complex one. For studying **RQ2**, we test both of the systems without presence of manipulations. We use the System Usability Scale [BKM09] to evaluate the usability of the system and the Trust in Voting Systems (TVS) [AKO22] questionnaire to evaluate the perceived trust in both systems. We find that both systems are perceived in a similar way: both achieved high scores in usability and a moderate-high level of perceived trust.

Overall, our work concludes that the tracking code-based systems are indeed promising in terms of being easy to use, perceived as transparent and trusted by the voters. However, we also show that an attacker controlling the voting client can use deceptive voter interface modifications to prevent voters from detecting vote manipulations, which is contrary to the goal of achieving verifiability via tracking codes. We therefore conclude that while using tracking codes for verification can indeed be a promising direction, more research is required into protection against the manipulation tactics we investigated.

2 Related Work

The usability evaluation of end-to-end verifiable voting systems has been a subject of several studies. In particular, the ability of voters to detect various types of manipulations during vote casting is investigated, e.g. in [Ac14; Ka11; Ku20; Ma18; Vo22a]. There are those manipulations in which only the chosen candidate is altered before sending the encrypted vote to the server (simple manipulation types) and those in which the user interface of the voting client is altered in a way that any information related to verification is removed (deceptive manipulation). In particular, the deceptive manipulation show high efficacy of such attacks; i.e. only a minority of voters being able to detect such manipulations, see, e.g. [Ku20; Vo22a]. None of these studies, simulating the deceptive manipulation, however, studied verification using tracking codes, but studied other types of verification techniques.

Systems relying on tracking code verification have, however, been a subject of empirical studies not involving deceptive manipulation of user interfaces – while studying manipulation detection efficacy and/or how the system was perceived, e.g. from usability point of view: Marky et al. [Ma21] have studied the ability of voters to detect simulated simple manipulations, with 84% of participants being able to detect that their vote was altered (while there was no change in the UI). The study furthermore demonstrated high user satisfaction as measured by System Usability Scale. In addition, in their study, only 44% of voters expressed trust in the system by answering that they are confident that they can verify the integrity of their votes. Other user studies focused on Selene [RRI15], a system that employs a variant of tracking code-based verification that distributes the tracking codes after the tally [AS20; Di19; Zo21]. Similar to the study by Marky et al [Ma21], the studies have demonstrated high usability of the evaluated system, yet low levels of perceived trustworthiness and lack of understanding of the verification process.

Overall, the related work on tracking codes is limited compared to the amount of research on the other type of verification approaches and the various ways to implement tracking codes (see Abschnitt 3). In particular, verification efficacy of such systems given an attacker capable of modifying user interfaces of the voting client has not been studied yet. However, those that exist show that such an approach can be promising in terms of ease of use. Therefore, our research contributes in studying two types of tracking codes regarding perceived usability and manipulation detection efficacy for three different attacks. In order to address the issues with perceived trustworthiness and understandability issues, identified by related work, we developed our proposals in an iterative way.

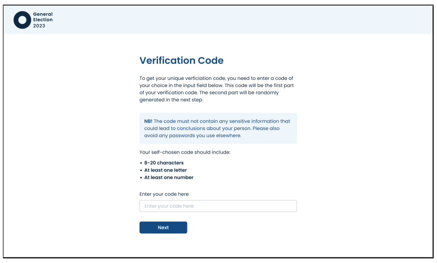
3 Proposed Voting Systems

In this section we describe the prototypes for two tracking code-based systems that we used in our evaluations[4]. As such, we chose to compare two methods of assigning a tracking code to the voter: *partly self-chosen*, where the voter chooses half of the code themselves and the other half is generated by the system similarly to [Kü16b] (*System-Self*, see Figure 1) and *auto-generated* where the entire code is generated by the system (*System-Auto*, see Figure 2) [5]. We furthermore decided to assign tracking codes to the voters before voting in order to mitigate clash attacks [KTV12], because adversaries cannot be sure which option a voter is going to choose, unlike if tracking codes were assigned after vote casting. After being assigned the tracking code and voting, the voter in both systems gets an option to download their tracking code as a PDF file, also containing the tracking code encoded in a

[4] For a more detailed description of the design process of both systems, see the extended version of our paper here [Ni]

[5] We decided against implementing fully self-generated codes due to usability issues and concerns regarding voters' ability to generate unique codes [Bo12]

QR code to facilitate later verification with a mobile device for ensuring better usability and device independence. After the voting phase is finished, the voters are given an option of either visiting the bulletin board website listing all the votes with tracking codes, or scanning the QR code that forwards them to a website showing only the vote assigned to their code [6]. The screenshots showing all the steps in both *System-Self* and *System-Auto* are provided in the extended version of our paper [Ni].



Verification Code

To get your unique verficiation code, you need to enter a code of your choice in the input field below. This code will be the first part of your verification code. The second part will be randomly generated in the next step.

NB! The code must not contain any sensitive information that could lead to conclusions about your person. Please also avoid any passwords you use elsewhere.

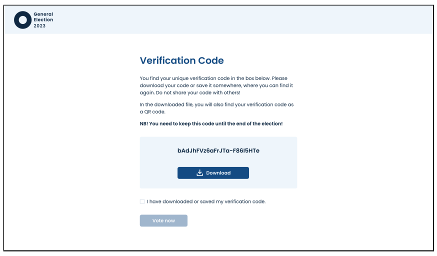
Your self-chosen code should include:

- **8-20 characters**
- **At least one letter**
- **At least one number**

Enter your code here

Enter your code here

Abb. 1: System-Self



Verification Code

You find your unique verification code in the box below. Please download your code or save it somewhere, where you can find it again. Do not share your code with others!

In the downloaded file, you will also find your verification code as a QR code.

NB! You need to keep this code until the end of the election!

bAdJhFVz6aFrJTa-F86I5HTe

□ I have downloaded or saved my verification code.

Abb. 2: System-Auto

After designing initial prototypes of *System-Self* and *System-Auto*, we conducted two rounds of lab usability testing ($N = 4$ and $N = 6$ respectively) and one round of remote moderated testing using video and chat ($N = 4$). Following the feedback from the testing, we made several enhancements to functionality and usability, such as relaxing the requirements for partly self-chosen tracking codes.

4 Methodology for the evaluation of the actual system

We describe the manipulation tactics we simulated to study **RQ1**, the measurements we used for answering both **RQ1** and **RQ2** and the study conducted to answer these research questions.

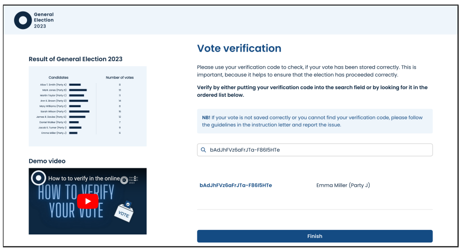
4.1 Manipulation tactics

We consider an attacker whose goal is to either change the vote cast by the voter into a vote for another candidate of attacker's choosing or to nullify the vote, making sure that it is not included in the tally. We assume that the attacker is capable of at least partial control over the voting client, being able to modify the cast votes as they are being sent to the voting system, as well as make changes to the user interface of the voting client. In order to prevent the voter from detecting a manipulation, such an attacker can therefore be assumed to attempt various tactics to prevent the voter from verifying their vote. Following previous research [Ku20; Vo22b], we study the following tactics.

[6] Note that in the latter case, the device scanning the QR code needs to be trusted for vote secrecy; on the other hand, the voter can potentially let it scan a QR code with a tracking code of another voter in case of concerns

Replace-Vote In the simplest tactic, the attacker does not attempt to do anything beyond manipulating the cast vote, so that the voter still gets their tracking code assigned, and the tracking code is published next to the vote sent by attacker. Once the voter attempts to verify their vote, they would therefore find their tracking code next to “Emma Miller”, which is different to their intended vote for Sarah Wilson (see Abb. 3). Detecting the manipulation would then require noticing this mismatch.

Remove-Vote In the next tactic, the attacker removes the voter’s vote from the bulletin board. Note, this tactic can be applicable in an attack where the adversary replaces the vote and alters the voting client UI so that the manipulated vote is not displayed, or an attack where the adversary removes or blocks the transmission of the vote entirely. Detecting this attack would require voter to understand that the absence of their code on a bulletin board is a sign of a problem with verification (see Abb. 4).



General Election 2023

Result of General Election 2023

Candidates	Number of votes
Alice T. Smith (Party A)	8
Mark Jones (Party B)	13
Martin Taylor (Party C)	11
Ann K. Brown (Party D)	14
Mary Williams (Party E)	8
Sarah Wilson (Party F)	15
James R. Davies (Party G)	12
Daniel Walker (Party H)	7
Jacob S. Turner (Party I)	9
Emma Miller (Party J)	6

Demo video

Vote verification

Please use your verification code to check, if your vote has been stored correctly. This is important, because it helps to ensure that the election has proceeded correctly.

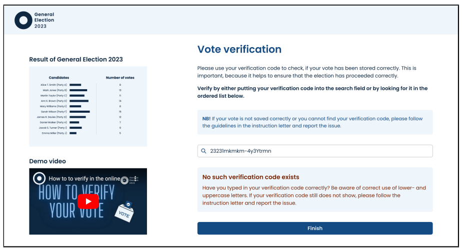
Verify by either putting your verification code into the search field or by looking for it in the ordered list below.

NB! If your vote is not saved correctly or you cannot find your verification code, please follow the guidelines in the instruction letter and report the issue.

bAdJhFVz6aFrJTa-F86I5HTe

bAdJhFVz6aFrJTa-F86I5HTe Emma Miller (Party J)

Abb. 3: Replace-Vote



General Election 2023

Result of General Election 2023

Candidates	Number of votes
Alice T. Smith (Party A)	8
Mark Jones (Party B)	13
Martin Taylor (Party C)	11
Ann K. Brown (Party D)	14
Mary Williams (Party E)	8
Sarah Wilson (Party F)	15
James R. Davies (Party G)	12
Daniel Walker (Party H)	7
Jacob S. Turner (Party I)	9
Emma Miller (Party J)	5

Demo video

Vote verification

Please use your verification code to check, if your vote has been stored correctly. This is important, because it helps to ensure that the election has proceeded correctly.

Verify by either putting your verification code into the search field or by looking for it in the ordered list below.

NB! If your vote is not saved correctly or you cannot find your verification code, please follow the guidelines in the instruction letter and report the issue.

23231mkmkm-4y3Ytrmn

No such verification code exists

Have you typed in your verification code correctly? Be aware of correct use of lower- and uppercase letters. If your verification code still does not show, please follow the instruction letter and report the issue.

Abb. 4: Remove-Vote

Remove-Process The third tactic involves a comprehensive manipulation of the UI with attackers removing everything related to the verification process. Thus, the voter does not receive any tracking code at all and is directed to vote casting straight after being welcomed by the voting system. The texts and graphics on the voting website are furthermore altered to remove any mentioning of the verification. Detecting such a manipulation requires the voter to be aware of the verification possibility – e.g. from thorough reading of the information materials related to the election – and therefore noticing that necessary steps are missing in the voting system.

The manipulation tactics require different levels of involvement on behalf of the attacker – as such, while the *Replace-Vote* tactic does not require any additional action from the attacker aside from manipulating the cast vote, the *Remove-Process* tactic requires a much more thorough modification of the user interface. Hence, depending on the level of access the attacker has to the voting client, some of the tactics might be much more challenging for them to perform. On the other hand, applying the tactics *Replace-Vote* and *Remove-Vote* has an advantage for the attacker in that the manipulation would only be detected after the election (i.e. when the voter either sees the wrong vote next to their code or cannot find their code at all in the list of all published votes), at which point it can become much more damaging to the election integrity. The *Remove-Process* manipulation, on the other hand, can be detected already during vote casting, which can lead to a more timely mitigation on behalf of election authorities.

4.2 Measurements

In order to answer our research questions, we collect the following measurements.

Manipulation detection (RQ1) In order to answer RQ1, we evaluate the manipulation detection rates as the share of participants who reported detecting manipulation when interacting with both systems with all three manipulations tactics.

Usability and trust (RQ2) We measure both usability and trust in the evaluated systems, given the scenario where the voters were not subjected to any manipulations. For measuring usability of the system, we rely on the System Usability Scale. For measuring trust, we use the Trust in Voting Systems questionnaire [AKO22]. In order to get further insights on trust perceptions of both systems, we furthermore use the TDIV scale [Ag23] to measure perceived transparency of the systems.

4.3 Study Procedure

We conducted a between-subject online experiment, resulting in overall seven groups of participants. Of these groups, five were subjected to one out of three manipulation tactics with one out of two systems, allowing us to investigate RQ1. Note, since the *Remove-Process* manipulation removes any references to vote verification from the system, the resulting user interface is indistinguishable between *System-Auto* and *System-Self*; hence, we decided to form one group of participants interacting with this manipulation tactic. For each of the remaining two manipulation tactics (that is, *Remove-Vote* and *Replace-Vote*), two groups of participants were formed, interacting with *System-Auto* and *System-Self* for the corresponding manipulation tactic. Two more participant groups interacted with either *System-Auto* or *System-Self* system without vote manipulation, allowing us to investigate RQ2.

The data collection and recruitment in two phases. In the first recruitment phase (conducted in April 2023), the participants were randomly assigned into one out of four groups, that is, being assigned either *System-Auto* with *Replace-Vote* manipulation, *System-Auto* with *Remove-Vote* manipulation, *System-Self* with *Replace-Vote* manipulation and *System-Self* with *Remove-Vote manipulation*. In the second recruitment phase (conducted in December 2023), participants were randomly assigned to one out of three groups, that is, being assigned either the *Remove-Process* manipulation, *System-Auto* without manipulation or *System-Self* without manipulation. For an overview of the study procedure, see Abb. 5.

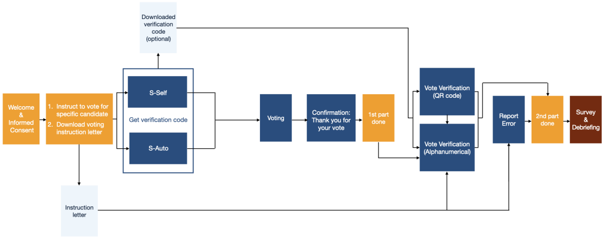

Downloaded verification code (optional)
Welcome & Informed Consent
1. Instruct to vote for specific candidate
2. Download voting instruction letter
S-Self
Get verification code
S-Auto
Voting
Confirmation: Thank you for your vote
1st part done
Vote Verification (QR code)
Vote Verification (Alphanumerical)
Report Error
2nd part done
Survey & Debriefing
Instruction letter

Abb. 5: Study procedure

The study consisted of two parts. In the first part, the participants were instructed to interact with their assigned voting system and cast a vote for a fictional candidate [7]. After completing the vote casting, the participants in groups with verification process intact were forwarded to the mock results page, where they were able to view the final tally of the election as well as verify that their own vote has been recorded correctly. For the participants in a group where the verification process was removed (that is, simulating the *Remove-Process* manipulation tactic), this step was omitted. For the second part, after the voting (for participants in groups subjected to *Remove-Process* manipulation) or after the verification (for participants in all of the other groups), the participants were forwarded to the study questionnaire. We explain the individual study steps in more details below.

Welcome and informed consent: Upon landing on the initial page redirected from the recruiting system, participants were prompted to provide informed consent. They were briefed on the study's purpose, focusing on exploring their user experience and perceived trustworthiness of an online voting system designed for general elections. Following this, participants were randomly directed to one of the four prototypes in the first run and one of the three prototypes in the second run.

Voting: Before entering the actual voting process of the prototype, participants were guided to an information page, urging them to vote for the candidate "Sarah Wilson" and confirm their commitment. Simultaneously, participants were instructed to download an election instruction letter, mimicking how they would receive it by letter or email in a real-world election [8]. The letter, among other instructions, furthermore included a link to a

[7] In order to make it clear for the participants to distinguish between study instructions and parts of the mock voting system, we employed two distinct color schemes for these two types of pages.

[8] The full text of the letters is provided in the extended version of our paper [Ni]

website the participants could use to report any problems they experience with the system. After downloading the letter, they were forwarded to their corresponding voting system where they cast their vote.

After participants completed the voting, the participants who interacted with prototypes that left the verification process intact (that is, participants in groups not subjected to manipulation as well as participants in groups subjected to manipulation tactics *Replace-Vote* and *Remove-Vote*) were directed to another information page. This page explained that, in a real election, the verification phase would only begin after the voting and tallying phases are complete. However, as a part of the study, they could verify their vote immediately after voting. Once the participants verified their vote, they were guided to the study questionnaire. to As the participants in groups subjected to *Remove-Process* were not subjected to any verification process within the voting system, they were forwarded directly to the questionnaire after they completed the voting.

Questionnaire: In the final questionnaire, the participants were asked questions about their experience with the system, including SUS questionnaire for evaluating usability, the "overall trust" part of the Trust in Voting Systems [AKO22] questionnaire for evaluating perceived trust and the Transparency Dimensions in Internet Voting questionnaire [Ag23] for evaluating perceived transparency. The participants in groups subjected to manipulations were asked whether they have experienced any problems with the voting system and asked to elaborate on the problems they experienced via an open-ended question. They were afterwards debriefed about the manipulation and asked whether they have detected the manipulation and whether they reported it; if they answered that they detected the manipulation but did not report it, they were furthermore asked about their reasons for not reporting via a multiple-choice question of (pre-selected reasons based on previous research) and an open-ended "Other" option. The questionnaire furthermore included an attention check, as an item "This question is not part of the survey and just helps us to detect bots and automated scripts. To confirm that you are a human, please choose 'Disagree' here".

4.4 Recruitment and ethics

We used Prolific platform for recruitment, using the gender-balanced option. While Prolific samples are known to be biased to a younger, educated and digitally savvy population, previous research shows that the platform can provide sufficient validity for studies related to security and privacy [TBL22]. We paid participants 4.5 GBP (April 2023) for the data collection as conducted in April for an estimated 30 minutes of participation, which is above the hourly rate recommended by Prolific. As the actual median duration of the study ended up being lower than expected, the reimbursement for the study in December has been lowered to 3 GBP for an expected duration of 20 minutes, keeping the hourly rate the same. Moreover, we had a filter that only recruited participants who were fluent in English.

While there is no mandatory ethical review process at our institutions, measures were taken to ensure participants' informed consent and confidentiality. Before accessing the voting system, participants signed a consent form providing details on the study's purpose, withdrawal options, and data handling. Researchers' contact details were also disclosed for inquiries. Given the sensitivity of political beliefs, all parties and candidates were fictional, a fact communicated at the study's outset. As the study involved deception with regards to vote manipulation, the participants subjected to manipulation were debriefed in the final study questionnaire about the real purpose of the study and the reason for deception. No personal identifying information about participants has been collected within the study.

5 Results of the user study

After excluding participants who either voted for a candidate other than Sarah Wilson (as they were instructed) or failed an attention check within the survey, a total of 306 participants were included in the data analysis. 149 of the participants (49%) identified as female, 155 as male, one as non-binary and one preferred not to answer. Most of the participants (195, 63%) were between 18 and 30 years old, and most (209, 69%) had at least a Bachelor's degree [9]. Table Tab. 1 shows the distribution of participants depending on the system and manipulation type they interacted with, including groups that interacted with a system that did not include a vote manipulation.

System	Manipulation	N
System-Auto	*Remove-Vote*	25
System-Auto	*Replace-Vote*	25
NA	*Remove-Process*	25
System-Self	*Remove-Vote*	25
System-Self	*Replace-Vote*	25
System-Self	None	92
System-Auto	None	89

Tab. 1: Distribution of participants by groups depending on system/manipulation. Note that since both *System-Auto* and *System-Self* look the same under the *Remove-Process* manipulation, we do not mention a specific system for this manipulation tactic.

5.1 RQ1 - Manipulation Detection

A total of 56 participants (44%) who were subjected to a manipulation reported it using the form on the website referenced in the study instructions. In the survey, 61 participants (49%) reported having problems with verifying their vote when asked a question about experiencing any issues with the system before debriefing. When asked whether they detected the manipulation after debriefing, 71 participants (57%) responded that they

[9] The full breakdown of the demographics is included in the extended version of our paper [Ni]

detected and reported the manipulation, while 31 (25%) more answered that they detected the manipulation but did not report it. Tab. 2 shows a breakdown of manipulation detection rates using different metrics by the system and manipulation tactic.

System	Manipulation	Voting System	Survey before Debriefing	Survey after Debriefing
System-Auto	*Remove-Vote*	13 (52%)	16 (64%)	21 (84%)
System-Auto	*Replace-Vote*	15 (60%)	16 (64%)	25 (100%)
System-Self/System-Auto	*Remove-Process*	0 (0%)	0 (0%)	16 (64%)
System-Self	*Remove-Vote*	9 (36%)	16 (64%)	17 (68%)
System-Self	*Replace-Vote*	19 (76%)	13 (52%)	23 (92%)
	Total	56 (45%)	61 (49%)	102 (82%)

Tab. 2: Overview of manipulation detection rates reported either via the voting system, survey before debriefing, or survey after debriefing (including participants answering that they detected the manipulation but did not report it within the system), separated by system/manipulation.

In order to understand the effect of either system or manipulation tactic on the detection rate, we decided to use the rate of participants reporting the manipulation during voting as our main measurement, in order to avoid the inaccuracies resulting from self-reporting within the survey. Figure 6 shows 95% confidence intervals [10] for all combinations of system/manipulation tactic. As such, our results show that while detection rates were similar between the two systems for each manipulation, the differences between manipulations are more pronounced, with *Remove-Process* manipulation being particularly hard to detect.

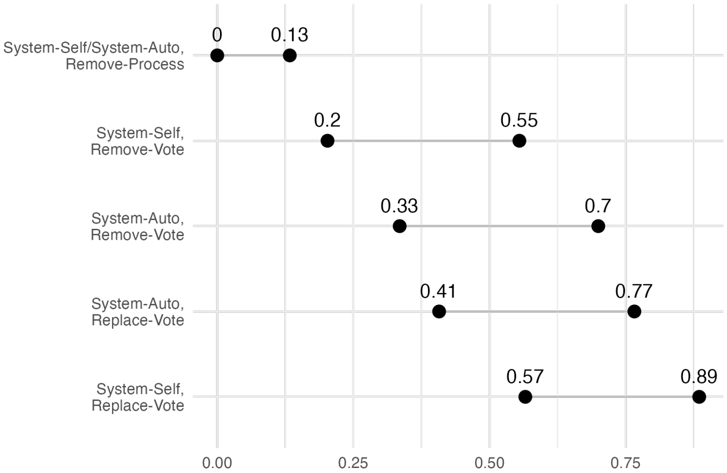


Abb. 6: 95% confidence intervals for manipulation detection rate, for all combinations of system and manipulation tactic.

[10] For all calculations of the confidence intervals, we used R package “DescTools”

5.2 RQ2 - Usability and Trust

Tab. 3 provides an overview of the mean scores for usability (using the SUS scale, ranging from 0 to 100) and trust (using the TVS “overall trust” scale, ranging from 1 to 7) for both *System-Self* and *System-Auto* (among the participants that were not subjected to any manipulation). Both of the systems fall within the range of “Good” to “Excellent” grades for usability [BKM09] and elicit a moderate to high level of trust.

	System-Auto	System-Self
Usability	M = 81.01	M = 83.07
	SD = 13.54	SD = 14.22
	CI = $78.23; 84.02$	CI = $80.11; 85.98$
Trust	M = 5.02	M = 5.35
	SD = 1.41	SD = 1.46
	CI = $4.72; 5.31$	CI = $5.06; 5.67$

Tab. 3: Overview of usability and trust scores for each system (M: mean, SD: standard deviation, CI: 95% confidence interval)

We furthermore calculated the scores from the scales for the perceived information availability, understandability, verifiability and general transparency of the systems, see Tab. 4, resulting in a moderate to high level of fulfillment across all the dimensions for both systems. While both of the systems had similar scores for perceived information availability, understandability and general transparency of the systems, the difference in scores for participants' perceived capability of the system to enable verification of cast votes was more pronounced, hinting at the need for further investigation in this direction.

	System-Auto	*System-Self*
Information availability	M = 4.92	M = 4.97
	SD = 1.1	SD = 1.44
	CI = $4.7; 5.15$	CI = $4.7; 5.25$
Understandability	M = 5.7	M = 5.76
	SD = 0.78	SD = 0.98
	CI = $5.54; 5.87$	CI = $5.59; 5.96$
Verifiability	M = 5.72	M = 5.98
	SD = 0.77	SD = 0.73
	CI = $5.56; 5.88$	CI = $5.83; 6.14$
General transparency	M = 5.4	M = 5.69
	SD = 1.27	SD = 1.28
	CI = $5.16; 5.68$	CI = $5.45; 5.95$

Tab. 4: Overview of scores TVID Dimensions for *System-Self* and *System-Auto* (M: mean, SD: standard deviation, CI: 95% confidence interval). The scores are calculated as mean of Likert scale items, ranging from 1 (low) to 7 (high). The 95% confidence intervals are calculated using the bootstrapping method, due to non-normal distribution of data.

6 Discussion

Study limitations Our research was carried out among participants, most of whom lacked familiarity with internet voting systems. Consequently, it raises the question of how applicable our results are to different demographics. This includes voters in nations like Estonia or Switzerland where internet voting is established, and the representation of various demographics beyond the younger, educated Prolific sample. Furthermore, as the data collection has been conducted in two phases, it is not clear whether this separation has contributed to a difference between *Remove-Process* and the other two manipulation tactics. Furthermore, in our study, the participants voted for a fictional candidate, minimizing personal data collection and promoting broader participation. While this approach reduced potential dropout due to privacy concerns, it's essential to note that using fictional candidates may impact participant seriousness in reporting.

Manipulation detection (RQ1) Our study shows relatively low verification detection rates for our systems. As such, none of the voters reported detecting manipulation using the link in the election instructions for the *Remove-Process* manipulation. Even if self-reporting of manipulation detection is taken into account (that is, assuming that the participants are telling the truth about detecting the manipulation but not reporting it), the rate of undetected manipulations remains up to 64% depending on the manipulation tactic. Furthermore, the verification process using tracking codes is optional – that is, the voter has to actively choose to verify their vote after the election is over. This can lead to low verification rates, as demonstrated by Internet voting in Estonia, where only around 5% of voters choose to verify their votes [Es24] [11]. Combined with low manipulation detection rates, the risk of undetected manipulation can be high if only a small percentage of voters chooses to verify their votes, and among the ones that do, a large share does not detect their votes being manipulated.

Our findings show that the *Remove-Process* manipulation was by far the hardest to detect. Since this manipulation implies that no hints are given to the voter via the voting client user interface, it is critical to communicate the need to verify one's vote via alternative channels (e.g. media campaigns) and provide instructions outside of the voting system specifying how to verify one's vote. Since similar studies of other verification techniques show improvement in verification rates given properly designed instructions available to voters as paper materials [Vo22a], designing such instructions for tracking code-based verification is an important direction of future work.

Overall, all three types of simulated attacks can become threats in a real election. Therefore, additional education is needed on a societal level outside the voting system, as especially instruction letters are not sufficient to detect different kind of manipulations. Additionally, ensuring the existence of reliable reporting channels is essential for the voters to report

[11] Note that the Estonian voting system relies on a different verification approach.

manipulations. Investigating most appropriate ways to establish and communicate such channels, as well as processes for handling reported manipulations (in particular, also accounting for voters who might lie about verification failures, aiming to create distrust in the election result) is an important direction of future work.

Our study furthermore has shown a large discrepancy between voters detecting and reporting the manipulation during or directly after interacting with the voting system, and self-reporting detecting the manipulation after being debriefed about its presence. Such discrepancy can be explained by several reasons, such as social desirability bias (i.e. participants not wanting to admit that they missed the manipulation), participants not feeling necessary to report a manipulation in a study setting, or not being able to find a link to the reporting form. Nonetheless, the real manipulation detection rates can be only roughly estimated, and a consistent metric for such an estimation has to be applied for further studies on manipulation detection rates, to ensure that the results of the studies are comparable with each other.

Usability and trust (RQ2) Our findings show moderate to high scores of usability and trust in both of our two proposed prototypes (with no manipulations). However, our proposed prototypes do not provide any information about how such types of systems are secured, nor do they inform voters of remaining risks such as voter coercion or violations of eligibility due to insufficiently secure voter authentication. Consequently, it remains an open question how trust will be affected if we notify people about the potential security risks related to electronic voting, including manipulations. Furthermore, our findings show a difference in perceived verifiability, with the prototype having a partly self-chosen tracking code perceived as more verifiable, as indicated by voters being more likely to agree with statements such as “I can confirm that the voting system accurately recorded my vote”. One possible explanation could be voters feeling a greater sense of control over their tracking code, thus becoming more engaged in the verification process. Nonetheless, this disparity did not result in a notable variance in perceived overall transparency or trust. As studies indicate that there are additional factors affecting both transparency and trust that were not addressed in this research [Ag22], further investigation is needed.

Further limitations of tracking code-based verification Our proposed voting systems, as well as systems relying on tracking code-based verification, do not address several critical risks that can be an issue in Internet voting. As such, while both *System-Self* and *System-Auto* offer an option to use a second device for verification using QR codes, such an option is not enforced, and the voters are furthermore not informed about the risks of using the same device for voting and verifying. Furthermore, as verification can only be done after tallying, vote manipulations would not be detected during vote casting, which might make it more challenging to support the voters (e.g. by telling them to cast their vote via an alternative voting channel in case of failed verification, as is the common practice in countries implementing Internet voting) or otherwise address them after the fact. This issue might become critical even in absence of actual vote manipulations, if there are voters who falsely claim verification failures. Finally, our study do not address the issue of coercion attacks which tracking code-based systems in general are prone to. While such attacks are addressed within the Selene voting system, the resulting system relies on complex cryptographic techniques, thus making the verification process potentially more difficult to understand (see Abschnitt 2). Therefore, future studies need to investigate how to address the risk of coercion attacks when developing tracking code-based systems.

7 Conclusion

Using tracking codes in Internet voting systems can provide an easy and intuitive way for voters to verify the integrity of their cast votes. Our study shows that systems relying on such verification can achieve a high level of usability and trust on behalf of the voters. However, we demonstrate that verification efficacy of tracking code-based systems can be lacking if the attacker can be assumed to have full or partial control over the voting client (e.g. the voting website) – an adversarial capability that cast-as-intended verification approaches, including tracking codes, were specifically designed to protect against. Depending on the specific manipulation tactic such an adversary can apply – such as removing all references of the verification process from the voting client user interface – the rate of voters who detect such a manipulation can be critically low, as shown by our study where none of the voters have noticed such a manipulation. Other kinds of manipulation tactics, involving attacker who is able to hide the vote from the voter's view (either by modifying the user interface or by blocking the vote from reaching the voting system, e.g. by interfering in the network communications), achieve a higher verification efficacy. However, even with such tactics, up to 64% of participants in our study failed to detect and report vote manipulation. Our findings imply that tracking code-based verification can potentially be used for low-stake elections where vote manipulation is not a critical issue. However, in more high-risk environments, additional care must be done to ensure that the verification efficacy of the implemented systems remains sufficiently high, e.g. with the use of properly designed and evaluated information materials educating voters on the verification process, as well as properly functioning and easy to use communication channels for reporting detected manipulations. Future research into development of such information materials and reporting channels is therefore required.

Acknowledgements

This work was supported by a research grant (40948) from VILLUM FONDEN.

Literaturverzeichnis

[Ac14] Acemyan, C. Z.; Kortum, P. T.; 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. In: EVT/WOTE. 2014.

[Ag22] Agbesi, S.; Dalela, A.; Budurushi, J.; Kulyk, O.: What Will Make Me Trust or Not Trust Will Depend Upon How Secure the Technology Is": Factors Influencing Trust Perceptions of the Use of Election Technologies. In: Proceedings of Seventh International Joint Conference on Electronic Voting (E-Vote-ID 2022). Seventh International Joint Conference on Electronic Voting : E-Vote-ID 2022 ; Conference date: 04-10-2022 Through 07-10-2022, University of Tartu, 2022.

[Ag23] Agbesi, S.; Budurushi, J.; Dalela, A.; Kulyk, O.: Investigating Transparency Dimensions for Internet Voting. In: International Joint Conference on Electronic Voting. Springer Nature Switzerland Cham, S. 1–17, 2023.

[AKO22] Acemyan, C.; Kortum, P.; Oswald, F.: The Trust in Voting Systems (TVS) Measure. International Journal of Technology and Human Interaction 18, S. 1–23, 2022, doi: 10.4018/IJTHI.293196.

[AS20] Alsadi, M.; Schneider, S.: Verify My Vote: Voter Experience. In. 2020.

[Be21] Beckert, B.; Budurushi, J.; Grunwald, A.; Krimmer, R.; Kulyk, O.; Küsters, R.; Mayer, A.; Müller-Quade, J.; Neumann, S.; Volkamer, M.: Aktuelle Entwicklungen im Kontext von Online-Wahlen und digitalen Abstimmungen, Techn. Ber., 46.23.01; LK 01, 2021, doi: 10.5445/IR/1000137300.

[BKM09] Bangor, A.; Kortum, P.; Miller, J.: Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. J. Usability Stud. 4, S. 114–123, 2009.

[Bo12] Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: Proceedings - 2012 IEEE Symposium on Security and Privacy, S and P 2012. Proceedings - IEEE Symposium on Security and Privacy, S. 538–552, 2012, doi: 10.1109/SP.2012.49.

[Di19] Distler, V.; Zollinger, M.-L.; Lallemand, C.; Rønne, P.; Ryan, P.; Koenig, V.: Security - Visible, Yet Unseen? How Displaying Security Mechanisms Impacts User Experience and Perceived Security. 2019, doi: 10.1145/3290605.3300835.

[Es24] Estonian National Electoral Committee: Statistics about Internet Voting in Estonia, Accessed: 16th of February, 2024, 2024, url: https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia.

[Ka11] Karayumak, F.; Olembo, M. M.; Kauer, M.; Volkamer, M.: Usability Analysis of Helios — An Open Source Verifiable Remote Electronic Voting System. In: 2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 11). USENIX Association, San Francisco, CA, 2011, url: https://www.usenix.org/conference/evtwote-11/usability-analysis-helios-%7B%5Ctextemdash%7D-open-source-verifiable-remote-electronic-voting.

[KTV12] Kusters, R.; Truderung, T.; Vogt, A.: Clash Attacks on the Verifiability of EVoting Systems. Proceedings - IEEE Symposium on Security and Privacy, S. 395–409, 2012, doi: 10.1109/SP.2012.32.

[Kü16a] Küsters, R.; Müller, J.; Scapin, E.; Truderung, T.: sElect: A Lightweight Verifiable Remote Voting System. In: 2016 IEEE 29th Computer Security Foundations Symposium (CSF). S. 341–354, 2016, doi: 10.1109/CSF.2016.31.

[Kü16b] Küsters, R.; Müller, J.; Scapin, E.; Truderung, T.: sElect: A lightweight verifiable remote voting system. In: 2016 IEEE 29th Computer Security Foundations Symposium (CSF). IEEE, S. 341–354, 2016.

[Ku20] Kulyk, O.; Volkamer, M.; Müller, M.; Renaud, K.: Towards Improving the Efficacy of Code-Based Verification in Internet Voting. In. S. 291–309, 2020, ISBN: 978-3-030-54454-6, DOI: 10.1007/978-3-030-54455-3_21.

[KV18] Kulyk, O.; Volkamer, M.: Usability is not Enough: Lessons Learned from 'Human Factors in Security' Research for Verifiability, Cryptology ePrint Archive, Paper 2018/683, https://eprint.iacr.org/2018/683, 2018, URL: https://eprint.iacr.org/2018/683.

[Ma18] Marky, K.; Kulyk, O.; Renaud, K.; Volkamer, M.: What did I really vote for? On the usability of verifiable e-voting schemes. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI Conference on Human Factors in Computing Systems : Engage with CHI, CHI 2018 ; Conference date: 21-04-2018 Through 26-04-2018, Association for Computing Machinery (ACM), United States, 2018, DOI: 10.1145/3173574.3173750, URL: %5Curl%20https://chi2018.acm.org/%22.

[Ma21] Marky, K.; Grube, T.; Kunze, K.; Zollinger, M.-L.; Roenne, P.; Ryan, P. Y. A.: Investigating Usability and User Experience of Individually Verifiable Internet Voting Schemes. ACM Trans. Comput.-Hum. Interact. 28, 2021.

[Ma22] Marky, K.; Gerber, P.; Günther, S.; Khamis, M.; Fries, M.; Mühlhäuser, M.: Investigating {State-of-the-Art} Practices for Fostering Subjective Trust in Online Voting through Interviews. In: 31st USENIX Security Symposium (USENIX Security 22). S. 4059–4076, 2022.

[Ni] Nissen, C.; Kulyk, O. K.; Volkamer, M.; Fredrich, L. E.; Hermansen, H.: Design and Evaluation of Verifiable Voting Systems Based on Tracking Code Verification. Extended version of the paper, URL: https://anonymous.4open.science/r/e-vote-id2024-0B62/.

[RRI15] Ryan, P. Y. A.; Roenne, P. B.; Iovino, V.: Selene: Voting with Transparent Verifiability and Coercion-Mitigation, Cryptology ePrint Archive, Paper 2015/1105, https://eprint.iacr.org/2015/1105, 2015.

[RRR21] Ryan, P. Y. A.; Rastikian, S.; Rønne, P. B.: Hyperion: An Enhanced Version of the Selene End-to-End Verifiable Voting Scheme. In: Proceedings of the Sixth International Joint Conference on Electronic Voting E-Vote-ID 2021. S. 285–287, 2021.

[TBL22] Tang, J.; Birrell, E.; Lerner, A.: Replication: How Well Do My Results Generalize Now? The External Validity of Online Privacy and Security Surveys. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). USENIX Association, Boston, MA, S. 367–385, 2022, ISBN: 978-1-939133-30-4, URL: https://www.usenix.org/conference/soups2022/presentation/tang.

[Vo22a] Volkamer, M.; Kulyk, O.; Ludwig, J.; Fuhrberg, N.: Increasing security without decreasing usability: A comparison of various verifiable voting systems. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). S. 233–252, 2022.

[Vo22b] Volkamer, M.; Kulyk, O.; Ludwig, J.; Fuhrberg, N.: Increasing security without decreasing usability: A comparison of various verifiable voting systems. In: Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022). USENIX Association, Boston, MA, S. 233–252, 2022, ISBN: 978-1-939133-30-4, URL: https://www.usenix.org/conference/soups2022/presentation/volkamer.

[Zo21] Zollinger, M.; Distler, V.; Rønne, P. B.; Ryan, P. Y. A.; Lallemand, C.; Koenig, V.: User Experience Design for E-Voting: How mental models align with security mechanisms. CoRR abs/2105.14901, 2021, URL: https://arxiv.org/abs/2105.14901.