Investigating Usability and User Experience of Individually Verifiable Internet Voting Schemes

KAROLA MARKY, Technical University of Darmstadt, Germany and University of Glasgow, United Kingdom
MARIE-LAURE ZOLLINGER, PETER ROENNE, and PETER Y. A. RYAN, University of Luxembourg, Luxembourg
TIM GRUBE, Technical University of Darmstadt, Germany
KAI KUNZE, Keio University, Japan

Internet voting can afford more inclusive and inexpensive elections. The flip side is that the integrity of the election can be compromised by adversarial attacks and malfunctioning voting infrastructure. Individual verifiability aims to protect against such risks by letting voters verify that their votes are correctly registered in the electronic ballot box. Therefore, voters need to carry out additional tasks making human factors crucial for security. In this paper, we establish a categorization of individually verifiable Internet voting schemes based on voter interactions. For each category in our proposed categorization, we evaluate a voting scheme in a user study with a total of 100 participants. In our study, we assessed usability, user experience, trust, and further qualitative data to gain deeper insights into voting schemes. Based on our results, we conclude with recommendations for developers and policymakers to inform the choices and design of individually verifiable Internet voting schemes.

CCS Concepts: • **Security and privacy → Usability in security and privacy**.

Additional Key Words and Phrases: E-Voting, Internet voting, individual verifiability, human factors

ACM Reference Format:
Karola Marky, Marie-Laure Zollinger, Peter Roenne, Peter Y. A. Ryan, Tim Grube, and Kai Kunze. 2021. Investigating Usability and User Experience of Individually Verifiable Internet Voting Schemes. *ACM Trans. Comput.-Hum. Interact.* 28, 5 (October 2021), 37 pages. https://doi.org/10.1145/3459604

1 INTRODUCTION

Elections are fundamental to democracy. With the drive to global digitization, some countries already utilize the Internet as an additional vote-casting channel, e.g., Estonia, Armenia, Canada, and Switzerland [34]. Internet voting brings various benefits, such as vote casting from any venue with Internet access, cost reduction [59], or faster announcement of the election result.

However, Internet voting also presents threats against the vote integrity, i.e., that the election result accurately reflects the voters' intentions. Lab testing the security of the voting software

Authors' addresses: Karola Marky, karola.marky@glasgow.ac.uk, Technical University of Darmstadt, Hochschulstr. 10, 64289, Darmstadt, Germany, University of Glasgow, School of Computing Science, 18 Lilybank Gardens, G12 8RZ, Glasgow, United Kingdom; Marie-Laure Zollinger, marie-laure.zollinger@uni.lu; Peter Roenne, peter.roenne.zollinger@uni.lu; Peter Y. A. Ryan, peter.ryan@uni.lu, University of Luxembourg, Maison du Nombre, 6, Avenue de la Fonte, L-4364, Esch-sur-Alzette, Luxembourg; Tim Grube, grube@tk.tu-darmstadt.de, Technical University of Darmstadt, Germany; Kai Kunze, kai@kmd.keio.ac.ja, Keio University, 4-1-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8526, Japan.

1073-0516/2021/10-ART $15.00
https://doi.org/10.1145/3459604

alone is not enough to ensure vote integrity [8, 9] because votes might be altered by malware on the voters' computers [38], by a malfunctioning voting software, or during transmission. Proof-of-concept attacks on politically binding elections have been demonstrated in France [40], the US [89], Australia [45] and Estonia [82].

To address these risks, the concept of verifiability has been proposed (cf. [2]). Verifiability[1] is typically described as a composition of three properties:

(1) **individual verifiability:** voters can verify that their ballots are indeed counted as intended,
(2) **universal verifiability:** anyone can verify that the result corresponds to published ballots,
(3) **eligibility verifiability:** anyone can verify that only eligible voters voted at most once.

While anyone can carry out the checks related to universal and eligibility verifiability, those related to individual verifiability are typically performed by the voters. In traditional paper-based schemes, voters verify by visually inspecting that their ballot represents their intention. Then, they insert it into the ballot box. Achieving both verifiability and privacy in Internet voting is, however, not trivial. Hence, the schemes have to be carefully designed to provide both properties.

Due to vote privacy, the voters' intents must be private, making it non-trivial to automate or delegate individual verifiability to a third party [30]. Voters have to validate the outcome (i.e., whether their votes are correct) by themselves. Related work demonstrated that maximizing the usability of individual verifiability constitutes a particular challenge (cf. [1, 35, 66, 93]). If voters are unable to verify, incorrect votes cannot be detected. Consequently, vote integrity cannot be guaranteed. Furthermore, if it is known that specific groups of voters do not verify, we run the danger of targeted attacks on such voters.

A variety of schemes for providing individual verifiability have been proposed in the scientific literature. The schemes differ regarding voter involvement, the voter's cognitive load, and the timing of verification. Existing evaluations of human factors strongly focused on particular verifiable Internet voting schemes, for instance, Selene [29, 93], the Benaloh challenge [66], or the Norwegian prototype [35]. However, what remains unexplored is how the schemes proposed in the literature compare to each other in terms of human factors and which type of schemes supports voters best in carrying out individual verification.

In this paper, we conduct a structured literature search to identify individually verifiable Internet voting schemes. Our literature search identified 34 schemes that we grouped into the five categories of 1) audit-or-cast, 2) verification devices, 3) tracking data, 4) code sheets, and 5) delegation.

We use our categorization as a basis to evaluate human factors of individual verifiable Internet voting schemes. This is achieved by a user study that assesses usability, user experience, trust, qualitative data concerning understandability, the adoption of verification, misconceptions, and voter concerns for each category with 100 participants where 25 participants interacted with each scheme. Our results depict that using schemes based on audit-or-cast is the least effective in detecting incorrect votes (28%). Those schemes do not align with voters' expectations. Using schemes based on code sheets, voters detected all incorrect votes but needed longer than all other schemes. Furthermore, code sheets have to be distributed before the election, adding organizational overhead. Verification device (64%) and tracking data (85%) schemes had lower detection rates. However, their overhead is lower because they rely on software only. Based on our results, we provide specific recommendations for the deployment of individually verifiable (Internet) voting schemes for developers and policymakers.

[1]For an overview of formal definitions, we refer the reader to the recent systematization of knowledge paper [25].

Research Contribution
Our work presents a structured literature research for verifiable Internet voting schemes that resulted in a categorization with the five categories audit-or-cast, tracking data, verification device, code sheets, and delegation. The categorization is based on the criteria of voter interactions and verification timing within the electoral process. We then conducted a user study with 100 participants to evaluate the perceived usability, user experience, and further aspects of the different categories of verifiable Internet voting schemes. Our results show that schemes based on audit-or-cast should be avoided outside of expert communities. Considering the other categories of schemes, the specific election should be taken into account for choosing a verification scheme. We conclude with recommendations for developers and policymakers for informing the design and choice of individual verifiable Internet voting schemes.

2 BACKGROUND AND RELATED WORK

In this section, we present related work and background. First, we summarize investigations of verifiable voting schemes by *user studies*. Then, we introduce *trust assumptions* that are an integral part of security.

2.1 User Studies

The usability of specific Internet voting schemes that offer some degree of individual verifiability has been investigated in a variety of user studies. The most thoroughly evaluated scheme is the Benaloh Challenge [10, 11]. The Benaloh Challenge is based on a challenge, thus, voters either cast the vote or challenge it for verification purposes. User studies of the challenge reveal two weaknesses: first, the effectiveness of detecting incorrect votes is rather low since only between 10 and 43% of participants were able to carry out verification successfully [1, 62, 66, 87]. Second, the challenge concept does not align with the voters' mental models, and therefore, they consider verification to be redundant because the verified vote cannot be cast [62, 66].

Verification can also be based on code sheets that the voters receive before the election. During vote casting, the voters use data on the code sheets for verification purposes. A direct comparison of the Benaloh Challenge to verification with codes demonstrated that code sheets offer better effectiveness [62]. Further schemes based on code sheets were investigated in a comparative study [17]. The participants interacted with three different voting schemes: (1) no verification, (2) return codes, and (3) a combination of return codes and code voting. The participants were informed about attacks that the respective scheme aims to prevent. The results show that voters are willing to sacrifice 26 points from the System Usability Scale [16] (scale from 0 up to 100) for the sake of security. That indicates that proper information of the voters might lead them to use more secure schemes, even if the usability is affected. The Swiss voting interface which implements code sheets was investigated in a series of user studies [68]. The study results show that the interface and information on the code sheet can impact the usability and user experience of the verification procedure. The Norwegian scheme is also based on code sheets [6]. Participants in a user study of its prototype could not determine whether their votes were submitted to the electronic ballot box [35].

The Selene scheme offers verification based on tracking codes that allow voters to identify their votes after the tally [78]. The tallied voting option is listed next to the voter's individual tracking code on a publicly available bulletin board. Displaying security-related information during voting and verification resulted in higher security perception but hampered understandability [29]. Studying mental models of Selene showed that voters are aware of potential security flaws in voting protocols but are not convinced by verifiability features [93].

Each of the publications mentioned above investigated a subset of the schemes available in the scientific literature. To our knowledge, our work offers the first comprehensive and comparative investigation of individually verifiable Internet voting schemes. Hereby, we do not only focus on usability since it has been repeatedly demonstrated that the usability of voting schemes is not enough to deliver effective security.

2.2 Trust Assumptions

Trust assumptions are used when designing a security protocol. Their purpose is to ensure that certain entities or components of a protocol are trustworthy. Trustworthy means that the entity or component functions without interference from adversaries. As long as the trust assumptions hold, the security of the protocol is assured. Consequently, Internet voting protocols also rely on trust assumptions to deliver specific security properties. In this section, we introduce trust assumptions related to individual verifiability. The trust assumptions were extracted from the verification protocols that we found in the structured literature search. To describe our proposed categorization in Section 4, we need the following trust assumptions[2]. A subset of them is required in each category to satisfy verifiability:

- **A1** The voting device is trustworthy.
- **A2** The supplementary device used for verification is trustworthy.
- **A3** The bulletin board is trustworthy.
- **A4** The component for code generation is trustworthy.
- **A5** Printing and distribution of auxiliary material is trustworthy.
- **A6** The electronic ballot box is trustworthy.
- **A7** The third party is trustworthy.

We use these trust assumptions in Section 4 for explaining trust properties of the voting scheme categorizes.

3 STRUCTURED LITERATURE SEARCH AND CATEGORIZATION METHODOLOGY

To identify individually verifiable Internet voting schemes, we conducted a structured literature search [85, 88]. The resulting literature list was then categorized. In the remainder of this section, we describe the methodology of the structured literature search and categorization.

3.1 Structured Literature Search

For the structured literature search, we used the keyword "individual verifiability". The terms "cast-as-intended" and "recorded-as-cast" were taken as further keywords since they denote components of individual verifiability. Furthermore, we searched for "end-to-end verifiability". The **search space** was given by the scientific databases ACM, IEEExplore, and SpringerLink. We also searched the proceedings[3] of conferences, workshops, and journals related to electronic voting but not published in the mentioned databases. We **excluded** a publication if it was not related to individual verifiability in the scope of Internet voting.

We then performed a forward and backward search. We examined whether the reference list of each paper contains references concerning individual verifiability. Publications found during this phase were not required to be published in one of the above-mentioned databases. Furthermore, we

[2]Considering the overall voting scheme, more trust assumptions have to be considered. However, we focus on those directly related to verification.

[3]The conferences were the International Conferences on Electronic Voting (EVote), Electronic Government (EGov), Electronic Participation (EPart), and Usenix Security. The workshops were the Electronic Voting Technology Workshop (EVT) and its successor Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE). Furthermore, the USENIX Journal of Election Technology and Systems (JETS) was searched.

examined which publications cite those we have found during the search. The structured literature search resulted in 34 publications with Internet voting schemes that provide individual verifiability. For a complete list of these schemes, the reader is referred to Table 1.

Table 1. List of publications with individually verifiable Internet voting schemes.

Category	Publication Title	First Author	Year	Ref.
Audit-or-Cast	Simple Verifiable Elections	Benaloh, Josh	2006	[10]
	Ballot Casting Assurance via Voter-Initiated Poll Station Auditing	Benaloh, Josh	2007	[11]
	BeleniosRF: A Non-interactive Receipt-Free Electronic Voting Scheme	Chaidos, Pyrros	2016	[19]
	BeleniosRF: A Strongly Receipt-Free Electronic Voting Scheme	Cortier, Véronique	2015	[24]
	Trivitas: Voters Directly Verifying Votes	Bursuc, Sergiu	2011	[18]
	From Helios to Zeus	Tsoukalas, Georgios	2013	[84]
	Apollo–End-to-End Verifiable Internet Voting with Recovery from Vote Manipulation	Gaweł, Dawid	2016	[37]
	PrivApollo–Secret Ballot E2E-V Internet Voting	Wu, Hua	2019	[90]
	Ordinos: A Verifiable Tally-Hiding E-Voting System	Küsters, Ralf	2020	[63]
Tracking Data	sElect: A Lightweight Verifiable Remote Voting System	Küsters, Ralf	2016	[64]
	Selene: Voting with Transparent Verifiability and Coercion-Mitigation	Ryan, Peter Y. A.	2016	[78]
	Using Selene to Verify Your Vote in JCJ	Iovino, Vincenzo	2017	[51]
	An Efficient E2E Verifiable E-voting System without Setup Assumptions	Kiayias, Aggelos	2017	[55]
Verification Device	Verifiable Internet Voting in Estonia	Heiberg, Sven	2014	[46]
	Estonian Voting Verification Mechanism Revisited Again	Kubjas, Ivo	2017	[61]
	An Overview of the iVote 2015 Voting System	Brightwell, Ian	2015	[15]
	How to Challenge and Cast Your E-Vote	Guasch, Sandra	2016	[42]
Code Sheets	Pretty Good Democracy	Ryan, Peter Y. A.	2009	[79]
	Remotegrity: Design and Use of an End-to-End Verifiable Remote Voting System	Zagórski, Filip	2013	[91]
	2015 Neuchâtel's Cast-as-Intended Verification Mechanism	Galindo, D.	2015	[36]
	Cast-as-Intended Mechanism with Return Codes Based on PETs	Brelle, Achim	2017	[14]
	Internet Voting System With Cast As Intended Verification	Allepuz, Jordi Puiggalí	2011	[5]
	Cast as Intended Verifiability for Mixed Array Ballots	Mateu, Víctor	2017	[70]
	Return Code Schemes for Electronic Voting Systems	Khazaei, Shahram	2017	[53]
	Cast-As-Intended Verification in Electronic Elections Based on Oblivious Transfer	Haenni, Rolf	2016	[44]
	Cast-as-Intended Verification in Norway	Allepuz, Jordi Puiggalí	2012	[6]
	The Norwegian Internet Voting Protocol	Gjøsteen, Kristian	2011	[39]
	Security and Trust for the Norwegian E-voting Pilot Project E-valg 2011	Ansper, Arne	2009	[7]
	Secure Internet Voting With Code Sheets	Helbach, Jörg	2007	[47]
	End-to-End Verifiable Elections in the Standard Model	Kiayias, Aggelos	2015	[54]
	D-DEMOS: A Distributed, End-to-End Verifiable, Internet Voting System	Chondros, Nikos	2016	[21]
Delegation	Du-Vote: Remote Electronic Voting with Untrusted Computers	Grewal, G. S.	2015	[41]
	To du or not to du: A Security Analysis of Du-Vote	Kremer, Steve	2016	[58]
	BeleniosVS: Secrecy and Verifiability Against a Corrupted Voting Device	Cortier, Véronique	2019	[23]
	Universal Cast-as-Intended Verifiability	Escala, Alex	2016	[30]

3.2 Categorization Methodology

To cluster the 34 schemes into categories, we used the following categorization criteria: verification timing and voter interactions. Verification timing refers to the timing of the verification within the electoral process, which can be before vote casting, during vote casting, after vote casting, and after vote tallying. Voter interactions are atomic tasks that the voters have to carry out by themselves to verify their votes successfully. Hence, a voter action is the smallest possible unit of action carried out by a voter.

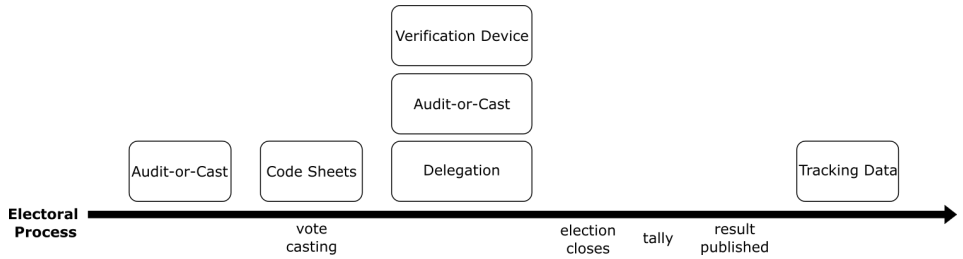


Fig. 1. The timing of verification within the electoral process differs in the categories extracted from the schemes in the scientific literature.

Using these two criteria, we followed the following methodology. We first constructed a sequence of required voter interactions for each scheme in the form of a sequence diagram. The voting system, which might consist of different components, such as bulletin boards or ballot box servers, was simplified to a black box that receives and sends data. This simplification was carried out whenever a voter action was not required.

In the second step, we started by grouping schemes with identical voter interactions into categories. Next, we followed an inductive categorization approach by combining categories with similar voter interactions to final categories. The resulting categorization was discussed with four researchers that are experts in Internet voting schemes, and a final categorization of five categories was agreed upon.

4 CATEGORIZATION OF INDIVIDUALLY VERIFIABLE VOTING

In this section, we describe our categorization of individual verifiable Internet voting schemes which consists of the categories 1) audit-or-cast, 2) tracking data, 3) verification device, 4) code sheets, and 5) delegation. Figure 1 provides an overview of the different timings of verification within the electoral process that we extracted from the schemes in the literature.

Each section is organized as follows. First, we describe the *voter interactions* that are required for successful verification. The voter interactions are formatted in bold letters and Figure 2 gives an overview of the voter interactions in each category. When describing the interactions we used common terminology for interactions that follow the same principles.

Next, we list the *trust assumptions* the categorization relies on as introduced in Section 2.2. Finally, we detail *schemes and implementations* of the categories. Screenshots of verification are provided in Appendix A.1. We used these screenshots for our user study. However, our participants interacted with a German version.

4.1 Category: Audit-or-Cast (AC)

Overall, the schemes based on audit-or-cast, voters can prepare additional encrypted ballots that they can challenge and spoil for auditing purposes before casting their votes. Individual verification is explicitly split into cast-as-intended and recorded-as-cast verification.

4.1.1 Voter Interactions. To carry out individual verification based on audit-or-cast, the voter Alice has to complete the following steps. After **preparing an encrypted vote** with the voting software, Alice either **casts the encrypted vote** as her final vote or **audits** it. Casting and auditing cannot be executed in parallel. That means Alice verifies *before* vote casting. At the time of vote preparation, the voting software must not know whether Alice will cast or audit this ballot. Otherwise, the software could cheat successfully.

Before indicating her cast or audit decision, Alice receives a vote identification code[4] that she **records**, e.g., by writing it down. Alice indicates whether she wants to cast or audit the encrypted vote. If Alice opts for auditing, she **launches a verifier**. This is an independent verification software that reveals the contents of the encrypted vote to her, and recalculates the vote verification code. Next, Alice **compares** her recorded vote identification code and the selection she made to those displayed by the verification software.

A verified vote cannot be cast because the verification could be used later on to prove Alice's vote choice. Furthermore, auditing is probabilistic. This means that Alice can only reach a certain level of confidence that her vote was cast-as-intended since verified votes cannot be cast. However, she can repeat the auditing process as often as she wishes.

The second step is verifying that Alice's vote was recorded-as-cast. After vote casting, Alice **accesses** a publicly available bulletin board, e.g., a website, that contains a list of voter pseudonyms and vote identification codes. Alice has to **find** her pseudonym and **compares** the listed vote identification code to verify that her vote has not been altered or deleted during transmission.

4.1.2 Trust Assumptions. If the verification software runs on the same device, the voting device must be trustworthy (A1). If the verification software runs on another device, the voting and verification devices must not be simultaneously corrupted (A1 or A2). Further, the bulletin board has to be trustworthy (A3).

4.1.3 Schemes and Implementations. The Benaloh Challenge [10, 11], BeleniosRF [19, 24], Trivitas [18], Zeus [84], Apollo [37], its extension PrivApollo [90], and Ordinos [63] belong to this category.

While the voter interactions are similar, the decision indication whether to cast or audit an encrypted vote depends on the voting scheme. It could be the submission of a previously received audit credential [18] or a button click in the voting software [10].

Audit-or-cast schemes have already been used in real elections. The Benaloh Challenge is implemented in Helios [3] which has been used in many academic elections, e.g., in the election for the board of the IACR (International Association of Cryptologic Research) [43], or the election of the university president at Université Catholique de Louvain [4]. The system Zeus was used in elections in Greek universities [84].

4.2 Category: Tracking Data (TD)

Schemes based on tracking data allow verification after the election tally based on a tracking code. These schemes are special because the individual verifiability check directly verifies the vote in the election result and can partially replace or strengthen universal verifiability.

4.2.1 Voter Interactions. Individual verifiability based on tracking data is the look-up of verification data on a bulletin board. Alice **prepares an encrypted vote** and **casts** it. After the election has closed and the tally has been completed, votes are anonymized and posted in clear text on a bulletin board. To identify her vote, Alice **receives an individual tracking code**. To verify her vote, Alice **accesses the bulletin board** and **locates her tracking code**. Then she **compares** the voting option next to the tracking code to her intended one. If the tracking code is present on the bulletin board and the voting option matches Alice's intent, her vote is correct.

4.2.2 Trust Assumptions. The bulletin board that displays the tracking codes has to be trustworthy (A3). If the tracking codes are generated, the component that generates them has to be trustworthy

[4]The purpose of this code is mathematically shortening the encrypted vote to enable better human handling, e.g., by hashing. Thus, all actions with the vote identification code could also be carried out with the encrypted vote.

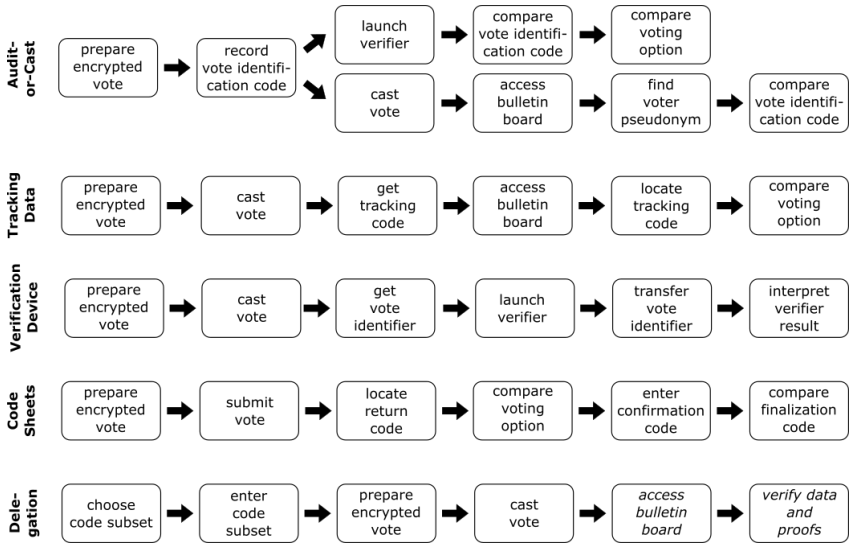


Fig. 2. Voter interactions in the different verification categories. A voter interaction is an atomic task that voters have to carry out to verify their votes.

(A4) to avoid collision attacks, or the creation of the codes has to be verifiable. The device that Alice uses to access the bulletin board has to be trustworthy (A1 or A2).

4.2.3 Schemes and Implementations. The following schemes use automatically issued tracking codes: the sElect system [64], the Selene system [78], the Selene extension of JCJ/Civitas [51] and an E2E system without setup-assumptions [55]. The Selene system allows the voter to obtain the tracking code *after* tallying to achieve receipt-freeness. The sElect system [64] offers the possibility to create tracking codes manually. The commercial voting system Polyas is based on tracking data and used by a variety of customers, such as banks [75].

4.3 Category: Verification Device (VD)

The schemes in this category rely on a supplementary device to carry out verification. Here, the verification is done after vote casting by inspecting the encrypted vote in the electronic ballot box.

4.3.1 Voter Interactions. First, Alice **prepares an encrypted vote** and **casts** it. After vote casting, she **gets a vote identifier** from the voting software. Next, Alice **launches a verifier** on her supplementary device. She **transfers** the identifier from the voting client on the voting device to the supplementary device. The supplementary device requests the encrypted vote from the electronic ballot box, inspects it and then displays the verification result. Depending on the specific scheme, the result might be a direct statement meaning that no further interpretation is needed, or Alice has to **interpret** it to determine whether her vote is correct. In the latter case, Alice has to compare a displayed voting option to her intended one.

4.3.2 Trust Assumptions. The supplementary device is assumed to be trusted if the voting device is corrupted (A1 or A2) and the electronic ballot box has to be trustworthy (A6).

4.3.3 Schemes and Implementations. The following schemes belong to this category: the Estonian scheme [46], the again-revised Estonian scheme [61], the iVote scheme [15] and challenge-and-cast [42]. Verification based on *SD* is used in Estonia for all types of political elections [31]. The

iVote scheme is used in the state of New South Wales in Australia for voters that fulfill certain criteria [73].

4.4 Category: Code Sheets (CS)

All schemes in this category use code sheets as auxiliary material to carry out verification.

4.4.1 Voter Interactions. Before the election, each voter receives an individual code sheet that is distributed over a trusted channel, such as trusted postal mail. The code sheet contains the list of available voting options and individual return codes for each voting option. Alice lets the voting software **encrypt** and **submit** her voting option. Note, that the vote is only submitted to the server but not yet inserted into the electronic ballot box and hence not yet cast. Next, Alice receives a return code from the voting system. She **locates the return code** on her code sheet and **compares** the voting option next to it to her intention.

If the received return code matches the one on the code sheet and the listed intention matches hers, Alice's vote is cast-as-intended. Additionally to the return codes, the code sheet can contain a confirmation code and a finalization code. After comparing the return codes, Alice **enters the confirmation code** into the voting software to confirm that she has compared the return codes and that the codes match. Depending on the used scheme, the voting system might confirm the receiving of the confirmation code by answering with the finalization code to confirm that the vote is recorded-as-cast. Alice **compares the finalization code** to the one on her code sheet.

4.4.2 Trust Assumptions. The printing and distribution of the code sheets have to be trusted (A5). Furthermore, the generation of the codes has to be trustworthy (A4). The server has to be trusted (A6) because of the submission of the return, confirmation, and finalization codes.

4.4.3 Schemes and Implementations. The usage of vote codes was initially proposed in the SureVote scheme for in-person voting [20]. Pretty Good Democracy introduced code voting for Internet voting via Plaintext Equivalence Tests (PETs) [79]. Remotegrity [91] only use a single return code, as it is also suggested in Pretty Good Democracy, and was employed in a municipal election in Takoma Park (US) in November of 2011. Further, the following schemes belong to this subcategory: the Neuchâtel scheme [36], PETs-based verification [14], revision of eValg2011 [5], verification for mixed-array ballots [70] and the schemes by Khazaei and Wikström [53], oblivious transfer based verification [44], secure Internet voting based on code sheets [47], DEMOS [54] and D-DEMOS [21]. While the voter interactions in these schemes are similar or identical, the calculation or generation of the codes differs among the schemes from an algorithmic perspective.

The Norwegian protocol specifically includes that the return codes are sent to the voters via SMSes [6, 7, 39]. *CS* schemes were used in Switzerland in several cantons [76], and in pilot elections in Norway [7].

4.5 Category: Delegation (DE)

There also exist schemes that aim to enable a delegation of the individual verifiability checks to a third party without violating vote privacy. Therefore, the process of vote casting differs substantially.

4.5.1 Voter Interactions. Before voting, Alice receives a list of voting codes. Instead of directly providing her choice, she **chooses a subset** of these codes that represent her choice and **enters** it into the voting software. Next, she **prepares an encrypted vote** using the voting software. The software forwards the encrypted vote along with mathematical proofs to the voting server to **cast** the vote. Alice can **access a bulletin board** to verify her vote, but she can also delegate these checks to a third party. If she **verifies** by herself, she checks the data and the mathematical proofs.

4.5.2 Trust Assumptions. For checking the data and the mathematical proofs on the bulletin board, the board is assumed to be trusted (A3). For the delegation to the third party, this party has to be trustworthy (A7).

4.5.3 Schemes and Implementations. The schemes Du-Vote [41, 58], BeleniosVS [23], and the universal cast-as-intended approach in [30] belong to this class of schemes. Du-Vote uses a trusted hardware token to compute the codes. BeleniosVS and universal cast-as-intended are based on code sheets to inform the voters about their codes. Delegation-based approaches have not been used in elections yet.

5 METHODOLOGY

To evaluate usability and user experience of our categorization[5], we conducted a user study with a total of 100 participants. Thus, the four conditions in the user study were: 1) audit-or-cast (*AC*), 2) tracking data (*TD*), 3) verification device (*VD*), and 4) code sheets (*CS*).

5.1 Apparatus

We designed voting and verification interfaces for each investigated category. For this, we did not implement cryptographic procedures but clickable interface prototypes. We made sure that differences between the conditions only resulted from differences between the verification schemes. In each condition, we used common terminology and design elements to avoid biases. For screenshots of verification, the reader is referred to Appendix A.1.

5.1.1 Voting Software. We implemented a voting software interface for each category in the form of a website. The voting software had a back-end in which the examiner could adjust and reset the voting software. Furthermore, the voting software was able to save timestamps of actions. To provide a realistic scenario, such that the participants would take the simulated election seriously [69, 81], we adapted each voting software to match a governmental election in Germany. The voting software could be accessed via login credentials in the form of a voter ID and a password. The mock election had two contests, and we used the ballot design and candidate list from the last election in Germany. We furthermore adjusted the texts of the respective voting software to match this election. All texts were presented in German.

5.1.2 Verification Material. For the conditions *VD*, *AC*, and *TD* we implemented verification apps that run on a smartphone. The reason for this is that we did not start our work from scratch. Instead, we built upon previous studies in the scientific literature on *AC* [66, 72] and *TD* [29, 93] schemes that specifically investigated the usage of an external device for verification and, based on that, recommend it. *VD*-based schemes are based on the usage of a verification device. For the *CS* condition, we designed a code sheet matching the election scenario based on recommendations from previous studies of *CS* schemes [68]. For verification in *TD*, the tally results need to be available. In the study, we considered a duration of two weeks between the election and result publication.

5.2 Captured Data

During our user study, we captured quantitative as well as qualitative metrics to assess usability (ISO 9241-11 [49]), user experience (ISO 9241-210 [50]), trust, and the participants' perceptions of the different schemes.

[5]We deliberately choose not to evaluate the delegation category for two reasons: 1) the voting process within this category is fundamentally different, 2) since verification is delegated, evaluating its effectiveness is not directly comparable since voters do not verify by themselves.

Usability is based on the criteria of effectiveness, efficiency, and satisfaction [49]. Assessing the *effectiveness* of verification constitutes a particular challenge since all schemes demand voters to perform mental tasks, such as comparisons, that cannot be measured directly. The self-reporting of participants, whether they indeed performed the mental task is not reliable enough, and even eye-tracking can only give information if the participant looked at the data but not if they performed the mental task. Therefore, we chose a proxy measure to assess effectiveness. In particular, we used deliberate manipulations of votes as recommended in the literature [69, 81]. Thus, effectiveness was captured by the share of participants that noticed incorrect votes.

To realize this, we manipulated votes to match a voting option different from the one chosen by the participant. To capture whether the participant observed an incorrect vote, we implemented buttons on the page with the data that the participants should verify. The buttons explicitly stated "yes, the data is correct" and "no, the data is incorrect" as recommended by related work [68]. The latter button forwarded participants to a page that instructed them to notify the examiner.

We assessed efficiency by the execution time required to vote and to verify. Since those two tasks cannot be clearly separated in each scheme, we included the time for voting. The measurements were taken by the voting software that stored timestamps in a database. The start and endpoints of all conditions were equal. Furthermore, we stored timestamps of each step and recorded the screens of all devices.

Satisfaction is assessed by the System Usability Scale questionnaire (SUS) [16]. User Experience is assessed by the User Experience Questionnaire (UEQ) [65] which assesses the six scales of

(1) attractiveness: the overall impression of an interface
(2) perspicuity: the ease of learning and getting familiar with the interface
(3) efficiency: the perceived effort spent to interact with the interface
(4) dependability: the perception of feeling in control
(5) stimulation: the perception regarding excitement and motivation during interaction
(6) novelty: the perceived creativity of the interface

To gain a deeper insight into the voters' trust in and perceptions of the different verification schemes, we asked eight additional questions. For the complete questionnaire, the reader is referred to Appendix A.

5.3 Study Design and Procedure

To prevent sequence effects, we opted for a between-subjects design. Thus, we had four groups, and each participant was randomly assigned to one scheme. To take our measurements and control the environment, we conducted a lab study and provided the participants with devices. An examiner was present in the room at all times but was positioned so that the screens of the participant's devices were not visible to them. Before the actual user study, we conducted a pilot study with three participants for each of the four conditions. We used these studies to refine the wording of our questions and their order. The results of the pilot study furthermore resulted in minor modifications to study materials. The procedure of our user study was as follows:

5.3.1 Consent Form & Demographics. We commenced by informing the participants about our study, the consent form as well as the study's data protection policy. After signing the consent form, the participants completed a demographics questionnaire. If a participant was a minor, the participant received the consent form before the study. In doing so, a legal guardian could review the consent form and sign it if they allowed the minor to participate.

5.3.2 Study Material. We proceeded by explaining the documents and materials to the participants. We provided a letter from the election authorities which contained sealed voting credentials.

Furthermore, we opted to provide written voting instructions to the participants to protect their vote privacy from our measurements [69, 81]. Each participant drew one of ten paper slips with different candidates. The participants were instructed to vote for the candidates on the paper slip and make sure that the vote is cast for that candidate. The paper slip was placed in front of the participants, such that they do not have to remember the voting instructions. In case the participant felt uncomfortable with the listed voting options, they could redraw.

5.3.3 Interaction and Questionnaires. The participants cast two votes with the voting scheme corresponding to their group. In the first round, the participants cast their votes, which corresponds to the usual voting scenario. We used this round to assess the efficiency, satisfaction, and user experience of the verification to avoid biases based on the experience of an incorrect vote. In the second round, we manipulated the cast vote to assess effectiveness, as detailed above. After each round, the participants were asked to fill in the SUS and UEQ questionnaires. We asked them to focus on verification specifically and provided them with screenshots from the verification steps.

5.3.4 Final Questionnaire & End. After completing the second round, the participants were given a final questionnaire with open-ended questions. In the last part, the participant was given the opportunity to ask questions. We also told the participants that we manipulated the voting option in the second round and that one of the study's goals was to investigate whether participants would find it. Furthermore, we provided them the opportunity to explore the voting software freely. Finally, the participant could fill in the consent form to participate in a raffle for an online shopping voucher.

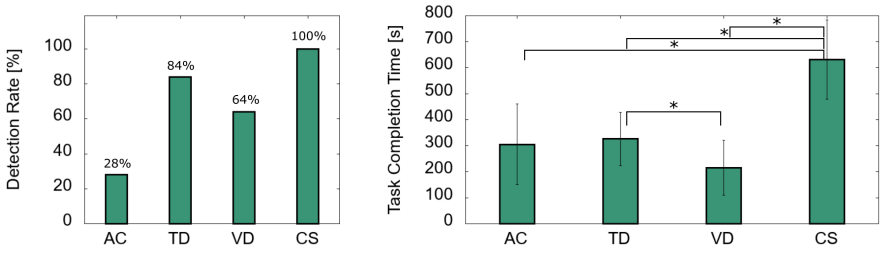
5.4 Recruitment and Participants

For our user study, we recruited 100 participants through different methods. In particular, we used mailing lists, flyers, poster advertisements, and snowball sampling. We did not reimburse the participants, but they could voluntarily participate in a raffle for four Amazon vouchers in the value of roughly 100 dollars. All participants possessed suffrage and had never participated in an Internet election before. Their mean age was 34.43 years ($SD = 15.59$, $Median = 28.00$, $Min = 17.00$, $Max = 72.00$). From the participants, 58% identified as male, 39% as female and 2% identified as diverse.

From the participants, 45% reported having a secondary school leaving certificate, 27% reported a university degree, 26% had a high school diploma, and 2% reported a doctoral degree or higher. All participants reported daily Internet usage.

5.5 Ethical Considerations

There is no requirement for following a formal IRB process for the kind of user study that we conducted in Europe. However, the ethics committee at our institution provides a set of guidelines, and our study procedure aligns with them. In particular, user studies at our institution have to preserve the participants' privacy and limit the collection of personal data to data that is necessary for the scope of the study. To preserve the participants' privacy, each participant received a randomly assigned identifier that we used throughout the study. Before participation, all participants received a consent form and signed it. The consent form was stored separately from all other captured data such that the captured data cannot be linked to the participants' identities. The consent form contained a description of our research goal, the procedure of the study, risks associated with participation, and information on how we store and analyze the captured data. It furthermore had a paragraph about the study's data protection policy. Our study complied with strict national privacy regulations. Since our measurements violate the vote privacy of the participants, we provided them with randomly assigned written voting instructions.



(a) Effectiveness (b) Efficiency

Fig. 3. Effectiveness and efficiency results. The asterisk * denotes significant differences.

6 QUANTITATIVE RESULTS

In this section, we present the quantitative results of our user study.

6.1 Effectiveness

Effectiveness is given by the share of the participants who reported an incorrect vote. Only 28% ($N = 7$) of incorrect votes in AC were reported. In VD, participants found 64% ($N = 16$), in TD they found 84% ($N = 21$), and 100% ($N = 25$) of incorrect votes were found in the CS condition (see also Figure 3a and Table 2 in the Appendix). A χ^2-test of independence was performed to examine the relationship between the scheme and the ability to detect an incorrect vote. The relation between these variables was significant ($\chi^2(3) = 33.80$, $p < .001$, $Cramer's\ V = .58$) between the four conditions. For the post-hoc tests, we looked at the adjusted residuals and used the Bonferroni correction to prevent the inflation of type I errors. The tests reveal that there is a relation between CS and the ability to detect an incorrect vote ($p = .0001$) and also between AC and the ability to detect an incorrect vote ($p < .0001$).

We furthermore analyzed the effectiveness results with a simplistic model that considers the detection of incorrect votes as a binomial experiment with a fixed detection probability. We then estimate the detection probability to be the average effectiveness and determine the confidence intervals via the Clopper-Pearson method. This results in the following 95% confidence intervals for the detection probabilities AC: $[0.12, 0.49]$, SD: $[0.42, 0.82]$, TD: $[0.63, 0.95]$, CS: $[0.86, 1.00]$. Based on this result, we conclude for the effectiveness that CS is better than both AC and VD, and TD is better than AC. However, TD and CS cannot be distinguished.

6.2 Efficiency of Non-Manipulated Trials

Since it is not possible in all conditions to clearly separate voting from verification, efficiency is assessed by measuring the execution time in seconds of the voting and the verification process in the non-manipulated trials. Participants in the VD condition were faster and needed on average 215.88s ($SD = 106.28$). Using AC participants needed 305.48s ($SD = 155.65$), and 326.68s ($SD = 103.12$) to use TD. In the CS condition participants took longest with 632.44s ($SD = 152.28$). Figure 3b provides an overview of the efficiency results, and descriptive statistics are given in Table 2 in the Appendix.

Since our data set met the assumptions of homogeneity of variances and normal distribution, we analyzed it with a one-way ANOVA. The test reveals that the execution time differed significantly

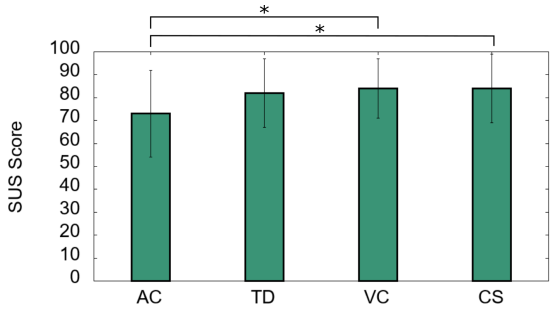


Fig. 4. Satisfaction scores given by the System Usability Scale. The asterisk * denotes significant differences.

between the conditions ($F(3, 96) = 47.18$, $p < .001$, $\eta^2 = .60$[6]). For the post-hoc tests, we used the Bonferroni correction to account for multiple testing. The post-hoc analysis revealed a significant difference between *CS* and all other voting schemes. In particular, participants using *CS* needed on average 326*s* longer than participants using *AC*, 416*s* longer than *VD* and 305*s* longer than *TD*, with $p < .001$ each. Participants using *VD* were on average 111*s* faster than participants using *TD* ($p = .02$). We could not find significant differences between participants using *AC* compared to the participants using *VD* ($p = .10$) and those using *TD* ($p = 1.00$).

6.3 Satisfaction of Non-Manipulated Trials

To assess satisfaction, we used the System Usability Scale (SUS) [16]. The scale ranges from 0 to 100 points, and higher values indicate better subjective usability.

The *AC* condition was rated lowest with an average SUS score of 73.00 ($SD = 19.57$), the *TD* condition received a SUS score of 82.10 ($SD = 15.08$), and the *CS* condition received 84.50 ($SD = 15.81$). Finally, the *VD* condition received the highest SUS score of 84.60 ($SD = 13.16$). The SUS scores are depicted in Figure 4. Descriptive statistics of the SUS scores are given in Table 3 in the Appendix.

First, we omitted two outliers because their SUS scores were more than three standard deviations away from the average. Then, we checked whether our data set met the assumptions of homogeneity of variances and normal distribution and analyzed it by a one-way ANOVA. The ANOVA revealed low but significant effects between the conditions with $F(3, 94) = 4.76$, $p = .004$, $\eta^2 = .13$. Bonferroni-corrected post-hoc tests were significant for *AC* compared to *CS* (*Difference* = 13.67, $p = .009$) and *VD* (*Difference* = 13.46, $p = .01$), indicating lower values for *AC* in both comparisons.

6.4 Subjective Trust

In the final questionnaire, we asked the participants whether they are confident that they can verify that their votes are correctly transmitted to and stored in the electronic ballot box using the provided software and materials. 44% of participants using tracking data, 64% of those using audit-or-cast, 84% of those using code sheets, and 88% of participants using the verification device answered this question affirmatively. A χ^2-test of independence was performed to examine the relationship between the scheme and the reported trust. The relation between these variables was significant ($\chi^2(3) = 14.67$, $p = .002$, $Cramer's\ V = .38$). For the post-hoc tests, we looked at the adjusted residuals and used the Bonferroni correction to prevent the inflation of type I errors. The post-hoc tests reveal a relation between *TD* and reported trust ($p = .0011$).

[6]All reported values of η^2 refer to the partial η^2.

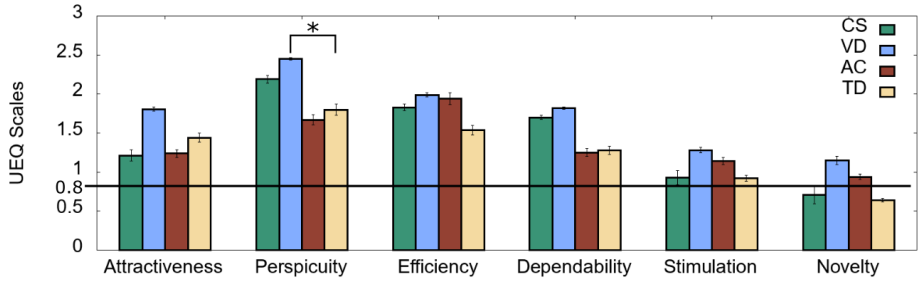


Fig. 5. Depiction of the user experience scales. Values below -0.8 represent a *negative evaluation*, between -0.8 and 0.8 a *neutral evaluation*, and values above 0.8 mean a *positive evaluation*. The asterisk * denotes significant differences. All error bars depict the standard error.

6.5 User Experience

User experience was measured by the user experience questionnaire, which assesses the six scales of 1) attractiveness, 2) perspicuity, 3) efficiency, 4) dependability, 5) stimulation, and 6) novelty [65]. Each scale ranges from -3 to +3, values below -0.8 represents a *negative evaluation*, values between -0.8 and 0.8 indicate a *neutral evaluation*, and values above 0.8 mean a *positive evaluation*. All schemes received a positive evaluation in all scales except for the novelty of TD (0.64) and CS (0.71). The individual UEQ results of the four conditions are depicted in Figure 5.

Because the assumption of homogeneity of variances was violated, we analyzed each scale with a Welch ANOVA which revealed significant differences in the scales of perspicuity (aspects like understandability, or difficulty) with $F(3, 96) = 2.98$, $p = .04$, $\eta^2 = .09$, and dependability (aspects like security, or predictability) with $F(3, 96) = 3.05$, $p = .03$, $\eta^2 = .09$. Using Bonferroni-corrected pairwise comparisons, we could not find significant differences in dependability (all *differences* < 0.78, $p > .05$). In perspicuity, we found significant differences between VD and TD (*difference* $= 0.65$, $p = .046$).

7 QUALITATIVE RESULTS

In this section, we report the results of the questionnaire analysis. We analyze the open-ended questions that we asked the participants after the interaction. For the list of questions, the reader is referred to Appendix A.

For our analysis, we used thematic analysis [13] and followed an open-coding approach [33]. Two authors of the paper were the coders. One coder developed a code dictionary by reviewing all transcripts. The coders then discussed it and agreed on a final dictionary with a total of 67 codes in 11 code categories (see Appendix B.1). The coders applied the codes to all transcripts independently. The agreement rate of the coders was 91%. To determine the interrater reliability, we calculated Cohen's κ, which is 0.894 referring to "almost perfect agreement" [22]. Finally, the results were discussed, and final code allocations for all transcripts were agreed upon.

We commence by presenting overall findings that are *condition-independent* since we did not observe differences in the answers between the different groups. We proceed with findings that are related to the *understandability* of the schemes. Finally, we present findings that are *category-specific*, meaning they differ between the groups. Whenever possible and meaningful, we provide quotes from the participants.

7.1 Condition-Independent Findings

In general, the answers regarding the specific reasons for using the presented verification scheme did not differ among the different groups except for why participants considered verification ineffective.

In the following, we report four categories of findings that were present in each group: 1) reasons for (not) adopting verification, 2) general trust perceptions, 3) general misconceptions, and 4) requesting independent software.

7.1.1 Reasons for (Not) Adopting Verification. We asked the participants whether they would use the presented verification mechanism to verify their vote in a real election. Participants that answered this question affirmatively gave the following three main reasons: 1) transparency that guarantees the integrity of the vote, 2) the ease of verification, and 3) they consider it as a duty. We continue by explaining these reasons in detail.

The participants pre-dominantly stated that the *transparency* which is gained by verifying the vote guarantees a vote integrity. In particular, the participants want to verify that their votes are cast correctly and for the candidate, they intend to vote for because they use an electronic system. For instance, P101 in the *CS* condition stated:

> *"I want to make sure that my vote indeed reached the ballot box."* (P101, CS)

Similar statements were given in the other groups, e.g.:

> *"I want to check whether everything functioned correctly."* (P751, AC)

Second, the participants commented on the *ease of verification*. They did not perceive it as a burden and considered a trade-off of usability and security, resulting in the conclusion that the ease of use enables this security feature:

> *"The app is easy to use and results in a feeling that my vote was stored correctly."* (P019, AC)

Some participants consider verification as a *duty* for voters because each voter should contribute to ensuring the integrity of an election:

> *"Because it is my duty as a voter to do this. As you can see a software that is made by a human can make mistakes or could be influenced by a third party [...]"* (P264, SD)

Now that we presented the three common reasons to adopt verification, we continue by providing four common reasons to abstain from it: 1) verification is only required once for familiarization, 2) participants trust the software and authorities, 3) they question the necessity, and 4) participants question the effectiveness of verification.

Participants in all conditions reported that they would use the verification feature only for *familiarization* purposes to audit once. In further usages, they do not consider verification required anymore because they already convinced themselves that the system works correctly.

> *"I might use it for my first Internet election, but once I'm familiar with the system and there's no evidence of programming errors or data entry errors in past elections, I'd drop the check."* (P636, AC)

Participants also provided reasons for not wanting to verify their votes at all. The most prominent finding was that participants consider verification to be not needed for several reasons. The first reason is a general *trust* in the provided software and the authorities that run the election. If a software is used in official elections, participants think that this particular software should be secure without the need for additional verification by the voters:

> *"I don't see a reason. It [the software] is used for elections and should, hence, fulfill software standards. Because of that, I don't have to carry out additional checks."* (P006, CS)

Based on that, participants also *questioned the necessity* of verification since it is not possible in other types of elections, such as postal voting:

> *"I carefully checked where I marked the ballot. If I use paper, I can't do that [verify] so why should I do that here."* (P313, AC)

Also, the *effectiveness of verification* was questioned. In general, participants were not convinced that verification could reveal incorrect votes since the verification mechanism might either be not executed properly or an attacker might control it:

> *"The display of the checking mechanism could be manipulated too."* (P361, TD)
>
> *"Voters can simply click the buttons without checking anything."* (P812, CS)

7.1.2 General Trust Perceptions. We also asked the participants whether they are convinced that their cast vote indeed corresponds to their intention. Participants that answered these questions affirmatively reported that a *confirmation via verification* convinces them that their votes are cast correctly:

> *"The data security is guaranteed by the provided confirmation [verification] methods."* (P670, SD)

On the order hand, several participants reported that they are not convinced that their cast vote indeed corresponds to their intention based on 1) the possibility that only the verification is manipulated and 2) intransparency of the voting software.

Participants reported that even if they can verify their votes, there is a possibility of a *manipulated verification* falsely showing that their vote is correct. This basically echoes the results of the adoption aspects:

> *"I believe that a sophisticated manipulation could also change the audit system."* (P703, AC)

The participants also reported a mistrust based on the *intransparency* of the software. In general, the purpose of verification-related tasks was unclear, and therefore participants struggled to base their trust on them:

> *"I don't know how the system internally works and how it counts my vote."* (P462, TD)

7.1.3 General Misconceptions. During our analysis, we found two distinct misconceptions that the participants reported in all groups. First, participants thought that verification could break their vote privacy, and second, they thought that the verification was only to check whether they themselves made a mistake.

Participants thought that *verification breaks vote privacy*. Since the votes should be private, but their contents are inspected, this inspection might have negative precautions. While this is true for *AC* schemes, the vote privacy is only broken towards the verification software in all other conditions. For instance, P876 in the *AC* condition gave a statement that represents the participants' impressions quite well:

> *"The following problem still remains: I am who I should be - only if I am personally identified. But that contradicts the secret ballot. Can I be sure that I have been identified as 'I' without my identity ending up in my vote, or that the vote cannot be traced back to me? If this is not the case, however, then even large quantities of forged votes can be introduced into the system undetected."*

When asked to explain the purpose of vote verification, many participants answered that this was to check whether they themselves made a mistake. Thus, they mistakenly thought that verification is *only to check for human errors*. We implemented a vote confirmation screen in each voting

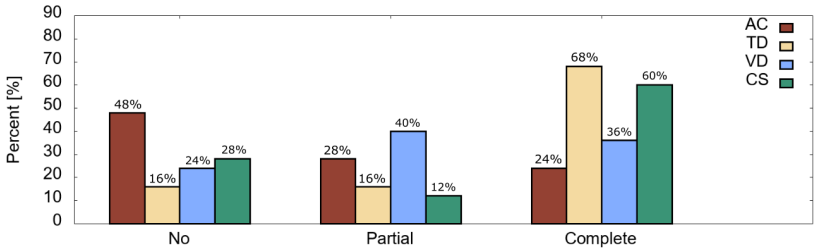


Fig. 6. Understanding of verification demonstrated by our sample.

software for this purpose. While verification could be seen as a double-check for the voters, they indeed verify the technology and not themselves:

> *"It's for me to check whether I indeed click on the correct column."* (P1, CS)

7.1.4 Requesting Independent Software. In *VD*, *AC*, and *TD*, which are the conditions with mobile apps to carry out verification, the participants requested the possibility to access the verification data from an independent software. P1607 from the *VD* group gave a representative statement:

> *"The provided app is okay, but why should I use this specific software. I would expect that there are several ones where I can choose from or even on option to implement my own one."*

7.2 Understandability-Specific Findings

After the interaction, we asked the participants to explain the general purpose of verification in their own words. In particular, we asked what they checked in their understanding.

We assigned one of the following three codes to the answers:

(1) *no understanding*: the participant gave none or an incorrect explanation.
(2) *partial understanding*: the participant's answer was correct but incomplete. For instance, if a participant only mentioned to verify the content of the vote but not whether it reached the electronic ballot box.
(3) *complete understanding*: the participant gave a complete answer in their own words.

An overview of the distribution of the codes in the individual conditions is given by Figure 6.

Participants in the *AC* condition demonstrated the lowest level of understanding. In their answers, participants wrote that it either is not possible to verify at all ($N=8$) or that verification serves to check whether the voters made a mistake ($N=4$):

> *"[To check that] you entered your vote correctly and that you didn't make any mistakes."* (P428, AC)

This is followed by the *VD* condition in which two participants mistakenly thought that they could verify whether their vote is included in the tally:

> *"The correctness of the vote whether it was indeed tallied."* (P684, SD)

Most participants in the *CS* condition demonstrated a partial understanding which was restricted to the registration in the ballot box but did not mention the correctness of the vote:

> *"I can check whether my vote reached the server."* (P118, CS)

Most of the participants in the *TD* condition demonstrated a complete understanding ($N = 17$):

> *"I can check the correctness of my cast vote and the result of the election because all votes are available in an anonymized way."* (P417, TD)

7.3 Condition-Specific Findings

The participants also gave answers that were different between the investigated conditions. We report these answers grouped by the condition that was investigated.

7.3.1 Audit-or-Cast (AC). When asked about trust, the participants in the *AC* condition expressed that the *usage of the second device* enhances their trust since it is harder to compromise both devices:

> *"Based on the QR-code on the website and the correct display on the smartphone I am certain that my vote was cast error-free."* (P701, AC)

Participants questioned the *effectiveness of verification* since the content of the cast vote cannot be verified. Based on that, they would refrain from using the challenge to verify their vote. This observation is not novel and confirms results from previous studies (e.g., [66]). The fact that the content of a cast vote cannot be verified also impacted the trust in *AC* schemes. Participants expressed not to be convinced that the cast ballot matches their intention if they cannot verify it.

A further misconception of *AC* expressed was that the participants thought *verification is impossible*. Three participants even demanded means to verify their votes, and two stated that the missing verification option impacts their trust:

> *"In principle, of course, I could do one check, but that was only a check for my first ballot. After that, I had to select a new one, and a review would have forced me to submit a new one. A continuation after the checking is missing here, in my opinion, therefore, I cannot examine my actually delivered choice. This leads to a high degree of uncertainty."* (P601, AC)

We also asked participants in this condition how often they would challenge the voting software before casting a vote since this is necessary for security purposes. The majority of participants ($N = 19$) prefers one verification only and questions the *necessity of multiple verifications*:

> *"Checking multiple times does not provide more security."* (P636, AC)

7.3.2 Tracking Data (TD). Participants using *TD* expressed *concerns about vote privacy*. In particular, since the vote on the bulletin board has on individual tracking code, participants thought that the tracking code might be linked to their identity:

> *"I am concerned that my vote can be linked to my identity."* (P302, TD)

In this condition, we asked the participants about the timing of the verification. Using *TD*, the verification is carried out after the announcement of the tally result, which could be up to two weeks after the election. Participants stated that they would carry out verification even at this late stage for *security purposes*.

> *"The effort is good since we can guarantee a fair and secure election based on it."* (P322, TD)

On the other hand, participants expressed *trust in official data* meaning that if the result has been announced, there is no need for further checking:

> *"I trust the official results, thus there is no need to use the app after two weeks."* (P297, TD)

Two participants expressed a *concern about the late verification*. Since the results are already published, voters that do not like the results or forgot what they voted for might use the verification feature to scrutinize the election based on false evidence:

> *"Could it be that people change their mind after the election result (or get insecure by people in the meantime) and then when checking the vote is no longer sure if that is really their vote and then contact support?"* (P297, TD)

7.3.3 Verification Device (VD). Similarly to the *AC* condition, participants in the *VD* condition expressed that the *usage of the second device* enhances their trust.

Since *VD*-based systems often limit the time of vote verification to mitigate voter coercion, we asked the participants about a verification limit. Therefore, we used the 30-minute limit from the Estonian system [46] and asked if they consider such a limit appropriate. The overwhelming majority expressed to *verify directly after voting* and, therefore, the limit would be acceptable to them:

> *"I only deal briefly with voting, then [I] check directly whether I have ticked the right one and what is done with my vote afterward I cannot check anyway."* (P935, SD)

Furthermore, the participants stated that *voting is absolute* using other voting channels such as paper and, therefore, such a limit represents this absolution:

> *"In the paper election it is also not possible."* (P1372, SD)

7.3.4 Code Sheets (CS). In the *CS* condition, we asked about receiving the code sheets since they have to be distributed before the election, making it impossible to make a spontaneous decision to vote online. The majority of participants said that based on the *similarity of postal voting* they consider the need to receive materials as acceptable:

> *"It's just like postal voting."* (P012, CS)

However, three participants stated that they wish to receive the code sheets via another channel, for instance, by e-mail:

> *"Then, I could do postal voting. If the system is Internet-based, why not doing everything online."* (P518, CS)

This shows that participants have to be informed that the complete code sheet content cannot be shared with the voting software and thus would have to be received on a separate device that is trusted.

8 DISCUSSION AND IMPLICATIONS

This section discusses the findings from our user study based on the categories of our proposed categorization. We conclude with final recommendations and directions for future studies.

8.1 User Study Results and Categorization

In this section, we discuss the individual categories based on our obtained study results and related work.

The effectiveness of detecting incorrect votes is crucial since it directly contributes to the security of the voting scheme. If incorrect votes are not detected, the integrity of the election result cannot be guaranteed. All categories of voting schemes in our proposed categorization contain mental tasks. These are tasks that cannot be measured directly, and as a proxy measure, we introduced the deliberate manipulation of votes. Thus, within the study, we changed the voting option marked by the participant in one contest to a random other one.

8.1.1 Audit-or-Cast. The participants detected only 28% of the incorrect votes in the *AC* condition. Our analysis indicates that using *AC* impacts the detection of incorrect votes. Other studies of *AC*-based schemes had various results for effectiveness from 43% [1], up to 81% [66]. However, these studies used another definition of effectiveness and evaluated the completion without specifically checking mental tasks. Consequently, our results reveal more accurate information on whether the participants indeed compared the voting options rather than just clicking through the procedure without paying attention to them.

By reviewing the screen-recordings and timestamps in our study, we determined that participants indeed clicked through the individual steps of verification but did not perform the mental tasks. 68% of all participants correctly clicked through the procedure without reporting an incorrect vote. 30% attempted verification without completing it, and 12% did not attempt verification at all. Thus, 30% of the participants did not detect the incorrect voting options displayed to them by the verification software. This confirms hints from previous studies of audit-or-cast [1, 62, 66]. The low rate of 28% is particularly alarming and shows that *AC*-based schemes are unlikely to be appropriate for elections outside expert communities. Furthermore, *AC* received the lowest SUS score of 73.00 among all tested schemes, and this was also significantly lower than *CS* (84.50) and *VD* (84.60).

The evaluation of the qualitative data further indicated that participants struggled the most with *AC*-based schemes. They questioned the schemes' effectiveness in detecting incorrect votes since the challenge-based approach did not align with their mental models of verification. Therefore, this category might be difficult to deploy in elections with voters without expert knowledge.

Another important aspect of *AC*-based schemes is their probabilistic nature. It must not be predictable whether the voters are going to cast or to verify. Our study results confirm that voters prefer to verify once indicating that additional information must be provided such that voters understand why it is advisable to verify multiple times. If voters verify only once, the verification mechanism loses effectiveness since the software could successfully cheat after the verification.

Furthermore, Culnane *et al.* investigated the *AC* scheme Benaloh Challenge [10, 11] from a game-theoretic perspective [26]. If the voting device has prior knowledge about the voter in Internet voting, the effectiveness of the challenge is weakened, and has to be adjusted to deliver the required effectiveness. Instead of a single encrypted vote, several should be prepared at once, and the voters choose which to cast and which to verify [26]. Considering the results from our user study, it is questionable whether the voters would indeed verify multiple times.

8.1.2 Verification Device. In the *VD* condition, the participants detected 64% of incorrect votes. Examining the screen-recordings showed that the remaining 36% of participants clicked through the procedure but did not pay attention to the voting options. This was even though the verification app, like all verification apps in our study, contained two buttons for proceeding with explicit statements about the codes being (in-)correct based on recommendations from related work [68]. Hence, further investigation of the interface design might be required since the comparison of the voting options cannot be automated. Participants in our study also questioned the effectiveness of verification because it is possible to "simply click the buttons".

Estonia uses a *VD* scheme for vote verification. Based on data published by the Estonian authorities, on average 4% of voters verify [32]. Verification based on *VD* is optional, and as the case of Estonia shows, only a small share of voters verifies. While we do not want to speculate why the share of voters is so low, one of the reasons may be that verification is optional.

The voters welcomed the usage of the second device, which also contributed to a feeling of security and to trusting the verification process. This represents the correct reason why such a second device is used in the first place and shows that even if such a second device is used, voters can handle verification with it. Furthermore, the execution time overall was the lowest.

To transfer the vote identifier from the voting software to the verification app, we used a QR-code similarly to the Estonian system [46]. Previous studies of QR-code scanning showed that scan reliability is dependent on the screen [66]. While we did not experience such problems in our study, interfaces that rely on QR-code scans should also provide means to adjust the code's size. QR-codes are susceptible to man-in-the-middle attacks. Humans cannot read the information encoded by QR-codes. As a result, humans cannot judge whether a QR-code indeed encodes the information required by them [56, 92]. Such attacks might impact schemes that rely on QR-codes.

VD schemes require a device that is different from the voting device to carry out verification. Therefore, voters without such a device might be disadvantaged. Thus, verification based on *VD* should only be deployed if each voter has at least access to such a verification device. However, even if each voter owns such a device, many vendors allow the synchronization of apps among different platforms. Konoth *et al.* have shown that such a synchronization can impact security [57]. This might require additional precautions.

8.1.3 Tracking Data. Using *TD*, 84% of incorrect votes were detected. The provided verification app highlights the tracking code for the voter. In a prior study of the *TD* scheme Selene, the voters could complete all verification steps [29]. However, the authors did not specifically evaluate the mental tasks. This might explain the different effectiveness results of 84% in our study. Similar to the *AC* and *VD*, the voters completed the procedure and likely did not pay attention to the tracking code. This highlights the importance of the mental task since verification is not effective without the comparison.

The tracking codes in *TD* schemes can either be generated by a trusted entity or by the voters themselves. Although we specifically investigated the code generation by a trusted entity, related work showed that humans perform very poorly in random number generation [86]. Furthermore, voters might accidentally include personal information into the codes or choose predictable codes. Thus, the code generation by the voter should be avoided.

Different from all other categories, the cast vote is verified *after* the tally. This means voters have to visit the bulletin board after the election result has been announced. While voters in our study were generally positive about verifying their votes even after two weeks, it should be investigated further whether voters would indeed verify. As mentioned by some participants, other aspects might impact verification after the tally. Voters might dislike the result and complain based on false evidence or be uncertain how they voted.

8.1.4 Code Sheets. Using *CS* all incorrect votes were detected confirming previous studies [62, 68]. Our analysis indicates that using *CS* impacts the detection of incorrect votes. *CS* schemes form the only category of schemes in which vote casting is impossible without verification, meaning that verification is mandatory. In our implementation, after inspecting the return codes, the voters had to enter a confirmation code to insert their vote into the electronic ballot box. This integration of the verification into the voting process might explain the high detection rate.

Participants using *CS* needed significantly longer for completion than any other category. Voting and verification took on average 10.5 minutes compared to 3.5 minutes in the *VD* scheme, which was fastest.

The UEQ scale of efficiency assesses the perceived efficiency, and here we could not find any statistical differences. This indicates that even if voting and verifying took longer than ten minutes, the participants still considered it efficient.

To carry out verification based on *CS*, the voters need auxiliary material, which must be generated and distributed before the election over a trusted channel. This introduces some restrictions into this category. Without a code sheet, voters cannot participate. Thus, the authorities have to spend extra effort to distribute and organize the material. This consumes more resources than the other categories that rely on the software distribution only (given the assumption that each voter has access to two devices).

8.1.5 Summary. Investigating human factors showed that *VD*, *TD*, and *CS* schemes are applicable for real elections while *AC* schemes are rather for expert communities. The choice of the specific scheme is dependent on the voting infrastructure, the availability of devices, and the trusted transmission channel.

8.2 Security – Usability Tradeoff

Within the description of our proposed categorization, we provided information about trust assumptions. It is not easy to judge whether trust assumptions can indeed be met outside the context of a specific election. Thus, we discuss the connection of the trust assumptions to human factors based on our findings.

8.3 Trusted Voting and Verification Devices

The goal of individual verifiability is to ensure the integrity of the election result by detecting incorrect votes. This is particularly challenging in Internet voting because of the secure platform problem [38]. Since the number of malware and infected devices is constantly increasing (cf. [48]), it should not be assumed that voting devices can be trustworthy (A1). This trust assumption is made by *AC*, *TD*, and *VD* schemes if voting devices are used for verification[7].

As stated above, if a second device is used for verification, the voting devices do not have to be trusted if the verification device is trustworthy (A2). This shifts trust from the voting to the verification devices. Even if both devices were infected by malware, the malware would have to be synchronized [57]. Thus, voters have to be informed that device synchronization might not be advisable.

Interacting with two devices for fulfilling one task is also utilized in related domains. The most prominent related domain is two-factor authentication (2FA). Our findings about verification devices relate to apps for 2FA in the following way. Our study participants handled two devices and did not experience any issues when scanning the QR-codes that encode the vote identifiers. This confirms investigations of 2FA [27, 60].

However, considering efficiency, our findings differ from 2FA since our participants did not perceive the procedure as too time-consuming. Online transactions have become part of daily life, while participating in an election is rare. Furthermore, our participants had no experience with Internet voting. Consequently, the tradeoff between security and usability is evaluated differently in voting and 2FA. As a result, we can conclude that a second device can be a viable solution to enhance the security of verification.

8.4 Central Trusted Entities

In each category, a central entity is required to be trustworthy. This is either the bulletin board (*AC*, *TD*, *DE*) or the electronic ballot box (*VD*, *CS*). The voting authorities could either control those entities or provide a verifiable component [71], such that targeted attacks against this infrastructure are more likely to be detected.

From the authorities' perspective, it can be decided whether the trust assumptions indeed hold similar to in-person or postal voting. From the voters' perspectives, however, trust in the authorities is required. In this context, *TD* schemes have an advantage over the other categories since those schemes also provide universal verifiability – anyone (voters or observers) can verify that the result corresponds to published ballots [80]. Thus, in *TD* schemes, any observer or voter could use the data published on the bulletin board to recalculate the election result independently. The other schemes require additional protocols to provide universal verifiability.

8.5 Trusted Transmission Channel

Code sheets performed best in terms of effectiveness, and verification is mandatory. However, a trusted transmission channel is required to deliver the code sheets to the voters before the election. Switzerland used code sheets based on the protocol of Neuchâtel [36] and delivered the code sheets

[7]Note, if voting devices are trustworthy, it is still possible to verify the voting software and the transmission channel.

via postal mail [76]. Several countries already permit postal voting and consequently have an infrastructure for delivering ballots. Therefore, postal mail might be a viable solution as a trusted transmission channel.

8.5.1 Summary. The context of a specific election is required to judge to which extend trust assumptions can be met. There are possibilities to assure trust assumptions, although a trusted voting device is unlikely. *AC*-based schemes demonstrated a rather high tradeoff between security and usability since voters have to be educated enough to carry out verification, and they have to do it multiple times. On the other hand, there is only a low tradeoff in the scope of *VD*, *TD*, and *CS* schemes. This shows that, in general, all of these schemes are applicable to real elections if their trust assumptions can be met.

8.6 Recommendations

Based on the study results and the security-vs-usability tradeoff, we conclude with seven recommendations for deploying individually verifiable schemes in elections for the developers and policymakers of Internet voting systems. Based on our investigation and the deployment requirements of the schemes, we cannot recommend a specific class of schemes in general. Developers and policymakers have to fit the verification mechanism to the election and decide on a mechanism based on the election's individual requirements.

(1) **Provide information about why verification is needed.** Participants in our study reported abstaining from verification because they viewed it as an extra task and could not determine why it is needed. Therefore, information on why verification is advisable and what it means should be available to voters, especially if verification not mandatory. This recommendation targets information that should be available to voters. In our study, we only provided information in the voting and verification software. However, it is crucial to provide information on different sources.

(2) **Provide information on vote privacy.** Some participants would abstain from verification because they fear that their vote privacy might be compromised. Therefore, information on how the verification mechanism preserves vote privacy should be available to voters. This recommendation also targets the information that should be available to voters. Again, in our study, we only provided information in the software. Information about vote privacy should be available here as well as on independent sources.

(3) **Verify the cast vote.** Based on the observations in our user study, we recommend using a verification scheme that is *not* based on audit-or-cast. Since verifying the cast vote better aligns with the voters' expectations, a scheme that enables this should be used. Not verifying the cast vote might lead voters to skip verification and question its necessity. This recommendation specifically targets the choice of the verification protocol.

(4) **Consider the impact of mandatory verification.** Voting is the primary task of the voters. Verification is only mandatory in *CS* schemes. Therefore, policymakers should decide whether they want verification to be carried out by all voters. If so, they should opt for a *CS*-based scheme. If not, they can opt for *TD* or *VD*. Based on our results, mandatory verification in *CS* schemes offers the best effectiveness. This recommendation specifically targets the choice of the verification protocol.

(5) **Consider the timing of verification.** Verification can be either performed during voting (*CS*), within a timeframe after voting (*VD*) or after the election's result have been announced (*TD*) (see also Figure 1). Therefore, policymakers should decide at what time they want the voters to carry out verification. The closer verification is to the time of voting, the more likely voters perform the task. This recommendation specifically targets the choice of the verification protocol.

(6) **Minimize human effort.** Participants in our study expected verification to happen automatically[8]. In general, it is neither completely possible nor advisable because it lowers the control of voters. We argue that the verification software should assist the voters in as many tasks as possible and guide them through the verification process. This recommendation specifically targets the choice of the user interface and process.

(7) **Provide an expert mode.** Voters might be familiar with verification or even wish to use their own software. Therefore, it should be possible to access the data needed for verification to use own verification software. This recommendation targets information that should be available to voters.

8.7 Limitations and Opportunities for Future Investigations

In this section, we reflect on the limitations of our investigation and discuss opportunities for future investigations.

8.7.1 Focus on Internet Voting. A first limitation to be mentioned is the focus on Internet voting. Individual verifiability is not limited to Internet voting schemes, it also available for polling station voting (cf. [77]) or postal voting (cf. [12]).

8.7.2 Investigated Sample. As a second limitation, one can argue that the sample of our user study was rather young. Consequently, our results might not be representative. However, our sample reflects the group of people that is most interested in Internet voting in the authors' country[9]. Future studies should investigate a more diverse sample that also focuses on groups of people that are not interested in Internet voting but would, for instance, use it when they are on vacation.

8.7.3 Realization of Schemes in the Study. Although our study was informed by the findings of existing investigations from the literature to realize the interfaces [1, 29, 52, 66, 67, 72, 74, 87, 93], verification apps [66, 72] and code sheets [72], a possible limitation is that the results have been impacted by the information that we presented to the participants. To mitigate this, all investigated interfaces shared common terminology, design and were tested in pilot studies. However, the information given in the interfaces could have impacted our findings.

When designing the voting interfaces for the individual voter interactions, we built upon previous work on code comparisons [28, 83]. However, we only investigated the verification process as a whole and did not capture results about individual voter interactions. Future work should investigate individual voter interactions and their impact on the overall verification process.

8.7.4 Improving code voter interactions. The categories audit-or-cast, tracking data, and code sheets contain voter actions in which the voter has to interact with a code. This code could either be a

[8]This is also realized in the sElect scheme [64], but it comes at the cost of not providing receipt-freeness.
[9]Reference removed for anonymity.

tracking code, a vote identification code, or a code listed on a code sheet. Such codes are security-critical parameters, and their length is required for sufficient security. Furthermore, codes can be complex strings consisting of numbers, letters, and sometimes symbols. Although participants in our study that interacted with the strings were generally able to compare them, the scalability of such comparisons is questionable.

8.7.5 Investigating scalability. As the voting scenario, we investigated a governmental election with two contests. The first contest had eight candidates, and the second one had 18. Referenda and elections might have more contests. Therefore, our study does not consider scalability aspects connected to the number of contests and candidates. In general, the visual comparison of complex strings is known to be problematic [28, 83], therefore, the improvement of schemes that require the comparison of many strings constitutes a particular challenge.

8.7.6 Alternatives to AC schemes. The *AC* category has been investigated by related work. We confirm but also extend these results. *AC* schemes performed worst in our investigation, and even good usability cannot mitigate the challenging nature of the schemes. Therefore, instead of further refining *AC* schemes, the focus should be on viable alternatives.

8.7.7 Delegating verification. Participants in our study also mentioned that verification should happen automatically. While it is not trivial to delegate the verification process, delegation-based schemes, which form the fifth category of our categorization, provide delegation for cast-as-intended verifiability. The recorded-as-cast verifiability cannot be automated since only the voters know that they voted. While we deliberately decided not to evaluate this category due to the fundamental differences in the voting process, we consider it an important part of future studies to investigate the feasibility of the voting process and the voters' perceptions of the delegation. Since voters in our study thought that verification breaks vote privacy, it is important to investigate this in the scope of delegation.

9 CONCLUSION

Individual verifiability provides measures for voters to verify that their votes have not been manipulated during vote casting. In this paper, we proposed a categorization of schemes that aim to provide individual verifiability in Internet voting. Our categorization is based on the voters' perspective, and the individual tasks voters have to carry out by themselves. In particular, we identified the categories of 1) audit-or-cast, 2) supplementary device, 3) tracking data, 4) code sheets, and 5) delegation. We investigated these categories in a comparative user study with 100 participants, where 25 participants interacted with each scheme. We captured quantitative and qualitative data to assess usability, user experience, trust, understandability, and individual user perceptions. Based on the results, we provide recommendations for the developers and policymakers and give directions for future investigations.

Our study indicates that audit-or-cast schemes are worrisome for remote elections because voters only detect 28% of incorrect votes. Furthermore, audit-or-cast schemes do not align with the voters' mental models resulting in voters questioning their necessity. Code sheet schemes supported the voter to detect all incorrect votes and require them to verify to cast a vote. However, the voters needed the longest when interacting with them, and the code sheets have to be distributed before the election over a trusted channel, which adds some organizational overhead. Supplementary devices and tracking data schemes had a better detection of incorrect votes than audit-or-cast. They rely on software for voting and verification. Thus, the overhead is lower than code sheet schemes, but verification is an extra task that might be skipped by a large share of voters.

ACKNOWLEDGMENTS

This research work has been co-funded by the Horst Görtz Foundation, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 251805230/GRK 2050, and by JST CREST Grant No. JPMJCR16E1. Further, the research was co-funded by the BMBF and the Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. The authors would further like to thank Christoph Schüßler, Felix Lange, Kira Bleck, Verena Zimmermann and Bane Janjus who supported the data acquisition of the study.

REFERENCES

[1] Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach. 2014. Usability of Voter Verifiable, End-to-End Voting Systems: Baseline Data for Helios, Prêt à Voter, and Scantegrity II. *The USENIX Journal of Election Technology and Systems* 2, 3 (2014), 26–56.

[2] B. Adida. 2006. *Advances in Cryptographic Voting Systems.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[3] Ben Adida. 2008. Helios: Web-based Open-Audit Voting. In *Proceedings of the USENIX Security Symposium (USENIX Security)*, Vol. 17. USENIX Association, Berkeley, CA, USA, 335–348.

[4] Ben Adida, Olivier De Marneffe, Olivier Pereira, Jean-Jacques Quisquater, et al. 2009. Electing a University President using Open-Audit Voting: Analysis of Real-World use of Helios. In *Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT '09)*. USENIX Association, Berkeley, CA, USA, 1–15.

[5] Jordi Puiggalí Allepuz and Sandra Guasch Castelló. 2011. Internet Voting System With Cast As Intended Verification. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer, Cham, Switzerland, 36–52. https://doi.org/10.1007/978-3-642-32747-6_3

[6] Jordi Puiggalí Allepuz and Sandra Guasch Castelló. 2012. Cast-as-Intended Verification in Norway. In *Proceedings of the 5th International Conference on Electronic Voting (EVOTE)*. Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn, Germany, 49–63.

[7] Arne Ansper, Sven Heiberg, Helger Lipmaa, Tom André Øverland, and Filip Van Laenen. 2009. Security and Trust for the Norwegian E-voting Pilot Project E-valg 2011. In *Proceedings of the Nordic Conference on Secure IT Systems (NordSec)*. Springer, Cham, Switzerland, 207–222. https://doi.org/10.1007/978-3-642-04766-4_15

[8] Davide Balzarotti, Greg Banks, Marco Cova, Viktoria Felmetsger, Richard Kemmerer, William Robertson, Fredrik Valeur, and Giovanni Vigna. 2008. Are Your Votes Really Counted?: Testing the Security of Real-world Electronic Voting Systems. In *Proceedings of the International Symposium on Software Testing and Analysis* (Seattle, WA, USA) *(ISSTA '08)*. ACM, New York, NY, USA, 237–248. https://doi.org/10.1145/1390630.1390660

[9] D. Balzarotti, G. Banks, M. Cova, V. Felmetsger, R. Kemmerer, W. Robertson, F. Valeur, and G. Vigna. 2010. An Experience in Testing the Security of Real-World Electronic Voting Systems. *IEEE Transactions on Software Engineering* 36, 4 (July 2010), 453–473. https://doi.org/10.1109/TSE.2009.53

[10] Josh Benaloh. 2006. Simple Verifiable Elections. In *Proceedings of the Electronic Voting Technology Workshop (EVT) (EVT)*. USENIX Association, Berkeley, CA, USA, Article 5, 10 pages.

[11] Josh Benaloh. 2007. Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. *Electronic Voting Technology Workshop* EVT '07 (2007).

[12] Josh Benaloh, Peter Y.A. Ryan, and Vanessa Teague. 2013. Verifiable Postal Voting. In *Proceedings of the Cambridge International Workshop on Security Protocols*. Springer, 54–65.

[13] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[14] Achim Brelle and Tomasz Truderung. 2017. Cast-as-Intended Mechanism with Return Codes Based on PETs. In *Proceedings of the International Joint Conference on Electronic Voting (E-Vote-ID)*. Springer, Cham, Switzerland, 264–279. https://doi.org/10.1007/978-3-319-68687-5_16

[15] Ian Brightwell, Jordi Cucurull, David Galindo, and Sandra Guasch. 2015. An Overview of the iVote 2015 Voting System. *New South Wales Electoral Commission, Australia, Scytl Secure Electronic Voting, Spain* (2015).

[16] John Brooke. 1996. SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7.

[17] Jurlind Budurushi, Melanie Volkamer, Oksana Kulyk, and Stephan Neumann. 2017. Nothing Comes for Free: How Much Usability Can You Sacrifice for Security? *IEEE Security & Privacy Special Issue on Electronic Voting* (2017). https://doi.org/10.1109/MSP.2017.265093646

[18] Sergiu Bursuc, Gurchetan S. Grewal, and Mark D. Ryan. 2011. Trivitas: Voters Directly Verifying Votes. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer, Cham, Switzerland, 190–207. https://doi.org/10.1007/978-3-642-32747-6_12

[19] Pyrros Chaidos, Véronique Cortier, Georg Fuchsbauer, and David Galindo. 2016. BeleniosRF: A Non-interactive Receipt-Free Electronic Voting Scheme. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. ACM, New York, NY, USA, 1614–1625. https://doi.org/10.1145/2976749.2978337

[20] David Chaum. 2001. Surevote: Technical Overview. In *Proceedings of the Workshop on Trustworthy Elections (WOTE)*.

[21] Nikos Chondros, Bingsheng Zhang, Thomas Zacharias, Panos Diamantopoulos, Stathis Maneas, Christos Patsonakis, Alex Delis, Aggelos Kiayias, and Mema Roussopoulos. 2016. D-DEMOS: A Distributed, End-to-End Verifiable, Internet Voting System. In *Proceedings of the 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Piscataway, NJ, USA, 711–720. https://doi.org/10.1109/ICDCS.2016.56

[22] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.

[23] Véronique Cortier, Alicia Filipiak, and Joseph Lallemand. 2019. BeleniosVS: Secrecy and Verifiability Against a Corrupted Voting Device. In *Proceedings of the IEEE Computer Security Foundations Symposium, CSF*. IEEE, Piscataway, NJ, USA, 367–381. https://doi.org/10.1109/CSF.2019.00032

[24] Véronique Cortier, Georg Fuchsbauer, and David Galindo. 2015. BeleniosRF: A Strongly Receipt-Free Electronic Voting Scheme. *IACR Cryptology ePrint Archive* 2015 (2015), 629.

[25] Véronique Cortier, David Galindo, Ralf Küsters, Johannes Mueller, and Tomasz Truderung. 2016. SoK: Verifiability Notions for E-voting Protocols. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE, 779–798.

[26] Chris Culnane and Vanessa Teague. 2016. Strategies for Voter-Initiated Election Audits. In *Proceedings of the International Conference on Decision and Game Theory for Security (GameSec)*. Springer, Cham, Switzerland, 235–247. https://doi.org/10.1007/978-3-319-47413-7_14

[27] Emiliano De Cristofaro, Honglu Du, Julien Freudiger, and Greg Norcie. 2014. A Comparative Usability Study of Two-Factor Authentication. In *Proceedings of the Workshop on Usable Security (USEC '14)*. Internet Society, Reston, VA, USA, 10. https://doi.org/10.14722/usec.2014.23025

[28] Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. 2016. An Empirical Study of Textual Key-Fingerprint Representations. In *Proceedings of the USENIX Security Symposium*. USENIX Association, 193–208.

[29] Verena Distler, Marie-Laure Zollinger, Carine Lallemand, Peter B. Roenne, Peter Y. A. Ryan, and Vincent Koenig. 2019. Security - Visible, Yet Unseen?. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 605, 13 pages. https://doi.org/10.1145/3290605.3300835

[30] Alex Escala, Sandra Guasch, Javier Herranz, and Paz Morillo. 2016. Universal Cast-as-Intended Verifiability. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*. Springer, Cham, Switzerland, 233–250. https://doi.org/10.1007/978-3-662-53357-4_16

[31] Estonian National Electoral Committee. 2010. E-Voting System General Overview. http://www.vvk.ee/public/dok/General_Description_E-Voting_2010.pdf. [Online; accessed 28-March-2019].

[32] Estonian National Electoral Committee. 2015. Statistics about Internet Voting in Estonia. https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia. [Online; accessed 28-November-2019].

[33] Uwe Flick. 2018. *An introduction to qualitative research*. Sage Publications Limited.

[34] International Institute for Democracy and Electoral Assistance. 2015. ICTs in Elections Database. https://www.idea.int/data-tools/data/icts-elections. [Online; accessed: 12-October-2019].

[35] Kristin Skeide Fuglerud and Till Halbach Røssvoll. 2012. An Evaluation of Web-Based Voting Usability and Accessibility. *Universal Access in the Information Society* 11, 4 (2012), 359–373. https://doi.org/10.1007/s10209-011-0253-9

[36] D. Galindo, S. Guasch, and J. Puiggalí. 2015. 2015 Neuchâtel's Cast-as-Intended Verification Mechanism. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer, Cham, Switzerland, 3–18. https://doi.org/10.1007/978-3-319-22270-7_1

[37] Dawid Gaweł, Maciej Kosarzecki, Poorvi L Vora, Hua Wu, and Filip Zagórski. 2016. Apollo–End-to-End Verifiable Internet Voting with Recovery from Vote Manipulation. In *International Joint Conference on Electronic Voting (E-Vote-ID)*. Springer, Cham, Switzerland, 125–143. https://doi.org/10.1007/978-3-319-52240-1_8

[38] Ed Gerck, C. Andrew Neff, Ronald L. Rivest, Aviel D. Rubin, and Moti Yung. 2001. The Business of Electronic Voting. In *Proceedings of the International Conference on Financial Cryptography (FC) (FC)*. Springer, Cham, Switzerland, 243–268. https://doi.org/10.1007/3-540-46088-8_21

[39] Kristian Gjøsteen. 2011. The Norwegian Internet Voting Protocol. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer, Cham, Switzerland, 1–18. https://doi.org/10.1007/978-3-642-32747-6_1

[40] Laurent Grégoire. 2012. Comment mon ordinateur a vote à ma place, 2012.

[41] G. S. Grewal, M. D. Ryan, L. Chen, and M. R. Clarkson. 2015. Du-Vote: Remote Electronic Voting with Untrusted Computers. In *Proceedings of the 28th Computer Security Foundations Symposium (CSF)*. IEEE, Piscataway, NJ, USA, 155–169. https://doi.org/10.1109/CSF.2015.18

[42] Sandra Guasch and Paz Morillo. 2016. How to Challenge and Cast Your E-Vote. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*. Springer, Cham, Switzerland, 130–145. https://doi.org/10.1007/978-3-662-54970-4_8

[43] Stuart Haber, Josh Benaloh, and Shai Halevi. 2010. The Helios E-Voting Demo for the IACR. International Association for Cryptologic Research.

[44] Rolf Haenni, Reto E. Koenig, and Eric Dubuis. 2016. Cast-As-Intended Verification in Electronic Elections Based on Oblivious Transfer. In *Proceedings of the International Joint Conference on Electronic Voting (E-Vote-ID)*. Springer, Cham, Switzerland, 73–91. https://doi.org/10.1007/978-3-319-52240-1_5

[45] J. Alex Halderman and Vanessa Teague. 2015. The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer, Cham, Switzerland, 35–53. https://doi.org/10.1007/978-3-319-22270-7_3

[46] Sven Heiberg and Jan Willemson. 2014. Verifiable Internet Voting in Estonia. In *Proceedings of the 6th International Conference on Electronic Voting, Verifying the Vote (EVOTE)*. IEEE, Piscataway, NJ, USA, 1–8. https://doi.org/10.1109/EVOTE.2014.7001135

[47] Jörg Helbach and Jörg Schwenk. 2007. Secure Internet Voting With Code Sheets. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer, Cham, Switzerland, 166–177. https://doi.org/10.1007/978-3-540-77493-8_15

[48] AV-Test The Independent IT-Security Institute. 2020. Malware. https://www.av-test.org/en/statistics/malware/, accessed: 31-Mai-2020.

[49] International Organization for Standardization. 1998. ISO 9241-11: Ergonomics of Human System Interaction – Part 11: Guidance on Usability.

[50] International Organization for Standardization. 2010. ISO 9241-210: Ergonomics of Human System Interaction - Part 210: Human-Centred Design for Interactive Systems.

[51] Vincenzo Iovino, Alfredo Rial, Peter B. Rønne, and Peter Y. A. Ryan. 2017. Using Selene to Verify Your Vote in JCJ. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*. Springer, Cham, Switzerland, 385–403. https://doi.org/10.1007/978-3-319-70278-0_24

[52] Fatih Karayumak, Maina M. Olembo, Michaela Kauer, and Melanie Volkamer. 2011. Usability Analysis of Helios - An Open Source Verifiable Remote Electronic Voting System. In *Proceedings of the Conference on Electronic Voting Technology/Workshop on Trustworthy Elections (EVT/WOTE)* (San Francisco, CA). USENIX Association, Berkeley, CA, USA, Article 5, 16 pages.

[53] Shahram Khazaei and Douglas Wikström. 2017. Return Code Schemes for Electronic Voting Systems. In *Proceedings of the International Joint Conference on Electronic Voting (E-Vote-ID)*. Springer, Cham, Switzerland, 198–209. https://doi.org/10.1007/978-3-319-68687-5_12

[54] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. 2015. End-to-End Verifiable Elections in the Standard Model. In *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Springer, Cham, Switzerland, 468–498. https://doi.org/10.1007/978-3-662-46803-6_16

[55] A. Kiayias, T. Zacharias, and B. Zhang. 2017. An Efficient E2E Verifiable E-voting System without Setup Assumptions. *IEEE Security Privacy* (2017), 1–1. https://doi.org/10.1109/MSP.2017.265093752

[56] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, and Edgar Weippl. 2010. QR Code Security. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia* (Paris, France) *(MoMM '10)*. Association for Computing Machinery, New York, NY, USA, 430–435. https://doi.org/10.1145/1971519.1971593

[57] Radhesh Krishnan Konoth, Victor van der Veen, and Herbert Bos. 2016. How Anywhere Computing Just Killed Your Phone-Based Two-Factor Authentication. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*. Springer, Cham, Switzerland, 405–421. https://doi.org/10.1007/978-3-662-54970-4_24

[58] Steve Kremer and Peter B Rønne. 2016. To du or not to du: A Security Analysis of Du-Vote. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, Piscataway, NJ, USA, 473–486.

[59] Robert Krimmer, David Duenas-Cid, Iuliia Krivonosova, Priit Vinkel, and Arne Koitmae. 2018. How much does an e-Vote cost? Cost comparison per vote in multichannel elections in Estonia. In *International Joint Conference on Electronic Voting*. Springer, Cham, Switzerland, 117–131.

[60] Kat Krol, Eleni Philippou, Emiliano De Cristofaro, and M Angela Sasse. 2015. "They brought in the horrible key ring thing!" Analysing the Usability of Two-Factor Authentication in UK Online Banking. In *Proceedings of the Workshop on Usable Security (USEC 2015)*. Internet Society, Reston, VA, USA. https://doi.org/10.14722/usec.2015.23001

[61] Ivo Kubjas, Tiit Pikma, and Jan Willemson. 2017. Estonian Voting Verification Mechanism Revisited Again. In *Proceedings of the International Joint Conference on Electronic Voting (E-Vote-ID)*. Springer, Cham, Switzerland, 306–317. https://doi.org/10.1007/978-3-319-68687-5_19

[62] Oksana Kulyk, Jan Henzel, Karen Renaud, and Melanie Volkamer. 2019. Comparing "Challenge-Based" and "Code-Based" Internet Voting Verification Implementations. In *Proceedings of the IFIP Conference on Human-Computer Interaction (INTERACT)*. Springer, Cham, Switzerland, 519–538. https://doi.org/10.1007/978-3-030-29381-9_32

[63] Ralf Küsters, Julian Liedtke, Johannes Mueller, Daniel Rausch, and Andreas Vogt. 2020. Ordinos: A Verifiable Tally-Hiding E-Voting System. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, Piscataway, NJ, USA, 216–235. https://doi.org/10.1109/EuroSP48549.2020.00022

[64] Ralf Küsters, Johannes Müller, Enrico Scapin, and Tomasz Truderung. 2016. sElect: A Lightweight Verifiable Remote Voting System. In *Proceedings of the Computer Security Foundations Symposium (CSF), 2016 IEEE 29th*. IEEE, Piscataway, NJ, USA, 341–354. https://doi.org/10.1109/CSF.2016.31

[65] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *Proceedings of the Symposium of the Austrian HCI and Usability Engineering Group (USAB)*. Springer, Cham, Switzerland, 63–76. https://doi.org/10.1007/978-3-540-89350-9_6

[66] Karola Marky, Oksana Kulyk, Karen Renaud, and Melanie Volkamer. 2018. What Did I Really Vote For? On the Usability of Verifiable E-Voting Schemes. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, New York, NY, USA, Article 176, 13 pages. https://doi.org/10.1145/3173574.3173750

[67] Karola Marky, Oksana Kulyk, and Melanie Volkamer. 2018. Comparative Usability Evaluation of Cast-as-Intended Verification Approaches in Internet Voting. In *Proceedings of the Jahrestagung des Fachbereichs Sicherheit – Schutz und Zuverlässigkeit der Gesellschaft für Informatik (SICHERHEIT)*. Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn, Germany, 197–208. https://doi.org/10.18420/sicherheit2018_15

[68] Karola Marky, Verena Zimmermann, Markus Funk, Jörg Daubert, Kira Bleck, and Max Mühlhäuser. 2020. Improving the Usability and UX of the Swiss Internet Voting Interface. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376769

[69] Karola Marky, Marie-Laure Zollinger, Markus Funk, Peter Ryan, and Max Mühlhäuser. 2019. How to Assess the Usability Metrics of E-Voting Schemes. In *Proceedings of Financial Cryptography and Data Security*. Springer, Cham, Switzerland, 257–271. https://doi.org/10.1007/978-3-030-43725-1_18

[70] Víctor Mateu and Magda Valls. 2017. Cast as Intended Verifiability for Mixed Array Ballots. In *Proceedings of the International Conference on Electronic Government and the Information Systems Perspective (EGOVIS)*. Springer, Cham, Switzerland, 206–218. https://doi.org/10.1007/978-3-319-64248-2_15

[71] Yomna Nasser, Chidinma Okoye, Jeremy Clark, and Peter YA Ryan. 2018. Blockchains and voting: Somewhere between hype and a panacea.

[72] S. Neumann, M. M. Olembo, K. Renaud, and M. Volkamer. 2014. Helios Verification: To Alleviate, or to Nominate: Is That the Question, or Shall we Have Both?. In *Proceedings of the International Conference on Electronic Government and the Information Systems Perspective (EGOVIS)*. Springer, Cham, Switzerland, 246–260. https://doi.org/10.1007/978-3-319-10178-1_20

[73] NSW Electoral Commission. 2019. iVote online and telephone voting. https://www.elections.nsw.gov.au/Voters/Other-voting-options/iVote-online-and-telephone-voting. [Online; accessed: 18-November-2019].

[74] Maina M. Olembo, Steffen Bartsch, and Melanie Volkamer. 2013. Mental Models of Verifiability in Voting. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer, Cham, Switzerland, 142–155. https://doi.org/10.1007/978-3-642-39185-9_9

[75] POLYAS GmbH. 2019. Overview of Polyas Costumers. POLYASKundenÜbersicht. [Online; accessed: 30-September-2019].

[76] Programme Office eGovernment Switzerland. 2019. Electronic Voting. https://www.egovernment.ch/en/umsetzung/schwerpunktplan/vote-electronique/. [Online; accessed: 18-November-2019].

[77] Peter Y. A. Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. 2009. Prêt à voter: a voter-verifiable voting system. *IEEE Transactions on Information Forensics and Security* 4, 4 (2009), 662–673.

[78] Peter Y. A. Ryan, Peter B. Rønne, and Vincenzo Iovino. 2016. Selene: Voting with Transparent Verifiability and Coercion-Mitigation. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*. Springer, Cham, Switzerland, 176–192. https://doi.org/10.1007/978-3-662-53357-4_12

[79] Peter Y. A. Ryan and Vanessa Teague. 2009. Pretty Good Democracy. In *Proceedings of the International Workshop on Security Protocols (SPW)*. Springer, Cham, Switzerland, 111–130. https://doi.org/10.1007/978-3-642-36213-2_15

[80] Kazue Sako and Joe Kilian. 1995. Receipt-Free Mix-Type Voting Scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, Cham, Switzerland, 393–403. https://doi.org/10.1007/3-540-49264-X_32

[81] Ted Selker, Elizabeth Rosenzweig, and Anna Pandolfo. 2006. A Methodology for Testing Voting Systems. *Journal of Usability Studies* 2, 1 (Nov. 2006), 7–21. http://dl.acm.org/citation.cfm?id=2835536.2835538

[82] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. 2014. Security Analysis of the Estonian Internet Voting System. In *Proceedings of the Conference on Computer and Communications Security (SIGSAC)*. ACM, New York, NY, USA, 703–715. https://doi.org/10.1145/2660267.2660315

[83] Joshua Tan, Lujo Bauer, Joseph Bonneau, Lorrie Faith Cranor, Jeremy Thomas, and Blase Ur. 2017. Can Unicorns Help Users Compare Crypto Key Fingerprints?. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, New York, NY, USA, 3787–3798. https://doi.org/10.1145/3025453.3025733

[84] Georgios Tsoukalas, Kostas Papadimitriou, Panos Louridas, and Panayiotis Tsanakas. 2013. From Helios to Zeus. In *Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE)*. USENIX Association, Berkeley, CA, USA.

[85] Jan Vom Brocke, Alexander Simons, Bjoern Niehaves, Kai Riemer, Ralf Plattfaut, Anne Cleven, et al. 2009. Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In *Proceedings of the European Conference on Information Systems (ECIS)*, Vol. 9. 2206–2217.

[86] Willem A. Wagenaar. 1972. Generation of Random Sequences by Human Subjects: A Critical Survey of Literature. *Psychological Bulletin* 77, 1 (1972), 65–72. https://doi.org/10.1037/h0032060

[87] Janna-Lynn Weber and Urs Hengartner. 2009. Usability Study of the Open Audit Voting System Helios. http://www.jannaweber.com/wpcontent/uploads/2009/09/858Helios.pdf. [Online; accessed: 22-December-2017].

[88] Jane Webster and Richard T. Watson. 2002. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *Management Information Systems Quarterly* 26, 2 (2002), xiii–xxiii.

[89] Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. 2012. Attacking the Washington, DC Internet Voting System. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*. Springer, Cham, Switzerland, 114–128. https://doi.org/10.1007/978-3-642-32946-3_10

[90] Hua Wu, Poorvi L. Vora, and Filip Zagórski. 2019. PrivApollo–Secret Ballot E2E-V Internet Voting. In *Proceedings of the 4th Workshop on Advances in Secure Electronic Voting (VOTING)*. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-43725-1_21

[91] Filip Zagórski, Richard T Carback, David Chaum, Jeremy Clark, Aleksander Essex, and Poorvi L Vora. 2013. Remotegrity: Design and Use of an End-to-End Verifiable Remote Voting System. In *Proceedings of the International Conference on Applied Cryptography and Network Security*. Springer, Cham, Switzerland, 441–457.

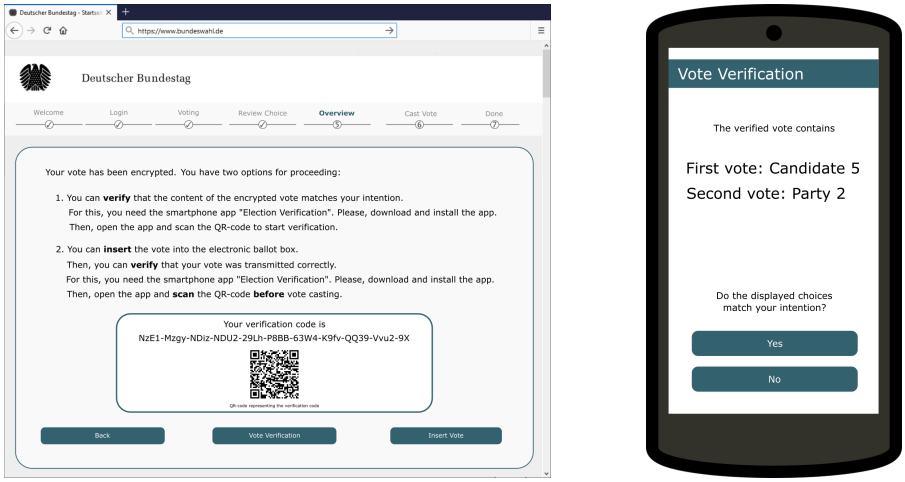
[92] Anfu Zhou, Guangyuan Su, Shilin Zhu, and HuaDong Ma. 2019. Invisible QR Code Hijacking Using Smart LED. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 126 (Sept. 2019), 23 pages. https://doi.org/10.1145/3351284

[93] Marie-Laure Zollinger, Verena Distler, Peter B. Roenne, Peter Y. A. Ryan, Carine Lallemand, and Vincent Koenig. 2019. User Experience Design for E-Voting: How Mental Models Align with Security Mechanisms. In *Proceedings of the International Joint Conference on Electronic Voting (E-Vote-ID)*. TalTech, 187–202.

A APPENDIX

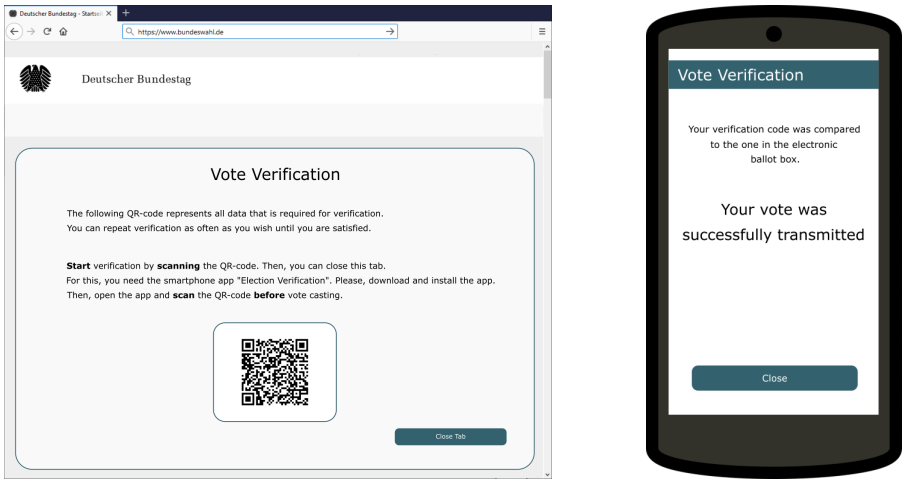
A.1 Screenshots

In this appendix section, we provide screenshots of verification in each category. We adapted the screenshots to show a generic election. In the study, we used data from the last election in Germany.



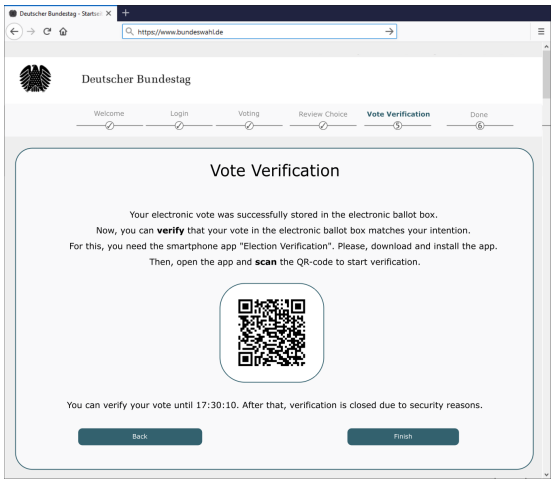
(a) Voting Website (b) Verification App

Fig. 7. Cast-as-intended verification using audit or cast.



(a) Voting Website (b) Verification App

Fig. 8. Recorded-as-cast verification using audit or cast.



(a) Voting Website

(b) Verification Device

Fig. 9. Individual verifiability using verification devices.

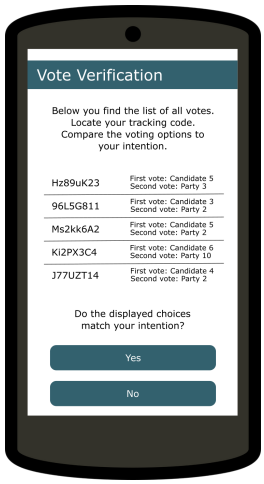
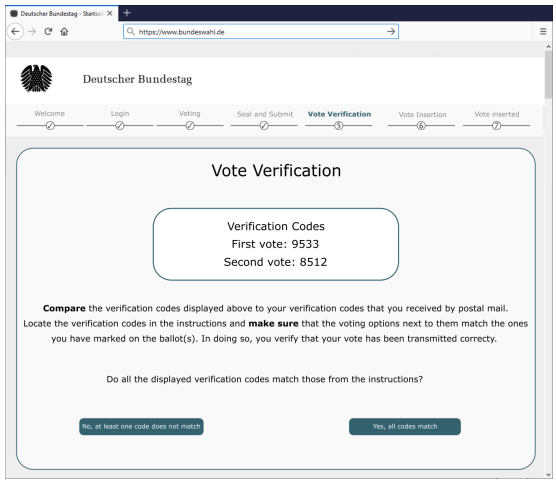
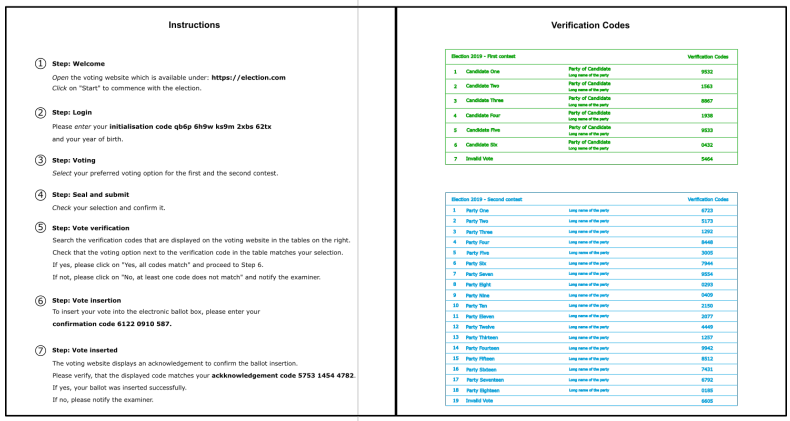


Fig. 10. Individual verifiability using tracking data.



(a) Voting Website



(b) Code Sheet

Fig. 11. Individual verifiability using code sheets.

A.2 Final Open-Ended Questionnaire

This questionnaire was given to the participant after the interaction with the voting scheme.

1) Would you use the presented voting system in a real political election? (Answer options: yes, no, not sure) Please explain your answer.
2) Would you like to check your vote for correctness in general independently from the software you have just used in a real political election? (Answer options: yes, no, not sure) Please explain your answer.
3) Would you use the software that you have just used to check your vote for correctness in a real political election? (Answer options: yes, no, not sure) Please explain your answer.
4) Do you think that you successfully audited your vote using the presented software? (Answer options: yes, no, not sure) Please explain your answer.
5) Which properties could you check or audit with the presented software to your understanding?
6) Would using the presented software convince you that your vote was stored in the electronic ballot box and your intention? (Answer options: yes, no) Please explain your answer.

7*) Audit-or-cast: Vote auditing can be done multiple times before casting a vote. How often would you carry out verification? Please, explain your answer.
7*) Supplementary device: Vote auditing typically can be done only in a limited time-frame for security purposes (e.g., 30 minutes). What do you think about this limitation? Please, explain your answer.
7*) Tracking data: Vote auditing can only be done once the official election result is announced. Typically, this is two weeks after elections day. After this period of time would you check your vote? Please, explain your answer.
7*) Code sheets: Maybe: To participate in the election, you need to register in advance to the authorities in order to receive the letter containing credentials and codes. Do you consider this effort as appropriate? Please, explain your answer.

8) If you have any further feedback, you can note it here.

B USER STUDY RESULTS

In this section, we provide additional descriptive data of the data collected in our user study.

Table 2. Descriptive statistics of effectiveness and efficiency metrics.

	Effectiveness [%]	**Efficiency [s]**				
		∅	*Median*	*SD*	*Min*	*Max*
Audit-or-Cast	28	305.48	263	155.65	99	812
Verification Device	64	215.88	175	106.28	127	535
Tracking Data	84	326.68	298	103.12	201	663
Code Sheets	100	632.44	614	152.28	429	937

Table 3. Descriptive statistics of satisfaction metrics in terms of SUS scores.

	∅	*Median*	*SD*	*Min*	*Max*
Audit-or-Cast	73.00	75.00	19.57	27.50	100.00
Verification Device	84.60	85.00	13.16	40.00	97.50
Tracking Data	82.10	90.00	15.08	40.00	100.00
Code Sheets	84.50	87.50	15.81	32.50	100.00

B.1 Code Dictionary

Final code dictionary used for the final round of coding.

- Reasons for not adopting verification
  - Security concerns
  - Trust in authorities
  - Trust in provided software
  - Violation of vote privacy (TD)
  - Usability aspects
  - Backup solution
  - Auditing option
  - Complexity
  - Missing information about the system
  - Humans perceived as more secure
  - Software secure by design
  - Verification not effective
  - Verification not needed
  - Familiarization once but not needed after
  - Should be automatic
  - Devices not available
  - Lack of motivation
  - Cast vote cannot be verified
  - No verification possible (AC)
- Reasons for adopting verification
  - Controllability
  - Ensuring integrity
  - Feeling of security
  - Importance of elections
  - Security concerns
  - Duty
  - Ease of verification (usability)
  - Transparency
  - Missing information about the system
  - Humans perceived as more secure
  - General trust aspects
  - Trust in officially published data (TD)
  - Not possible alternatives
  - Security by second devices (AC, VD)
- General trust perceptions
  - Confirmation via verification
  - Manipulation of verification software possible
  - Intransparency
- Requesting software independence
- No understanding of verification
- Partial understanding of verification
- Complete Understanding of verification
- Audit-or-Cast question of verification
  - Verification not effective
  - Verification not needed
  - Intransparency
  - Sufficient
  - Personal preference
  - No necessity of more than once
  - Security aspects
  - Exerting control
  - Familiarization
- Supplementary device question
  - Totality (there must be an end like in the paper election)
  - Verification directly after casting
  - Feasibility within time-frame
  - No necessity to do it later
  - Not available if forgotten
  - Doubts later on
  - Helplessness
- Tracking data question
  - Trust in official results
  - Similarity to other voting channels (postal/paper)
  - Similarity to other service (banking)
  - Necessity
  - Scrutinizing based on false evidence
  - Time consuming

  - Security
  - No alternatives
  - Impact on vote privacy
- Code sheets question
  - Similarity to other voting channels (postal/paper)
  - Security aspects
  - Time aspects
  - Comfort aspects
  - Personal preference
  - To much effort for internet voting