Improving the Usability and UX of the Swiss Internet Voting Interface

Karola Marky[1,2], **Verena Zimmermann**[1], **Markus Funk**[1,3], **Jörg Daubert**[4], **Kira Bleck**[1] **and Max Mühlhäuser**[1]
[1]Technische Universität Darmstadt, Darmstadt, Germany, [2]Keio University, Yokohama, Japan, [3]Cerence GmbH, Ulm, Germamy, [4]Philipps-Universität, Marburg, Germany
{marky, bleck, max}@tk.tu-darmstadt.de, zimmermann@psychologie@tu-darmstadt.de, markus.funk@cerence.com, daubert@mathematik.uni-marburg.de

ABSTRACT
Up to 20% of residential votes and up to 70% of absentee votes in Switzerland are cast online. The Swiss scheme aims to provide individual verifiability by different verification codes. The voters have to carry out verification on their own, making the usability and UX of the interface of great importance. To improve the usability, we first performed an evaluation with 12 human-computer interaction experts to uncover usability weaknesses of the Swiss Internet voting interface. Based on the experts' findings, related work, and an exploratory user study with 36 participants, we propose a redesign that we evaluated in a user study with 49 participants. Our study confirmed that the redesign indeed improves the detection of incorrect votes by 33% and increases the trust and understanding of the voters. Our studies furthermore contribute important recommendations for designing verifiable e-voting systems in general.

Author Keywords
E-Voting; Individual Verifiability; Usability Evaluation

CCS Concepts
•**Security and privacy** → **Usability in security and privacy;**

INTRODUCTION
Switzerland introduced e-voting via the Internet as a possible vote-casting channel in the early 2000s for citizen-driven referenda and has gradually expanded its usage since then [47]. In 2008, the canton of Neuchâtel introduced Internet voting for expats as the first canton [21]. Over a decade later in 2019, 12 of the 29 cantons permitted e-voting for up to 50% of the electorate in the respective canton [16]. Up to 20% of residential votes and up to 70% of expat votes were cast online in past elections and referenda in the cantons that allow Internet voting [47].

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
DOI: https://doi.org/10.1145/3313831.3376769

Designing and implementing secure e-voting systems is challenging and many weaknesses and vulnerabilities of e-voting systems that are used in practice have been uncovered [27, 48, 53, 12, 54]. Due to the secure platform problem, malware on the voters' computers that manipulates or deletes votes, can also compromise the integrity of these systems [20].

Switzerland follows the principle "security before speed" and the deployed systems have to comply with security requirements mandated by the Swiss jurisdiction [42, 26]. The security requirements include *individual verifiability* for voters: 1) voters must receive proof that the electronic ballot box has registered the vote as it was entered by the voter into the voting software — so-called *cast-as-intended verifiability*; and 2) a proof of correct vote registration must be provided for each vote — the *recorded-as-cast verifiability*.

In traditional paper-based voting schemes, voters can easily verify these properties: they visually inspect that their paper ballot represents their intention, and then insert it into the ballot box. Achieving such verifiability in e-voting is not trivial, and e-voting schemes have to be designed specifically to support individual verifiability.

The Neuchâtel Internet voting scheme [19] is used in several Swiss cantons [15]. The scheme aims to provide individual verifiability by return codes. Before the election or referendum, the voters receive individualized paper code sheets. After they cast their votes, the voting software displays a return code that corresponds to the cast vote. The voter then compares this displayed code to the code that appears next to their intended voting option on their paper code sheet. While the individual verifiability of the Neuchâtel scheme has been formally proven [11], there is another potential security threat: usability issues.

Maximizing the usability of e-voting schemes constitutes a particular challenge. Vote secrecy is one of the cornerstones of democratic elections and this requires voters' intentions to be kept secret. This makes it difficult to automate or delegate individual verifiability. The voters have to validate the outcome of the verification i.e., checking the correctness of the cast vote. If voters are unable to verify successfully, incorrect votes cannot be detected. Verification is an unfamiliar action that is not required by traditional paper-based schemes. Because voting is typically not an everyday activity, and any election

includes new voters, we cannot expect training and learning to mitigate any usability issues. Investigations of Internet voting schemes show that usability issues prevent voters from determining whether their votes have been cast correctly [18] and only a small percentage of voters (between 10% and 43%) were able to verify successfully [52, 1, 37]. Existing evaluations of the Neuchâtel scheme focus on formal proofs [11] or on the usability at the protocol level [38]. The user interface greatly impacts the usability of Internet voting systems [37, 32]. Hence, the usability has to be evaluated with end-users.

Research Contributions

- We conducted an expert evaluation of the Swiss interface of the Neuchâtel scheme revealing six weaknesses that impact the usability of vote casting and individual verification.
- Based on the expert study and an exploratory user study, we propose a redesign of the voting software and supportive materials that we evaluated with 49 participants. It assists the voters significantly better in detecting incorrect votes, provides a better understanding of verification and significantly enhances voter trust.
- Our results do not only impact the tested Neuchâtel scheme. Based on the results of our studies, we conclude with five recommendations to inform the design of individual verifiable Internet voting schemes in general.

VERIFICATION IN INTERNET VOTING

In this section, we give an overview of verification in Internet voting schemes. We commence with the general properties of verification followed by a description of five classes of verification schemes in the literature.

Verifiability can be divided into the subroutines of 1) cast-as-intended – the cast vote corresponds to the voter's intention, 2) recorded-as-cast – the registered vote corresponds to the cast vote, and 3) counted-as-recorded – the registered vote was included in the tally [2]. Vote casting refers to the act of submitting a ballot while registering means that it is stored in the electronic ballot box. While counted-as-recorded can be verified by anyone, the voters have to carry out the other subroutines by themselves. The general-purpose to let the voters perform such checks is ensuring the integrity of the election result. Since an electronic system and infrastructure are used, cast votes might be incorrect due to a malfunctioning voting software or Internet connection, furthermore, third parties might maliciously influence the election.

Verification Schemes

There are five classes of verification schemes that are available in the literature. In all classes of verification schemes, the voters have to carry out verification by themselves. They have to compare data and have to make sure that they fulfilled all steps correctly. The recommendations we provide based on our user studies impact all of these classes by showing how voters can be effectively supported during verification.

The first class is *decryption-based* verification in which the ciphertext of the registered vote is inspected. Alice receives a vote identifier that she uses in combination with a verification software to verify that her ballot in the electronic ballot box matches her intention. Decryption-based verification is, for instance, used in the Estonian voting scheme [28].

The second class is a set of probabilistic schemes which are called *challenge or cast*. The voting software prepares an encrypted vote and issues a verification code that Alice has to store. Then, she can either cast the encrypted vote or she verifies it. Due to the fact that an adversary does not know whether Alice will verify or cast, the adversary cannot be certain whether a manipulation will be undetected. If Alice verifies, the randomness value used during encryption and her choice are transmitted to a verification software. The software calculates an encrypted vote and Alice compares both her verification code and the choice displayed by the verification software. Challenge or cast is used in the Helios scheme [3].

The third class is *hardware-based* verification in which a trusted device is used during the generation of the encrypted vote. Alice uses previously received voting codes and a trusted device to generate a vote code. To cast her vote Alice enters the vote code into the voting software. The electronic ballot box publishes a list of all received vote codes. Alice accesses a bulletin board to verify that her vote code is on the list. The scheme Du-Vote is based on a trusted device [23].

The fourth class of schemes is based on *tracking codes*. After casting a vote, Alice is notified with a tracking code. After the election, all votes are published in plain text together with the tracking code. To verify Alice accesses a bulletin board and looks for her tracking code. The commercial voting system Polyas is such a tracking code scheme [43].

In the class of *return code schemes* each voting option has a return code which is listed on a code sheet. Code sheets are distributed before the election over a trusted channel (e.g., postal mail). Voters are advised to verify if the codes returned by the voting software match the ones from their personalized code sheet. In the next section, we detail the return code scheme Neuchâtel which is used in Switzerland [19].

Neuchâtel Scheme

The Neuchâtel Scheme is a return code scheme as described above. Switzerland uses the postal office to distribute paper-based code sheets, to avoid the voting software gaining knowledge of voters' codes. If a compromised voting software gained this knowledge, it could successfully manipulate votes without fearing detection. To verify and cast a vote, a voter Alice has to carry out the following five steps (see also Fig. 1):

Step 1 – Authentication. Alice opens the voting software and authenticates. In the Swiss case, she does this by entering an individual unique *initialization code*, that she obtains from her *code sheet* and her year of birth. This initialization code has 25 digits and consists of numbers and lower case letters.

Step 2 – Voting. Alice makes her selections and confirms them on a review-screen. The voting software encrypts the selections and transmits the encrypted vote to the ballot box server. The vote is not stored in the electronic ballot box yet.

Figure 1. To cast and verify a vote, these five steps have to be completed in return code schemes.

Step 3 – Return Codes. After receiving the encrypted vote, the ballot box server requests the *return codes* that belong to Alice's selections from a code generator and sends them to the voting software. The software displays the return codes and Alice compares them to the ones on her code sheet next to her selections. If any return code does not correspond to her selections, Alice has to cancel the voting process and to use another computer or voting channel (e.g., postal or in-person polling station). Swiss return codes consist of four digits.

Step 4 – Confirmation Code. Alice confirms the correctness of the return codes by entering her *confirmation code* which is also listed on her code sheet but protected by a scratch field. The voting server checks Alice's confirmation code and, if it is correct, Alice's encrypted vote is now stored in the electronic ballot box and will be included in the tally. Swiss confirmation codes have nine digits.

Step 5 – Finalization Code. To confirm the storage in the electronic ballot box, the voting client displays a *finalization code* that it received. Alice compares the displayed finalization code to the one on her code sheet to confirm that the vote has been recorded-as-cast. If they do not match, Alice can trigger an alarm by calling a support hotline. The Swiss finalization code has eight digits and consists of numbers.

Further return code schemes are the Norwegian scheme [4, 22], code sheet-based schemes [29], DEMOS [35], D-DEMOS [10], verification for mixed-array ballots [40], oblivious transfer [25], and the schemes by Khazaei and Wikström [34]. While the computational generation of return codes differs in the schemes the interaction from the voter's perspective is either very similar or even identical to Neuchâtel.

USABILITY IN INTERNET VOTING

The usability of e-voting schemes is important because errors triggered by usability issues might lead to an incorrect election result. The usability of several schemes that offer some degree of individual verifiability has been investigated.

The protocol Benaloh Challenge [7, 6] in the Helios voting scheme [3] is a challenge or cast scheme. A user study of Helios and the University of Lleida scheme revealed that participants did not realize that they needed to challenge the system by verifying their vote [44]. Further usability studies of Helios show that between 57% and 90% of participants attempted verification but were unable to complete it [1, 52]. Karayumak *et al.* did a cognitive walkthrough of Helios and proposed refinements for verification [33], which they evaluated in a user study [32]. Neumann *et al.* also investigated Helios and proposed the usage of an independent mobile device to carry out verification [41]. A comparative usability study of Helios, the refinement by Karayumak *et al.* [33, 32] and the refinement by Neumann *et al.* [41] reveals that verification with a mobile device offers the best usability [37]. Although the majority of participants were able to verify successfully, they struggled with the Benaloh Challenge, expressed confusion because their verified votes could not subsequently be cast, and even considered verification to be redundant [37]. The studies of the Benaloh Challenge show that the usability of vote verification can impact the scheme's effectiveness and security. Usability also depends on the interface design which can be adjusted to provide better support.

The Selene scheme is based on tracking codes to enable individual verifiability [45]. After the tally, the voting option is listed next to the voter's individual tracking code. Displaying security-related information during voting and verification results in a higher security perception but hampers understandability [14]. The Neuchâtel scheme aims to provide individual verifiability based on return codes. The mCESG return code scheme for mobile devices issues return codes via SMS [49]. A user study revealed that participants tended to appreciate being able to verify, but struggled to trust the implemented scheme [50]. The Norwegian scheme is a predecessor of Neuchâtel that also uses return codes [4]. Participants in a user study of its prototype could not determine whether their votes had been submitted to the electronic ballot box [18].

The usability of four protocols aiming to provide cast-as-intended verifiability have been investigated in an expert study [38]. The authors analyzed the usability of Neumann *et al.'s* approach of the Benaloh Challenge [41], the Estonian cast-as-intended mechanism [28], Du-Vote [23] and Neuchâtel 2015 [19] on a protocol level by identifying tasks that the voters have to carry out by themselves. The authors conclude that the Neuchâtel protocol places the lowest burden on the voters. On the protocol level, the voters have to compare the return and finalization codes and enter their confirmation codes. The Du-Vote scheme, by comparison, places the highest burden on the voter. Voters have to choose specific codes based on a coin flip, enter them into a trusted device and record verification-related data and subsequently search for these data on a bulletin board. The expert study described above revealed that the Neuchâtel protocol is promising in terms of usability and individual verifiability has already been proven formally [11]. The specific interface of Switzerland, to our knowledge, has not been investigated with voters, yet.

EXPERT EVALUATION

In this section, we report the expert evaluation of the Swiss Post interface, which aims to answer the following two research questions: 1) What are the usability weaknesses of the interface that impact vote casting and verification, and 2) how can the identified weaknesses be mitigated and/or addressed?

Study Procedure, Analysis, and Participants

Our expert evaluation was split into the interaction with the interface and a semi-structured interview. After completing a consent form and providing demographics, each expert cast a vote with the Swiss Post interface. Swiss Post provides a demo[1], we used a copy of its interface and code sheet. The voting and verification procedure corresponds to the five steps in the Neuchâtel scheme detailed above. All interaction on the screen was recorded and we asked the experts to think aloud [8] and to comment on potential usability issues with a focus on vote casting and vote verification. All comments were recorded. We explained the voting protocol to them before the interaction commenced and instructed them to consider all tasks that are necessary to cast and verify a vote. After the interaction, we conducted semi-structured interviews to gain a deeper understanding of the identified weaknesses. We opted for semi-structured interviews because they offer a degree of standardization while allowing flexibility to gain a deeper understanding. We asked the experts to detail the weaknesses mentioned earlier on, their impact, and to make suggestions for mitigating or addressing them without changing the voting protocol. For the full interview guide, the reader is referred to the paper's supplementary material. On average, an interview took 18 minutes ($Min = 8$, $Max = 35$). All recordings were transcribed and analyzed with an open-coding approach [17]. One coder developed an initial codebook by reviewing the transcripts. Two researchers then coded all transcripts independently with an 85.7% agreement. In a follow-up discussion, new categories for inconclusive codings were developed. The researchers then applied the final codebook and remaining ambiguities were solved by discussion.

We recruited twelve experts in human-computer interaction and (usable) security using mailing-lists and poster advertisements. We kept recruiting until the answers of the experts showed repetitions and saturation was reached. On average, the experts were 31.2 years ($SD = 5.5$, $Median = 30$, $Min = 26, Max = 45$) old and had an average of 4.8 ($Min = 1$, $Max = 10$) years experience in the field. Nine came from an academic context and three were industry experts. Eight were computer scientists, two psychologists, and two designers. Half of them reported profound knowledge in IT-security.

Usability Weaknesses

The experts uncovered the following six weaknesses related to the existing code sheet, and the corresponding voting software.

W1: Wrong Procedure Indication. The code sheet contains graphical elements (e.g., arrows) that are intended to depict voting and verification procedures. This depicted procedure does not match the five steps from the Neuchâtel protocol because it does neither include the second step "voting and confirmation" nor the return codes. Comparing the return codes is security-critical, but a mental task, that is not required to proceed. Voters might skip it if they are not properly informed about it. Therefore, voters might miss out on verifying the return codes or struggle to locate them on the code sheet.

W2: Missing/Confusing Common Elements. The code sheet and the voting software do not share a common design. The only common elements are four shapes (triangle, pentagon, star, and diamond) that link the required code(s). These shapes do not depict the actions or items that they represent. They might confuse voters or appear to be random. Thus, voters are not supported in locating the required code(s) on the code sheet and might miss interacting with them.

W3: Missing Information. The code sheet contains the codes and information in which step they are required but does not provide further instructions. Hence, the code sheet does not guide voters through the procedure and they themselves have to ensure that they performed all steps. Without guidance on the code sheet, they might miss essential steps, such as checking the return or finalization codes. Furthermore, voters might be overwhelmed by all the different codes appearing on the code sheet. To get information about the procedure, voters need to access the voting software and can therefore not familiarize themselves with the procedure beforehand. The information on how to act if a problem (e.g., an incorrect return code) was uncovered is available after clicking a link. Voters might overlook this information and not act properly.

W4: Incorrect Code Sheet Labeling. A letter from the authorities accompanies the code sheet. The letter is labeled as "suffrage credentials", although the letter itself is not required for voting. The letter also contains a barcode with numbers that look similar to the initialization code. Voters could mistakenly think that the letter is the code sheet or confuse the barcode numbers with the initialization code.

W5: Human Errors during Code Interaction. Voters have to interact with at least one initialization code, one return code per vote, one confirmation code and one finalization code and might be overwhelmed by the number of codes. This could lead to the missing of essential steps (e.g., verifying the return codes) or the interchanging of codes. Voters have to visually compare the return and finalization codes and determine the result of the verification by themselves. Comparing and entering sequences of numbers is an error-prone process [51]. These errors might lead voters to uncover an incorrect code that is correct in reality and thus trigger a false alarm or to overlook an incorrect code. This weakness is not specific to the Neuchâtel scheme and concerns all verification schemes.

W6: Non-Scalability of Return Codes. Each voter needs to compare one return code per vote and the finalization code. In small elections, voters are likely to compare all codes while it is debatable that they would compare all return codes in an election with many contests. Comparing a large number of codes might also lead to fatigue and consequent errors. Therefore, voters might overlook an incorrect vote. This weakness concerns all return code and tracking code schemes.

[1] Available on https://www.evoting.ch/en accessed: 14-May 2019

EXPLORATORY STUDY

To address the weaknesses, we propose an interface informed by the expert findings. We investigated it in a study with 36 participants and gathered mainly qualitative data.

Proposed Interface

The Swiss code sheet does not correctly reflect the procedures (W1: wrong procedure indication). To implement a clear procedure indication, that includes the return codes, the proposed interface has a progress bar with six steps. The same six steps with identical labels are listed on the code sheet.

The redesign introduces common design elements by progress icons to address W2 (missing/confusing common elements). As a further common element, we introduce colors: each vote has a different color and the return codes on the code sheet are colored accordingly. The Swiss code sheet contains the codes but no instructions to guide voters through the process (W3: missing information). The proposed interface includes step-by-step instructions on the code sheet that encompass the complete procedure, such that voters can familiarize themselves with the procedure before voting commences. To address another aspect of W3, we added the information on how to act in case of incorrect codes. W4 describes the incorrect labeling of the code sheet. Each part of the documents, the code sheet and letters from authorities, need to be labeled in such a way that their purpose is clear. Therefore, the text "suffrage credentials" is removed from the letter and we added a sentence to the letter's text, which states that the credentials can be found on the following page.

Voters might make errors during code interaction (W5) and should, therefore, be supported. All codes in the redesign are divided into groups of four characters as they are in the original interface. There are other alternative code representations, such as optic hashes or sentences that offer a high error-detection rate [30, 51, 13]. We deliberately did not include those into our redesign although the benefit of the techniques mentioned above is clearly given. In our study, we aimed to investigate our proposed interface changes, but not the effects of different hash presentations.

Addressing the non-scalability of return codes (W6) constitutes a particular challenge since the reduction of return codes influences the security of the verification protocol. Therefore, we cannot adapt the redesign to address this weakness directly without an in-depth protocol evaluation. The only way to address this weakness without changing the protocol is by improving voter awareness. This means that voters have to be clearly informed about the importance of checking all return codes. We realized this by an adaption of the instructions. Screenshots can be found in the supplementary material.

Method

To assess the usability of the proposed interface and the original interface, we conducted a user study with 36 participants. We implemented a prototype for the proposed interface detailed above, and a copy of the Swiss Post interface. We adapted both to match the 2017 election for the German federal government "Bundestag" to provide a realistic scenario with a realistic look and feel so that participants would take the simulated election in the study seriously [46, 39]. Therefore, we adjusted the ballot, logos and the texts to match this election. The instructions and arrangements of elements in the original condition were taken from the Swiss Post interface. The election in our user study had two contests, the first ballot had eight candidates and the second 18 candidates.

During the interaction, we encouraged the participants to think aloud [8] to gain an understanding of their interaction with the respective interface. We furthermore assessed the effectiveness of each interface, meaning whether participants were able to successfully verify their votes. To capture this, we introduced deliberate manipulations as recommended in related work [46, 39]. The adversary in our user study cannot alter the user interface by removing or adding user interface elements, but can manipulate the content of an encrypted vote before encryption by changing the voting option. The codes displayed on the voting software are manipulated to simulate an adversarial attack or voting software malfunction. To achieve this, the participant receives a return code that does not map to their selected voting option. Instead, the return code maps to another voting option or a blank vote, which is randomly chosen. Effectiveness is determined by the share of participants that detected and reported a manipulation.

Study Design and Procedure

We opted for a within-subjects study to compare the schemes with the same participants. Each participant interacted with four conditions in counterbalanced random order to mitigate sequence effects. In the study, we tested the following four conditions: Original, Original with manipulation, proposed interface and proposal with manipulation. To control external influences from the environment, we opted for a lab study and provided our participants with a laptop.

An average study session took 30 minutes. The materials that we used in the study can be found in the paper's supplementary material. The procedure aligns with the recommendations and standards of the ethics committee at our institution:

1) Welcome and Demographics. We started by explaining the consent form and data protection policy, that each participant had to sign. The consent form informed the participants that their participation was voluntary and that they could abort the study at any time. The consent form also contained details on the recordings that took place and explained that all collected data was only processed after anonymization. The audio recordings were transcribed and then deleted. All statements that might contain a link to the participant were anonymized by replacing parts of the statement by neutral placeholders. Then, the participant provided demographics.

2) Study Material. Each participant received an intent card with the voting options that the participant should vote for in the two contests. Otherwise, our recordings would have compromised the participants' vote secrecy. We had ten intent cards with different voting option possibilities. The participants drew one of them and could redrew if they felt uncomfortable with the given voting options. The examiner explained that they should vote according to the intend card and make sure, that a vote for the listed voting options was cast. The

examiner also provided an envelope with a mock letter from the election authorities and a code sheet for each condition.

3) Interaction and Questionnaires. We asked the participants to interact with the software and fulfill all steps as if this was a real election until the software told them to notify the examiner. The participants were asked to imagine that it was election day and that they were alone at home. The participants interacted with the conditions and were encouraged to think aloud. Their comments were recorded. The interface instructed the participants to notify the examiner if they found an incorrect return code or if they completed vote casting. If so, the examiner proceeded to the next condition.

4) Final Questionnaire. The participants received a final questionnaire with open-ended questions about all conditions. We asked which of the presented systems they like the most. Furthermore, we gave the opportunity to provide additional comments and feedback.

5) End. The examiner explained that manipulations were hidden in two conditions and that one of the study's goals was to investigate whether those were detected. The participant was offered the opportunity to go through the system again and the examiner answered the participant's questions.

Participants

We recruited 36 participants by advertising via mailing-lists, social networks, flyers, poster advertisements, and word-of-mouth. We did not reimburse participation. All participants possessed suffrage and had never participated in an Internet election. Their mean age was 35 years ($SD = 15$, $Median = 31$, $Min = 18$, $Max = 72$) and 53% ($N = 19$) identified as female. Half of the participants ($N = 18$) either studied a technical subject or worked a technical-related job, the other half did not. All participants reported daily Internet usage.

Effectiveness and Questionnaire Results

The effectiveness was calculated as the share of the $N = 36$ participants who reported an incorrect vote. 58% of participants detected the manipulation using the original, and 94% of participants detected it using the proposed interface. Thus, 36% detected the manipulation using the proposed interface even though they had not spotted it in the original. To account for the dependent measures design and the sample size, we used a McNemar test for the analysis. The test revealed that a significantly larger proportion of participants detected the manipulation when using the proposed interface, as compared to the original ($p < .001$, one-tailed, $w = 1.0$, $1\text{-}\beta = .95$).

Based on the questionnaires, 53% of participants preferred the proposed interface, 39% favored the original and 8% did not favor a scheme. 88% of those who favored the original gave the number of pages as an explanation. Of those who favored the proposed interface, 63% pointed at the clarity of the instructions, which supported them better and led to less confusion, as compared to the original.

Think Aloud and Open-Ended Questions Results

Throughout the study, the participants were encouraged to think aloud. We also provided a questionnaire to collect responses to open-ended questions. We analyzed the transcripts and the open-ended questions, using an open-coding approach [17], informed by following two questions: 1) What problems did the participants experience, and 2) What aspects were welcomed and liked?

Analogously to the expert study, one coder proposed an initial code book by reviewing the transcripts. Then two coders coded all transcripts independently with a 76.9% agreement level. During the coding process, the coders could add new codes to the initial code book that were discussed and agreed upon. The final code book was used to review all codings again. Finally, all assigned codes were discussed and final codes agreed upon. We present the findings in three groups that concern the 1) original, 2) the proposal, and 3) the general condition-independent process. Whenever meaningful we provide participant comments. The participants furthermore reported the weaknesses W1, W2, W3, W4 and W5 that we have uncovered in the expert study.

Original-Specific Findings

In the original condition, participants reported *initialization code localization problems*, *procedure problems* and *comprehension-related problems*. They positively commented on the fact that the code sheet only takes up *one page*.

Initialization code localization problems. 36% of participants struggled to locate the initialization code on the documents. The letter from the authorities is titled "suffrage credentials" and participants assumed this to be the code sheet. They tried to enter the barcode number or even the support hotline number as initialization codes. The problem is connected to weakness W4 (incorrect code sheet labeling).

Procedure Problems. The code sheet uses arrows to indicate the following procedure: initialization code $\rightarrow$ confirmation code $\rightarrow$ finalization code. The return codes are listed below this procedure indication in a table. 28% of participants did not notice the presence of the return codes on the code sheets. They focused on locating and entering the confirmation code. This could explain the low manipulation detection rate of 58%. Furthermore, 6% of participants wrote down the return codes, but did not use them later on.

Comprehension-Related Problems. 39% of participants mentioned a problem in comprehending the few given instructions and confusion about the next required task. Therefore, participants searched the voting software and the code sheet on what to do next. As mentioned above in the procedure problems, 28% of participants did not notice the return codes.

One Page Preference. 39% of participants specifically mentioned a preference towards the one-page code sheet.

Proposal-Specific Findings

In the proposed interface, participants *confused the confirmation and finalization codes*, but valued the *instructions*.

Confusion of Confirmation and Finalization Codes. The confirmation code was listed in the instructions above the return code table. 14% of participants searched the following page first and confused confirmation and finalization codes, because only the codes are printed bold and the code length is similar.

Instructions. 33% of the participants perceived the detailed instructions as helpful and used their guidance through the procedure.

Condition-Independent Findings

In all tested conditions participants struggled with the *initialization code*, and the used *terminology*. Furthermore, participants questioned the *necessity* of the verification tasks.

Initialization Code Complexity Problems. The initialization code is composed of numbers and lower-case letters and has a length of 20 characters, divided into chunks of four. 78% of participants considered this code too confusing and too long. They frequently made errors entering the code or struggled to read the printed code. Sample comments are:

> *"[...] it is long with letters and numbers. I find that quite cumbersome."* (P4)
> *"I immediately thought that the initialization code was extremely long and complicated."* (P21)

Terminology problems. 28% of participants expressed confusion about the terminology used to denote the codes. The terminology was perceived to be too technical and did not reflect the purpose of the codes. This was particularly the case in terms of the purpose of the finalization code, which was questioned based on the terminology. Sample comments are:

> *"The purpose of this finalization code is not apparent to me. Maybe it's connected to security."* (P4)
> *"I thought that the initialization code was extremely long, and is it like a password?"* (P25)

Questioning Necessity. 33% of participants questioned the necessity of the different codes while interacting with them. For 28% of participants the codes' purposes were completely unclear and they wished a deeper understanding and more information about the codes' purposes. Sample comments are:

> *"I apparently can use it and it looks easy, but why do I have to screw around with so many codes?"* (P27)
> *"What do all these codes mean? This feels more like banking than voting."* (P35)

REDESIGN USER STUDY

Based on the data we collected in our first user study, we propose a redesign, which we tested in a second user study.

Redesign Interface

6% of participants in the first study did not detect the incorrect return codes in the proposed interface. Since the return codes and the entry field for the confirmation code were in the same step, voters had two tasks to fulfill. Only confirmation code entry is required to proceed. Therefore, this page is split into two separate steps. The first one instructs voters to compare the return codes. It specifically asks them whether the codes match. A button with the label *"yes, all codes match"* forwards to the confirmation code page. A button with the label *"no, at least one code does not match"* forwards the voter to a page with instructions on actions what to do. The finalization code page is designed analogously.

33% of participants preferred the original code sheet because it takes up a single page of paper. At the same time, 39% of participants mentioned comprehension-related problems related to the instructions and 33% of participants welcomed the redesign instructions. 33% of participates favored the redesign because of the detailed instructions. The provision of one return code per voting option is essential and, therefore, the return codes can take up a lot of space. Since all instructions can fit on a single DIN A5 page, the return codes are listed in a table next to the instructions[2].

The participants in both schemes confused the finalization and confirmation codes, because of their placement on the code sheet and the label not being formatted in bold. To reduce this confusion, the codes are labeled in bold letters. The terminology was perceived as too technical and too confusing. This is because the terminology does not specifically reflect the purpose of the code in a natural language. Therefore, we use the following code terminology: The initialization code is denoted as *login code*, the confirmation code is denoted as *vote insertion code* and the finalization code is denoted as *acknowledgement code*. For screenshots of the redesign as well as the original interface the reader is referred to the supplementary material.

User Study

To test the redesign, we conducted a user study with 49 participants. The election was identical to the one in the exploratory study. The study was between-subjects with two groups. One group interacted with our redesign, the second group was the control group that interacted with the Swiss interface. To assess *effectiveness*, we measured the detection rate of manipulations as in the exploratory study. For *efficiency*, we considered the execution time of the verification procedure. *Satisfaction* was measured with the System Usability Scale (SUS) [9]. We furthermore assessed *user experience* with the user experience questionnaire (UEQ) [36]. Based on our previous studies, we derived the following hypotheses:

H 1 More of the voters using the redesign are able to successfully verify than voters using the original (Effectiveness).

H 2 The voters using the redesign require less time than voters using the original (Efficiency).

H 3 The redesign receives a higher SUS score than the original (Satisfaction).

We also wanted to investigate the participants' *understanding* of the verification procedure and their *subjective trust* in it. Therefore, we asked additional questions. The code sheets, authority letters and questionnaires that we used in the study can be found in the paper's supplementary material. The procedure follows the recommendations of the ethics committee at our institution and took on average 25 minutes:

1) Welcome and Demographics. Analogously to the exploratory study, we explained the consent form and study's data protection policy that each participant had to sign. Then the participants provided demographics.

[2]This does not scale for elections with many voting options, thus the return codes might be listed on separate pages.

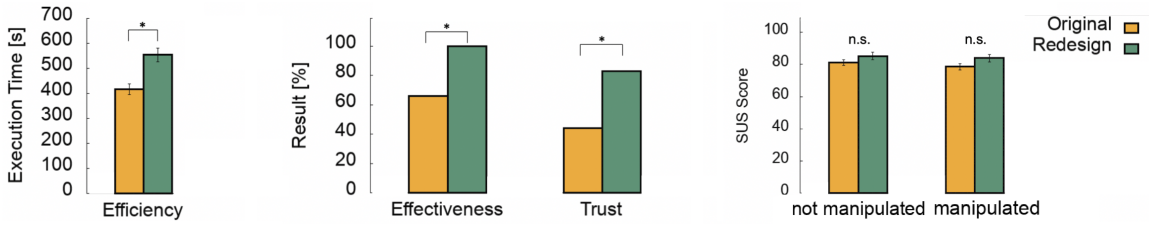

Figure 2. Quantitative results of our user study (a) execution time and, (b) effectiveness and trust, and (c) SUS score. All error bars indicate the standard error. The asterisk (*) indicates a statistically significant difference between the groups.

2) Study Material. Each participant was randomly assigned to one group. To ensure an equal group size, we generated a set participant IDs for all participants before the study and randomly drew one ID from the set. Analogously to the exploratory study, each participant received an intent card, could draw a voting option and a closed envelope containing a mock letter from the election authorities and the code sheet.

3) Interaction and Questionnaires. The participants should imagine that it was election day and interact with the voting software until it instructed them to notify the examiner. Each participant interacted twice with the voting software. In the second round, we manipulated the encrypted vote. After each round, the participants were asked to fill in the SUS and UEQ questionnaires. We asked the participants to rate only the verification procedure and provided them with screenshots.

4) Final Questionnaire. Then, we provided the final questionnaire to investigate the subjective trust and understanding of the verification procedure. We told the participants to consider both rounds when filling in the questionnaire.

5) End. The examiner explained the hidden manipulations. The participant was offered the opportunity to go through the system again and the examiner answered their questions.

Participants

We recruited 49 participants via mailing-lists, social networks, flyers, poster advertisements, contacting secondary schools, and snowball sampling. The participants could participate in a raffle for an Amazon 100€-voucher. 25 participants interacted with the original and 24 interacted with our redesign. All participants possessed suffrage for upcoming elections and had no previous experience with Internet elections. Their mean age was 27 years ($SD = 12$, $Median = 24$, $Min = 17$, $Max = 60$). Participants that were under-age received a consent form for their parents that had to be signed before the study. 47% ($N = 23$) of them identified as female, and one as diverse. Three participants were unemployed, two had a technical-related job, six were from academia, nine worked a non-technical job, 14 were students of various subjects and 15 were school students. All participants reported daily Internet usage.

Results

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

Effectiveness

As effectiveness, we consider the share of the participants who reported an incorrect vote. All of the 25 participants (100%) in the redesign group reported an incorrect vote. In the original, 8 of 24 participants (33%) reported an incorrect vote (see also Figure 2b). Because two expected cell frequencies were below 5, Fisher's exact test was used. A significance level of $p = .002$ (one-sided) supports hypothesis H1 (more of the voters using the redesign are able to successfully verify than voters using the original) with a medium high effect ($Cramer's\ V = 0.43$).

Efficiency of Unmanipulated Trials

Efficiency is measured as the execution time of the verification process. Participants using the redesign needed on average $554s$ to carry out verification ($SD = 134s$, $Median = 537s$, $Min = 357s$, $Max = 831$), while participants using the original needed $416s$ ($SD = 109s$, $Median = 399s$, $Min = 257s$, $Max = 599s$). Figure 2a depicts the execution times. We analyzed the collected duration with a t-test and the redesign group needed significantly longer than the original group ($t(47) = -3.94$, $p < .001$, $d = 1.12$). Based on that, hypothesis H2 (the voters using the redesign do not require more time than voters using the original) cannot be supported.

We also compared the time needed for checking the return codes and entering the confirmation codes between the redesign ($M = 141s$, $SD = 43s$) and the original ($M = 48s$, $SD = 13s$). Again, the time needed in the redesign is significantly longer $t(26.92) = -10.13$, $p < .001$, $d = 2.89$). The durations needed to mark the ballot, however, do not demonstrate differences ($M_{redesign} = 85s$, $SD_{redesign} = 33s$, $M_{original} = 90s$, $SD_{original} = 24s$, $t(47) = 0.53$, $p = .60$, $d = 0.15$).

Trust

In the final questionnaire, we asked: *"Are you confident that you can verify whether your votes are correctly transmitted to and stored in the electronic ballot box using the provided software and materials?"*. 44% of participants using the original, and 83% of those using the redesign answered this question affirmatively (see Figure 2b). A χ^2-test reveals significant differences ($\chi^2(1) = 9.69$, $p = .002$, $Cramer's\ V = 0.45$).

SUS Scores

The System Usability Scale (SUS) [9] ranges from 0 to 100 points, higher values indicate a better subjective usability. The SUS scores from our study are given by Figure 2c. The redesign received a mean score of 85.2 ($SD = 15.7$, $Median = 90.0$, $Min = 32.5$, $Max = 100.0$) when no manipulation was induced and a mean score of 83.9 ($SD = 15.9$, $Median = 90.0$, $Min = 32.5$, $Max = 100.0$) in the manipulated trials. The original received a mean score of 81.1 ($SD = 12.6$, $Median = 82.5$, $Min = 52.5$, $Max = 97.5$) in not manipulated trials and a

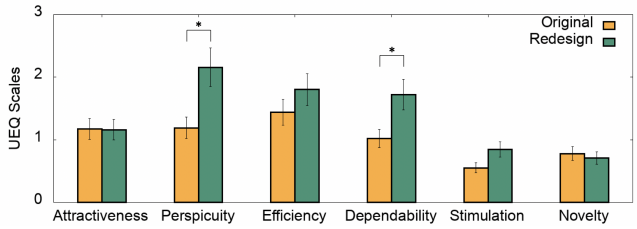

Figure 3. User experience scales.

mean score of 78.6 ($SD = 14.1$, $Median = 82.5$, $Min = 52.5$, $Max = 100.0$) in manipulated trials. No significant difference was found between the original and the redesign, neither for not manipulated trials ($t(47) = -1.01$, $p = .31$) nor for manipulated trials ($t(45.88) = -1.25$, $p = .22$). Thus, H3 (the redesign receives higher SUS scores) cannot be supported. The scores of all schemes can be interpreted as an "A" of the grade scale [5].

User Experience

The user experience was measured by the user experience questionnaire with the six scales 1) attractiveness, 2) perspicuity 3) efficiency 4) dependability 5) stimulation, and 6) novelty [36]. The UEQ results of our study are depicted by Figure 3. We analyzed each scale with a t-test and found significant differences in the scales of perspicuity ($M_{redesign} = 2.16$, $SD_{redesign} = 1.14$, $M_{original} = 1.19$, $SD_{original} = 1.43$, $t(45.48) = -2.62$, $p = .01$, $d = 0.75$) and dependability ($M_{redesign} = 1.72$, $SD_{redesign} = 0.81$, $M_{original} = 1.02$, $SD_{original} = 1.16$, $t(43.02) = -2.46$, $p = .02$, $d = 0.70$). In both cases, the redesign was better.

Intuitive Understanding

Based on the instructions given on the codes sheets and in the voting software, we asked the participants the following question: *"Please explain your understanding of the verification procedure. Which properties can you verify with the help of the code sheet?"*. Based on the definition of individual verifiability in the Neuchâtel protocol, we categorized the answers of the participants into three groups: 1) complete understanding, 2) partial understanding and 3) no understanding.

37% of all participants correctly explained the concept of individual verifiability. 58% of participants using the redesign gave a complete explanation whereas only of those 16% using the original did. Two participants using the redesign correctly explained the purpose of each code. 68% using the original and 25% using the redesign had an incomplete understanding. Finally, 16% using the original and 16.7% using the redesign demonstrated no understanding. Differences between the redesign and the original were significant ($\chi^2(2) = 9.45$, $p = .009$, $Cramer's\ V = 0.44$).

DISCUSSION

In this section, we discuss the results from our user studies and explain why usability alone is not sufficient in verifiable Internet voting schemes. We conclude with final recommendations that affect verifiable Internet voting schemes in general.

User Study Results

The effectiveness of the detection of incorrect votes is important because it directly contributes to the security of the voting scheme. Using the original interface, between 42% (exploratory study) and 33% (main study) of incorrect votes were not detected because the participants did not check the return codes. This might be rooted in 1) the misleading procedure indication on the original code sheet, 2) the participants not reading the instructions in the voting software and/or 3) in the fact that checking return codes is a mental task that is not required to proceed. To proceed to the next step, voters only have to provide the confirmation code. The checking of the return codes is easily omitted, as our studies confirm. The redesign improved the detection rate to 100%. Although the improved code sheet depicts the correct procedure, there are two further reasons for the increased effectiveness: 1) there is only one task per step, and 2) the buttons for proceeding contain a clear statement. Instead of neutral "next"-button, we opted for the statements "yes, all codes match" and "no, at least one code does not match". This introduces the task of choosing one option and likely prompted the participants to read the instructions more carefully.

The participants in the redesign group needed on average 2.5*min* longer than those who used the original. This increment is rooted in the additional screens and the information but also in the aspect that more participants checked the return codes. Since the redesign prompts voters to compare the codes, this takes longer than proceeding to the next step. As our results show, the participants using the redesign spent on average 141*s* on the return and confirmation code pages compared to 48*s* in the original. We argue that the increment of time does not constitute an issue because it reflects the duration required for a successful verification more accurately. Furthermore, the participants did not make any negative statements regarding the duration. Switzerland follows the principle "security before speed" showing that security has to be paramount.

Both interfaces received similar SUS scores. This shows that the additional instructions that lead to increased effectiveness and execution time do not negatively impact subjective usability. Considering user experience, no difference in the perceived effectiveness was found although the participants using the redesign needed significantly longer. This indicates that the user experience is not negatively impacted by the redesign. The user experience scale of perspicuity refers to the ease of getting familiar with a new product and learning how to use it [36]. The redesign was rated significantly better considering the perspicuity which confirms that the instructions and interface elements better support the voters during verification. Dependability shows whether users feel in control and their perceived security and predictability of a product [36]. Also in this domain, the redesign received significantly better ratings. This shows that the transparency given by a successful verification procedure and the access to information in the redesign impacts the security perception of voters.

Usability is Not Enough

In individual-verifiable Internet voting schemes, voters play an active part in assuring the integrity of the election result. This is not restricted to the evaluated Neuchâtel scheme but is also the case for the further classes of schemes presented above. Thus, every voter has to be able to verify. Our redesign

enabled all study participants to detect incorrect votes, but it also revealed limitations of mere usability: voters also need 1) to *understand* why they verify and 2) to *trust* verification.

Without understanding, participants might consider verification to be redundant and skip steps [14, 37]. Our participants could only consider the information available during the study. Thus, most of them only had a limited understanding of what they verified. In a real election, more information material would be available as the case of Switzerland shows [15]. The participants using the redesign demonstrated a better understanding indicating that detailed instructions deliver value. If voters do not trust the voting scheme, they might refrain from using it. The gap between the perceived and real security level of a system can influence users. If security is perceived as too low, users might refuse to adopt a technology, on the other hand, a too high perceived security may lead to engaging in insecure practices [31].

Significantly fewer participants using the original interface were confident that their vote was correctly transmitted and registered. This might be connected to the effectiveness results. Participants who were prompted by the interface to carry out all (mental) tasks, were shown more specifically whether their submitted vote was correct. This might have been more convincing because they were shown that it is possible to check the correctness of a transmitted vote. Participants using the original interface simply interacted with the interface without convincing themselves about the correctness of their transmitted votes. This is also supported by the perceived differences in the user experience scale of dependability.

Our investigation confirms that the interface of voting software and the provision of information on the code sheet influences both: understanding and trust. Therefore, 1) it is crucial for voters to be given the opportunity to gain a deeper understanding why verification is needed and 2) access to information in order to build trust in the provided system.

Final Recommendations

While we investigated the specific interface that is used in Switzerland, our results also affect further individual verifiable Internet voting schemes that are available in the literature [24]. Besides Neuchâtel, there are further return code schemes [4, 22, 29, 35, 10, 40, 25, 34]. While they differ on the protocol level, the user interaction is either similar or even identical to Neuchâtel, thus, our redesign can be adapted for all of these schemes. Furthermore, we conclude with five recommendations to inform the design of individual verifiable Internet voting schemes in general.

1) Make it clear how to act in case of an incorrect vote. The purpose of verification, in general, is to detect incorrect votes. Therefore, voters must be informed on how to act in this case. If the auxiliary material, such as code sheets or other credentials, is distributed to the voters this material should also contain information on how to act if a problem occurs.

2) Ask for one task per verification step. Voters might overlook tasks, such as code comparisons if they have to perform multiple tasks within one step. Therefore, each verification step in the software should contain exactly one task.

3) Ask directly for the outcome of mental tasks. Previous evaluations of voting schemes show that voters frequently have to compare data during verification [38]. This data is either a verification code or a voting option. Such comparisons are mental tasks. Our study indicates that isolating a comparison to a specific step with buttons directly asking for the result of the comparison is likely to deliver improved effectiveness. Therefore, we recommend asking voters directly for the outcome of mental tasks. The interface should also offer immediate action options for all action possibilities. The buttons for proceeding to the next steps should specifically be labeled with the result of the mental task to prompt the voter to act.

4) Provide complete verification information and instructions. Verification is an extra task that is not present in traditional paper-based voting schemes and Internet voters might not anticipate it. Therefore, they need means to familiarize themselves with the procedure before voting commences. Those instructions should match the steps from voting software to ensure good guidance. If a return code scheme is used, the code sheet should also contain instructions to guide voters through the process. In the other verification categories, the instructions should be delivered with the voting credentials or the trusted device. Since code sheets, devices and credentials are delivered via a trusted channel, it is crucial from a security perspective to provide such instructions on another channel because the voting software might be compromised.

5) Include clear labels for verification data. Each code required for verification should be labeled in such a way that the label reflects its purpose. For verification with codes, we recommend the following terminology: The initialization code should be denoted as *login code*, the confirmation code as *vote insertion code* and the finalization code as *acknowledgement code*. Other verifiable voting schemes use different codes for providing verifiability. The purpose of these codes has to be reflected by their label without being too technical. Each document, such as credential letters or code sheets, should be labeled in such a way that their purpose is clear.

LIMITATIONS AND FUTURE DIRECTIONS

The study samples do not represent the general population of voters, because our sample was rather young. Therefore, future studies should extend the sample in terms of diversity of participants to increase generalizability.

In our study, we investigated an election with two contests but referenda and elections might have more. Therefore, our study does not consider scalability aspects connected to the number of contests. Scalability (weakness W6) is particularly challenging since addressing it requires a protocol change. Optical hashes have been demonstrated to improve the usability of code comparisons [51]. This representation might be a viable enhancement and should be reflected in future studies. Therefore, we consider the following three aspects to be of great importance: 1) an investigation into the number of return codes that voters are willing to verify, 2) means to assist the voters in code comparison, and 3) investigating means for reducing the number of return codes, without violating security properties.

ACKNOWLEDGMENTS

This research work has been funded by the Horst Görtz Foundation and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 251805230/GRK 2050. The authors would further like to thank Bane Janjus who supported the data acquisition of the study and Marie-Laure Zollinger and Peter Rønne for providing feedback.

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 (JETS)* 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*, Vol. 17. USENIX Association, Berkeley, CA, USA, 335–348.

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

[5] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies* 4, 3 (2009), 114–123.

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

[7] Josh Benaloh. 2007. Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. In *Proceedings of the Electronic Voting Technology Workshop (EVT)*. USENIX Association, Berkeley, CA, USA, Article 14, 8 pages.

[8] Ted Boren and Judith Ramey. 2000. Thinking Aloud: Reconciling Theory and Practice. *IEEE transactions on professional communication* 43, 3 (2000), 261–278. DOI:http://dx.doi.org/10.1109/47.867942

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

[10] 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 International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Piscataway, NJ, USA, 711–720. DOI: http://dx.doi.org/10.1109/ICDCS.2016.56

[11] Véronique Cortier, David Galindo, and Mathieu Turuani. 2018. A Formal Analysis of the Neuchatel e-Voting Protocol. In *Proceedings of the IEEE European Symposium on Security and Privacy (S&P) ((Euro S&P))*. 430–442. DOI: http://dx.doi.org/10.1109/EuroSP.2018.00037

[12] Véronique Cortier and Ben Smyth. 2013. Attacking and Fixing Helios: An Analysis of Ballot Secrecy. *Journal of Computer Security* 21, 1 (2013), 89–148. DOI: http://dx.doi.org/10.1109/CSF.2011.27

[13] 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, Berkeley, CA, USA, 193–208.

[14] 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 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 605, 13 pages. DOI: http://dx.doi.org/10.1145/3290605.3300835

[15] Swiss Post E-Voting. 2019. Swiss Post. https://www.evoting.ch/. (2019). [Online; accessed: 18-January-2019].

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

[17] Uwe Flick. 2014. *An Introduction to Qualitative Research* (5 ed.). Sage, London.

[18] 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. DOI: http://dx.doi.org/10.1007/s10209-011-0253-9

[19] David Galindo, Sandra Guasch, and Jordi Puiggalí. 2015. Neuchâtel's Cast-as-Intended Verification Mechanism. In *Proceedings of the International Conference on E-Voting and Identity (VoteID)*. Springer-Verlag, Cham, Switzerland, 3–18. DOI: http://dx.doi.org/10.1007/978-3-319-22270-7_1

[20] 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)*. Springer-Verlag, Cham, Switzerland, 243–268. DOI: http://dx.doi.org/10.1007/3-540-46088-8_21

[21] Micha Germann and Uwe Serdült. 2014. Internet Voting for Expatriates: The Swiss Case. *JeDEM-eJournal of eDemocracy and Open Government* 6, 2 (2014), 197–215. DOI: http://dx.doi.org/10.29379/jedem.v6i2.302

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

[23] Gurchetan S. Grewal, Mark D. Ryan, Liqun Chen, and Michael R. Clarkson. 2015. Du-Vote: Remote Electronic Voting with Untrusted Computers. In *Proceedings of the Computer Security Foundations Symposium (CSF)*. IEEE, Piscataway, NJ, USA, 155–169. DOI: http://dx.doi.org/10.1109/CSF.2015.18

[24] Sandra Guasch Castelló. 2016. *Individual Verifiability in Electronic Voting*. Ph.D. Dissertation. Universitat Politécnica de Catalunya.

[25] 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*. Springer-Verlag, Cham, Switzerland, 73–91. DOI: http://dx.doi.org/10.1007/978-3-319-52240-1_5

[26] Rolf Haenni, Reto E. Koenig, Philipp Locher, and Eric Dubuis. 2017. CHVote System Specification. Cryptology ePrint Archive, Report 2017/325, (2017), 1–325. https://eprint.iacr.org/2017/325.pdf.

[27] 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. DOI: http://dx.doi.org/10.1007/978-3-319-22270-7_3

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

[29] 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-Verlag, Cham, Switzerland, 166–177. DOI:http://dx.doi.org/10.1007/978-3-540-77493-8_15

[30] Hsu-Chun Hsiao, Yue-Hsun Lin, Ahren Studer, Cassandra Studer, King-Hang Wang, Hiroaki Kikuchi, Adrian Perrig, Hung-Min Sun, and Bo-Yin Yang. 2009. A Study of User-Friendly Hash Comparison Schemes. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*. IEEE, Piscataway, NJ, USA, 105–114. DOI: http://dx.doi.org/10.1109/ACSAC.2009.20

[31] Ding-Long Huang, Pei-Luen Patrick Rau, Gavriel Salvendy, Fei Gao, and Jia Zhou. 2011. Factors Affecting Perception of Information Security and Their Impacts on IT Adoption and Security Practices. *International Journal of Human-Computer Studies* 69, 12 (2011), 870–883. DOI: http://dx.doi.org/10.1016/j.ijhcs.2011.07.007

[32] Fatih Karayumak, Michaela Kauer, M. Maina Olembo, Tobias Volk, and Melanie Volkamer. 2011a. User Study of the Improved Helios Voting System Interfaces. In *Proceedings of the 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST)*. IEEE, Piscataway, NJ, USA, 37–44. DOI: http://dx.doi.org/10.1109/STAST.2011.6059254

[33] Fatih Karayumak, Maina M. Olembo, Michaela Kauer, and Melanie Volkamer. 2011b. 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)*. USENIX Association, Berkeley, CA, USA, Article 5, 16 pages. DOI: http://dx.doi.org/10.5445/IR/1000081859

[34] 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-Verlag, Cham, Switzerland, 198–209. DOI: http://dx.doi.org/10.1007/978-3-319-68687-5_12

[35] 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-Verlag, Cham, Switzerland, 468–498. DOI: http://dx.doi.org/10.1007/978-3-662-46803-6_16

[36] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *Proceedings of the HCI and Usability for Education and Work*. DOI: http://dx.doi.org/10.1007/978-3-540-89350-9_6

[37] Karola Marky, Oksana Kulyk, Karen Renaud, and Melanie Volkamer. 2018b. What Did I Really Vote For? On the Usability of Verifiable E-Voting Schemes. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 176, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3173750

[38] Karola Marky, Oksana Kulyk, and Melanie Volkamer. 2018a. Comparative Usability Evaluation of Cast-as-Intended Verification Approaches in Internet Voting. In *Proceedings of the "SICHERHEIT 2018"*. Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn, Germany, 197–208. DOI: http://dx.doi.org/10.18420/sicherheit2018_15

[39] Karola Marky, Marie-Laure Zollinger, Markus Funk, Peter Ryan, and Max Mühlhäuser. 2019. How to Assess the Usability Metrics of E-Voting Schemes. In *Proceeding of the 4th Workshop on Advances in Secure Electronic Voting (VOTING '19)*. Springer.

[40] 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-Verlag, Cham, Switzerland, 206–218. DOI: http://dx.doi.org/10.1007/978-3-319-64248-2_15

[41] Stephan Neumann, Maina M. Olembo, Karen Renaud, and Melanie 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-Verlag, Cham, Switzerland, 246–260. DOI: http://dx.doi.org/10.1007/978-3-319-10178-1_20

[42] Federal Chancellery of Switzerland. 2018. Federal Chancellery Ordinance on Electronic Voting (VEleS). https://www.bk.admin.ch/dam/bk/de/dokumente/pore/Federal_Chancellery_Ordinance_on_Electronic_Voting_V2.0_July_2018.pdf.download.pdf/Federal_Chancellery_Ordinance_on_Electronic_Voting_V2.0_July_2018.pdf. (2018). [Online; accessed: 18-January-2019].

[43] POLYAS GmbH. 2019. Overview of Polyas Costumers. POLYASKundenÜbersicht. (2019). [Online; accessed: 26-December-2019].

[44] Paulo Realpe-Muñoz, César A. Collazos, Julio Hurtado, Toni Granollers, Jaime Muñoz-Arteaga, and Jaime Velasco-Medina. 2017. Eye Tracking-Based Behavioral Study of Users Using E-Voting Systems. *Computer Standards & Interfaces* 55 (2017), 182–195. DOI: http://dx.doi.org/10.1016/j.csi.2017.08.004

[45] 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, 176–192. DOI: http://dx.doi.org/10.1007/978-3-662-53357-4_12

[46] 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

[47] U. Serdült, M. Germann, F. Mendez, A. Portenier, and C. Wellig. 2015. Fifteen Years of Internet Voting in Switzerland [History, Governance and Use]. In *Proceedings of the Second International Conference on eDemocracy & eGovernment (ICEDEG)*. IEEE, Piscataway, NJ, USA, 126–132. DOI: http://dx.doi.org/10.1109/ICEDEG.2015.7114482

[48] 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 (CCS '14)*. ACM, New York, NY, USA, 703–715. DOI: http://dx.doi.org/10.1145/2660267.2660315

[49] Tim Storer and Ishbel Duncan. 2004. Polsterless Remote Electronic Voting. *Journal of E-government* 1, 1 (2004), 75–103. DOI:http://dx.doi.org/10.1300/J399v01n01_07

[50] Tim Storer, Linda Little, and Ishbel Duncan. 2006. An Exploratory Study of Voter Attitudes Towards a Pollsterless Remote Voting System. In *Pre-Proceedings of the IAVoSS Workshop on Trustworthy Elections (WOTE)*. 77–86.

[51] 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 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3787–3798. DOI: http://dx.doi.org/10.1145/3025453.3025733

[52] Janna-Lynn Weber and Urs Hengartner. 2009. Usability Study of the Open Audit Voting System Helios. http://www.jannaweber.com/wp-content/uploads/2009/09/858Helios.pdf. (2009). [Online; accessed: 12-June-2018].

[53] Scott Wolchok, Eric Wustrow, J. Alex Halderman, Hari K. Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, and Rop Gonggrijp. 2010. Security Analysis of India's Electronic Voting Machines. In *Proceedings of the Conference on Computer and Communications Security (CCS '10)*. ACM, New York, NY, USA, 1–14. DOI: http://dx.doi.org/10.1145/1866307.1866309

[54] 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-Verlag, Cham, Switzerland, 114–128. DOI: http://dx.doi.org/10.1007/978-3-642-32946-3_10