# What did I really vote for? On the usability of verifiable e-voting schemes

Karola Marky
Oksana Kulyk
Karen Renaud
Melanie Volkamer

# What Did I Really Vote For?
# On the Usability of Verifiable E-Voting Schemes

**Karola Marky**[1], **Oksana Kulyk**[1], **Karen Renaud**[2,3], **Melanie Volkamer**[4,1]
[1]Technische Universität Darmstadt, Darmstadt, Germany, [2]Abertay University, Dundee, Scotland,
[3]University of South Africa, Pretoria, South Africa, [4]Karlsruhe Institute of Technology, Karlsruhe, Germany
{karola.marky, oksana.kulyk, melanie.volkamer}@secuso.org
k.renaud@abertay.ac.uk

## ABSTRACT

E-voting has been embraced by a number of countries, delivering benefits in terms of efficiency and accessibility. End-to-end verifiable e-voting schemes facilitate verification of the integrity of individual votes during the election process. In particular, methods for *cast-as-intended verification* enable voters to confirm that their cast votes have not been manipulated by the voting client. A well-known technique for effecting cast-as-intended verification is the *Benaloh Challenge*. The usability of this challenge is crucial because voters have to be actively engaged in the verification process. In this paper, we report on a usability evaluation of three different approaches of the Benaloh Challenge in the remote e-voting context. We performed a comparative user study with 95 participants. We conclude with a recommendation for which approaches should be provided to afford verification in real-world elections and suggest usability improvements.

### ACM Classification Keywords

H.5.m. Security and privacy: Usability in security and privacy

### Author Keywords

E-voting; End-to-end verifiability; Cast-as-intended Verifiability, Usability Evaluation; Benaloh Challenge

## INTRODUCTION

Elections are the cornerstone of modern democracies. With the explosion of digitization, governments have embraced electronic solutions in many areas, and elections are no exception. Prime examples are Switzerland [43] and Estonia [19], which allow voters to cast votes via remote e-voting channels during national elections.

There are clear advantages to employing remote e-voting as a vote-casting channel. For example, it is trivial for voters living abroad to cast votes, and tallying is more efficient and accurate. Yet it has to be acknowledged that the deployment of technology introduces risks of deliberate vote manipulation. There may be those who wish to illegally influence the outcome of elections using a range of possible attacks [28, 44]. To increase the possibility that such vote manipulations are detected, a number of assurance measures are required, similar to election-monitoring processes for paper-based voting.

Such assurance can be provided if e-voting schemes implement *end-to-end verifiability* [3, 23]. End-to-end verifiable e-voting schemes enable monitoring of each vote processing step. As such, these schemes enable verification that: (1) the voting client encoded the vote matching the voter's intent (*cast-as-intended verifiability*), (2) the vote recorded by the voting system matches the cast vote (*recorded-as-cast verifiability*), and (3) the recorded vote is correctly included in the election result (*tallied-as-recorded verifiability*).

The first of these, cast-as-intended verifiability, presents a particular challenge from the human-computer interaction perspective, since there is currently no way to exclude the voter from the verification process. If the verification process is unusable voters will not be able to verify their vote, and might abort verification because it takes too long. This potentially jeopardizes the integrity of the election.

In this paper, we focus on the usability of a widely-adopted technique for supporting cast-as-intended verifiability: the so-called *Benaloh Challenge* [10, 11], which is implemented by several voting schemes, e.g. [4, 8, 9]. When this challenge has been implemented, voting proceeds as follows. Voters commence voting by making their choice, which the voting client then encrypts. The Benaloh Challenge subsequently gives voters two options: *(1) to vote*, or *(2) to verify*. In the first case they cast the encrypted vote. In the second they verify that the encrypted vote accurately reflects their expressed choice. Because verification, as implemented by the Benaloh Challenge, is not compatible with vote secrecy, verified votes must be discarded. Verification is performed with the assistance of a so-called *verifier*: software running either on the voting or supplementary device, such as a Smartphone.

Three possibilities for performing the Benaloh Challenge in the context of remote e-voting have been proposed in the literature [4,32,38]. The first of these is a *manual approach* [4] which has already been used in legally binding elections, e.g.

*CHI'18 April 21 – 26, 2018, Montreal, QC, Canada.*
Copyright © 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5620-6/18/04...$15.00
https://doi.org/10.1145/3173574.3173750

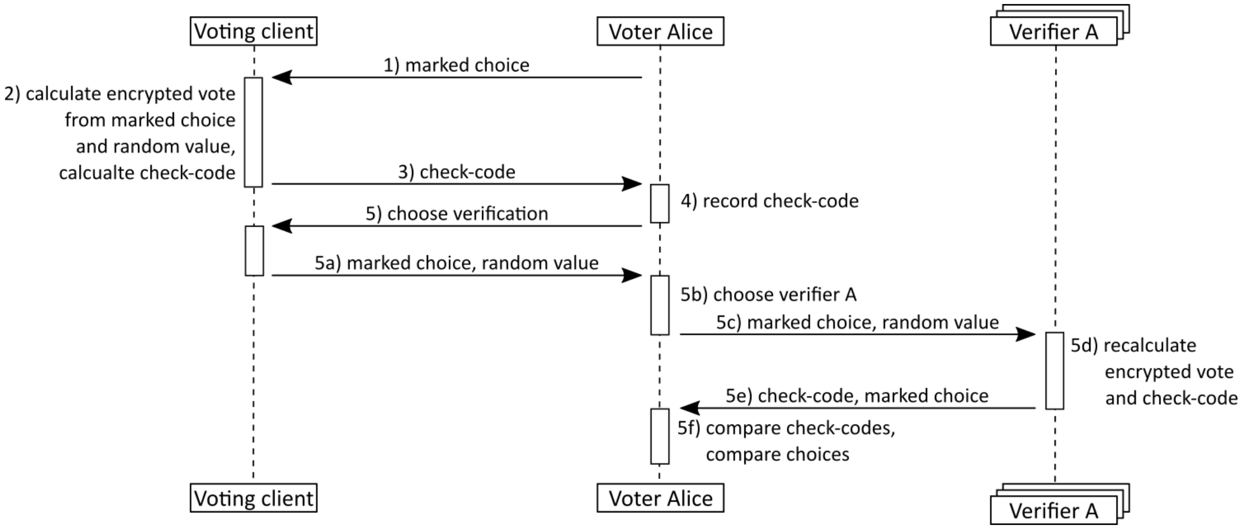

Voting client
Voter Alice
Verifier A
1) marked choice
2) calculate encrypted vote from marked choice and random value, calcualte check-code
3) check-code
4) record check-code
5) choose verification
5a) marked choice, random value
5b) choose verifier A
5c) marked choice, random value
5d) recalculate encrypted vote and check-code
5e) check-code, marked choice
5f) compare check-codes, compare choices
Voting client
Voter Alice
Verifier A

Figure 1: Sequence diagram of one verification with the Benaloh Challenge.

[5, 27, 30]. The second *automatic approach* [32] is a research proposal that offers the same level of security as the manual approach while automating some of the manual steps required by the first approach. Both of these approaches reveal nefarious activities of a malicious voting client, i.e. the software that is responsible for encrypting the vote. Unfortunately, they do not mitigate against other corrupted software on the voter's personal computer. A possibility is that the browser used to access the voting website has been compromised. To offer protection against such an eventuality, the third approach, the *mobile approach* [38], proposes the use of a supplementary device to carry out verification.

In order to determine which approach is most usable, we performed a comparative evaluation in a between-subjects study with 95 participants, which we analyzed quantitatively as well as qualitatively. The quantitative evaluation showed that the automatic and mobile approaches are significantly more efficient than the manual approach. Our qualitative evaluation also revealed that the participants using the mobile approach encountered problems that can be mitigated by improving the implementation. The other approaches, however, suffer from problems that cannot be mitigated by implementation or interface improvements, because they are approach-specific. We thus recommend that the mobile approach is made available during elections. Because the mobile approach requires access to a supplementary device, we recommend also offering the automatic approach as an alternative, to enhance inclusiveness. We conclude this paper by providing recommendations for usability improvements.

## BACKGROUND: THE BENALOH CHALLENGE

A well-known technique for confirming that votes have been "cast-as-intended" is the *Benaloh Challenge* [10, 11]. In a voting scheme that utilizes the Benaloh Challenge, voters have two options after choosing a voting option and having the voting client encrypt their choices. They can either (1) *vote* by casting the encrypted vote, or (2) *verify* that the encrypted vote does indeed reflect their expressed choice.

The power of the Benaloh Challenge is that a manipulator can not predict whether the voter will vote or verify, nor how often he or she will choose to verify. Hence, even if a manipulator has been able to corrupt a voter's device, he or she does not know when it is safe to manipulate the voter's choice. If the manipulator changes the vote and the voter verifies, he or she will detect the manipulation. If the manipulator does not change the vote, and the voter chooses to cast the vote, the opportunity for manipulation is lost. It is this unpredictability that gives the Benaloh Challenge its power. Voters are advised to repeat the challenge until they are satisfied and convinced that the voting client is behaving correctly. They are told that such repetition solidifies assurance, as explained above. The voter, Alice, proceeds as follows (the steps refer to the sequence diagram of the protocol depicted in Figure 1):

1) Alice commences by selecting a voting option, and indicating her choice via the voting client.

2) The voting client then encrypts the choice. During encryption, the voting client generates a *random value* on the fly, ensuring that encryption is individualized.

3) To ease verification, a hash, further denoted as a *check-code*, is derived from the encrypted vote and displayed to the voter. (This code is shorter than the fully encrypted vote, and thus easier for the voter to manage.)

4) Alice takes note of the displayed check-code[1]. She can then choose between one of two actions:

   **(a) to vote:** to complete the process by casting the encrypted vote, she proceeds to Step 6.

   **(b) to verify:** to check that the encrypted vote does indeed reflect her choice, she proceeds to Step 5.

[1] Even if Alice does not verify, she can use the noted check-code later on to ensure that her cast vote has been recorded and tallied correctly.

5) **Verify**. The Benaloh Challenge executes as follows:

   (a) The voting client provides the *verification data*, consisting of Alice's marked choice and the random value.

   (b) Alice chooses a *verifier*.

   (c) Alice transfers the verification data to the verifier.

   (d) The verifier independently calculates an encrypted vote and check-code by using the provided verification data.

   (e) The verifier displays a check-code, together with the choice that was encrypted.

   (f) Alice compares the original and verifier-displayed check-codes and her intended choice to the verifier-displayed one. If they match, she knows that all is well: she has verified that her choice was encrypted correctly.

   The verified vote cannot be cast, because the verification data could later on be used to break vote secrecy. Alice must now discard the verified vote and return to Step 1 to cast her actual vote.

6) **Vote**. Alice casts the encrypted vote, and finalizes the voting process.

Steps 1 to 5 can be traversed as many times as desired, until Alice is satisfied. Step 6 can only be carried out once.

## RELATED WORK

Paper-based voting, as well as e-voting schemes, have been subjected to usability evaluations. Such evaluations include investigations into the usability of paper ballots, paper punch cards and lever machines [14, 20, 26]. Further studies [7,13,16,21,29,48] have focused on the usability of Direct Recording Electronics (DREs), the computers used for voting in polling stations. Smartphone-based systems have also been investigated [15]. The accessibility and usability of E-voting schemes have been evaluated [24, 34, 35]. The usability of polling station schemes has also been investigated [37,50]. All of these studies acknowledge usability as a crucial feature of voting systems, as the voters might make errors that influence the integrity of the election result.

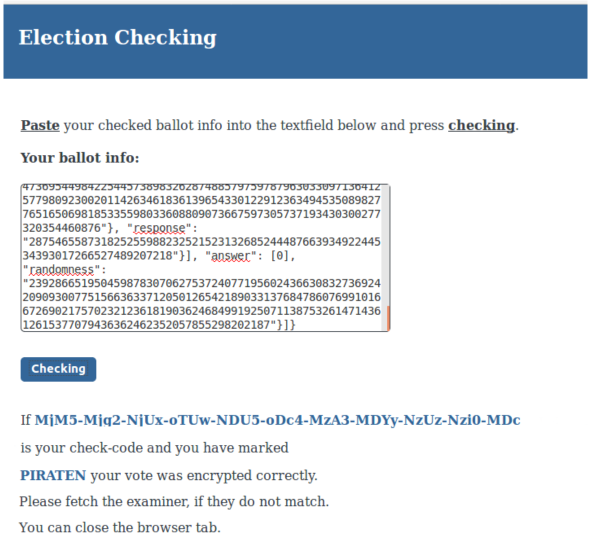
Other research has focused on end-to-end verifiable e-voting schemes which assist voters in "tracing" their vote throughout the election process. Realpe-Muñoz *et al.* performed a user study of the Helios voting scheme and the University of Lleida (UdL) [41] remote e-voting system using an eye-tracking device. Their participants did not realize that they needed to verify their votes. Weber and Hengartner [49] performed a usability study of the Helios implementation of the Benaloh Challenge with 20 participants. Only two of the participants were able to verify successfully, being overwhelmed by the displayed verification data. Acemyan *et al.* [1] investigated the usability of three end-to-end-verifiable voting systems: Helios (incl. the Benaloh Challenge), Prêt á Voter and Scantegrity II in a lab study with 37 participants. Only 43% of participants using the Benaloh Challenge were able to verify their vote successfully. The verification process of the Benaloh Challenge also received a very poor SUS score [12] of 20 out of 100. Neumann *et al.* [38] addressed participants' concerns that a verifier in the Benaloh Challenge might be able to break vote secrecy [31]. Neumann *et al.* aimed to maintain secrecy by hiding the verified choice. Only the random value is transferred. The verifier uses the random value to encrypt all voting options and compute all possible check-codes. Then it displays all possible pairs of voting options and check-codes. Voters are instructed to open the browser's search bar and to search for their check-codes. This approach, however, does not achieve its aims. The verifier's website could use JavaScript to track which check-codes voters search for and thereby detect their choices. It also introduces additional steps into the verification process, potentially impacting usability.

## EVALUATED APPROACHES

We present three possible approaches for conducting the Benaloh Challenge from the literature, which we evaluated and compared in our study.

### Manual Approach

The first evaluated approach is the Helios-implemented Benaloh Challenge [4] further denoted as the *manual approach*. In this case the check-code is a 43-digit cryptographic hash over the encrypted vote. The voter, Alice, has to write it down and choose to verify. The verification data, consisting of her marked choice and the individual random value, are displayed (Step 5(a) in Figure 1). To transfer the verification data to the verifier, Alice selects the data and copies it to her clipboard. The verifier, in this case, is a trusted verification institute that Alice chooses from a list (Step 5(b)). Any research group, company or individual can be a verification institute. The verifier's website is automatically launched in another browser tab and Alice pastes the copied verification data into a text box (Step 5(c)). After the verifier has recalculated the encrypted vote and check-code (Step 5(d)), it displays the check-code and the encrypted choice to Alice (Step 5(e)). She can then compare those to the check-code issued by the voting client and her choice (Step 5(f), see also Figure 2).



Election Checking

**Paste** your checked ballot info into the textfield below and press **checking**.

**Your ballot info:**

```
4736954498422544573898326287488579759787963033097136412
5779809230020114263461836139654330122912363494535089827
7651650698185335598033608809073667597305737193430300277
320354460876"}, "response":
"2875465587318252559882325215231326852444876639349224445
343930172665274892072l8"}], "answer": [0],
"randomness":
"2392866519504598783070627537240771956024366308327369244
2090930077515663633712050126542189033137684786076991016
6726902175702321236181903624684991925071138753261471436
126153770794363624623520578552982021 87"}]}
```

If **MjM5-Mjq2-NjUx-oTUw-NDU5-oDc4-MzA3-MDYy-NzUz-Nzi0-MDc**

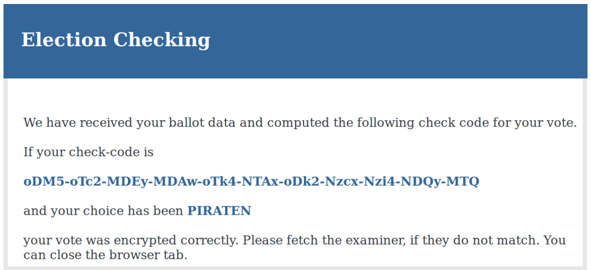
is your check-code and you have marked

**PIRATEN** your vote was encrypted correctly.

Please fetch the examiner, if they do not match.

You can close the browser tab.

Figure 2: Result of the verification in the manual approach.



Election Checking

We have received your ballot data and computed the following check code for your vote.

If your check-code is

**oDM5-oTc2-MDEy-MDAw-oTk4-NTAx-oDk2-Nzcx-Nzi4-NDQy-MTQ**

and your choice has been **PIRATEN**

your vote was encrypted correctly. Please fetch the examiner, if they do not match. You can close the browser tab.

Figure 3: Result of the verification in the automatic approach.

### Automatic Approach

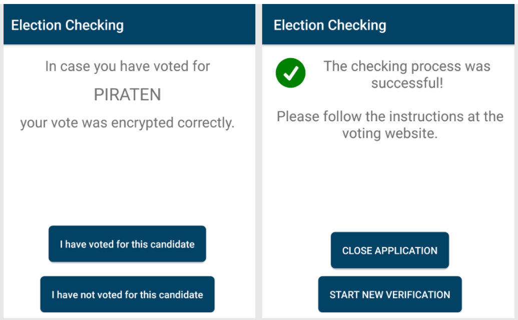
This approach eliminates the copying step required by the manual approach. Karayumak *et al.* [32] performed a cognitive walkthrough of the manual approach and proposed the removal of the copy-paste step from the verification process. Instead of having the voter copy-and-paste the verification data, this data is automatically transmitted to the verifier in Step 5(c) via HTTPS. Thus, after choosing a verifier, the voter, Alice, immediately sees the re-calculated check-code and choice, but still has to manually check that the two are the same. The proposal was implemented and evaluated by Karayumak *et al.* [31] in a user study with 34 participants, but not compared to the manual approach.

For our study, the automatic approach is executed as follows. First Alice records the check-code and opts to verify. Then she chooses a verifier (verification institute) from a list. The verification data is then automatically transmitted to the verifier and the result is displayed (see Figure 3). The subsequent steps do not differ from the manual approach.

### Mobile Approach

The third investigated approach is proposed by Neumann *et al.* [38]. Their approach targets the assumption that the integrity of the election result relies on the use of supplementary devices. Therefore, Neumann *et al.* propose the use of a supplementary device as verifier to carry out verification.

Thus, instead of writing down the check-code in Step 4, Alice scans a QR-code representing the check-code, using a supplementary device such as a Smartphone, further denoted as *mobile device*. In doing so Alice also transfers the check-code to the mobile device. Alice opts for verification and is redirected to a second QR-code containing the verification data. The mobile device scans the second QR-code (Step 5(c)) and uses the data to recalculate the check-code (Step 5(d)). The recalculated check-code is then automatically compared to the previously scanned one. Thus, the comparison of the check-code that Alice performs in Step 5(f) is no longer required. Alice still needs to check that the mobile device used her correct choice for the verification, that is, that the choice displayed on the mobile device matches her intended one (Step 5(f), see also Figure 4). Neumann *et al.* claim that automating the verification steps removes some of the burden from the voter, thereby improving usability. However, they did not formally evaluate their proposal.



Election Checking

In case you have voted for

PIRATEN

your vote was encrypted correctly.

I have voted for this candidate
I have not voted for this candidate

Election Checking

The checking process was successful!

Please follow the instructions at the voting website.

CLOSE APPLICATION
START NEW VERIFICATION

Figure 4: Result of the verification in the mobile approach.

### Comparison of Approaches

The automatic and manual approaches differ only with respect to the transfer of the random value and the marked choice. The manual approach requires deliberate copying and pasting, whereas in the automatic approach it is transmitted automatically. In both approaches Alice has to manually compare her recorded check-code with the one computed by the verifier. Because the same device is used for voting and for verification, the assumption of a trusted device must also hold in the automatic approach. Thus, the automatic and manual approaches offer the same level of security[2]. In the mobile approach all data is transferred by scanning QR-codes. Thus, Alice does not have to record the check-code manually. The mobile device verifier compares the check-codes and informs Alice of mismatches. The deployment of a supplementary device to effect verification weakens the assumption of a trusted voting device with respect to vote integrity and makes the mobile approach more secure than the manual and automatic approaches. Even if the initial voting device (i.e. not just the voting client) has been compromised, manipulations will be revealed if the supplementary device is trustworthy. For further screenshots of the approaches the reader is referred to the paper's supplementary material.

## METHOD

The motivation for our user study was to identify the most usable of three Benaloh Challenge approaches. According to the ISO 9241-11 [45], usability is based on the criteria of effectiveness, efficiency and satisfaction. Based on this ISO standard we formulate our underlying research questions:

- Which of the approaches enables the most voters successfully to verify their vote? [Effectiveness]
- Which of the approaches enables voters to verify their vote more quickly? [Efficiency]
- Which of the approaches leads to the highest levels of user satisfaction? [Satisfaction]

We do not assess the usability of the vote casting process; only that of the different vote verification approaches. Hence, we assess the effectiveness, efficiency and satisfaction of the

[2] The manual approach does not exclude the usage of a supplementary device for verification. The transfer of the verification data to such a device, however, is not supported.

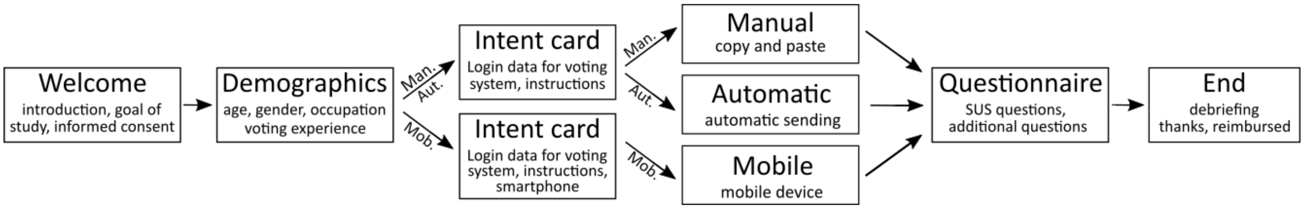

Welcome
introduction, goal of study, informed consent
Demographics
age, gender, occupation voting experience
Man.
Aut.
Mob.
Intent card
Login data for voting system, instructions
Intent card
Login data for voting system, instructions, smartphone
Man.
Aut.
Mob.
Manual
copy and paste
Automatic
automatic sending
Mobile
mobile device
Questionnaire
SUS questions, additional questions
End
debriefing thanks, reimbursed

Figure 5: Study procedure including the paths for the three different groups.

tasks necessary for performing vote verification. We rely on a number of metrics to achieve this. For effectiveness, we consider the share of participants that were able to successfully verify their votes (completion rate). For efficiency, we measure the time that participants required to successfully complete the verification (speed). For satisfaction, we collect SUS scores by asking the questions from the System Usability Scale [12].

The automatic approach does not require the voter to interact with the verification data. Hence, it is anticipated that voters will make fewer mistakes, resulting in higher completion rates. Fewer interactions involving the voter ought also to be less time consuming. We therefore formulate the following hypotheses:

*$H_{1.1}$: The automatic approach is more effective than the manual approach.*

*$H_{1.2}$: The automatic approach is more efficient than the manual approach.*

*$H_{1.3}$: The automatic approach leads to more satisfaction than the manual approach.*

In the mobile approach the voter does not have to record and compare the check-code manually, so a more effective and more efficient execution is anticipated. The corresponding hypotheses are:

*$H_{2.1}$: The mobile approach is more effective than the manual approach.*

*$H_{2.2}$: The mobile approach is more efficient than the manual approach.*

*$H_{2.3}$: The mobile approach leads to more satisfaction than the manual approach.*

There is no research comparing the automatic and mobile approaches. Based on the descriptions in the literature, we formulate the following hypotheses:

*$H_{3.1}$: There are significant differences between the automatic and mobile approaches in terms of effectiveness.*

*$H_{3.2}$: There are significant differences between the automatic and mobile approaches in terms of efficiency.*

*$H_{3.3}$: There are significant differences between the automatic and mobile approaches in terms of satisfaction.*

### Study Scenario

We chose the German Federal Parliament (*Bundestag*) election, which took place on September 24$^{th}$ 2017, as the scenario for our user study. Due to the significance of this election, we anticipated that participants would appreciate the importance of verifying their votes and therefore perform the tasks diligently. Furthermore, this election was current and open to all citizens. We chose to include the BSI, Federal Office for Information Security in Germany, and the OSCE, the Organization for Security and Co-operation in Europe as verification institutes suggested by [38] as being the ones participants were most likely to trust. We included only two of the trusted institutions so as not to overwhelm participants with too many options.

### Study Design and Procedure

We conducted a between-subjects study, meaning that each participant interacted with only one approach, in order to prevent training effects. This resulted in three groups, one group per evaluated approach. We opted for a lab study to ensure we were able to control the environment and so that we could record the screen during interactions with the voting system. The participants were observed by the researcher, who positioned herself so that she could see the participant's actions but not the computer screen. The study follows the guidelines provided by the ethics commission at the author's institution. To test the study design we performed two pre-tests: the first with two participants and the second with one participant. The instructions and intent cards, which are detailed below, were subsequently refined and improved. Our study required participants to step through the following (see also Figure 5 for an overview):

**(1) Welcome and Demographics:** To commence, participants were requested to provide their age, so that we could ensure that they were over 18, or that they had a declaration of consent from a legal guardian to participate. Having ensured that, we then proceeded to inform participants of the purpose and the process of the study, and the fact that screen-recording would take place. The participant was requested to sign a declaration of consent. Subsequently, they were asked for to record age, gender, occupation and previous voting experiences in a questionnaire.

**(2) Intent Card:** In order to determine whether the participant could perform a successful verification in an objective way, we chose to use screen-recordings. However, if the participants cast votes according to their own political preferences, screen-recordings would violate the vote privacy. Hence, we issued participants with an intent card telling them which party to vote for. Because the study aimed to evaluate the usability of verification, and not the true intentions of the participants, the instructions explicitly instructed participants to carry out the verification procedure. The intent card did not specify the steps required for a successful verification, but instructed participants to verify using either the website or mobile device. Participants also received a *faux* letter from the election authority containing login credentials and some space for them to use to write down the check-code. The mobile approach group received a slightly different version mentioning the verification device. If the participant was randomly allocated to the mobile approach group, a Smartphone was provided with the app pre-installed.

**(3) Voting & Verification:** The participants interacted with the approach corresponding to their group, performing the verification procedure and casting a vote as dictated by their intent card. Time measurements were captured by the voting client to assess efficiency. Participants informed the examiner when they had completed all their assigned tasks.

**(4) Questionnaire:** After the participants reported completion, they were handed a questionnaire containing the SUS questions [12]. We also wanted to gain insight into the participants' impressions of the evaluated approaches. Therefore, we included some open-ended questions. For example, we asked whether they experienced any problems while performing verification, whether they thought they would perform verification during an actual election and if so, how often they would carry out verification. Each question contained some space for the participants to justify their answers.

**(5) End:** After completing the questionnaire, participants were given the opportunity to ask questions and were informed that our research was not connected with officials of the German parliament. Finally, the examiner thanked and reimbursed them.

During Steps 3-5 the examiner recorded all participant comments as well as the questions they asked during Step 5. All documents and materials used in our study, as well as the source code of the software and screenshots, are provided in this paper's supplementary material.

## Prototypes and Setup

The manual approach is implemented in Helios and has an open-source implementation. Although the source code is publicly available[3], the development commenced from scratch. The main reason for this was that we wanted to match the interface design to the election scenario from the study, namely, to the German Federal Parliament election. Furthermore, the system used in our study required additional functionality to support data collection, such as timing all interactions. At the same time, the cryptographic functionality of Helios was not essential to the study's aims, and, as such, could be simulated in order to ensure the system had a realistic look and feel. Hence, developing a new system was preferable to adjusting the existing Helios implementation.

A voting website was implemented for each of the three tested approaches and populated with texts and instructions matching the study scenario. All three interfaces were identical except for the verification process. We made the following adjustments to the check-codes in the manual and automatic approaches. We adopted Karayumak *et al.*'s suggestion that the check-code should not contain characters which are hard to distinguish. We did not follow further suggestions by Karayumak *et al.* regarding the representation of check-codes, namely, the suggestion to shorten the check-code to 16 characters. The length of the check-code is a security-critical parameter, and a shorter check-code is unlikely to provide the security required by a real-life election scenario. Hence, we decided to rely on the original Helios check-codes that were 43 characters long. In order to make recording and comparison of the check-code easier, we split the check-code into four character chunks, as shown in [18]. Moreover, in the manual and automatic approaches we added some information about the verification institutes to support the voters in choosing an institute. Finally, in the mobile approach we added functionality to detect incorrect QR-codes, providing instructions on how to proceed. The back-end of the voting websites was implemented in Python (Django v1.11). The front-end was written in JavaScript (Angular). Voter actions were logged in the database as time-stamped actions. The verification application was implemented for Android, no controls were part of the application and no hardware buttons were used. Because verification would always be positive in the experiment, we did not implement the management of a negative verification outcome[4]. The voting website and the verification institutes' websites were locally available via HTTPS. To be able to use HTTPS locally, and to have the browser display it as trusted website, we used a self-issued certificate. The verification application was installed on a lab Android device provided to participants.

## Participants

We recruited 100 participants by contacting secondary schools, advertising on social networks, mailing-lists, flyers, posters and by word-of-mouth. Table 1 provides an overview of our sample. The advertisement included a short description of the study explaining that it was a usability study related to the German Bundestag election.

| *Approach* | *∅ Age* | *Gender [%]* | *Occupation [%]* |
|---|---|---|---|
| Manual<br>($N = 31$) | 18.5 | Male: 38.7<br>Female: 61.3<br>N/A: 0.0 | School: 61.3<br>Univ.: 29.0<br>Other: 9.7 |
| Automatic<br>($N = 32$) | 18.6 | Male: 62.5<br>Female: 37.5<br>N/A: 0.0 | School: 59.4<br>Univ.: 31.2<br>Other: 9.4 |
| Mobile<br>($N = 32$) | 18.4 | Male: 37.5<br>Female: 62.5<br>N/A: 0.0 | School: 53.1<br>Univ.: 34.4<br>Other: 12.5 |
| Overall<br>($N = 95$) | 18.5 | Male: 46.3<br>Female: 53.7<br>N/A: 0.0 | School: 57.9<br>Univ.: 31.6<br>Other: 10.5 |

Table 1: Overview of our sample.

[3] https://github.com/benadida/helios-server, access 03-10-17

[4] The participants were instructed to notify the examiner if the verification step revealed a problem.

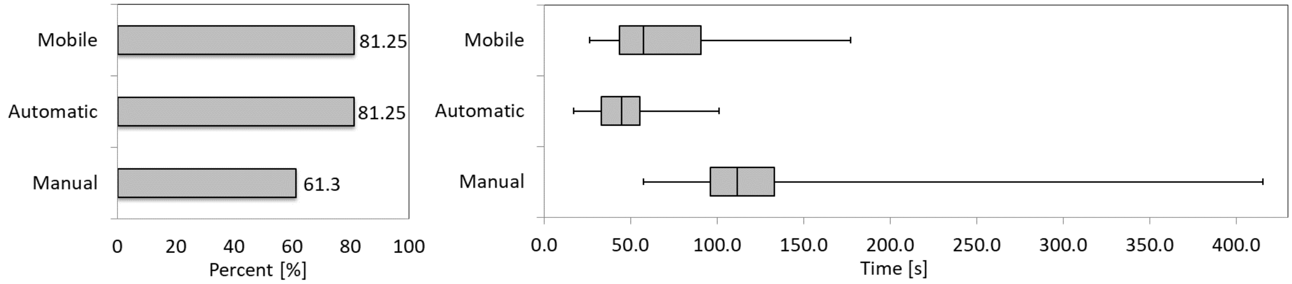


Figure 6: Left: completion rates of the three approaches; Right: box plot of the time needed for a successful verification.

The participants either took part in the study at school or university or received a financial reimbursement of € 10. We restricted the participation to first-time voters (i.e. ages 17-22) for the election of the German Bundestag in 2017 for the following reasons. A survey in 2017 reported that voters between 18 and 29 are the most likely to use remote e-voting in a real election [25]. Hence, if remote e-voting were introduced as an alternative channel to paper voting, young voters would be one of its target users. Our participant choice also allowed us to investigate the usability of the verification approaches without encountering the effects of pre-existing mental models related to Bundestag election experiences.

From the 100 participants, we had to exclude five from further processing, as they either aborted or experienced technical difficulties that prevented us from collecting the necessary data. Of the remaining 95 participants, 46.3% ($N = 44$) were male and 53.7% ($N = 51$) were female, with a median age of 18.5 years. Overall, 2.1% ($N = 2$) had previous e-voting experience and 12.1% ($N = 12$) were technically-related students. The rest were either school pupils, apprentices, university students of non-technical specialties or unemployed.

## QUANTITATIVE RESULTS

We now report the quantitative results of our study, considering the hypotheses outlined above.

### Effectiveness

Overall, the majority of our participants could verify their vote successfully. 61.3% ($N = 19$) were able to perform a successful verification in the manual approach. The automatic approach has a completion rate of 81.25% ($N = 26$). The mobile approach received the same completion rate of 81.25% ($N = 26$). Figure 6 (left side) depicts the completion rates of the three approaches. Chi-Square tests do not reveal significant differences between the three tested approaches ($p > .05$). Therefore, the hypotheses $H_{1.1}$ (the automatic approach is more effective than the manual approach) and $H_{2.1}$ (the mobile approach is more effective than the manual approach) cannot be supported. Furthermore, $H_{3.1}$ (there are significant differences between the automatic and mobile approaches regarding effectiveness) can also not be supported, since they received the same results. From the screen-recordings we observed the progress of participants that commenced verification but aborted, which allowed us to identify points of failure that need to be addressed. 12.5% ($N = 4$) in the automatic approach only opened the list of institutes and then aborted. In the mobile approach 9.4% ($N = 3$) scanned the QR-code containing the check-code but did not proceed further. 6.25% ($N = 2$) aborted verification because, according to the feedback provided in the questionnaire, the scan took too long. One participant per group (3.1%) did not attempt verification at all.

### Efficiency

We recorded the time taken for a successful verification as an interval that starts when the participant presses the verification button and ends when the participant returns to the voting client. On average, the manual approach took roughly three times as long as verification using the automatic approach (see Table 2, also Figure 6 on the right). The manual approach shows a high standard deviation, which is rooted in the results of two participants. Participants using the mobile approach were, on average, twice as fast as participants using the manual approach. Efficiency was analyzed by a Welch ANOVA, including only participants who successfully verified their vote. The Welch ANOVA shows significant differences between the three test groups ($F(2,68) = 17.668,\ p < .001$). In order to locate the differences we ran post-hoc tests using the Bonferroni error correction. The post-hoc tests show that the manual approach is estimated to be on average 83.9 seconds slower than the automatic approach ($p < .001,\ \eta^2 = .364$). This supports $H_{1.2}$ meaning the automatic approach is more efficient than the manual approach. Furthermore, the post-hoc tests show that the manual approach is estimated to be, on average, 70.2 seconds slower than the mobile approach ($p < .001,\ \eta^2 = .260$). This supports $H_{2.2}$ showing that the mobile approach is also more efficient than the manual approach. The post-hoc tests for the automatic, versus the mobile approach, do not reveal significant differences between these proposals ($p > .05$). Thus, the hypothesis $H_{3.2}$ (there are significant differences between the automatic and mobile approaches regarding efficiency) cannot be supported.

| *Approach* | *N* | *Time [s]* | *SD* |
|---|---|---|---|
| Manual | 19 | 131.3 | 83.3 |
| Automatic | 26 | 47.3 | 20.1 |
| Mobile | 26 | 61.1 | 33.8 |
| Overall | 71 | 74.8 | 59.6 |

Table 2: Durations for a successful verification.

### Satisfaction

The automatic approach received the highest mean SUS score with 79.4 ($N = 26$, $SD = 15.5$) out of 100 points. The manual approach ($Mean = 75.4$, $N = 19$, $SD = 15.6$) and the mobile approach ($Mean = 75.8$, $N = 26$, $SD = 12.4$) received relatively similar mean SUS scores. A one-way ANOVA did not show significant differences between the three approaches ($p > .05$). Thus, the hypotheses $H_{1.3}$, $H_{2.3}$ and $H_{3.3}$ cannot be supported. The mean SUS score of the three approaches of 76.5 points to a "C" on the grade scale (with as "A" best and as "F" worst result) [6].

## QUALITATIVE RESULTS

In this section we report the results of the questionnaire analysis, namely, the open-ended questions that we asked after the SUS-related questions. Our goal thereby was to gain a deeper insight into the usability problems that the participants experienced during verification.

The questions that we evaluated were: *"Did you experience any problems during vote verification? If yes, which ones?"* and *"Do you have any additional comments or feedback?"*. We analyzed the answers to these questions using an open-coding approach [22]. There were three coding phases: (1) developing the dictionary, (2) coding the answers by approach, and (3) analysis of the results.

To develop the code dictionary two coders independently reviewed the answers of the open-ended questions and proposed a list of categories. The list was discussed by the two coders and agreed upon. The final dictionary contained the following five codes: (1) *copy-paste issues, (2) check-code issues, (3) missing feedback issues, (4) comprehension-related issues and (5) QR-code issues*. The code *copy-paste issues* is specific to the manual approach. The code *QR-code issues* is specific to the mobile approach. The code *check-code issues is specific to the manual approach and automatic approach.* All other codes apply to all approaches. Two of the paper's authors coded all answers independently (grouped by the three investigated approaches). The level of agreement was 76.5%. The findings were discussed and final allocations agreed upon. We report the results by approach.

### Manual Approach

Based on the answers given by the participants, we identified four distinct issues: (1) copy-paste issues, (2) check-code issues, (3) missing feedback issues, and (4) comprehension-related issues.

#### *(1) Copy-Paste Issues*

The participants reported difficulties understanding what they had to copy and paste and what the displayed verification data is. They either misunderstood the instruction and thought that they had to copy and paste the check-code, or they did not understand what to copy and paste at all. Sample comments given by the participants are[5]:

- "*I thought that I had to copy and paste the check-code [...]*"
- "*I did not immediately know what to copy to where.*"

#### *(2) Check-Code Issues*

The participants' answers revealed usability problems related to the check-code. It has 43 characters comprising both numbers and letters. This sequence is admittedly long and caused confusion. Hence, mistakes in writing it down and comparing it in the last verification step are understandable. These mistakes can lead to a failure to notice a mismatch i.e. failing to detect a vote manipulation. Alternatively, if voters mistakenly write down the wrong check-code, and therefore notice a mismatch during comparison, it could lead them to conclude that a manipulation had occurred, thus undermining their trust in the system. Note that this issue has also been reported in other usability studies of the Benaloh Challenge [31, 32, 49]. Sample comments given by the participants are:

- "*The verification with the code is annoying. It is too long.* "
- "*I think it is easily possible to misread or mistype the check-code, because of its length [...]*"

#### *(3) Missing Feedback Issues*

The participants reported that they were unsure about the current state of verification: the interface did not report the status of verification clearly. Furthermore, the lack of format checking during copy-pasting was problematic. A sample comment is:

- "*At first I pasted [the choice] SPD, which did not work. Then I entered my check-code, also didn't work. In all cases I did not receive a notification whether or not the verification was successful.*"

#### *(4) Comprehension-Related Issues*

Some participants expressed dissatisfaction with the interface instructions. They considered some instructions to be difficult to understand. Sample comments are:

- "*The wording of some instructions was not understandable/inconclusive.*"
- "*I had to read it [the instructions] several times.*"

### Automatic Approach

All the answers related to usability problems in the automatic approach group were related to check-code issues. The check-code issues reported by the participants are analogous to those reported in the manual approach group. Sample comments are:

- "*Copying and comparing the code is cumbersome. Mistakes regarding long character sequences are human, thus likely to occur [...]*"
- "*I found that the code was too long and the repeating letters are confusing.*"

### Mobile Approach

The usability problems of the mobile approach were related to the use of QR-codes. In particular, the participants reported problems regarding scanning duration. A voter willing to perform verification might be discouraged by a too-long scan duration and abort the verification. While the majority of scans were very fast, the scanning duration was characterized by large variations from 2 up to 150 seconds per QR-code. Sample comments given by the participants are:

[5] The comments are translated from the original German.

- "*It takes relatively long until the check-code is scanned and not every Smartphone has such a good camera to recognize the info on the screen.*"
- "*I aborted verification because scanning the code took extremely long.*"

### Further Findings

The open-ended questions offer two prominent further findings: (1) some participants mistakenly thought that they had completed verification, and (2) some were confused by the fact that verified votes could not subsequently be cast.

*(1) Participants mistakenly thought that they had verified*

According to the screen-recordings, 18.75% of participants in the manual approach and 38.7% in the automatic and mobile approaches were unable to complete verification. The open-coding analysis provides insights into the problems the participants experienced, but does not uncover whether the participants mistakenly thought that they had verified. Hence, we compared the results from the screen-recordings with the answers from the open-ended questions to find out about this misperception. In the manual approach 25.8% ($N = 8$) of participants stated in the questionnaire, that they have verified successfully although the screen-recordings clearly show they have failed. In fact, they have started, but aborted the verification process. We also observed this misperception in the automatic approach (12.5%, $N = 4$) and mobile approach (9.4%, $N = 3$). This is evidence of understandability issues or due to insufficient feedback being provided by the interface.

*(2) Confusion that a verified vote cannot subsequently be cast*

Many participants expressed confusion about the nature of the Benaloh Challenge in the open-ended questions. This was not covered by the open-coding analysis, since it not a usability-related problem. In particular, participants were surprised that it was not possible to cast their verified vote. They considered this counter intuitive, because the vote they actually cast is submitted without verification. This shows that participants either need more information about the Benaloh Challenge or an approach in which the cast vote can be verified.

## DISCUSSION & REFLECTION

In considering our results, we first identify and discuss several important areas where the verification was shown to be particularly challenging to the voters and where improvements might be particularly useful. We furthermore make recommendations related to *effectiveness improvements* that apply to all approaches. We conclude with a number of *final recommendations* on which approaches to use in elections.

### Evaluation Results

All three approaches received very similar SUS scores. A possible explanation for this might be that we have captured the SUS score of the overall voting system (i.e. verifying and vote casting). Hence, as vote casting is performed in the same way within all three systems, the SUS scores converge. A study from related work [1], that measured the SUS score of Benaloh Challenge specifically, shows that Benaloh Challenge using the manual approach has very poor satisfaction (SUS score of 20), while having a similar overall SUS score to the one we measured. Considering that, the satisfaction from the verification specifically could significantly differ from the satisfaction related to the overall voting system.

Verification using the automatic and mobile approaches is significantly faster than verification using the manual approach. This makes both approaches more efficient than the manual approach. This is important because verification can, and ought, to be repeated several times and this makes efficiency important.

Our qualitative results reveal a number of problems participants experienced during the evaluation. Participants in the manual approach experienced copy-paste, missing feedback and comprehension-related issues. Of these, missing feedback and comprehension-related issues can be mitigated by improved implementation and a better user interface (i.e. a more prominent indicator of the verification status) or more detailed explanations of the procedure in voting materials. The copy-paste issues, however, are inherent in the manual approach. While the usability of the copy-paste step can be improved with additional instructions, the participants can still be overwhelmed by the displayed cryptographic verification data they have to interact with. This was, among others, one of the findings of the study performed by Weber and Hengartner [49]. Hence, as the verification data would always be displayed if the manual approach is used, the copy-paste issues cannot be fully mitigated.

Participants in the manual and automatic approaches experienced problems with the check-code, in particular, in writing it down and comparing it with a displayed value. These tasks cannot be omitted in either of these approaches, since they are crucial for the verification to take place. Hence, for the sake of improving the usability of the manual and automatic approaches, an easier way to enable comparison of check-code should be provided to the voters. As visual comparisons of complex strings is known to be problematic in a range of disciplines [17,36,47], improvements of the manual and automatic approaches would present a particularly difficult challenge.

The mobile approach automates recording and comparison of check-codes, which removes this burden from the voter. Still, in preparing the prototype and conducting our user study with the mobile approach we uncovered issues related to the use of QR-codes which were not investigated by Neumann *et al.* [38] when they proposed the mobile approach. We describe the QR-codes-related issues and possible solutions to them in more details below.

One issue with the usage of QR-codes is that the durations of QR-code scans are unreliable and exhibit an undesirable variation. Scanning a QR-code is dependent on the display, the ambient light and the quality of the Smartphone camera. Voters might abstain from verification if the scan takes too long. In real-life, a fall-back method for transferring the data from the voting to the verification device will have to be provided to engender resilience. The QR-code size on the computer screen could be adjusted to maximize the chance of a successful scan. The QR-code issue in the mobile approach might be mitigated by an improved implementation.

Another QR-code related issue, which we uncovered while preparing the prototype for the user study, is the amount of data being coded. The verification data in the current Helios implementation comprises a number of cryptographic components which results in $75,309$ characters[6] to be transferred for verification in the scenario from the user study (i.e. an election with one question and 21 voting options). A QR-code, however, has an upper bound of only $7,089$ characters for numeric content and an upper bound of $4,296$ characters in case of alpha-numeric content [40]. Since the prototype in our study is only a click dummy, we simplified the data, to be able to use only one QR-code. In a realistic scenario the QR-code would be composed of alpha-numeric characters, because the data needs to be labeled to distinguish between the check-code, the option and the individual random value. Hence the current data format would not fit into one QR-code. To reduce the data in the QR-code it could contain only a link referring to the verification data, as applied in the Estonian system [44].

### Effectiveness Improvements

Since a successful verification contributes to the integrity of the election result, voters who are willing to perform verification should be able to do so. Although we could not determine significant effectiveness differences between the three approaches, not all participants could successfully verify their votes. Even though between 61.3 and 81.25% were indeed able to verify successfully, the fact that the remaining 18.75 to 38.7% were unable to do so suggests that there is potential for further improvement. It is particularly alarming that in all investigated approaches participants thought they had verified although they have failed. They thought that verification had already occurred: a clear usability failure. It is necessary to convey, more clearly, the status of verification to the voter, ensuring that the need for extra steps is clear in all three approaches. This communication could be achieved by providing a status or progress bar. The current instantiation contains a button labelled "back to voting". If voters click on this button they should be informed that this, in effect, aborts the verification.

### Final Recommendations

Based on our usability evaluation, we recommend the mobile approach for deployment during elections. Many of the problems that participants reported in the questionnaire could be mitigated by improved implementations. However, the mobile approach's use is limited to supplementary device owners (e.g. a Smartphone), of which there were only 78% in Germany, according to the latest survey [46]. Therefore, for greater inclusivity we recommend offering the automatic approach as a fall-back. The automatic approach can serve as an alternative for those who do not own Smartphones.

It has to be acknowledged, however, that the verification data is no longer displayed to the voter in the automatic and mobile approaches. As already proposed by Karayumak *et al.* [32], the verification data should still be available in an auxiliary expert mode. Expert voters can then conduct manual verification, perhaps by using their own programmed verifier.

## LIMITATIONS AND FUTURE DIRECTIONS

The user study was restricted to first-time voters (age 17-22 years) of the German Federal Parliament. This affects generalizability to all eligible German voters. Further studies should therefore include different age groups and bigger samples.

In order to better study the effects of different verification approaches on the satisfaction of the voters, it would be beneficial to capture the SUS score of the verification task only. Since the user study presented in [1] shows significant differences between the overall voting system and the verification task, it should be investigated in future research for the three approaches that we investigated in this paper.

With respect to efficiency, the automatic and the mobile approaches demonstrate a significant improvement over the manual approach. However, to perform verification using the mobile approach, the voter has to search, download and install the verification application. The time taken for these tasks is not included in our timings. Furthermore, for better assurance a voter might be willing to use more than one verification application from different institutions. Thus, even if the mobile approach was, on average, 70.2 seconds faster than the manual approach, it is likely that voters would require additional time to search, download and install the verification application. Because of that, we can only conclude that the actual verification, using the mobile approach, is more efficient than the manual approach.

The focus of this study was solely on usability. As such, the participants were not incentivized to verify and did not receive any instructive or informative material apart from the information provided by the voting website. Results from the additional questions underline the importance of voter motivation and the understandability of the voting system. Participants who thought that verification was complex or time-consuming did not want to verify. Furthermore, participants stated that they do not wish to verify, because they did not understand the purpose of it. Especially, the fact that the verified vote could not be cast was confusing for voters. 22.1% of overall participants mentioned this in the questionnaire. It is unlikely that voters would perform verification in a real election if these perceptions persist. Therefore, investigating understandability, and its impact on verification, should be the focus of future research.

While mental models of verifiability in electronic voting have already been explored e.g. by [2, 39], research in this direction should be extended further in order to investigate the understandability of verification approaches and its effects on the voters' willingness to verify. Finally, alternatives to the Benaloh Challenge that provide cast-as-intended verifiability, such as return code approaches [42], should be considered in future investigations. Previous research has already started to investigate the usability of this approach [33], and as such, a comparison of its usability with the usability of the Benaloh Challenge should be performed.

## REFERENCES

1. Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach. 2014. Usability of voter verifiable,

[6] Number obtained from the Helios Online Demo https://heliosvoting.org/, accessed 08-25-17

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. Claudia Z. Acemyan, Philip Kortum, Michael D. Byrne, and Dan S. Wallach. 2015. Users' Mental Models for Three End-to-End Voting Systems: Helios, Prêt à Voter, and Scantegrity II. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 463–474.
3. Ben Adida. 2006. Advances in Cryptographic Voting Systems. https://dspace.mit.edu/handle/1721.1/96589. (2006). [Doctoral Dissertation].
4. Ben Adida. 2008. Helios: Web-based Open-Audit Voting. In *USENIX Security Symposium*, Vol. 17. USENIX Association, Berkeley, CA, USA, 335–348.
5. Ben Adida, Olivier De Marneffe, Olivier Pereira, Jean-Jacques Quisquater, and others. 2009. Electing a University President using Open-Audit Voting: Analysis of Real-World use of Helios. *Electronic Voting Technology Workshop/Workshop on Trustworthy Elections* EVT '09 (2009).
6. 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.
7. Benjamin B. Bederson, Bongshin Lee, Robert M. Sherman, Paul S. Herrnson, and Richard G. Niemi. 2003. Electronic Voting System Usability Issues. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 145–152.
8. Susan Bell, Josh Benaloh, Michael D. Byrne, Dana DeBeauvoir, Bryce Eakin, Gail Fisher, Philip Kortum, Neal McBurnett, Julian Montoya, Michelle Parker, and others. 2013. STAR-Vote: A Secure, Transparent, Auditable, and Reliable Voting System. *USENIX Journal of Election Technology and Systems (JETS)* 1, 1 (2013), 18–37.
9. Jonathan Ben-Nun, Niko Fahri, Morgan Llewellyn, Ben Riva, Alon Rosen, Amnon Ta-Shma, and Douglas Wikström. 2012. A New Implementation of a Dual (Paper and Cryptographic) Voting System. In *Electronic Voting*. 315–329.
10. Josh Benaloh. 2006. Simple Verifiable Elections. *Electronic Voting Technology Workshop* EVT '06 (2006).
11. Josh Benaloh. 2007. Ballot Casting Assurance via Voter-Initiated Poll Station Auditing. *Electronic Voting Technology Workshop* EVT '07 (2007).
12. John Brooke. 1996. SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7.
13. Jurlind Budurushi, Karen Renaud, Melanie Volkamer, and Marcel Woide. 2016. An Investigation into the Usability of Electronic Voting Systems for Complex Elections. *Annals of Telecommunications* 71, 7-8 (2016), 309–322.
14. Michael D. Byrne, Kristen K. Greene, and Sarah P. Everett. 2007. Usability of Voting Systems: Baseline Data for Paper, Punch Cards, and Lever Machines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 171–180.
15. Bryan A. Campbell, Chad C. Tossell, Michael D. Byrne, and Philip Kortum. 2014. Toward more Usable Electronic Voting: Testing the Usability of a Smartphone Voting System. *Human Factors* 56, 5 (2014), 973–985.
16. Frederick G. Conrad, Benjamin B. Bederson, Brian Lewis, Emilia Peytcheva, Michael W Traugott, Michael J. Hanmer, Paul S. Herrnson, and Richard G. Niemi. 2009. Electronic Voting Eliminates Hanging Chads but Introduces New Usability Challenges. *International Journal of Human-Computer Studies* 67, 1 (2009), 111–124.
17. Lynn A. Cooper and Peter Podgorny. 1976. Mental Transformations and Visual Comparison Processes: Effects of Complexity and Similarity. *Journal of Experimental psychology: Human perception and performance* 2, 4 (1976), 503.
18. Sergej Dechand, Dominik Schürmann, Karoline Busse, Yasemin Acar, Sascha Fahl, and Matthew Smith. 2016. An Empirical Study of Textual Key-Fingerprint Representations. In *USENIX Security Symposium*. USENIX Association, 193–208.
19. Estonian National Electoral Committee. 2015. Statistics about Internet Voting in Estonia. https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia. (2015). [Online; accessed 28-August-2017].
20. Sarah P. Everett, Michael D. Byrne, and Kristen K. Greene. 2006. Measuring the Usability of Paper Ballots: Efficiency, Effectiveness, and Satisfaction. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50. SAGE Publications Sage CA: Los Angeles, CA, 2547–2551.
21. Sarah P. Everett, Kristen K. Greene, Michael D. Byrne, Dan S. Wallach, Kyle Derr, Daniel Sandler, and Ted Torous. 2008. Electronic Voting Machines versus Traditional Methods: Improved Preference, Similar Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 883–892.
22. Uwe Flick. 2014. *An Introduction to Qualitative Research* (5 ed.). Sage, London.
23. Rojan Gharadaghy and Melanie Volkamer. 2010. Verifiability in Electronic Voting-Explanations for Non Security Experts.. In *Electronic Voting*. 151–162.
24. Juan E. Gilbert, Yolanda McMillian, Ken Rouse, Philicity Williams, Gregory Rogers, Jerome McClendon, Winfred Mitchell, Priyanka Gupta, Idong Mkpong-Ruffin, and E Vincent Cross. 2010. Universal Access in E-Voting for the Blind. *Universal Access in the Information Society* 9, 4 (2010), 357–365.

25. Kaspersky Labs GmbH. 2017. Stimmabgabe per Klick - So steht Deutschland zum Thema Online-Wahl. http://newsroom.kaspersky.eu/fileadmin/user_upload/de/Downloads/PDFs/Kaspersky-Studie_Stimmagbabe_per_Klick.pdf. (2017). [Online; accessed: 13-September-2017].

26. Kristen K. Greene, Michael D. Byrne, and Sarah P. Everett. 2006. A Comparison of Usability Between Voting Methods. *Electronic Voting Technology Workshop* EVT '06 (2006).

27. Stuart Haber, Josh Benaloh, and Shai Halevi. 2010. The Helios E-Voting Demo for the IACR. International Association for Cryptologic Research. (2010).

28. J. Alex Halderman and Vanessa Teague. 2015. The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election. In *International Conference on E-Voting and Identity*. Springer, 35–53.

29. Jeffrey Hsu and Gary Bronson. 2018. E-Voting Technologies Usability: A Critical Element for Enabling Successful Elections. In *Emerging Challenges in Business, Optimization, Technology, and Industry*. Springer, 61–78.

30. IACR. 2016. IACR Elections. http://www.iacr.org/elections. (2016). [Online; accessed 19-January-2017].

31. Fatih Karayumak, Michaela Kauer, M. Maina Olembo, Tobias Volk, and Melanie Volkamer. 2011a. User Study of the Improved Helios Voting System Interfaces. In *2011 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST)*. IEEE, 37–44.

32. 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 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections (EVT/WOTE'11)*. USENIX Association, Berkeley, CA, USA.

33. Oksana Kulyk, Stephan Neumann, Jurlind Budurushi, and Melanie Volkamer. 2017. Nothing Comes for Free: How Much Usability Can You Sacrifice for Security? *IEEE Security & Privacy* 15, 3 (2017), 24–29.

34. Seunghyun Tina Lee, Yilin Elaine Liu, Ljilja Ruzic, and Jon Sanford. 2016. Universal Design Ballot Interfaces on Voting Performance and Satisfaction of Voters with and without Vision Loss. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 4861–4871.

35. Seunghyun Tina Lee, Yilin Elaine Liu, Xiao Xiong, and Jon Sanford. 2013. Development of a More Universal Voting Interface. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 57. SAGE Publications Sage CA: Los Angeles, CA, 1624–1628.

36. Robert H. Logie. 2014. *Visuo-spatial Working Memory*. Psychology Press.

37. Damien Mac Namara, Ted Scully, and Paul Gibson. 2011. DualVote Addressing Usability and Verifiability Issues in Electronic Voting Systems. (2011). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.399.7284.

38. 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 *International Conference on Electronic Government and the Information Systems Perspective*. Springer, 246–260.

39. Maina M. Olembo, Steffen Bartsch, and Melanie Volkamer. 2013. Mental Models of Verifiability in Voting. In *International Conference on E-Voting and Identity*. Springer, 142–155.

40. QR4. 2017. QR Code Data Capacity - How much information can a QR code have ? http://blog.qr4.nl/page/QR-Code-Data-Capacity.aspx. (2017). [Online; accessed: 30-August-2017].

41. 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.

42. Peter YA Ryan and Vanessa Teague. 2009. Pretty Good Democracy. In *Security Protocols Workshop*, Vol. 17. Springer, 111–130.

43. Uwe Serdult, Micha Germann, Fernando Mendez, Alicia Portenier, and Christoph Wellig. 2015. Fifteen Years of Internet Voting in Switzerland [History, Governance and Use]. In *ICEDEG 2015: 2nd International Conference on eDemocracy & eGovernment*. IEEE, 126–132.

44. 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 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 703–715.

45. International Organization For Standardization. 1998. ISO 9241-11: Ergonomics of Human System Interaction – Part 11: Guidance on Usability. (1998).

46. Statista. 2017. Statista - Anteil der Smartphone-Nutzer in Deutschland 2017. https://de.statista.com/statistik/daten/studie/585883/umfrage/anteil-der-smartphone-nutzer-in-deutschland/. (2017). [Online; accessed: 30-March-2017].

47. 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 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3787–3798.

48. Michael W. Traugott, Michael J. Hanmer, Won-ho Park, Paul S. Herrnson, Richard G. Niemi, Ben B. Bederson, and Frederick G. Conrad. 2005. The Impact of Voting Systems on Residual Votes, Incomplete Ballots, and

Other Measures of Voting Behavior. In *Annual Conference of the Midwest Political Science Association, Chicago, IL, April*. 7–10.

49. 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. (2009). [Online; accessed: 22-December-2017].

50. Marco Winckler, Regina Bernhaupt, Philippe Palanque, David Lundin, Kieran Leach, Peter Ryan, Eugenio Alberdi, and Lorenzo Strigini. 2009. Assessing the Usability of Open Verifiable E-Voting Systems: a Trial with the System Prêt à Voter. *Proceedings of ICE-GOV* (2009), 281–296.