Usability Analysis of Helios - An Open Source Verifiable Remote Electronic Voting System

Fatih Karayumak, Maina M. Olembo, Michaela Kauer and Melanie Volkamer
CASED
Technische Universität Darmstadt
{*fatih.karayumak, maina.olembo, michaela.kauer, melanie.volkamer*}@*cased.de*

Abstract

Currently, rather secure cryptographic voting protocols providing verifiability exist. However, without adequate usability and abstraction concepts to explain the voting process and, in particular, the verifiability steps, they are not ready for legally binding elections. This holds in particular for remote electronic voting systems because of the absence of poll workers who can support voters by explaining single steps. In this paper, the usability of the ballot casting and verifiability procedures of the Helios open source end to end verifiable remote electronic voting system is analyzed using the cognitive walkthrough approach by security, electronic voting and usability experts. We demonstrate the need for improvements to the usability and verifiability of Helios, before it is used in large scale elections outside of an academic context. Based on our results, we propose new interfaces for improved usability of Helios and future end to end verifiable electronic voting systems.

1 Introduction

Past approaches to create trust in electronic voting systems and in particular in remote electronic voting were based on system evaluation and certification. Due to negative criticism of existing electronic voting systems, future approaches will additionally be based on the implementation of verifiability to enable voters, election commissions and election observers to verify the integrity of the election results and thus increase transparency and trust in the election.

Verifiability is subdivided into individual verifiability where voters can verify that the ballot is cast as intended and stored as cast, and universal verifiability where everyone can verify that the tallying is correct. Systems providing both types are called E2E (end to end) verifiable. Examples of systems implementing this property in the context of remote electronic voting are proposed in [3, 4, 10, 17] and in the context of polling station election systems in [9, 20].

Many E2E verifiable electronic voting systems have been proposed in cryptography conferences while only a few have been implemented and used for real or test elections. In general, underlying voting protocols are based on complex cryptographic schemes in order to implement verifiability and to ensure election secrecy at the same time. In order to verify the electronic election result, voters need to take additional steps. For instance, to check that the ballot is stored as cast (individual verifiability), the voter gets an encrypted receipt after having cast a ballot. Therefore, a major challenge for developers is to provide a user-friendly and comprehensible interface for ballot casting and processing of verifiability – in particular for an average voter without any background in cryptography.

While several E2E verifiable voting protocols exist, few have been carefully studied in terms of usability for the end-user. It remains unclear if the voters understand the verifiability aspect of these systems and whether they are able to both cast and verify their ballots. This is especially true in the case of remote electronic voting where voters cast their ballot alone from their home computers. With these considerations, the usability of such E2E verifiable voting systems needs further research. We therefore analyzed the Helios remote electronic voting system described in [3]. It is an open source E2E verifiable voting system and enables us to implement an improved interface. Moreover, it has already been used for legally binding elections in academic contexts.

The focus of this paper is the voter's interaction with the system and in particular on ballot casting combined with verifiability mechanisms to verify whether the ballot is cast as intended. Security, electronic voting and usability experts applied the cognitive walkthrough method to assess this interaction. Based on our results we propose improvements to the usability of Helios' interface, which might also be applicable for other E2E verifiable electronic voting systems.

The rest of this paper is structured as follows: In section 2 we discuss related work. We introduce Helios and describe the ballot casting and verifiability processes in section 3, while in section 4 we discuss the cognitive walkthrough approach used. We then provide the results of our usability analysis of the voter interfaces for the ballot casting procedure including individual verifiability in section 5. We propose improvements in section 6, analyze possible security problems which may be caused by these recommendations and discuss countermeasures in section 7 and conclude the paper with a short summary and presentation of future work in section 8. Screenshots of the Helios system and our proposed interface improvements are available in Appendix A.

2 Related Work

There is some research that has been undertaken in assessing ballot design (including instructions to the voter) and its effects on elections (see e.g. [12, 18, 19]). This research has an indirect effect on the design of interfaces for electronic voting systems. While this is important to further improve any voting system, our focus is on verifiability mechanisms, specifically in remote electronic voting.

There also exists a series of papers on the usability of non-remote electronic voting systems (see e.g. [5, 11, 14]). However, most of them analyze voting systems that do not provide verifiability. The few papers which analyze verifiable electronic voting systems include e.g. paper audit trails in [1, 15] and ballot scanning techniques in [7, 8]).

A usability study has already been undertaken on Helios version 1.0 by Weber and Hengartner [22]. In their paper, a mock student government election was created in order to analyze the usability of Helios. The behavior of 20 voters was observed and a user study including an interview and a survey conducted. The work shows two main results: first, half of the participants did not complete the ballot casting process and gave up before submitting their vote. There were various reasons for this outcome, for example script timeout warnings confused voters so that they aborted the voting process. Second, most voters did not understand the security features implemented in the system to ensure integrity either because of the jargon-laden language used or the lack of adequate information. For example, the auditing section did not provide enough information such that only two out of twenty voters actually verified their ballot.

In this paper we analyze the usability of Helios version 3.1. We take into account the results proposed in [22]. However, we use a different evaluation method namely, the cognitive walkthrough method, which allows us to work in an interdisciplinary team and thus identify many more findings and make suggestions for improvements.

3 Helios Voting System

Based on pre-existing cryptographic and web development technologies, the Helios system was designed to provide an accessible E2E verifiable electronic voting solution. It was implemented and presented by Ben Adida.

Helios is far from being just a research project and an experimental prototype of a remote electronic voting system. Different custom deployments of Helios have been used in legally binding elections in academic contexts: the presidential election at the Universitê Catholique de Louvain in March 2009 [4], the undergraduate student government at Princeton University in 2009, as well as a demo election for the International Association of Cryptographic Research (IACR) in 2010 [13]. User guidelines and videos of Helios as used in the Princeton University election are available in [2].

In subsection 3.1 we provide some technical information explaining Helios' security features. After this, in subsection 3.2, we describe the ballot casting process as well as the steps for individual verifiability.

3.1 Technical Description and Security Issues

Helios is an open-source voting system distributed to the public under GPL v3 license. Anyone who is willing to test the system can register on the Helios website and set up an election. The ballot casting application is implemented as a single-page web application using JavaScript. In particular, all necessary data is preloaded into the browser's memory and the JavaScript code updates the rendered HTML user interface when any links or buttons are clicked. As a result, no Internet access is required from the time the data is loaded onto the web browser, until one is ready to cast their vote. Anyone using a modern web browser running JavaScript (e.g. Firefox 2 or later), is able to cast a vote.

Ballot preparation and casting including individual verifiability are based on Benaloh's Simple Verifiable Voting Protocol [6], which is based on two aspects - separating ballot preparation/encryption and ballot casting as well as on Benaloh's challenge. The idea of separation means that the ballot can be viewed, selections can be made, the ballot can be encrypted and the encryption can be verified without having to authenticate oneself and, thus, without being an eligible voter. The voter only needs to be authenticated for the final ballot casting. An advantage of this approach is that everyone (including election observers) can verify the ballot preparation mechanism.

In Benaloh's challenge, the system commits to the encrypted vote and then voters can decide whether they want to verify or cast the vote. The software cannot falsify information by encrypting the wrong candidate since it does not know whether the voter will either verify or cast the encrypted vote. Voters will notice during verification if the wrong candidate name is encrypted. In order to ensure that the software provides the same ciphertext for verifiability and ballot casting (instead of sending the properly encrypted vote to the system in case of verifiability and the ciphertext of a wrong candidate in case of ballot casting), it commits to its encryption by displaying a hash value of the ciphertext, which is the smart ballot tracker.

Privacy requirements make it impossible to cast a verified vote. The verified encrypted vote therefore has to be re-encrypted. Correspondingly, a new hash value is computed and displayed. Thus, the voter cannot verify the encrypted vote they finally cast but must trust the system due to previous checks.

Note that Helios simplifies Benaloh's protocol with the consequence that it looses the benefit of coercion-resistance. As Helios has been designed specifically for use in elections that do not suffer from high coercion risks such as student governments, local clubs or online groups, this is acceptable.

3.2 Voting Procedure

The whole ballot casting and verification procedure from the voter's perspective of Helios (3.1) is depicted in Figure 1 and described below. Note, the corresponding screenshots of all steps can be found in Appendix A and are referenced correspondingly.

First, an invitation email containing the URL of the election page, an election fingerprint, the Voter-ID and the assigned password is sent to the voter (Figure 7). The voter clicks on the URL to open the 'Voting Booth' web-page which contains instructions on the voting procedure. After reading the instructions the voter presses the 'Start' button on this page (Figure 8) to invoke a JavaScript session on their computer. The JavaScript code will lead the voter through the ballot casting procedure without a connection to the Internet. An empty ballot is first displayed (Figure 9). Upon making a selection, depending on the maximum number of candidates allowed, the voter gets a warning message pointing to the limit of the options (Figure 10). To continue the ballot casting process the voter clicks the 'Proceed' button and is forwarded to a page where all selected candidates are displayed. The voter can review their selection here, having then the chance to change this selection by clicking the link 'Update' or seal the ballot by pressing the button 'Confirm Choices and Encrypt Ballot' (Figure 11). To continue the process (Figure 12), the voter can either submit or verify their encrypted vote by pressing the 'Proceed to Cast' button or the 'Audit' link respectively. Note that on this page, the smart ballot tracker is displayed along with two links to 'print' or 'email' the smart ballot tracker. The voter should use either of these options or record the smart ballot tracker for later verification. The voter may also copy and paste this information elsewhere.

If the voter opts to verify, and clicks the 'audit' link, the JavaScript then displays a new page with the mathematical proofs of the encryption as well as instructions on how to verify the encrypted vote (Figure 2). The voter should copy the displayed information, click on the 'Ballot Verifier' link and paste the copied information into the 'Helios Single-Ballot Verifier' page which pops up (Figure 3). On clicking the 'Verify' button, they receive the results at the bottom of the page. As the 'Helios Single-Ballot Verifier' is an independent application with its own window, the voter can go back to the main ballot casting application at any time (e.g. by closing the verification application). In order to continue with the voting process, the voter clicks the 'back to voting' button. After that they will need to re-encrypt the ballot. The voter can then decide whether to cast the encrypted vote or to verify it again.

If voter clicks the 'Proceed to Cast' button they are forwarded to the login page (Figure 13) where the smart ballot tracker is displayed again. The voter enters the Voter-ID and the assigned password from the invitation email in order to proceed. On pressing the 'check credentials' button, the system confirms eligibility. If they are authenticated, the voter can in the next step finally cast the vote (by pressing the 'I am —, cast this ballot' button). Alternatively they can click the 'cancel' button and cancel the election (Figure 14). A success message is displayed once the encrypted vote is successfully cast (Figure 15). The voter is redirected to the election information page (Figure 16) by clicking the link 'return to election info'. This page contains information on the election, verifying procedure and provides a link labeled 'Vote in this election' for the voter to start the election process afresh if they so choose. Finally, a confirmation email is sent to the voter (Figure 17).

4 Cognitive Walkthrough

We applied the cognitive walkthrough technique according to [21] in our analysis. This is a usability inspection technique that uses exploration by experts in the field to evaluate a design for the ease of its learning. Security, electronic voting and usability experts inspected the Helios user interface by going through a fictitious university president election and evaluating the understandability

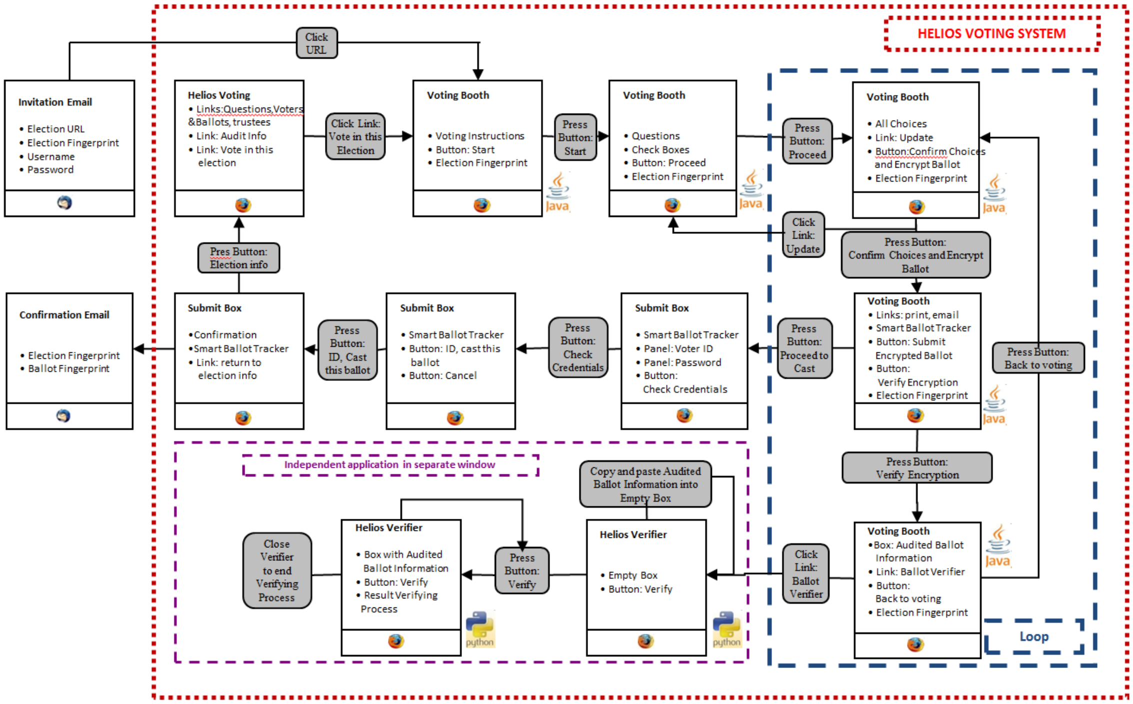


Figure 1: Ballot Casting and Verification Procedure

of the voting procedure. The group of experts included one post-doctoral fellow whose research areas include E-voting and usable security, a usability expert with a background in psychology, two graduate students, and an undergraduate computer science student. The usability expert had no prior exposure or experience with Helios or verifiable remote electronic voting. The two graduate students are familiar with HCI concepts. We observe that the views of the researchers may differ from other researchers due to the differing voting experiences, for example between countries in Europe and North America. In our work, the European perspective on electronic voting is dominant and mostly considered.

During several sessions, each step of the ballot casting process was carefully considered from the point of view of the voter, analyzing the steps they would have to follow and the instructions and clues that would help them along the way. The goal was to capture the functionality provided to the voter in each step and assess whether the provided instructions support the voter to decide which functionality to use and to understand the corresponding next step.

We considered the voters to be non-computer security experts and not to have a background in cryptography. In addition, we assumed that voters are generally more concerned about the functions of the ballot casting itself, and act in four steps:

1. Select a task and identify necessary single steps.
2. Explore the product/system and look for the action which enables the task performance.
3. Select the seemingly most fitting action.
4. Interpret the system response and continue.

To perform the cognitive walkthrough all possible tasks and single steps were collected and noted down. For Helios, the tasks were: ‘cast a ballot’ and ‘verify the ballot’. As an example, the first three steps of the task ‘cast a ballot’ are presented:

1. Read the invitation-to-vote email.
2. Click on the link to access the election website.
3. Press the button ‘Start’ after reviewing the Instructions.

Overall, the task ‘cast a ballot’ consisted of twelve single steps and separately, the task ‘verify the ballot’ consisted of eight steps.

Three questions for each step of each task were then raised:

1. How do the voters know what to do next and is the correct answer sufficiently evident to them?
2. Can the voters connect what they are trying to do with the correct action?

3. Can the voters see if they have made progress?

These questions in combination with an assessment of the steps involved in ballot casting and verifiability led to a collection of challenges that map to the principles of design in the ISO 9241-210 standard [16]. These principles are: suitability for the task, self-descriptiveness, conformity with user expectations, suitability for learning, controllability, error tolerance and suitability for individualization.

We list our findings in section 5 and document recommendations for system improvements in section 6.

5 Results of Analysis

This section presents our analysis of the Helios version with a closed voter register, that is, a voter needs to authenticate themselves using login credentials sent via email. The open voter register of Helios that uses Twitter, Google, Facebook and Yahoo accounts for login was not considered. We assess the challenges voters would face and how these challenges affect their ability to cast a ballot.

Our findings are divided into three categories: general usability (GUF), verifiability procedure (VPF) and usable security (USF) findings. In the first category are the findings that result from the Helios implementation being a work in progress. These are easy to correct. Those in the remaining two categories are more challenging to resolve as average voters are not used to verifiability and may not be familiar or comfortable with computer security issues.

Under each category, we further present our findings based on:

- Wording (W): areas where the voter may not understand the terms used (maps to question 1 in section 4);
- Misleading Information (MLI): here the voter is misled or misinformed by the instructions given (maps to question 1 in section 4);
- Missing Information (MI): where the voter receives little or no instructions or explanations (maps to question 2 in section 4);
- User Interface Elements (UIE): findings relevant to the design aspects of the Helios user interface (maps to question 3 in section 4).

The findings are labeled using the abbreviation <Category>-<Group>.<Number>. A general observation is that the verifiability steps are very tiring and complex for the voter, requiring at least eight steps. It is likely that a voter is unable to verify their ballot or, worse, does not finish the ballot casting procedure.

5.1 General Usability Findings

We identified two findings in the context of wording (W), one regarding misleading information (MLI), two for missing information (MI) and five for usability interface elements (UIE).

GUF-W.1: There are terms used in the ballot casting process that are technical and unfamiliar to the average voter, with the result that they are unable to cast their ballot. Examples are the terms ‘fingerprint’ (Figure 7), ‘encrypt’ (Figure 8 and Figure 11), ‘audit’ and ‘verify’ (Figure 12), and ‘trustees’ (Figure 16). This also holds for whole phrases like ‘encrypted safely inside the browser’ and ‘smart ballot tracker’ (Figure 8) as well as ‘check credentials’ (Figure 13).

GUF-W.2: There is lack of consistency in the terms used, e.g. ‘audit’ and ‘verify’ (Figure 12) refer to the same action. This can be confusing to the voter, causing them to doubt the reliability of the voting system.

GUF-MLI.1: ASCII characters are used in the password and the election fingerprint, both provided in the invitation-to-vote email (Figure 7) as well as in the smart ballot tracker (Figure 12). Consequently, some letters look similar for example, ‘I’ and ‘l’ as well as ‘O’ and ‘0’. Generally voters may be confused if they cannot uniquely identify characters and may be unable to log in to cast their ballot.

GUF-MI.1: The username and password provided in the invitation-to-vote email (Figure 7) are only for use during final vote casting, yet no instructions are provided to this effect. The voter may expect to be authenticated first, as in other secure applications such as online-banking and online payments. This lack of conformity to expectations may lead to confusion and mistrust in Helios and interfere with the voter casting their ballot.

GUF-MI.2: In the instructions to voters (Figure 8), no mention is made whether a voter can deliberately cast a blank ballot, and no check is carried out to confirm whether a blank ballot cast by the voter is intentional or in error. Without a warning message on submission the voter may use the system incorrectly and cast a blank ballot unintentionally.

GUF-UIE.1: The step ‘Submit your encrypted ballot’ in the instructions to voters page (Figure 8) encompasses many more steps than are indicated, such as verifiability. Furthermore, the navigation buttons in the menu bar at the top of the web page (e.g. Figure 9) do not inform the voter how many steps they have left to finish the ballot casting process. The voter may therefore be surprised and confused to find more steps are required.

GUF-UIE.2: A ‘Help’ link is provided (Figure 8), but it is impractical as sending an email may not offer the voter immediate assistance. They may back out and not complete the ballot casting process.

GUF-UIE.3: Features for directing the voter through the vote casting and verifiability processes differ across the web pages. For instance, the proceed button is placed on the right hand side on some web pages (Figure 9), while on others it appears on the left (Figure 12), the voter sometimes has to click on a button to proceed while other times they click on a link for the same function, and the design of the button on the login page (Figure 13) differs from other buttons the voter has previously encountered, e.g. on the candidate selection page (Figure 9). This may result in the voter getting confused while casting their ballot, and they may doubt the reliability of the system. Additionally, the user interface does not easily direct the voter to the right destination, making interaction with the system difficult. For example, there are no back buttons for navigation and the browser back button leads the voter to the very first page and not one step back as is expected.

GUF-UIE.4: After the voter makes the candidate selection (Figure 10), they are informed that the maximum number of candidates has been selected. This message is not necessary and may be confusing to the voter.

GUF-UIE.5: The login page (Figure 13) allows the voter to authenticate themselves and submit the encrypted ballot. The webpage after it (Figure 14) is unnecessary, as the voter should not need to click yet again on a button to submit their ballot. The voter may be confused as they expect that the ballot will be cast as soon as they log in successfully.

5.2 Verifiability Procedure Findings

We identified two findings in the context of wording (W), six regarding misleading information (MLI), ten for missing information (MI) and five for usability interface elements (UIE).

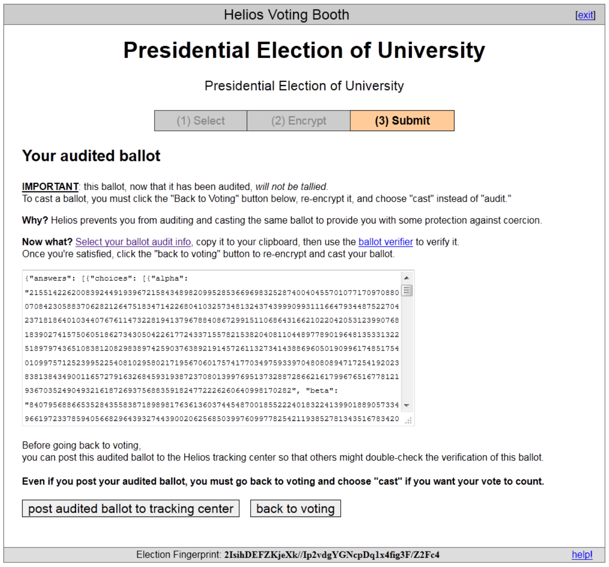
VPF-W.1: There is the use of unfamiliar terms and lack of precise explanations to describe the verifiability process (Figure 12). As a result voters may not be able to either verify the ballot or finish the ballot casting process.

VPF-W.2: Diverse information is displayed in the verification results (Figure 3). First, the election fingerprint and the ballot fingerprint are displayed with the message 'election fingerprint matches the ballot', which could be incomprehensible to the voter, as it is unclear what in particular should be matching. Second, the ballot content is displayed as well as the message 'Encryption Verified', and 'Proofs ok'. The voter may not understand this output. Furthermore, the message 'Proofs ok' does not mean that a manipulation has not taken place. This is only the case if the ballot fingerprint and the ballot content are verified by the voter. This could result in the voter accepting this result and not going on to compare the smart ballot tracker.

VPF-MLI.1: The smart ballot tracker is displayed when the voter is prompted to log in to cast their vote (Figure 13), yet it neither provides any advantages for security nor improves the level of verifiability at this stage. This may result in the voter ignoring the information, disregarding information necessary for verifiability.

VPF-MLI.2: On clicking 'Audit' (Figure 12), the voter is given a lot of information which contains terms they may not understand, e.g. '...reveal how your choices were encrypted' and 're-encrypt'. No indication is given what happens after verification. Further, the voter is given instructions using phrases that may be unclear. They are informed that they will be guided to re-encrypt their choices for 'final casting', but this choice of phrasing may unintentionally influence the voter to only audit the ballot once.

VPF-MLI.3: The voter accesses the auditing webpage (Figure 2), and notes that it bears the title message 'Your audited ballot', this before the voter has carried out the verifiability process. This can cause the voter to think that their ballot is automatically audited, and so they do not go ahead to audit it. Second, the instruction '... to cast a ballot, you must...' may cause the voter to verify the ballot only once.



Presidential Election of University

Presidential Election of University

Your audited ballot

IMPORTANT: this ballot, now that it has been audited, *will not be tallied*.
To cast a ballot, you must click the "Back to Voting" button below, re-encrypt it, and choose "cast" instead of "audit."

Why? Helios prevents you from auditing and casting the same ballot to provide you with some protection against coercion.

Now what? Select your ballot audit info, copy it to your clipboard, then use the ballot verifier to verify it.
Once you're satisfied, click the "back to voting" button to re-encrypt and cast your ballot.

```
{"answers": [{"choices": [{"alpha":
"21551422620083924491939672158434898209952853669698325287400404557010771709708800
070842305883706282126475183471422680410325734813243743999099311166479344875227042
371818640103440767611473228194137967884086729915110686431662102204205312399076818
390274157506051862734305042261772433715578215382040811044897789019648135331322
518979743651083812082983897425903763892191457261132734143886960501909961748517540
109975712523995225408102958021719567060175741770349759339704808089471725419202383
813843490011657279163268459319387237080139976951373288728662161799676516778121
93670352490493216187269375688359182477222626064099817 0282", "beta":
"840795688665352843558387189898176361360374454870018552224018322413990188905733496
619723378594056682964393274439002062568503997609977825421193852781343516783420
```

Before going back to voting,
you can post this audited ballot to the Helios tracking center so that others might double-check the verification of this ballot.

Even if you post your audited ballot, you must go back to voting and choose "cast" if you want your vote to count.

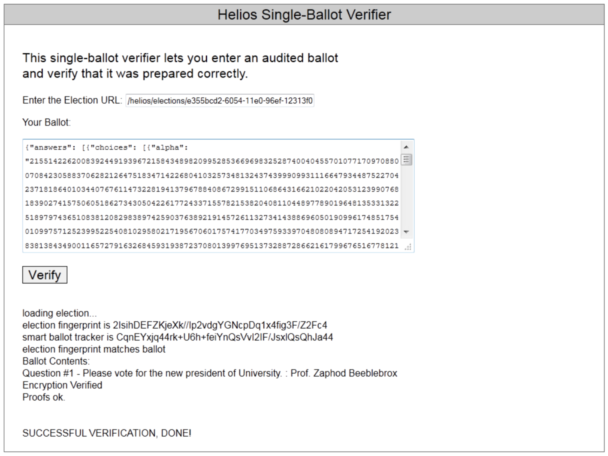
Election Fingerprint: **2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4**

Figure 2: Voting Booth - Audited Ballot

VPF-MLI.4: On choosing to verify the encryption (Figure 12), the voter is directed to the verification page (Figure 2) where the ballot information is presented as a long mysterious string. They may find this string too long and not make the comparison, thereby not verifying the correctness of the ballot.

VPF-MLI.5: After successful verification (Figure 3) the voter may be misled to think that they have successfully cast their ballot since there is no information reminding them to do so. This may result in the voter not verifying or casting their ballot.

VPF-MLI.6: In the verification results page (Figure 3), the voter's candidate selection is displayed in clear text. The voter may have fears regarding their privacy in that the verifier is aware of their vote and may not proceed to cast their ballot.



Helios Single-Ballot Verifier

This single-ballot verifier lets you enter an audited ballot
and verify that it was prepared correctly.

Enter the Election URL: /helios/elections/e355bcd2-6054-11e0-96ef-12313f0

Your Ballot:

```
{"answers": [{"choices": [{"alpha":
"215514226200839244919396721584348982099528536696983252874004045570107717097088O
07084230588370628212647518347142268041032573481324374399909931116647934487522704
23718186401034407676114732281941379678840867299151106864316621022042053123990768
18390274157506051862734305042261772433715578215382040811044897789019648135331322
51897974365108381208298389742590376389219145726113273414388696050190996174851754
01099757125239952254081029580217195670601757417703497593397048080894717254192023
83813843490011657279163268459319387237080139976951373288728662161799676516778121
```

Verify

loading election...
election fingerprint is 2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4
smart ballot tracker is CqnEYxjq44rk+U6h+feiYnQsVvI2IF/JsxIQsQhJa44
election fingerprint matches ballot
Ballot Contents:
Question #1 - Please vote for the new president of University. : Prof. Zaphod Beeblebrox
Encryption Verified
Proofs ok.

SUCCESSFUL VERIFICATION, DONE!

Figure 3: Verification Results

VPF-MI.1: There are no instructions given to the voter (Figure 8) regarding verifiability although this feature is provided by the system. A lack of information may cause the voter to get confused later in the voting process or ignore the verification steps as they have not been made aware of them.

VPF-MI.2: A smart ballot tracker is given after the encryption of the ballot (Figure 12) and appears in several other webpages (Figures 13, 14 and 15), yet the importance and purpose of this information is not indicated, and as a result the voter may ignore it. The ballot fingerprint itself is long and the voter may not be motivated to either copy it or write it down, or makes errors when they do so.

VPF-MI.3: The voter is instructed to keep a record of the smart ballot tracker with the option to note it down, print it or email it (Figure 12). However, they may not understand the significance of this action and may ignore information important for verifiability.

VPF-MI.4: The action of the voter auditing their ballot (Figure 12) and the importance of this step is not emphasized. There is a high likelihood that the voter may ignore it and complete the voting process without verifying their vote.

VPF-MI.5: The voter is informed that their audited ballot will not be tallied (Figure 2). No explanation is given why this is the case. The voter may be deterred from continuing with the process.

VPF-MI.6: The voter is instructed to post the audited ballot to the tracking center (Figure 2), yet neither is information given why they should do this, nor the purpose and location of the tracking center indicated. The voter may ignore this step although it is important for verifiability.

VPF-MI.7: The voter is not directed to compare the election fingerprint (Figure 3) and the fingerprint received in the invitation-to-vote email (Figure 7), with the result that they might not know what to do and therefore ignore this information.

VPF-MI.8: Without instructions on the purpose and function of the smart ballot tracker displayed in the verifiability results (Figure 3), the voter may also ignore it.

VPF-MI.9: If the voter is dissatisfied with the results of the verifiability procedure (Figure 3), there is no information provided on how they can proceed and there is no possibility to contact the election authorities to file a complaint. This lack of support may disillusion voters, who may decide not to continue casting their ballot.

VPF-MI.10: After casting their ballot, the voter is informed that they have done so successfully (Figure 15). This however may cause the voter to trust the results prematurely and not verify the results.

VPF-UIE.1: It is unclear what the link 'Select your ballot audit info' (Figure 2) does. The voter may miss some steps or assume that the information is copied into the verifier automatically, which is not the case.

VPF-UIE.2: On clicking the link 'ballot verifier' (Figure 2), a new window pops up displaying an empty box with the heading 'Your Ballot'. However there is no verified ballot information displayed and the voter can make an error here in the absence of the required guidelines.

VPF-UIE.3: After verification the voter may not know how to continue with the ballot casting process, and how to move between the two pages (Figure 3 and Figure 11). This is because there is no link, button or text explaining how to either continue or terminate the process. The voter would therefore back out of the voting process.

VPF-UIE.4: The verification results webpage (Figure 3) contains an election URL which the voter can edit, but whose purpose is not stated. The voter may as a result be uncertain how to proceed.

VPF-UIE.5: The fact that the verifier (Figure 3) has the same interface design as the voting system is likely to evoke doubts in the voter as the ballot will be verified by the same system used to cast it.

5.3 Usable Security Findings

We identified two findings in the context of misleading information (MLI) and one finding regarding missing information (MI).

USF-MLI.1: After successfully encrypting their ballot, the voter is given information regarding protecting their privacy (Figure 12), that is, they will log in only once the ballot is encrypted. The voter may not understand the difference and expect to be authenticated prior to interacting with the system. Second, the voter is informed that the system will only remember the encrypted vote, but if the JavaScript is manipulated it can display this message and yet go on to record the voter's identity.

USF-MLI.2: No means of authenticating the identity of the election server is provided, for example, a fingerprint of the server certificate. The consequence is that these issues may cause distrust since they are not addressed, and the voter may not go ahead to cast their ballot. As an example: phishing emails could be distributed with faked links to manipulated web pages.

USF-MI.1: In the invitation-to-vote email (Figure 7) the voter receives their password in cleartext with no encryption applied. This is a security threat that can be exploited, thereby causing the voter to question the security of the voting system.

Note: some of the findings presented here were noted as challenges in the UCL elections [4] and the IACR demo election [13], for example, *GUF-W.1*, *GUF-W.2*, *GUF-MLI.1*, *GUF-MI.2* and *VPF-W.1*. A few were addressed for example: in the UCL election, a service desk number was availed at the bottom of the voting interface.

6 Recommendations For An Improved Interface

In total we identified ten general usability findings, 23 verifiability procedure findings and three usable security findings. In this section we present our proposed improvements to Helios in order to enhance its comprehensibility and usability, particularly the usability of verifiability. We refer to the findings made in section 5.

Our improvements are categorized as follows:

- the different setting we use, i.e., a fictitious university president election run at our university on our own servers, (under subsection 6.1)
- main improvements to the interface and ballot casting and verifiability process, (under subsection 6.2)
- minor changes, such as wording, (under subsection 6.3).

We have developed a prototype to demonstrate the proposed improvements and plan to run a user study in the near future. The improvements were made to meet the comprehensibility, usability and verifiability challenges in Helios. Due to these changes, a new voting process is necessary and can be seen in Figure 4. Screenshots of the new interface are available in Appendix A.

6.1 Different Setting

In the proposed process we use postal mail to send letters containing login credentials to individual voters (Figure 18). The electorate in this environment are familiar with postal voting and may be comfortable with this. In preparing the password, we propose that it be encoded in a font that clearly distinguishes the different characters. An alternative is to leave out the problem characters to avoid confusing the voters (GUF-MLI.1).

If the voter is unable to successfully complete the verifiability process, rather than writing an email, they can immediately contact the election authorities (Figure 5) using a hotline with 24-hour service. This telephone number is displayed on the web page (GUF-UIE.2; VPF-MI.9). Implementation of this 24-hour hotline can integrate both human and machine support to improve feasibility.

We propose that the verifiability process is handled by independent parties in order to enhance the trust a voter has in the results of the verifiability as seen in Figure 5. The voter is free to select which party they want to verify their vote. They no longer have to copy and paste the ballot information, thereby reducing the chance of errors. Once the ballot is verified, the voter is only given relevant and simple information as feedback. Additionally, we provide the option for a voter to manually verify their ballot is correctly encrypted (VPF-W.2; VPF-MLI.3; VPF-MLI.5; VPF-MLI.7; VPF-UIE.2; USF-UIE.1; VPF-UIE.5). Having several independent institutions to verify the voter's ballot, as well as allowing the voter to select any of the institutions provided, adds greater credibility to the verified results. Use of multiple institutions will minimize the possibility of collusion between a corrupt auditing institution and a manipulated JavaScript. Note: The criteria used to select these institutions depend on the election in question.

Since in our setup only one election is run at a given time and on only a single server, the election fingerprint is not necessary in identifying the election (Figure 19) and is therefore removed (VPF-UIE.4). As such, the voter no longer has to compare the fingerprint information (VPF-MI.7).

We propose use of three distinct websites, namely www.election.university.org (Figure 19) with information on the election, www.votecasting.university.org (Figure 20) for the actual ballot casting and www.electionresults.university.org (Figure 27), where election results and information for verifiability will be posted. This will enhance usability for the voter as they have one web page per action; one for general information, one to cast their ballot and one to verify if their vote appears on the bulletin board.

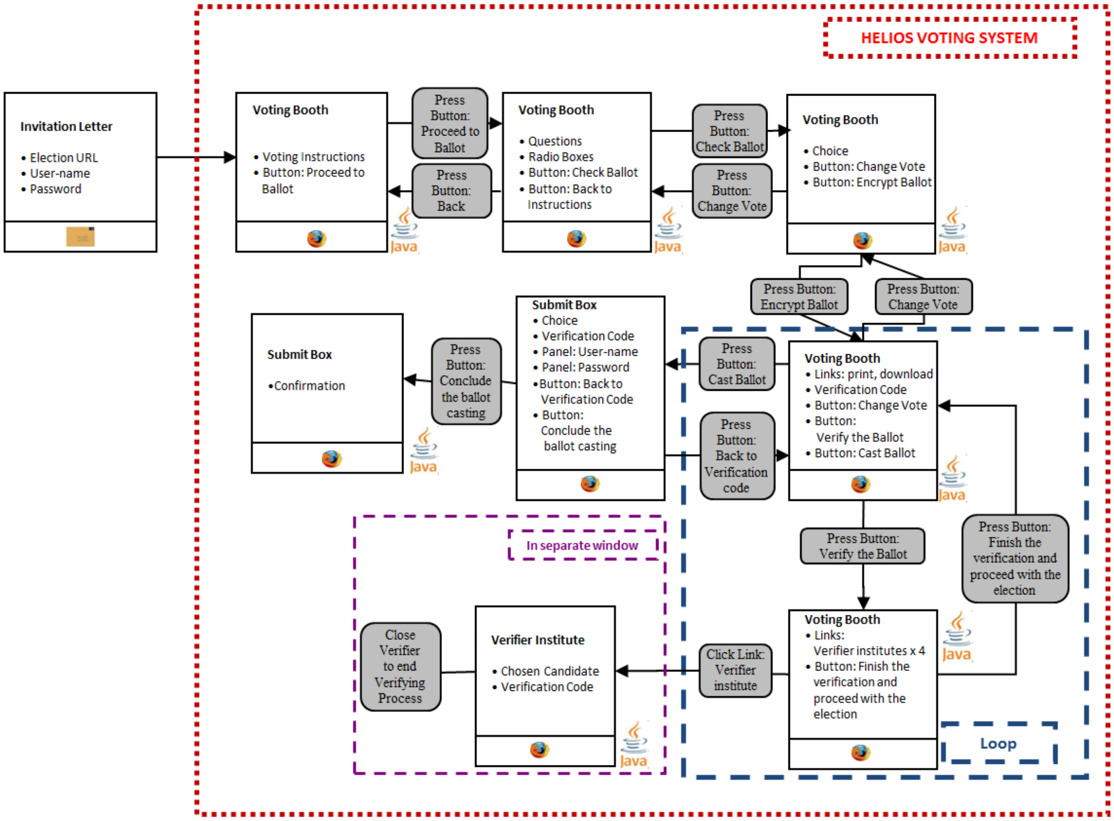

HELIOS VOTING SYSTEM
Invitation Letter
• Election URL
• User-name
• Password
Voting Booth
• Voting Instructions
• Button: Proceed to Ballot
Press Button: Proceed to Ballot
Press Button: Back
Voting Booth
• Questions
• Radio Boxes
• Button: Check Ballot
• Button: Back to Instructions
Press Button: Check Ballot
Press Button: Change Vote
Voting Booth
• Choice
• Button: Change Vote
• Button: Encrypt Ballot
Java
Press Button: Encrypt Ballot
Press Button: Change Vote
Submit Box
• Confirmation
Press Button: Conclude the ballot casting
Submit Box
• Choice
• Verification Code
• Panel: User-name
• Panel: Password
• Button: Back to Verification Code
• Button: Conclude the ballot casting
Press Button: Cast Ballot
Press Button: Back to Verification code
Voting Booth
• Links: print, download
• Verification Code
• Button: Change Vote
• Button: Verify the Ballot
• Button: Cast Ballot
Press Button: Verify the Ballot
Press Button: Finish the verification and proceed with the election
In separate window
Close Verifier to end Verifying Process
Verifier Institute
• Chosen Candidate
• Verification Code
Click Link: Verifier institute
Voting Booth
• Links: Verifier institutes x 4
• Button: Finish the verification and proceed with the election
Loop

Figure 4: Improved Ballot Casting and Verification Procedure

6.2 Main Improvements

Since the verification-code used in the original version is too long, we have shortened it, making it less intimidating and more usable for the voter (Figure 23). However, identifying an appropriate length remains work in progress. The verification-code can be shortened as a collision is only relevant for four days (duration of the election). Finding a collision after this is not useful. We have also avoided using characters that are likely to be difficult to distinguish (VPF-MLI.1; VPF-MI.3).

In Figure 18 we provide a fingerprint of the SSL certificate of the election server at the back of the invitation-to-vote letter (USF-MI.1).

The verification process has been made more user-friendly. The voter no longer has to select and copy the ballot audit information as can be seen in Figure 6 (VPF-UIE.1). The option for self-verification of ballot information is retained.

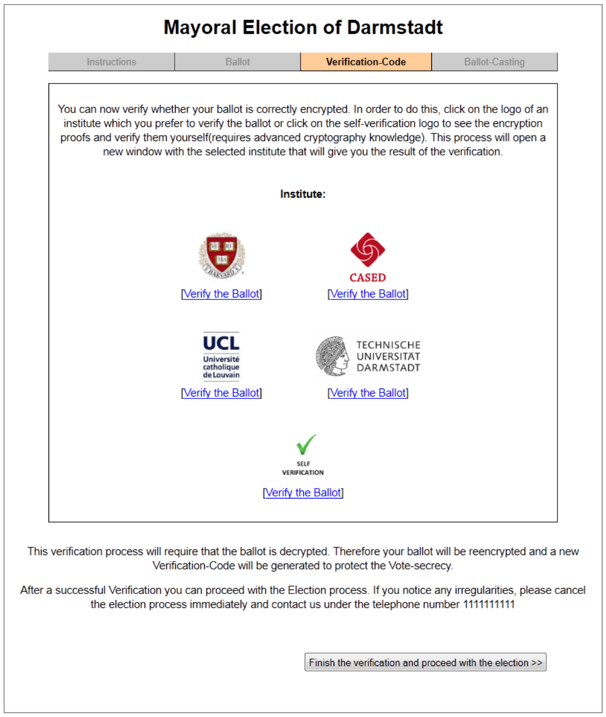
We integrate an additional step ‘verification-code’ in the menu bar (Figure 19) to adequately inform the voter of all the steps that will be carried out, and also to draw attention to the verifiability process (GUF-UIE.1).

The interface has been redesigned to be consistent, with each webpage providing forward and back buttons (Figure 20) enabling the voter to easily proceed with the ballot casting process. There is also consistency with regards to the terminology used (GUF-UIE.3).

The voter’s ballot is immediately sent to and stored at the server upon successful authentication (Figure 26) and a simple confirmation message is displayed (GUF-UIE.5; VPF-MI.10).

We aim to maintain consistency such that one word has the same meaning throughout the whole process. In addition we have used terms that are familiar to voters (Figure 19), e.g. we have renamed the ‘smart ballot tracker’ to ‘verification-code’ (Figure 23), a term more understandable to the voters (GUF-W.1; GUF-W.2). The aspect of wording familiar to voters will later be analysed in a user study.

The improved interface gives adequate explanation in areas where voters might be confused, for example what they should do with the verification-code after successful ballot encryption in Figure 23 (VPF-W.1; VPF-MLI.2; VPF-MLI.3) The explanations hold in particular for the verification steps: In Figure 6 the voter is given clear instructions how to proceed with ballot-casting after verification (VPF-MLI.6; VPF-UIE.3). The instructions to



Mayoral Election of Darmstadt

You can now verify whether your ballot is correctly encrypted. In order to do this, click on the logo of an institute which you prefer to verify the ballot or click on the self-verification logo to see the encryption proofs and verify them yourself(requires advanced cryptography knowledge). This process will open a new window with the selected institute that will give you the result of the verification.

Institute:

This verification process will require that the ballot is decrypted. Therefore your ballot will be reencrypted and a new Verification-Code will be generated to protect the Vote-secrecy.

After a successful Verification you can proceed with the Election process. If you notice any irregularities, please cancel the election process immediately and contact us under the telephone number 1111111111

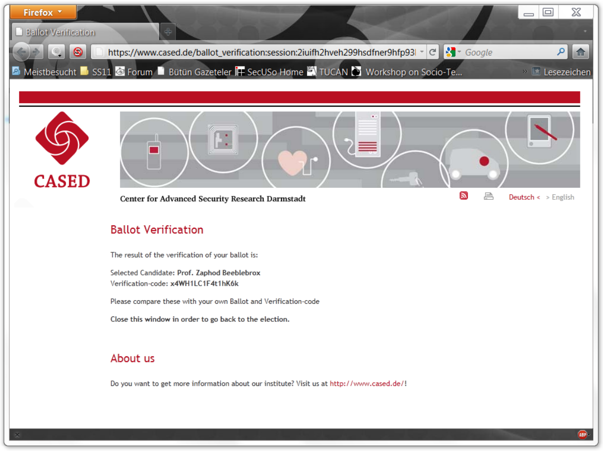
Figure 5: Improved System - Verification Institutes

the voter are deliberately simple and clear in order to give them adequate support to complete the voting process (VPF-MI.1). The need for verification (Figure 23) is explained to the voter using terms that they can understand (VPF-MI.2; VPF-MI.4; VPF-MI.5).

The ballot tracking center is no longer incorporated in the new interface. The voter can check for their verification-code by accessing the bulletin board (Figure 27) after the end of the election (VPF-MI.4; VPF-MI.6). This verification at the end of the election is necessary as the voter can detect if their ballot has been modified. They would do this by checking that the verification-code posted on the bulletin board matches what they received during vote casting. The information on this website would be availed in the invitation-to-vote email.

We inform the voter that they will only be authenticated at the end after selecting their candidate, and in the instructions to voters (Figure 19) we give explanations for this difference from what the voter may be familiar with (GUF-MI.1). In addition, before the voter logs in to cast a ballot we inform them (Figure 25) where to obtain their username and password voting credentials, which were sent earlier in the invitation-to-vote letter (USF-MLI.2).

In the improved screenshots, the option to cast a blank ballot is made clearly visible to the voter. If they make this selection, a message is displayed to inform them that



Ballot Verification

The result of the verification of your ballot is:

Selected Candidate: **Prof. Zaphod Beeblebrox**
Verification-code: **x4WH1LC1F4t1hK6k**

Please compare these with your own Ballot and Verification-code

Close this window in order to go back to the election.

About us

Do you want to get more information about our institute? Visit us at http://www.cased.de/!

Figure 6: Improved System - Verification Result

they will cast an invalid vote (GUF-MI.2). For an example, see Figure 20.

In order to ensure that the voter knows what to do with the verification-code, we remind them to compare the value given at one stage to that obtained in an earlier one (VPF-MLI.1; VPF-MI.8). See for example Figure 25.

6.3 Minor Changes

While making a candidate selection (Figure 20), the voter is informed that they can only select one candidate. The word 'one' is highlighted in order to catch the voter's attention. Once the voter has made their selection, they are instructed to proceed to encrypt the vote (GUF-UIE.4).

In Figure 19 we do not mention the aspect of protecting the voter's privacy in the instructions to the voters since it is likely to be confusing (USF-MLI.1).

All the findings indicated earlier in section 5 have been discussed in this section and matched with proposed improvements. Those not dealt with require further research and are considered future work.

7 Security analysis of the Proposed Modifications

We have proposed and implemented in a prototype several improvements to the Helios interface and ballot casting and verifiability process. Here we briefly review the security of Helios given the proposed modifications.

- We have proposed use of a shortened verification-code in order to improve the usability. The chances of an attacker finding a collusion are higher, however they would have a limited amount of time (from the start of the election when public key parameters are released, to the close of the election period) in which to do this. The exact amount of time depends on the election and the time available for vote casting and correspondingly the size of the verification-code depends on this. The exact length that provides an acceptable amount of security and usability requires further research.

- When the voter is ready to verify their ballot, the ballot information is forwarded directly to the institution selected. This will require Internet connectivity earlier in the ballot casting process. Besides this, one might argue that in our proposed version, a manipulated JavaScript would send modified verifiability information to the institutions. This is true, however, in the current Helios version a manipulated JavaScript would behave in a similiar way: it simply displays modified verification information to the voter (to copy and paste this to the external verification service). In both versions, this can only be detected if a voter stores and compares their verification-code.
- After the voter logs in to cast their ballot, the encrypted ballot is immediately captured and the voter is given feedback to this effect. This is done to reduce the number of single steps the voter has to carry out to complete a task. If any modifications are made to the ballot at this stage, the voter will detect this when they access the bulletin board at the end of the election.

No major modifications have been made related to security other than shortening the fingerprint, which has been done to improve the usability of the verification-code. Thus the security is not reduced either in practice or in theory by the shortened fingerprint.

8 Conclusions and Future Work

E2E verifiable electronic voting systems are the most promising approaches to enable electronic voting and in particular remote electronic voting. While a large number of cryptographic protocols have been proposed in conferences, only few protocols for E2E verifiable remote electronic voting have been implemented and used for legally binding elections and only in academic contexts. The Helios voting system is one of the exceptions. Its implementation is a first step in the right direction from theoretical approaches to practical solutions. So far the focus has mainly been on security issues of Helios and less on the usability shortcomings of Helios and verifiable electronic voting schemes in general.

In this paper we used the cognitive walkthrough approach to analyze the usability of the Helios voting system, specifically the usability of the verifiability process. We have proposed an improved interface corresponding to the findings made. There still exist, in the latest version of Helios, a number of usability flaws in the general design of the voter interface (like a mix of links and buttons, inconsistent wording, no back buttons provided e.t.c). Also, the functionality of individual verifiability and the re-encryption/modified commitments after the verifiability has been carried out are too complicated, time intensive and error-prone. The consequence of these flaws and complexity is that very few voters can make use of the verifiability. Even with the note that not all voters have to verify, a requirement of the equal election principle is that all voters have the same opportunities and can all verify.

More research is needed to find out how many people verify their ballot in elections run with the Helios voting system. The prototype developed will be used to run several user tests to improve it further. Furthermore, we plan to carry out a user study using the improved interfaces to determine what terms voters are more familiar or comfortable with, for instance between ‘audit’ and ‘verify’ and the effect of renaming ‘smart ballot tracker’ to ‘verification-code’. This user study will also test if in general average voters opt to verify their ballot using the new interfaces and how to get them to use it. In addition, the user study will help determine which institutions are known and trusted by the voter to carry out the verifiability process. Cultural issues and background will be taken into account in selecting participants for the user study. This is because these are likely to be factors affecting the participants’ perception of the voting interfaces. Finally, the effect of shortening the verification-code on the security of Helios requires further investigation regarding which length is appropriate for which election and vote casting timeframe as well as the possibility of using QR (quick response) codes.

We plan to cooperate with the developers of Helios to have these improvements integrated in the official version. As we concentrated in this paper only on the voter’s side and only on the first part of the individual verifiability, the next steps will consist of a usability analysis of the remaining voter interfaces and the interfaces for the election commission.

Acknowledgments

This work was supported by CASED (www.cased.de) and Micromata (www.micromata.de).

References

[1] A Study of Vote Verification Technology Conducted for the Maryland State Board of Elections Part II: Usability Study. Tech. rep., CAPC Report on Vote Verification Systems, 2006.

[2] Guide to helios (user guide). http://usg.princeton.edu/usg-senate/elections-center/guide-to-helios.html, 2010.

[3] ADIDA, B. Helios: Web-based Open-Audit Voting. In *In Proceedings of the 17th Symposium on Security* (Berkeley, CA, USA, 2008), Usenix Association, pp. 335 – 348.

[4] ADIDA, B., PEREIRA, O., MARNEFFE, O. D., AND JACQUES QUISQUATER, J. Electing a University President using Open-Audit Voting: Analysis of real-world use of Helios. In *Electronic Voting Technology/Workshop on Trustworthy Elections (EVT/WOTE* (2009).

[5] BEDERSON, B. B., LEE, B., SHERMAN, R. M., HERRNSON, P. S., AND NIEMI, R. G. Electronic Voting System Usability Issues. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2003), ACM, pp. 145–152.

[6] BENALOH, J. Simple verifiable elections. In *EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (Berkeley, CA, USA, 2006), USENIX Association.

[7] BUECHLER, J., EARNET, T., AND SMITH, B. Voting System Usability: Optical Scan, Zoomable, Punchscan. *UMBC CMSC 691/491V* (2007).

[8] BYRNE, M. D., GREENE, K. K., AND EVERETT, S. P. Usability of Voting Systems: Baseline Data for Paper, Punch Cards, and Lever Machines. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), ACM, pp. 171–180.

[9] CHAUM, D., CARBACK, R., CLARK, J., ESSEX, A., POPOVENIUC, S., RIVEST, R. L., RYAN, P. Y. A., SHEN, E., SHERMAN, A. T., AND VORA, P. L. Corrections to scantegrity II: End-to-end Verifiability by Voters of Optical Scan Elections Through Confirmation Codes. *IEEE Transactions on Information Forensics and Security 5*, 1 (2010), 194.

[10] CLARKSON, M. R., CHONG, S., AND MYERS, A. C. Civitas: Toward a Secure Voting System. *Security and Privacy, IEEE Symposium on 0* (2008), 354–368.

[11] EVERETT, S. P. *The Usability of Electronic Voting Machines and How Votes Can Be Changed Without Detection*. PhD thesis, Rice University, 2007.

[12] EVERETT, S. P., BYRNE, M. D., AND GREENE, K. K. Measuring The Usability of Paper Ballots: Efficiency, Effectiveness and Satisfaction. In *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting* (2006), pp. 2547–2551.

[13] HABER, S., BENALOH, J., AND HALEVI, S. The Helios e-Voting Demo for the IACR. http://www.iacr.org/elections/eVoting/heliosDemo.pdf, 2010.

[14] HERRNSON, P. S., BEDERSON, B. B., LEE, B., FRANCIA, P. L., SHERMAN, R. M., CONRAD, F. G., TRAUGOTT, M., AND NIEMI, R. G. Early Appraisals of Electronic Voting. *Soc. Sci. Comput. Rev. 23*, 3 (2005), 274–292.

[15] HERRNSON, P. S., NIEMI, R. G., HANMER, M. J., BEDERSON, B. B., CONRAD, F. G., AND TRAUGOTT, M. The Importance of Usability Testing of Voting Systems. In *EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (Berkeley, CA, USA, 2006), USENIX Association.

[16] INTERNAIONAL ORGANIZATION FOR STANDARDIZATION. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems .

[17] JUELS, A., CATALANO, D., AND JAKOBSSON, M. Coercion-Resistant Electronic Elections. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society - WPES '05* (2005), ACM Press, pp. 61 – 70.

[18] KIMBALL, D. C., AND KROPF, M. Ballot Design and Unrecorded Votes on Paper-based Ballots. Vol. 69 4, Public Opinion Quarterly, 2005.

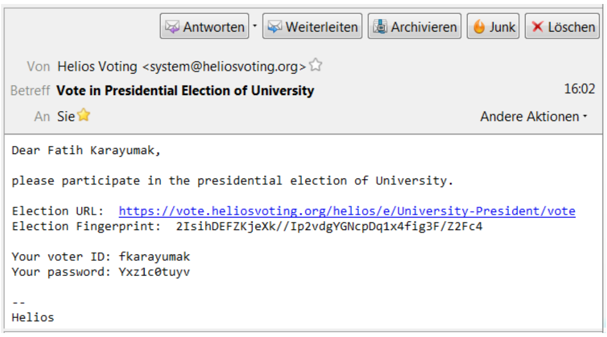
[19] LASKOWSKI, S. J., AND REDISH, J. Making Ballot Language Understandable To Voters. In *EVT'06: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (Berkeley, CA, USA, 2006), USENIX Association.

[20] RYAN, P. Y., BISMARK, D., HEATHER, J., SCHNEIDER, S., AND XIA, Z. The Prêt à Voter Verifiable Election System. In *IEEE Transactions in Information Security and Forensics* (2009).

[21] STONE, D., JARRETT, C., WOODROFFE, M., AND MINOCHA, S. *User Interface Design and Evaluation*. Elsevier, San Francisco, 2005.

[22] WEBER, J., AND HENGARTNER, U. Usability Study of the Open Audit Voting System Helios. http://www.jannaweber.com/wp-content/uploads/2009/09/858Helios.pdf, 2009.

A Screenshots



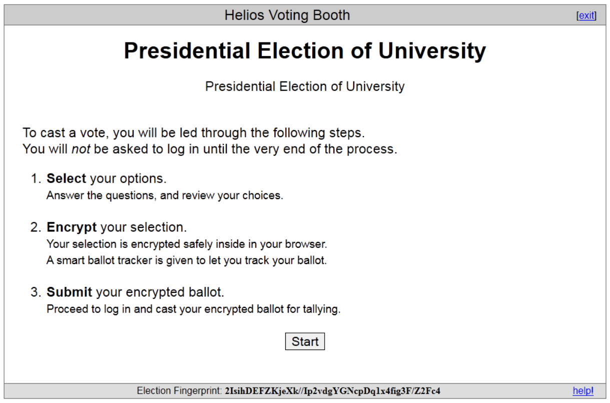
Von Helios Voting <system@heliosvoting.org>

Betreff **Vote in Presidential Election of University** 16:02

An Sie

```
Dear Fatih Karayumak,

please participate in the presidential election of University.

Election URL:  https://vote.heliosvoting.org/helios/e/University-President/vote
Election Fingerprint:  2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4

Your voter ID: fkarayumak
Your password: Yxz1c0tuyv

--
Helios
```

Figure 7: Invitation-to-Vote Email



Presidential Election of University

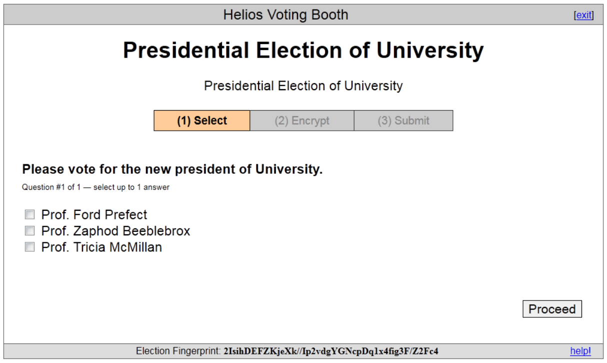
Presidential Election of University

To cast a vote, you will be led through the following steps.
You will *not* be asked to log in until the very end of the process.

1. **Select** your options.
   Answer the questions, and review your choices.
2. **Encrypt** your selection.
   Your selection is encrypted safely inside in your browser.
   A smart ballot tracker is given to let you track your ballot.
3. **Submit** your encrypted ballot.
   Proceed to log in and cast your encrypted ballot for tallying.

Election Fingerprint: **2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4**

Figure 8: Voting Booth - Information Page



Presidential Election of University

Presidential Election of University

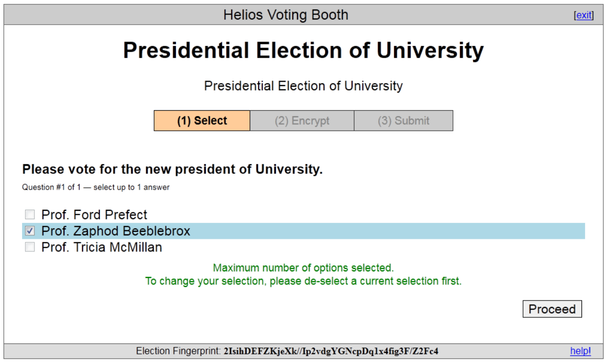
Please vote for the new president of University.

Question #1 of 1 — select up to 1 answer

☐ Prof. Ford Prefect
☐ Prof. Zaphod Beeblebrox
☐ Prof. Tricia McMillan

Election Fingerprint: **2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4**

Figure 9: Voting Booth - Empty Ballot



Presidential Election of University

Presidential Election of University

Please vote for the new president of University.

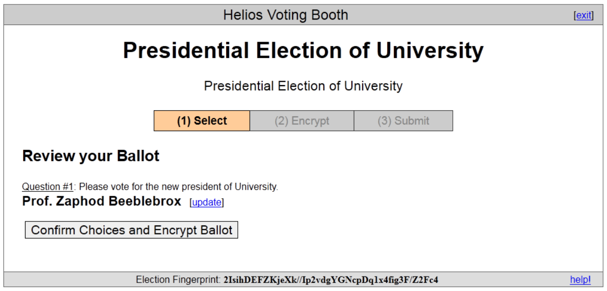
Question #1 of 1 — select up to 1 answer

☐ Prof. Ford Prefect
☑ Prof. Zaphod Beeblebrox
☐ Prof. Tricia McMillan

Maximum number of options selected.
To change your selection, please de-select a current selection first.

Election Fingerprint: **2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4**

Figure 10: Voting Booth - Ballot with a Selection



Presidential Election of University

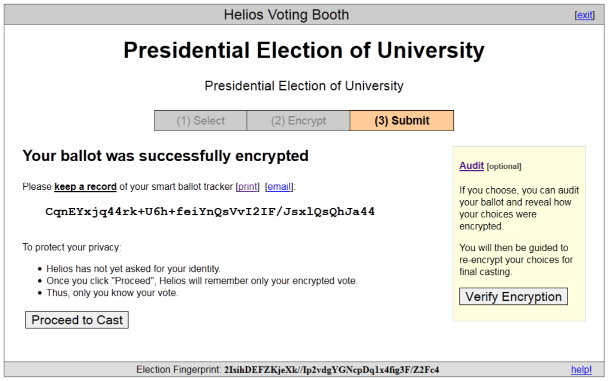
Presidential Election of University

Review your Ballot

Question #1: Please vote for the new president of University.
Prof. Zaphod Beeblebrox [update]

Election Fingerprint: **2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4**

Figure 11: Voting Booth - Review the Ballot



Presidential Election of University

Presidential Election of University

Your ballot was successfully encrypted

Please **keep a record** of your smart ballot tracker [print] [email]:

```
CqnEYxjq44rk+U6h+feiYnQsVvI2IF/JsxlQsQhJa44
```

To protect your privacy:

- Helios has not yet asked for your identity.
- Once you click "Proceed", Helios will remember only your encrypted vote.
- Thus, only you know your vote.

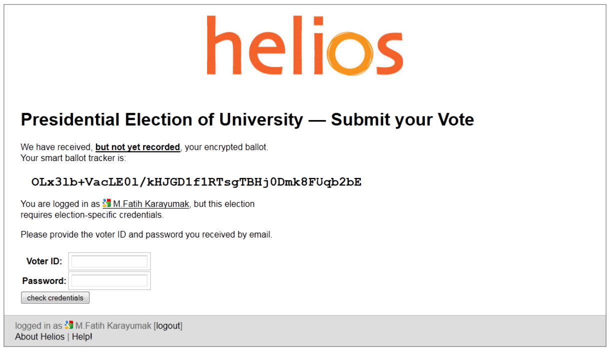
Audit [optional]

If you choose, you can audit your ballot and reveal how your choices were encrypted.

You will then be guided to re-encrypt your choices for final casting.

Election Fingerprint: **2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4**

Figure 12: Voting Booth - Smart Ballot Tracker and Audit Option



Presidential Election of University — Submit your Vote

We have received, **but not yet recorded**, your encrypted ballot.
Your smart ballot tracker is:

```
OLx3lb+VacLE01/kHJGD1f1RTsgTBHj0Dmk8FUqb2bE
```

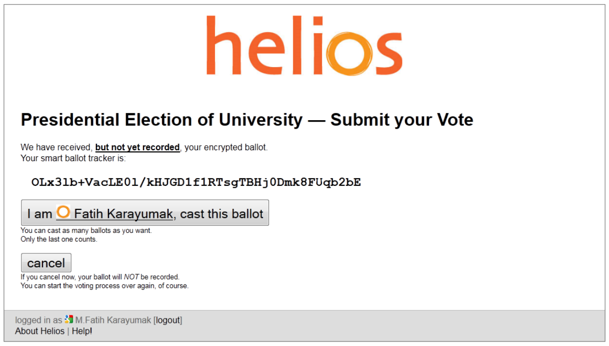
You are logged in as M.Fatih Karayumak, but this election requires election-specific credentials.

Please provide the voter ID and password you received by email.

Voter ID:
Password:

logged in as M.Fatih Karayumak [logout]
About Helios | Help!

Figure 13: Voting Booth - Authentication



Presidential Election of University — Submit your Vote

We have received, **but not yet recorded**, your encrypted ballot.
Your smart ballot tracker is:

`OLx3lb+VacLE01/kHJGD1f1RTsgTBHj0Dmk8FUqb2bE`

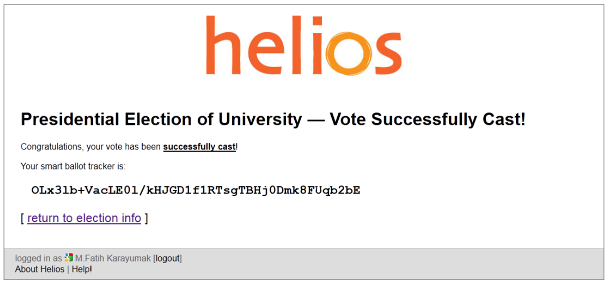
I am Fatih Karayumak, cast this ballot

You can cast as many ballots as you want.
Only the last one counts.

If you cancel now, your ballot will *NOT* be recorded.
You can start the voting process over again, of course.

logged in as M.Fatih Karayumak [logout]
About Helios | Help!

Figure 14: Voting Booth - Last Check before Casting



Presidential Election of University — Vote Successfully Cast!

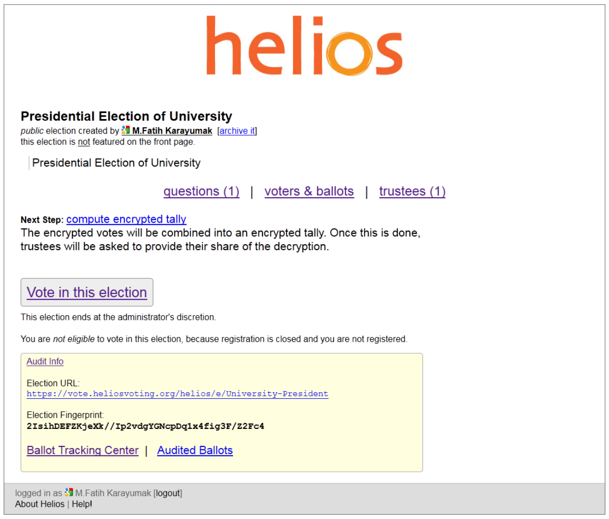
Congratulations, your vote has been **successfully cast**!

Your smart ballot tracker is:

`OLx3lb+VacLE01/kHJGD1f1RTsgTBHj0Dmk8FUqb2bE`

[return to election info]
logged in as M.Fatih Karayumak [logout]
About Helios | Help!

Figure 15: Voting Booth - Confirmation Message



Presidential Election of University
public election created by **M.Fatih Karayumak** [archive it]
this election is not featured on the front page.

Presidential Election of University

Next Step: compute encrypted tally
The encrypted votes will be combined into an encrypted tally. Once this is done, trustees will be asked to provide their share of the decryption.

This election ends at the administrator's discretion.

You are *not eligible* to vote in this election, because registration is closed and you are not registered.

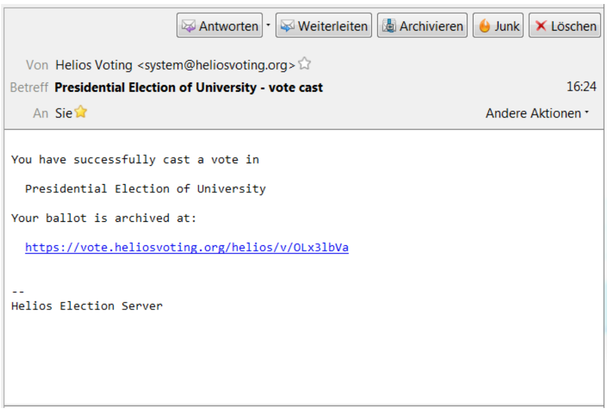
Audit Info

Election URL:
https://vote.heliosvoting.org/helios/e/University-President

Election Fingerprint:
2IsihDEFZKjeXk//Ip2vdgYGNcpDq1x4fig3F/Z2Fc4

Ballot Tracking Center | Audited Ballots
logged in as M.Fatih Karayumak [logout]
About Helios | Help!

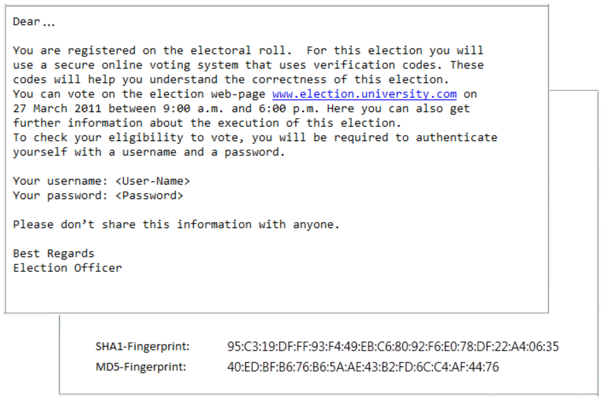
Figure 16: Election Information Page



Von Helios Voting <system@heliosvoting.org>
Betreff **Presidential Election of University - vote cast** 16:24
An Sie

```
You have successfully cast a vote in

  Presidential Election of University

Your ballot is archived at:

  https://vote.heliosvoting.org/helios/v/OLx3lbVa


--
Helios Election Server
```

Figure 17: Confirmation Email



Dear ...

You are registered on the electoral roll. For this election you will use a secure online voting system that uses verification codes. These codes will help you understand the correctness of this election.
You can vote on the election web-page www.election.university.com on 27 March 2011 between 9:00 a.m. and 6:00 p.m. Here you can also get further information about the execution of this election.
To check your eligibility to vote, you will be required to authenticate yourself with a username and a password.

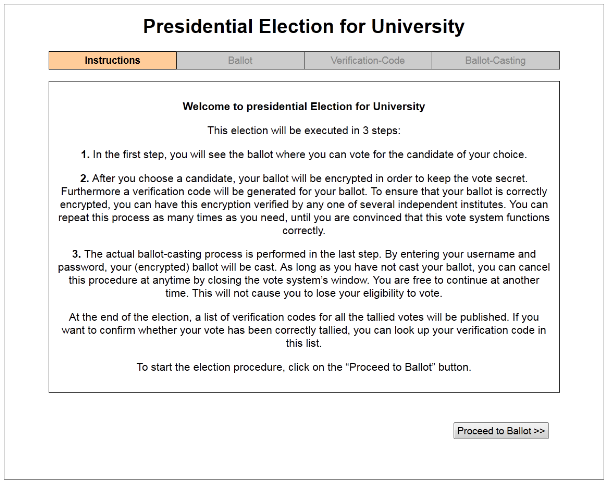
Your username: <User-Name>
Your password: <Password>

Please don't share this information with anyone.

Best Regards
Election Officer

SHA1-Fingerprint:	95:C3:19:DF:FF:93:F4:49:EB:C6:80:92:F6:E0:78:DF:22:A4:06:35
MD5-Fingerprint:	40:ED:BF:B6:76:B6:5A:AE:43:B2:FD:6C:C4:AF:44:76

Figure 18: Improved System: Invitation-to-Vote Letter



Presidential Election for University

Welcome to presidential Election for University

This election will be executed in 3 steps:

1. In the first step, you will see the ballot where you can vote for the candidate of your choice.

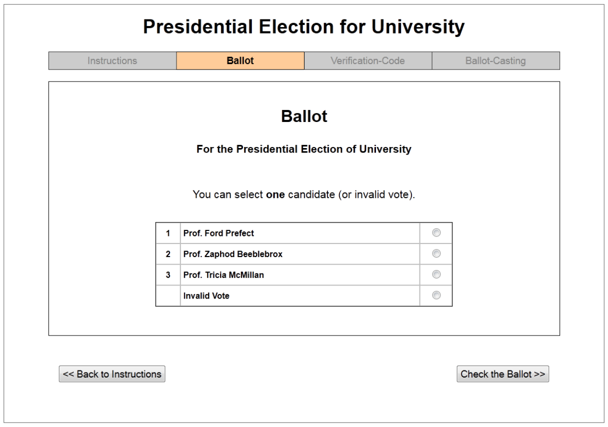
2. After you choose a candidate, your ballot will be encrypted in order to keep the vote secret. Furthermore a verification code will be generated for your ballot. To ensure that your ballot is correctly encrypted, you can have this encryption verified by any one of several independent institutes. You can repeat this process as many times as you need, until you are convinced that this vote system functions correctly.

3. The actual ballot-casting process is performed in the last step. By entering your username and password, your (encrypted) ballot will be cast. As long as you have not cast your ballot, you can cancel this procedure at anytime by closing the vote system's window. You are free to continue at another time. This will not cause you to lose your eligibility to vote.

At the end of the election, a list of verification codes for all the tallied votes will be published. If you want to confirm whether your vote has been correctly tallied, you can look up your verification code in this list.

To start the election procedure, click on the "Proceed to Ballot" button.

Figure 19: Improved System: Election Information Page



Presidential Election for University

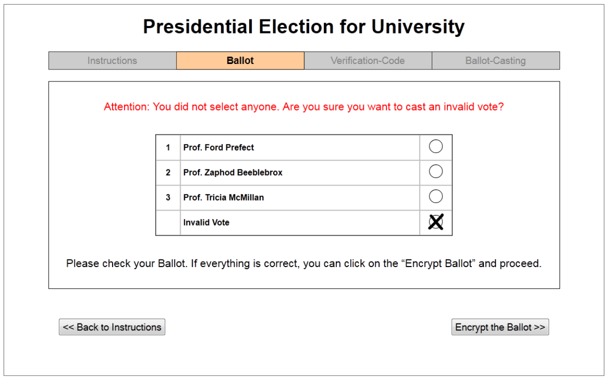
Ballot

For the Presidential Election of University

You can select **one** candidate (or invalid vote).

1	Prof. Ford Prefect	○
2	Prof. Zaphod Beeblebrox	○
3	Prof. Tricia McMillan	○
	Invalid Vote	○

Figure 20: Improved System: Empty Ballot



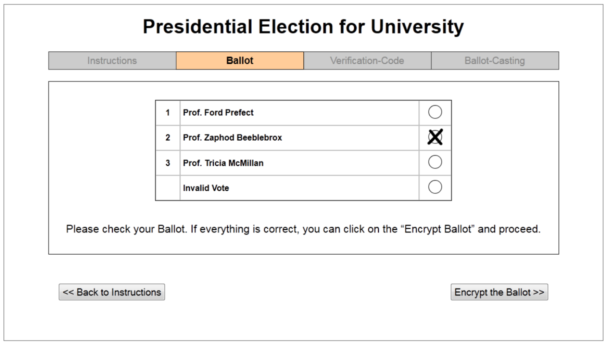
Presidential Election for University

Attention: You did not select anyone. Are you sure you want to cast an invalid vote?

1	Prof. Ford Prefect	○
2	Prof. Zaphod Beeblebrox	○
3	Prof. Tricia McMillan	○
	Invalid Vote	☒

Please check your Ballot. If everything is correct, you can click on the "Encrypt Ballot" and proceed.

Figure 21: Improved System: Invalid Vote

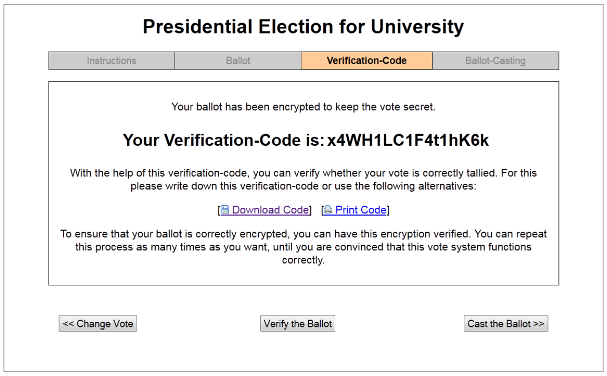


Presidential Election for University

1	Prof. Ford Prefect	○
2	Prof. Zaphod Beeblebrox	☒
3	Prof. Tricia McMillan	○
	Invalid Vote	○

Please check your Ballot. If everything is correct, you can click on the "Encrypt Ballot" and proceed.

Figure 22: Improved System: Ballot with a Selection



Presidential Election for University

Your ballot has been encrypted to keep the vote secret.

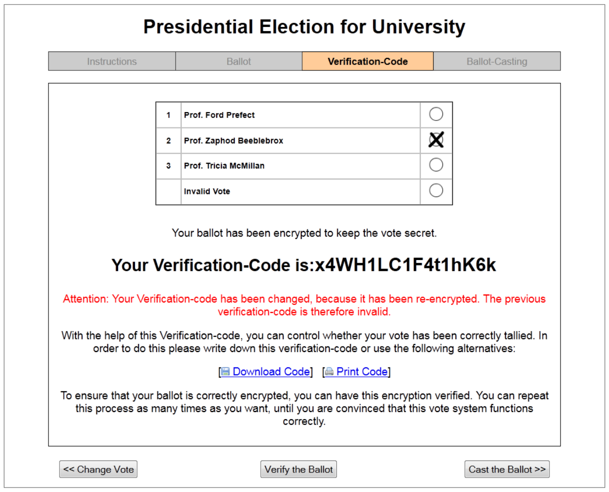
Your Verification-Code is: x4WH1LC1F4t1hK6k

With the help of this verification-code, you can verify whether your vote is correctly tallied. For this please write down this verification-code or use the following alternatives:

[Download Code] [Print Code]

To ensure that your ballot is correctly encrypted, you can have this encryption verified. You can repeat this process as many times as you want, until you are convinced that this vote system functions correctly.

Figure 23: Improved System: Verification-Code and Verify Option



Presidential Election for University

1	Prof. Ford Prefect	○
2	Prof. Zaphod Beeblebrox	☒
3	Prof. Tricia McMillan	○
	Invalid Vote	○

Your ballot has been encrypted to keep the vote secret.

Your Verification-Code is:x4WH1LC1F4t1hK6k

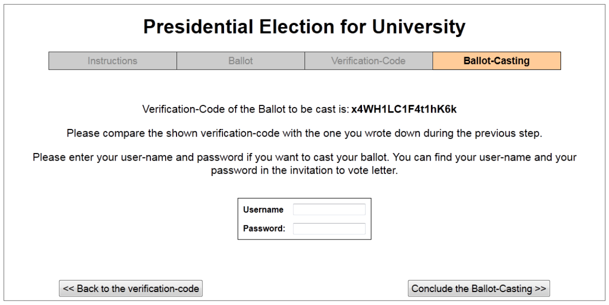
Attention: Your Verification-code has been changed, because it has been re-encrypted. The previous verification-code is therefore invalid.

With the help of this Verification-code, you can control whether your vote has been correctly tallied. In order to do this please write down this verification-code or use the following alternatives:

[Download Code] [Print Code]

To ensure that your ballot is correctly encrypted, you can have this encryption verified. You can repeat this process as many times as you want, until you are convinced that this vote system functions correctly.

Figure 24: Improved System: New Verification-Code


Instructions
Ballot
Verification-Code
Ballot-Casting
<< Back to the verification-code
Conclude the Ballot-Casting >>

Presidential Election for University

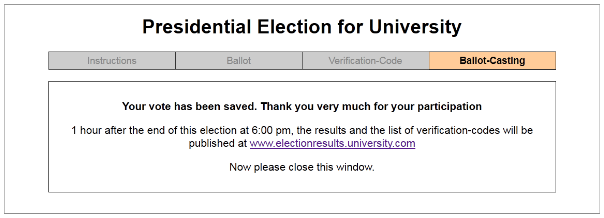
Verification-Code of the Ballot to be cast is: **x4WH1LC1F4t1hK6k**

Please compare the shown verification-code with the one you wrote down during the previous step.

Please enter your user-name and password if you want to cast your ballot. You can find your user-name and your password in the invitation to vote letter.

Username:
Password:

Figure 25: Improved System: Authentication



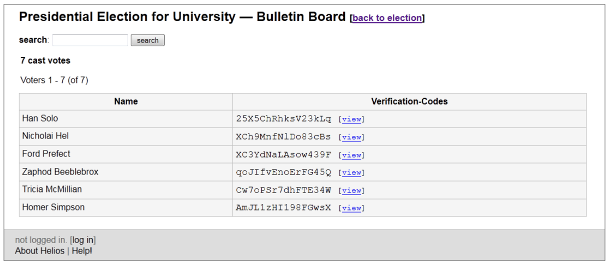
Presidential Election for University

Your vote has been saved. Thank you very much for your participation

1 hour after the end of this election at 6:00 pm, the results and the list of verification-codes will be published at www.electionresults.university.com

Now please close this window.

Figure 26: Improved System: Confirmation Message



Presidential Election for University — Bulletin Board [back to election]

search:

7 cast votes

Voters 1 - 7 (of 7)

Name	Verification-Codes
Han Solo	25X5ChRhksV23kLq [view]
Nicholai Hel	XCh9MnfNlDo83cBs [view]
Ford Prefect	XC3YdNaLAsow439F [view]
Zaphod Beeblebrox	qoJIfvEnoErFG45Q [view]
Tricia McMillian	Cw7oPSr7dhFTE34W [view]
Homer Simpson	AmJL1zHI198FGwsX [view]

Figure 27: Improved System: Bulletin Board