Analysis and Implementation of a Suitable E-Voting Solution for Universidade de Lisboa

Eduardo Alexandre Silva da Costa
Instituto Superior Técnico, Universidade de Lisboa

***Abstract*—This thesis consists of the analysis and implementation of a suitable e-voting system for Universidade de Lisboa. It is required that this system satisfies most of the voting system properties such as integrity, privacy and verifiability while also providing a simple interface to allow inexperienced users with no technical knowledge to use it correctly and confidently. It is also acceptable that the system does not guarantee non-coercibility since most of the elections that are expected to be performed in it have a low-coercive risk. This implementation is based on a version of Helios: a voting system that allows any willing observer to audit the entire process of an election and has already been used several times in real-world elections. The thesis then details modifications that were made in Helios in order to comply with the requirements specified by the Universidade de Lisboa. After adapting and customizing the system, a field test with real voters was orchestrated together with a form in order to evaluate and detect problems and adversities related to security, authentication, usability and accessibility that may have gone unnoticed. As such, further modifications are performed in the system to mitigate any detected problem and every unresolved issue is properly documented. This project has already been used in a real-world election which details can be found in this thesis. Finally, the conclusion and the current system limitations are presented, as well as future work to further improve the system.**

***Index Terms*—E-Voting; Elections; Helios; Cryptosystems**

1. Introduction

Elections have a great impact on developing strong democracies, giving people a say in the way that they want to be governed. To do so, however, it is required that the election system guarantees the correctness of its results while still preserving the voter's privacy. The traditional paper based voting methods have no difficulty in guaranteeing these properties. Privacy is guaranteed because the voter marks his ballot alone in a private voting booth. Correctness is also guaranteed if the election's officials are trustful, since every vote is recorded by them.

Performing an election on an internet based remote voting system is a much greater challenge than traditional paper based systems, since it's more complex to guarantee both the voter's privacy and the correctness of the results.

1.1. Motivation and Objectives

The Universidade de Lisboa's (UL's) rectory was searching for a way to simplify the logistic operation of their periodic elections. The solution agreed was to implement and adapt an already existing e-voting system to solve this problem. Furthermore, the adapted e-voting system should also be capable of allowing any approved member of the UL community to create smaller elections, either for faculties, courses, nucleus or any other group of members.

This thesis focus on finding an already existing e-voting system, implementing and then adapting it, followed by a field test and further improvements of the solution, in order to mitigate any issues detected and to fulfill the UL's community needs.

1.2. Requirements

The solution chosen, besides following the standard e-voting properties stated in Section 2.1.1 must also follow some specific requirements imposed by the UL:

- The final solution must authenticate users via the UL's centralized authentication system and make it possible to create elections without requiring to upload a list of voters. In order to do so, the system must be able to read and save the voter attributes provided by the authentication system.
- The solution may allow hybrid elections, i.e., elections that are both available to vote electronically and in paper, but it must also be able to perform completely electronic and remote elections.
- The chosen solution must have its technical documentation available and legitimate to ease its implementation and also to help identify any possible security issues and limitations.

Although it is not a requirement, it is valued that the solution is open-source, does not require the use of security tokens or code systems and has also been successfully used several times to run real-world elections.

2. E-voting Concepts and Related Work

Some general concepts that are used in the construction of cryptographic voting systems are detailled in this section, followed by the types and a list of researched e-voting systems.

2.1. E-voting Concepts

2.1.1. Voting System Properties. Every voting system is expected to satisfy some specific criteria, such as eligibility, authentication, uniqueness, accuracy, integrity, verifiability, auditability, reliability, secrecy, non-coercibility, flexibility, convenience, certifiability, transparency and cost-effectiveness.

2.1.2. Public Key Cryptosystem. The goal of a public key cryptosystem is to allow anyone to send an encrypted message that can only be decrypted by one entity. For that, it uses asymmetric pairs of keys:

- *Public Key* - This key is known widely and used to encrypt the message;
- *Secret Key* - This key is known only by the entity able to decrypt the ciphered messages.

2.1.3. ElGamal Encryption. The ElGamal cryptosystem [1] is an asymmetric key encryption algorithm that relies on the assumption that discrete logarithm problems with carefully chosen groups have no efficient solution. The exponential ElGamal cryptosystem is a variant in which the difference lies on the message M that is encrypted as g^M.

2.1.4. End-to-End Verifiability. End-to-end (E2E) verifiability aims to allow individual voters to verify the election results without requiring them to trust other entities. It can be divided in two different components:

- Cast as Intended: voters can verify that their selections are correctly recorded;
- Counted as Cast: any member can verify that every recorded vote is correctly included in the tally.

2.2. Types of E-voting Systems

2.2.1. Homomorphic Voting System. A homomorphic voting system uses a homomorphic cryptosystem to compute an election result without the need of decrypting individual votes. This systems guarantee that an algebraic operation performed on a plaintext corresponds to another algebraic operation performed on its correspondent ciphertext.

2.2.2. Mix-nets Voting Systems. A mix-nets voting system anonymously mixes the encrypted votes in order to generate a shuffled election's result, so that the privacy of the voters is assured. For that it uses a mix-net, which is composed by a chain of mix servers. Each mix server encrypts the votes and shuffles them, proceeding to pass them to the next server on the chain.

2.2.3. Blind Signatures Voting Systems. A blind signatures voting system authenticates the votes with a digital signature of an election authority. However, to maintain the voters' privacy, the election authority signs a blinded vote. The vote is then unblinded by the voter which is then encrypted and sent anonymously to the ballot box. To compute the results, every vote is decrypted and the validity of the votes can be verified, since they must all have the authority's signature.

2.3. E-voting Systems

In this section the e-voting systems researched are presented, including a brief description about each one of them. Every e-voting system presented here complies with most of the main security properties of e-voting stated in Section 2.1.1.

2.3.1. VeryVote. The VeryVote system [2] is a mix-net code voting system that adapts MarkPledge's cryptographic technique in order to achieve a cast-as-intended verification. The election server creates and sends to every voter a code sheet before the polls open. The election key is created by the trustees which is signed by the electoral commission and published in the bulletin board. On the election day, the voter types the code that corresponds to their favourite candidate.

It is then used the MarkPledge's encryption technique to create a receipt for the cast vote. The voter then checks the receipt to confirm that the vote confirmation code is associated with the selected candidate. This receipt is also signed by the electoral commission and published in the bulletin board to allow a claiming stage, during which the voters can check and revoke their votes.

Every validated vote goes through a mix-net protocol in order to be anonymized. After this process, the trustees decrypt and publish the votes in a shared and verifiable way which allows for other entities to verify the correctness of the vote decryption.

2.3.2. EVIV. The End-to-end Verifiable Internet Voting (EVIV) system [3], much like VeryVote, integrates the MarkPledge technique with a homomorphic code voting protocol. The main difference between both voting systems is the use of voter security tokens (VST's) that contain unique cryptographic key pairs used to encrypt the votes.

The voter gets registered in the electoral roll by presenting himself to a local authority office, getting a VST. When every voter is registered, the election parameters, an electoral roll containing the list of voters and their public keys are published on the bulletin board. Then the trustees get an ElGamal shared threshold key distributed between them which is verified by the Electoral Commission.

The voter can now register for the election by connecting his VST to a computer and to the Election Registrar. Then the VST receives the candidate list, the election public key, and creates a ballot encryption, signing it and sending it to the Election Registrar. If the ballot is correct, it's published in the Bulletin Board, that will contain all the valid ballots and be signed by the Election Registrar. Finally, the VST creates a code card that must be kept secret by the voter.

To vote, the voter connects his VST to the Ballot Box and introduces the code associated to his intended candidate. The receipt is then presented to the voter which allows the voter to immediately prevent the vote to reach the Ballot Box if he notices something wrong. At the end of the voting phase, all the data received in the Ballot Box is signed by the Electoral Commission and then published in the Bulletin Board.

In the last phase, anyone is able to verify all of the election public data without compromising the voters privacy.

2.3.3. Helios. Helios [4] is a homomorphic voting system that allows any willing observer to audit the entire process of an election.

An election in Helios has only one administrator that creates and is responsible for it. Its ballot is composed by multiple questions, each one having multiple choices which must be setup by the administrator. The administrator is also responsible to setup the trustees for the election, and he has the option to add Helios as one of the trustees. The trustees generate a shared election key pair using ElGamal Encryption and publish the shared election public key to the server. The administrator can add, update and remove voters at will. Each voter is identified by a name and an e-mail address, to where the voter's credentials (username and a randomly generated password), the hash of the election and a link to the voting booth will be sent. After everything is set, the administrator can freeze the ballot and open the election.

When the election begins, voters can enter the voting booth by accessing the link received by e-mail and begin the voting process. After the voter selects their desired choices for the ballot questions, their ballot is encrypted (using Exponential ElGamal Encryption) and a hash of the ciphertext (known as the ballot tracker) is displayed. The voter is now presented with a cast-as-intended verification. If the voter chooses to audit the ballot, the ciphertext and the randomness used to encrypt the selected choices are revealed which allows for the voter to verify the correct encryption of the ballot. After auditing, the voter will have to confirm once again their desired answers and have them encrypted in a new ballot. The voter may audit the ballots successively until being satisfied with the encryption process. Alternatively to auditing, the voter can choose to seal the ballot discarding all randomness and plaintext information, leaving only the ciphertext. The voter is then prompted to authenticate using the credentials received by e-mail and if successful, the encrypted vote will be recorded in the server, which in turn acknowledges the vote reception by sending an e-mail to the voter's e-mail address with the ballot tracker.

A voter may vote multiple times, only the last cast ballot will be counted. The ballot trackers are then displayed in a bulletin board, next to the correspondent voter's name or an alias. Anyone can access this bulletin board and find the encrypted votes posted there.

When the administrator closes the election, the encrypted tally is computed by aggregating every encrypted vote, making use of the additive homomorphism property of the exponential ElGamal cryptosystem used to previously encrypt the individual ballots. To decrypt the tally, the trustees must submit their decryption factors which are computed using the election secret key previously created. Once every trustee has submitted the decryption factors, the tally can be decrypted and the administrator can release the results to the public.

After the results have been released, it is possible for any observer to verify the tally. The verification program downloads every needed parameter, verifies every proof and re-performs the tally based on the decryptions.

The main issue with Helios is coercion. A voter that shows their selected choices together with the ballot tracker to a third party cannot vote again without the third party knowledge. Helios is also vulnerable to ballot stuffing attacks, since a dishonest bulletin board could add ballots to the tally without anyone noticing.

2.3.4. Belenios. Belenios [5] is a homomorphic e-voting system that partly implements the Helios-C protocol which makes use of voter signatures to avoid ballot stuffing attacks.

When setting up an election, the registrar generates and sends privately a signing key to each voter and their corresponding verification keys to the voting server. The server publishes the list of verification keys on the bulletin board, generates and sends a password for each voter in private.

The election key is generated using a threshold public key cryptosystem using ElGamal encryption. Each trustee sends his public key to the server together with a proof of knowledge (POK) of the secret key. The public election key is then published on the public bulletin board by the voting server.

During the voting phase, the voters select their vote which in turn is encrypted and signed by their voting device. The resulting ballot is sent via an authenticated channel to the voting server using a login and password mechanism. The server performs several checks in order to verify the validity of the ballot and adds it to the bulletin board if everything is in order. Voters can then check if their last submitted ballot appears in the bulletin board.

When the election is closed, the trustees contribute to the decryption of the list of the accepted ballots in the bulletin board by providing their decryption factors together with a POK of correct decryption.

Like Helios, Belenios also does not provide any coercion resistance since voters may provide the randomness used to produce their ballot or sell their voting material.

2.3.5. EPFL E-voting System. The EPFL E-voting System [6] is a mix-net laid upon a decentralized and distributed architecture that makes use of blockchain technology and verifiable cryptographic shuffles of ElGamal ciphertext pairs.

The system is handled by the cothority (short for collective authority) that provides a platform to handle arbitrary blocks of data through the use of an alternative implementation of blockchain, named skipchain.

Skipchain is used to store all the election related data. The master skipchain handles configuration data that is common to a set of elections, and its genesis block stores a list of servers that are meant to handle the protocols and skipchains, a list of election administrators, a public key of a front-end application and its own skipchain identifier.

When a new election is created an election skipchain is also generated. This skipchain contains all the data related to its election including the actual election data and the ballots cast by the voters. Each election skipchain has an identifier that is appended to the master skipchain in a separated block called ”link” and its genesis block contains the server list, its own skipchain identifier and more importantly, the election's public key. The correspondent secret key is shared among the different servers.

When a voter casts a ballot to the cothority, it is previously encrypted using the election public key. The ballot only contains its correspondent user's identifier and the encrypted vote in the form of an ElGamal ciphertext. When the administrator closes the election the submitted ballots are re-encrypted and permuted several times by the various servers. Finally, each server is prompted to decrypt the shuffled ballots with their own secrets and the final result is appended in the concluding block of the election skipchain. At no point during the encryption protocol is it necessary for one server to accumulate the shared secrets.

3. Methodology

This section presents the chosen solution and the reasoning behind its choice along with some modifications and improvements to it. It also details a field test that serves the purpose of identifying any problems and possible improvements to the implemented voting solution.

3.1. Helios

The solution that was chosen to be implemented from the systems detailed in Section 2.3 is the Helios Voting System [4]. From all the systems analyzed, Helios is the one that has been mostly used to run real-world elections. Helios has a robust front-end and doesn't require VST's or makes use of code voting systems which simplifies the voting process. Therefore, even being subject to coercion and ballot stuffing attacks, Helios is a great choice for low-risk, small scale environments such as university student governments.

Helios, with some modifications, can fulfill every requirement stated in Section 1.2. It also satisfies most of the criteria listed in Section 2.1.1, with the exception of non-coercibility and flexibility.

The Helios version used in this project was obtained from the Instituto Federal de Santa Catarina's GitHub repository in January of 2020 and its map is depicted in Figure 1. This map is organized by blocks each representing a major page of the platform. Each block also has a colour associated which represents the permission required to access that page or to perform that action in the platform. White blocks are accessible by any user including non-authenticated ones. Orange blocks are accessible only by registered voters of a given election. Purple blocks can only be accessed by platform administrators. These blocks correspond to the administration tool of the platform which provides an interface where trusted users can manage sensitive content on the platform site. Only a handful of users should have access to this tool and all their actions performed in it are recorded in logs. It's in this tool that any previously authenticated user can be given election administrator privileges which allows them to create and administer elections. Finally, the election administrators are the ones with access to the blue blocks.

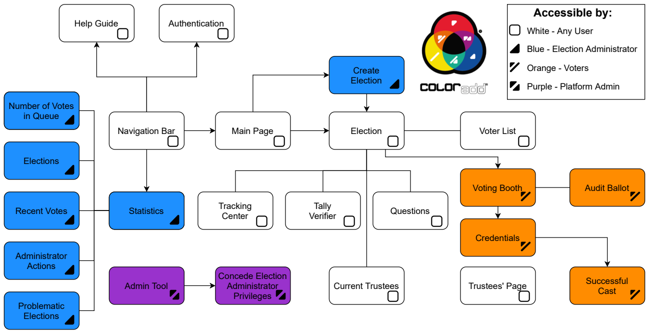

Figure 1. Helios Platform Original Map [7]

The user accesses the platform and is presented with its main page. Here, he has access to the navigation bar, a list of featured elections and, if the user is an election administrator, he will also have the option to create a new election. The navigation bar allows the user to authenticate, to visit the help guide and to return to the main page of the platform. Also, an election administrator has access to a menu called statistics in it, that contains diverse information about every election. To access an election, a user chooses an election from the list of featured elections in the main page which will bring him to the selected election's page. In there, the user has access to the election's voter list, the questions present in its ballot, a list of its current trustees, the tracking center and, if the tally is computed, the tally verifier. The election's page also gives access to the voting booth where the voting process will take place. The voting booth has an auditing tool, that allows the voters to verify if their choices were correctly encrypted into the ballot if they wish to. It also gives the option for the voter to post their audited ballot in the previously mentioned tracking center. The tracking center contains every audited ballot that was posted in it, allowing every voter to audit them. After encrypting and sealing the ballot, the user is prompted to insert credentials that were sent to him by the election administrator via e-mail and only after submitting them, the ballot is successfully cast. After casting the ballot, an e-mail confirming the cast of the ballot is sent to the voter's e-mail address. At last there is the trustees' page which is only accessible via URL. Each trustee has its own trustee page and accesses it by using an URL sent by the election's administrator by e-mail.

3.2. Implemented Changes

This subsection states the changes implemented on the original state of the voting platform that aim to improve its security and usability and also to fulfill the requirements stated in Section 1.2.

3.2.1. Initial Configurations. Helios strongly relies on sending e-mails, which means that an e-mail address was created specially for the platform to send these.

To configure the intended authentication system, it is required to set up the connection with the UL's Identity Management System. To do so, it was needed to install and configure a Shibboleth SP.

It was originally possible to associate a user with one and only one faculty. As a consequence of this, by default, every election administered by this user would also be associated with that faculty. This behaviour was not intended, and to circumvent this problem, users cannot be associated with faculties anymore. Instead, every election has now a faculty associated that is manually chosen by its administrator.

The platform needs election administrators that can create elections, but for security reasons not everyone can be an administrator. The solution implemented is to create a new role: delegates. Delegates are users considered to be responsible and as such bear the duty of assigning and removing election administrators.

3.2.2. Modifications by Block.

Admin Tool - There is no longer the option to set users as election administrators in the admin tool. Instead, it is possible to set users as delegates which can in turn assign and remove election administrators, as stated before.

Main Page - The list of featured elections was replaced by a list of faculties. Choosing a faculty from the list will display every election that is associated with it.

Delegates can now concede and revoke election administrator privileges to users and this new functionality is accessible through the main page of the platform.

A list containing user's attributes was added to the main page.

Navigation Bar - The UL's logo was added to the platform's navigation bar which redirects to the UL's website.

Authentication - To guarantee the uniqueness of each voter it is required a unique identifier. The attribute used initially to guarantee this was EPPN, which was composed by a RDN (CN), and an e-mail domain. However, the EPPN could not be used as the unique identifier since it was not immutable and therefore a different attribute had to be used as the unique identifier. The chosen replacement is the DN, which in this case is formed by 3 RDN's: CN, OU and O, with an enforcement of OU="Users".

Help Guide - A new help guide was implemented in the platform to ease the use of it by every user.

Statistics - This tool includes the number of votes in queue, a list of every election, a list containing every vote cast during the last 24 hours, a log of election administrators actions and a list of problematic elections.

It has been renamed to "Administration Panel" to better reflect it's purpose and the problematic elections were removed from it. The tool can now also be accessed by both election administrators and delegates.

Credentials - The purpose of the previously mentioned credentials that are sent via e-mail by the election administrator is to guarantee that users are registered voters. This is successfully achieved since only users whom have received these e-mails would have the correct credentials, guaranteeing that the users were indeed registered voters. However, with the user authentication correctly configured, a user can only access the voting booth and cast a ballot if he's correctly authenticated in the platform and has been registered in the election by its administrator which makes the use of credentials redundant and therefore, they were removed.

Create Election - When creating an election, the administrator will be prompted to choose a faculty to be associated with the election, guaranteeing that one election is always associated to one faculty.

Election - A user is now required to authenticate before accessing an election.

It was created an administrator panel in the election's main page which is only visible to the administrator. This panel is separated from all the other information, includes some admin actions (copy, edit and archive election) and shows the "Next Step" of the administrator.

An administrator can now delete an election in the administration panel if the ballot box isn't frozen yet. Also the options for the administrator to archive and copy the election are now only available after the release of the results.

When an administrator finished setting up an election and every trustee has uploaded their public key, the ballot can be frozen. If the election only has 1 trustee in it, a warning will now be shown to the administrator stating that a minimum of 2 trustees is highly advised to guarantee the integrity of the election.

The administrator can only e-mail a trustee if he either hasn't submitted his public key, or if, with the encrypted tally locked, he hasn't uploaded his decryption factors.

If for some reason the administrator decides to end the election earlier than stipulated, this action will now be stated in the election's main page, specifying the time at which the administrator ended the election.

The preview/review voting booth and the election tally verifier links were moved from the bottom of the page and a reminder to verify the election tally was added.

It was also possible to review the voting booth before the administrator locked the encrypted tally if the election original ending time has finished. Since the administrator can extend the election's ending date, reviewing the voting booth is now only allowed after the encrypted tally is locked.

The platform did not always display correctly if a user was registered as a voter or not which was fixed.

Voter List - The administrator had 2 options to register voters in his election. The first was to allow every authenticated user to vote which has been removed. The other option was to upload a list of voters, composed by the users' names, unique identifiers and e-mail addresses. A new option was implemented and it works in conjunction with the voters' list, providing more versatility to the administrators. An administrator can now also register a voter based on his attributes by choosing one or more categories. These categories are composed by 8 fields, 3 of them being static and the other 5 being dynamic. The static fields are "students", "professors" and "employees" and the administrator must always choose at least one of these when adding a new category. The dynamic fields are optional and named ”faculty”, ”establishment”, ”management”, ”area” and ”nucleus”. The dynamic fields have a hierarchic relation, where ”nucleus” belongs to ”area” which by its turn belongs to ”management” and so on and so forth. The dynamic fields are obtained during users authentication where the user’s attributes are verified and if the combination of these 5 fields is unfamiliar to the server, they are saved and made available as categories. Every voter can still receive e-mails from the platform since the voters registered by categories have their e-mail addresses present in their attributes. The templates of these e-mails have been changed: they don’t contain user credentials anymore, they remind the voter that they can vote as many times as they want to and they also remind the voter to verify the election tally if it has been computed.

It was added a button that allows the administrator to remove every voter manually uploaded instead of only being able to remove them one by one.

A voter and consequently his cast votes could also be removed from an election with the ballot already frozen. This is not possible anymore, the administrator can only remove voters before freezing the ballot.

Current Trustees - The trustees are added by the administrator that needs to submit the trustee’s name along with his e-mail address. Then, the administrator sends an e-mail to the trustee that contains the URL to his trustee page, so he can submit his public key and his decryption factors.

A trustee does not need to be authenticated to access his page and theoretically, anyone with knowledge of the URL can impersonate the trustee and make the submissions as him. This was fixed by associating every trustee’s URL to a previously authenticated user, which is now also accessible to authenticated trustees via ”Current Trustees”.

The original e-mail template that was sent to the trustees by the administrator has been divided into 2 different templates, each containing a brief guide to help the trustees perform the uploads of both the public key and the decryption factors.

Trustees’ Page - A possible exploit was found concerning the submission of the trustee’s public key. If the trustee saved the URL while submitting the public key, he could re-submit another public key with the ballot already frozen. This compromises the tally’s integrity since the election’s public key would not correspond to the key formed by the trustees anymore. This issue was mitigated by blocking public key re-submissions after freezing the ballot.

Voting Booth - There was too much information exposed in the voting booth. This excess of information can confuse voters, thus a big part of it was moved to the help guide. Only indispensable information that allows a user to vote without using the guide was kept in it.

Audit Ballot - Excess information was also moved to the help guide only leaving indispensable information that allows a user to audit the ballot without using the guide.

Confirm Cast - After sealing the ballot, the user was prompted to insert the previously mentioned credentials that were sent to him via e-mail by the election administrator. Since these credentials are no longer used, the user is instead prompted to confirm the cast of his sealed ballot.

Tally Verifier - The result of the verification is now more organized and less crowded. A reminder was also added after the verification finishes for the voter to verify the presence of his ballot tracker in the results.

3.2.3. Solution Map. After the implementation of every solution, the platform map is slightly changed being depicted in Figure 2.

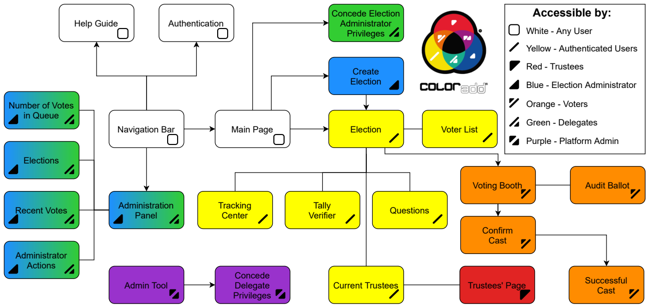

Figure 2. Helios Platform Solution Map [7]

3.3. Field Test

With the new solution implemented, it was time to conduct a field test with a significant number of users. The main objective of this field test was to identify any problems, oversights and ways of improving the platform. This was achieved by designing a feign election, gathering people to vote on it and to provide feedback detailing their voting experience.

The election took the form of an innocuous poll, asking the voters which was their preferred e-learning solution (from a short range of choices). Voters could be any UL’s member who belonged to any of the following groups: students, professors and employees from IST; students, professors and employees from Faculdade de Ciências (FC); employees from the UL’s rectory. Finally, 2 trustees were also chosen to follow and verify the election. The election was open during one day to vote from 9 am to 11 pm and a form was made available to obtain feedback from the participants that wished to share their voting experience.

4. Detected Issues

Every issue detected with the field test is depicted in this section. The field test had 97 participants, 51 of which were students, 25 professors and 21 employees. Of the 97, 36 filled the form, providing information about any issues that were detected. These issues are detailed in this section followed by any other issues detected and communicated by the participants of the field test.

4.1. Issues Encountered by Voters

Out of the 36 participants that filled the form there were 5 students, 17 professors and 14 employees and every issue detected has a code assigned to it, as well as the percentage of participants that encountered it and they are organized in 3 levels: *General*, *Voting* and *Auditing*. It should also be noted that every participant that filled the form was able to cast a vote.

4.1.1. General.

[G1] *19% did not find the platform easy and/or intuitive to use* - The main cause of this issue is that the platform's interface is only in English and the participants were mainly Portuguese. Even though the participants could understand English some technical and unfamiliar terms may difficult the use of the platform. There is also no visual indication that a user is authenticated and after the authentication some users couldn't tell what to do next. Finally, it is not clear that the elections are associated to one faculty - some users thought that by performing the authentication via their faculty, that the election would also be associated with it.
[G2] *3% had their attributes incorrect* - There are users with wrong attributes which can inhibit a user to vote in elections configured with categories.
[G3] *25% couldn't understand the meaning or usefulness of some technical terms* - Even though all the participants could cast a vote, some of them had difficult understanding some more technical terms such as *Audit*, *Tracking center* and *Ballot tracker*.
[G4] *47% did not find the help e-mail address in the main page of the election* - The help e-mail address can be found in the bottom of the election page and most users probably don't scroll through the whole page to be able to find it.
[G5] *36% did not know that he could cast a vote multiple times* - This feature of the platform is mentioned after casting a ballot, in the help guide and in the e-mail templates. However, this information can easily be overlooked since most users will not read all the information shown to them.

4.1.2. Voting Booth.

[V1] *19% felt excess or lack of information during the voting process* - Some users felt overwhelmed by the information displayed in the screen while others didn't find the information that they needed.
[V2] *31% felt that the voting process is too long* - Too many steps, verifications and "clicks" are needed to cast a vote. There are also too many successive reminders to save the ballot tracker. Also the buttons to go to the next step are often in different locations of the page.
[V3] *6% encountered problems while casting the voting ballot* - The button to cast the ballot is too small in comparison to the others. Also, the last page to confirm the submission of the cast ballot creates some confusion.
[V4] *25% did not receive the e-mail confirming the cast ballot* - The e-mail address created to send e-mails in bulk was not authorized to do so being limited to send 100 e-mails per day. Since every ballot cast would result in sending one e-mail and some voters has cast multiple ballots, the e-mail address was blocked from sending more e-mails which resulted in the latest participants not receiving the e-mails confirming the cast ballot.
[V5] *14% did not save the ballot tracker* - As previously mentioned in **[G3]**, many voters didn't know what to do with it. Others were just lazy or believed that the ballot tracker would be available to them if they needed to consult it.
[V6] *6% had concerns regarding their voting privacy* - Some voters felt that it wasn't clear enough that their vote was secret and that the platform didn't transmit enough robustness and confidentiality.

4.1.3. Audit Ballot.

[A1] *58% did not understand that an audited ballot isn't accounted for the tally* - Even though that every of the participants was able to cast a vote many didn't understand the difference between auditing and casting a ballot.
[A2] *50% did not audit a ballot* - Half of the users did not audit a ballot. Most of them simply weren't interested in doing so but some tried and gave up halfway through the process.
[A3] *28% of auditors felt excess or lack of information during the audit process* - The auditing tool is not intuitive enough for being used by the average voter since it involves copying the opened ballot and pasting it in the verifier.
[A4] *39% of auditors didn't understand the purpose of the tracking center* - The tracking center generated some confusion since some voters thought that their cast ballot would be registered in it.

4.2. Other Issues

Some issues were not possible to detect directly with the questions of the form being identified by the trustees, participants and the election administrator. This section describes these issues and when applicable, the causes of them.

[O1] *Redirect* - After authenticating via the navigation bar the user is not being correctly redirected always returning to the main page instead.
[O2] *Review the voting booth* - After the election's administrator closes the election and no more ballots can be deposited, it is possible for the participants to review the voting booth. This review is identical to the actual voting with the exception of no ballot being cast. However, a user who has not participated in the election (typically a late voter) and reviews the voting booth is still added to the voter list. Even though that this user cannot cast any ballot, any other participants may notice the increase of the number of voters after the closure of the election which in turn may lead to an incorrect assumption that people are still voting and breaking the confidence in the system.
[O3] *Aliases* - In the case of elections using aliases, only the administrator can see the names of the voters. This means that in practice, a malevolent administrator manually adding participants to the election that should not be able to cast a vote would go unnoticed by other participants and even trustees.
[O4] *URL manipulation* - By manipulating either the URL's of the pages that contain the voter list or the current

trustees, it is possible for any participant to see the names of the election voters (even with aliases enabled) and the trustees' e-mail addresses, respectively. Both of these URL's are necessary for the tally verifier and therefore cannot be deleted.

[O5] *Secret key submission* - Trustees are able to both generate the key pair on the browser or to submit a previously generated key pair. In either way, the voting system uses the trustee's submitted secret key to generate a POK with JavaScript that allows the system to verify if the trustee knows his own secret key without verifying the key directly. After the POK is generated, the trustee's secret key is discarded. This event has a direct impact on the trustee's confidence in the platform since they are obligated to submit their private key without knowing what is being done behind the scenes with it.

[O6] *Changing the voting choice* - Some voters had difficulty or were unable to change their voting choice in the voting booth. Since the election's question required exactly 1 choice, voters assumed that to change their choice it was enough to just select any other choice. However, the input type used for the ballot choices are check boxes that are designed to block when the predefined maximum number of choices is selected, in this case, blocking when a voter selects 1 choice and consequently not allowing voters to directly select another choice. To change the voting choice, voters had to deselect their original choice, and only then could they select a new choice which confused the voters.

[O7] *Auditing and Verification* - Users have both the single-ballot and tally verifiers to validate their encrypted choices and the integrity of the election, respectively. Although these tools are trustworthy they are still integrated in the voting system which means that voters may be reticent in trusting them.

[O8] *Changing the DN* - It is still possible for a user to change his unique identifier. This occurs when users activate their "ULisboa User Account" since doing so changes their DN and therefore their unique ID is altered. Even though each user can only perform this once (when activating his "ULisboa User Account"), it still is a very concerning vulnerability.

5. Issue Mitigation and Improvements

The solutions implemented for the problems stated in Section 4 and some other improvements are stated in this section organized blockwise, followed by all the other issues that are currently known but were not resolved and the details of a real world election that took place using the implemented final solution.

5.1. General Issue Mitigation

[G1] - The platform's whole interface is now also available in Portuguese and the elections are no longer associated to a faculty. The layout and usability was also reviewed with special attention to remove complex sentences with more technical terms, explaining it further or moving it to the help guide.

[V4] - The e-mail address is now, theoretically properly configured, being allowed to send 1000 e-mails per hour. The templates of the e-mails have more useful information and are now automatically signed by the election's administrator.

5.2. Issue Mitigation by Blocks

Main Page: [G1] - Since the faculties were removed from the platform, the main page now contains instead a list of elections that an authenticated user may participate. Also a list that contains every election (with its ballot frozen and not archived) is now accessible in the main page.

Entities List: This new block contains a list of entities that should be contacted by users who want to create and administer an election.

Navigation Bar: [G1] - The navigation bar now states if a user is authenticated by showing his name. It also contains a button to change the platform's language between Portuguese and English.

Authentication: [O1] - After authenticating, the user is now correctly redirected to the page that he was before.

Help Guide: [G3, V6] - The help guide is now more detailed, including explanations about more technical terms and a section that details how the voting system works and guarantees that the deposited votes are private.

Create Election: The elections are now private and have aliases enabled by default to comply with General Data Protection Regulation (GDPR), but the administrator can still create non-private elections or disable the aliases.

Election: [G1] - The election's main page has been streamlined with the objective of simplifying the voting process by diminishing the intensity of less crucial information.

[G4] - The name of the election's administrator and the help e-mail address were moved to the top of the election's page so that voters can find it more easily.

[G5] - A reminder stating that voters can cast a vote as many times as they want is now displayed in the election's page.

[O2] - The option to review the voting booth has been removed. The utility of the revision is limited to auditing ballots and therefore, it has been removed from the platform.

Tracking Center: [A4] - The tracking center does not bring much utility since its main purpose is to allow participants to audit ballots posted there by other voters but without giving any way to know what was the original vote intention of the voter, and therefore not allowing a valid audit. This resulted in the removal of the tracking center.

Voter List: [O3] - The voter list now contains a field stating if the voter was manually added by the election's administrator or if he was able to vote using his attributes.

[O4] - Since the voter names aren't used in the tally verifier they are no longer sent to the URL which means that they are no longer exposed in it.

Current Trustees: [O4] - Since the trustees' e-mail addresses aren't used in the tally verifier they are no longer sent to the URL which means that they are no longer exposed in it.

Trustees' Page: [O5] - Trustees can still generate the key pair directly in the browser and submit it if they wish to but it's no longer mandatory. It is now possible to generate the key pair in the browser in offline mode by providing the secret key - this will generate the public key and the POK. The trustee saves these, and can then manually upload them to the server with the browser back in online mode without ever having to submit his secret key to the server. Alternatively, the trustee may generate his public key locally using discrete logarithms and the election's fixed parameters and then submitting it to generate the POK. Then the trustee saves his secret key and submits the public key and the POK to the server.

Voting Booth: [V1] - The information that was more technical and not that useful for a common voter has been moved to the help guide.

[V2] - Every button to go "back" is now displayed on the left of the window while the buttons to "continue" are displayed on the right of the window adding some coherence to the voting booth. Also a button with the option to go "back" to confirm the answers before casting the ballot was added.

[V3] - The size of the button to cast the ballot was increased in comparison to the others.

[V5] - Before casting a ballot, the voters are informed that their ballot tracker will be sent by e-mail although they should still verify if that ballot tracker is equal to the one displayed in the voting booth.

[V6] - After confirming his choices for the questions of the ballot, the voter now clicks in "Encrypt Secret Ballot" instead of just clicking a generic "Next Step".

[O6] - When a question requires to choose exactly 1 choice, the check boxes are replaced by radio buttons to avoid any possible confusion while switching choices.

Audit Ballot: [A1] - Information was added to the audit tool stating that the audited ballot is discarded and cannot be counted to the tally. It is required to generate and encrypt a new ballot for it to be accounted for the tally.

[O7] - It is now possible to download some of the election's information necessary to audit ballots locally or in a different machine. The single ballot verifier was also exported and adapted in order to make this possible to happen.

When the ballot is incorrectly formatted, an error will now be displayed to the voter, and since the tracking center has been removed, the option for the voter to post their audited ballot in it is no longer available.

Confirm Cast: [V2, V3] - Voters that reach this step have already clicked to cast their ballot in the voting booth which renders pointless this cast confirmation. Therefore, this step is now skipped.

Successful Cast: [V2] - The reminder to save the ballot tracker was removed and now it is informed that it will be sent by e-mail.

Tally Verifier: [O7] - It is now possible to download some of the election's information necessary to verify the tally locally or in a different machine. The tally verifier was also exported and adapted in order to make this possible to happen.

5.2.1. Final Map.

After the mitigation of the issues, the platform map is again slightly changed being depicted in Figure 3.

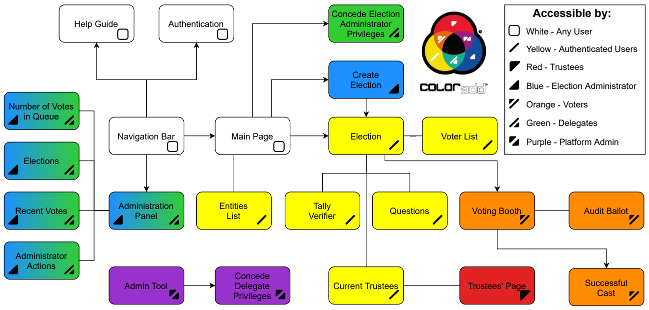

Figure 3. Helios Platform Final Map [7]

5.3. Unsolved issues

[A2, A3] - It is not expected that most voters will use the auditing tool. The main goal of the verifying tools is to give assurance to the common voters that someone can and will use these tools and not necessarily being used by every participant. Consequently, it is admissible that half of the participants do not audit any ballots.

[G2] - Voters with incorrect attributes can contact the election's administrator so that he can manually add them to the election. This works as a temporary patch but it's not a solution to this issue. The IdP has to rectify the attributes of every user to ensure that they are correct and perform a routine maintenance to guarantee the health of them.

[G1] - The platform's presentation can also be massively improved, either in layout, colours and fonts. It should also be taken into account that members of the UL community with disabilities have difficulty using the tool and therefore accessibility standards for them should also be implanted in the platform.

[O8] - This issue can be mitigated in the IdP or in the platform.

- *IdP* - The DN is frozen and cannot be changed no matter what (with the exception of "OU");
- *Platform* - A different unique, immutable attribute that does not contain legal identification is selected to be the unique identifier.

5.4. Real World Election

The final implemented solution has already proven its utility - in May of the current year, the elections for the corporate bodies of the AAUL took place with a mixed electoral system for 2 days allowing online voting on the first and presential voting on the second. The electoral roll consisted of every UL student, totaling just over 50,000 voters. The online voting took place on the implemented e-voting system and counted with the participation of 479 different voters, running with two different types of incidents reported:

- A few participants reported that they could not authenticate in the platform due to errors with the IdP;
- It was not possible to send e-mails informing the election results to every participant, because the quota of e-mails sent was, once again, exceeded.

The presential voting that took place on the next day counted with the participation of 26 voters. This means that the election counted with the participation of a total of 505 real voters, just over 1% of all the potential voters. It is also worth noting that 94.8% of the real voters chose to vote electronically rather than in person, highlighting both the preference that voters have in casting their votes electronically rather than having to go in person to the stipulated polling place, and the confidence in the electronic voting system for a small scale election.

It is not possible to be sure how many voters ran into problems that prevented them from voting electronically, since many of them may not have seeked for help. Either way, a mixed electoral system with 2 days, the first electronic and the next being in person, turns out to be ideal to bypass any authentication problem while the IdP does not mitigate the reported authentication errors.

6. Conclusions and Future Work

This thesis presents the implementation of an already existing e-voting system in order to serve the UL's community necessities in this matter allowing both the creation of fully online elections and if necessary, hybrid elections. Even though the chosen system, Helios, is not state of the art it has been proven along the years of its existence to be a reliable solution for low risk elections. This thesis also presents the modifications performed to the original solution in order to make use of the voters attributes defined in the IdP, followed by conducting a field test and a detailed analysis of every issue encountered in it along with the measures taken in order to mitigate them. It ends by stating the issues that could not be resolved and by detailing a real world election that took place using the final implemented e-voting system.

6.1. System Limitations and Future Work

The biggest limitation of the implemented solution is coercion. As stated before in this thesis, Helios does not offer reliable methods to resist coercion and therefore it should be used only in low-coercive and small scale environments, e.g., university student governments.

Helios is also vulnerable to ballot stuffing since a dishonest bulletin board could add ballots without anyone noticing.

Another current limitation is the necessity to trust the system when using both the audit and tally verifiers to ensure E2E verifiability. This limitation can be easily mitigated by distributing the verifiers across different machines which ensures a E2E verifiability if at least one of the machines is trustful. Both the single ballot and the tally verifiers have already been exported and adapted so that they can easily be implemented in the future.

Some users cannot authenticate in the platform via IdP which makes it completely impossible for them to participate in elections. Also, it has been reported that some users had their attributes configured incorrectly resulting in inaccurate permissions to access and vote on different elections. Finally, some users may still be able to change their unique identifier. These issues have to be solved directly by the IdP which must mitigate the errors that users encounter while trying to authenticate, assure the correct assignment and continual update of the users' attributes and send an attribute that may be properly used as the unique identifier.

As stated in Section 5.4, e-mails sent in bulk are still exceeding the previously set quota. It is therefore required further investigation to determine the reason for such occurrence.

If it is considered useful, a log out option may be added to the navigation bar for authenticated users. However, it is needed to be taken into account that the log out must be performed on the IdP since performing it directly on the SP is ineffective.

Another useful and interesting implementation would be, for example, Shamir's Secret Sharing Scheme. This cryptosystem would give more flexibility to the solution allowing the tally to be decrypted without requiring that every trustee submits their decryption factors.

The platform's front-end also needs to be updated in order to boost the platform's presentation providing a more user-friendly interface and an overall better voting experience.

Finally, it should also be taken into account that some members of the UL's community have disabilities that may difficult their use of the platform. With this in mind, accessibility standards should be implemented in the platform.

References

[1] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[2] R. Joaquim, C. Ribeiro, and P. Ferreira, "Veryvote: A voter verifiable code voting system," in *International Conference on E-Voting and Identity*. Springer, 2009, pp. 106–121.

[3] R. Joaquim, P. Ferreira, and C. Ribeiro, "EVIV: An end-to-end verifiable Internet voting system," *Computers & Security*, vol. 32, pp. 170–191, 2013.

[4] B. Adida, "Helios: Web-based Open-Audit Voting," in *USENIX security symposium*, vol. 17, 2008, pp. 335–348.

[5] V. Cortier, P. Gaudry, and S. Glondu, "Belenios: a simple private and verifiable electronic voting system," in *Foundations of Security, Protocols, and Equational Reasoning*. Springer, 2019, pp. 214–238.

[6] A. Caforio, L. Gasser, and P. Jovanovic, "A Decentralized and Distributed E-voting Scheme Based on Verifiable Cryptographic Shuffles," 2017.

[7] M. Neiva, "ColorADD, color identification system," 2010.