

# Enhancing Helios with Biometric Authentication and Smartphone-Automated Verifiability

Jurlind Budurushi<sup>1,\*</sup>, Khalid Abdallah<sup>2</sup>, Farhan Al Sadi<sup>2</sup>, Hosam Zarouk<sup>2</sup>,  
Abdelwahab Almasri<sup>2</sup> and Armstrong Nhlabatsi<sup>2</sup>

<sup>1</sup>Baden-Württemberg Cooperative State University Karlsruhe, Germany

<sup>2</sup>Qatar University, Qatar

## Abstract

Although extensively used, studied, and continuously improved, the current implementation of Helios suffers from various security and usability shortcomings. From a security point of view, the reliance on a single factor for voter authentication makes it susceptible to impersonation attacks. Furthermore, the exposure of voters' identities alongside their respective ballot trackers on the bulletin board compromises voter participation privacy. Regarding usability a significant issue resides in the complexity of the vote verification process, particularly challenging for non-technical voters dealing with large encrypted datasets. This paper tackles these shortcomings in the implementation of Helios, improving both its security and usability aspects. From a security perspective, we enhance the voter authentication mechanism by integrating biometric authentication into Helios. Additionally, we ensure voter participation privacy by dissociating voters' identities from their ballot trackers on the bulletin board, while mitigating potential clash attacks. In terms of usability enhancements, we introduce a QR code mechanism along with the implementation of a corresponding mobile application authenticator, making the vote verification process simpler and more efficient for voters.

## Keywords

Internet Voting, Biometric Authentication, Coercion Resistance, Usability, Verifiability

## 1. Introduction

Internet voting continues to attract the interest of the scientific community, decision-makers, and political spheres due to its potential to transform electoral processes. In theory, internet voting offers a spectrum of benefits, including improved voter accessibility and convenience. However, it also poses significant drawbacks, such as security vulnerabilities, usability issues, and technical complexities. Consequently, practical engagement with internet voting is imperative to explore and better understand its advantages, in particular for addressing existing challenges effectively.

Helios [1], an open-source internet voting system, has been instrumental in exploring practical aspects, enabling a deeper understanding of the benefits and drawbacks associated with internet voting. Hence, Helios has undergone extensive examinations, covering aspects of usability [2, 3] and security [4, 5]. Its refinement has addressed various factors, such as trust assumptions [6] and participation privacy [7]. Moreover, Helios has served as the foundation for the development of novel systems, e.g., the Zeus voting system [8], and the Apollo voting system [9]. Furthermore, Helios has been used in small-scale election settings, including the ACM general elections [10], and since 2010 in the elections of the International Association for Cryptologic Research (IACR)<sup>2</sup>. However, the current Helios implementation suffers from various security and usability shortcomings. One of the key security limitations involves its reliance on a single factor for voter authentication, which exposes the system to threats of impersonation [11]. To enhance authentication robustness, we incorporate

*Proceedings EGOV-CeDEM-ePart conference, August 31 - September 4, 2025, University for Continuing Education, Krems, Austria*

\*Corresponding author.

✉ jurlind.budurushi@dhbw-karlsruhe.de (J. Budurushi); ka2008033@student.qu.edu.qa (K. Abdallah);

fa1908826@student.qu.edu.qa (F. A. Sadi); hz1902702@student.qu.edu.qa (H. Zarouk); aa2003806@student.qu.edu.qa

(A. Almasri); armstrong.nhlabatsi@qu.edu.qa (A. Nhlabatsi)

🌐 <https://jurlindbudurushi.com> (J. Budurushi)

🆔 0000-0002-6732-4400 (J. Budurushi); 0000-0002-3407-7466 (A. Nhlabatsi)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

biometric verification as a second factor. To ensure the confidentiality of biometric data, our approach employs privacy-preserving cryptographic techniques [12]. Another concern arises from the public association of voter identities with their corresponding ballot trackers on the election bulletin board, which compromises voter anonymity and participation privacy [7]. To address this, we decouple voter identities from their ballot trackers on the bulletin board, while at the same time mitigating potential ballot copying [13] and clash attacks [14]. In terms of usability, the verification process can be complex and inaccessible, particularly for non-expert users who may struggle with cryptographic procedures [4, 3]. To improve usability and promote greater voter engagement, we introduce a QR code-based verification mechanism, supported by a dedicated mobile application. This enhancement simplifies the end-to-end verification workflow, making it more intuitive and efficient for all users.

## 2. Background

This section lays the foundation of our work. We introduce the concept of end-to-end verifiability in internet voting and provide an overview of the Helios voting system. Additionally, we review relevant literature and discuss related works.

### 2.1. End-to-end verifiability in Internet voting

*End-to-end verifiability* in internet voting, similar to traditional and electronic voting at polling stations, refers to the ability to verify the correctness of every step in the election process. *End-to-end verifiability* aligns with the *public nature of elections* principle established by the German Federal Constitutional Court in 2009. This principle mandates that any voter should be able to verify each essential step of the election and its outcome reliably and without needing expert knowledge [15].

Based on the systematization of knowledge on verifiability notions in electronic voting [16], *end-to-end verifiability* comprises two components: *individual verifiability* and *universal verifiability*. Individual verifiability includes *cast-as-intended*, which ensures the voter’s intention is correctly captured during casting, and *recorded-as-cast*, which guarantees the vote remains unaltered until tallying. Universal verifiability, or *tallied-as-recorded*, ensures that all recorded votes are accurately tallied and verified.

### 2.2. The Helios voting system

**Overview** – Helios is an open-source, end-to-end verifiable web-based voting system built on a traditional client-server architecture [1]. Users can access the system via any modern web browser, while the server can be either self-hosted or hosted by the Helios maintainers. Helios is designed for elections, where election integrity and ballot secrecy are important, but the risk of coercion is minimal. Typical use cases include student government elections and elections within non-governmental organizations. The election process using Helios can be separated into three steps: election management, voting, and auditing [17].

**Election management** – In this step, the election administrator creates the election by defining its name, the voter list, the voting options, the election configurations (such as closed or open voter lists, voter support contact address, use of voter aliases, and randomization of voting options), and the trustees<sup>1</sup>. Trustees are crucial for maintaining vote secrecy. Each trustee generates a public/private key pair and uploads the public key to the Helios server. During the tally, trustees use their private keys to decrypt the votes.

After setting up the election, the administrator freezes it and invites the voters to participate via email. The invitation includes the election description, a unique election fingerprint, a link to the online voting booth, and voter credentials. Helios also supports third-party authentication services like Google, Facebook, or custom SSO/LDAP servers, in which case the invitation email does not include voter credentials.

---

<sup>1</sup>The Helios server serves by default a trustee. However, the server can be removed from the trustees’ list and any number of new trustees can be added.

**Voting** – Upon receiving their invitation, voters can begin the voting process by visiting the online voting booth and selecting their preferred choices. After reviewing and confirming their selection, the ballot is encrypted to ensure vote secrecy and a unique ballot tracker is displayed. Voters should record this ballot tracker, which uniquely identifies their encrypted ballot, either by printing or writing it down.

Voters can then choose to submit the encrypted ballot to the Helios server or verify that it accurately reflects their selection. If the voter decides to verify, Helios reveals the randomness used in the ballot encryption. Thereby, the voter can use any third party tool to verify the correctness of the ballot encryption process. To verify and be convinced about the correctness of the ballot encryption process, voters repeat these steps (select and then verify) any number of times. This verification process, referred as the *Benaloh challenge* [18], ensures cast-as-intended verifiability. Note that the voter cannot submit the verified ballot, because this would allow voters to prove to a third party how they voted, and consequently allow vote selling or buying.

When ready to submit, voters enter their credentials and send the encrypted ballot to the Helios server. The ballot tracker allows voters to verify that their ballot was received and stored correctly, ensuring recorded-as-cast verifiability. Entering credentials at the final step prevents a compromised Helios system from targeting specific voter populations, e.g., elderly voters might not necessary verify or repeat the verification step more than once. To summarize, if the encrypted ballot contains a modified selection, the cast-as-intended verification would detect it. If the encrypted ballot is modified at submission, the change in the ballot tracker would reveal the discrepancy.

**Auditing** – The election audit step involves both individual and universal verifiability. Individual verifiability, ensured by voters during the voting step, allows them to verify their own ballots. Universal verifiability occurs during the tally process and can be audited by anyone.

After voting concludes, the Helios server computes the election tally by aggregating the cast ballots using the homomorphic properties of the encryption scheme, which ensures vote secrecy. Trustees then decrypt the encrypted tally using their private keys, performing partial decryption publicly. Once the tally is complete and decrypted, all information needed to verify the election tally is available from the Helios server, except for the trustees' private keys, which are not required for the audit. To minimize the risk of compromising vote secrecy in the future, it is recommended that trustees destroy their private keys after the tally.

### 2.3. Related work

Helios [1], is an open-source and well-established system designed for verifiable internet voting. It has been used in various electoral contexts, such as the University president election at UC Louvain [19], the ACM general elections [10], and since 2010 in the elections of the International Association for Cryptologic Research (IACR)<sup>2</sup>. Consequently, Helios has been extensively studied in the literature, and several usability and security enhancements, as well as feature extensions, have been proposed.

Many of the proposed enhancements focus on strengthening *vote integrity*. For example, the Zeus voting system [8], the Apollo voting system [9], the Selene voting protocol [20], and the proposals by Bernhard et al. [21], Escala et al. [22], and Guasch et al. [23] introduce different ways and new security mechanisms to ensure *individual verifiability*. In contrast to these proposals, we do not modify the current *individual verifiability* process in Helios, but rather improve it by using QR codes and implementing a mobile application that supports voters in executing the required verification steps [24, 4, 5]. Furthermore, contrary to [4], our integration of QR codes in the verification process not only goes beyond a theoretical proposal but also eliminates the need for a trusted third-party.

Other proposals focus on improving *eligibility* of voters, e.g., [6], [25], [26], [7] and [27]. While [6] and [25] introduce modifications concerning only *verifiable eligibility*, such as requiring voters to sign their ballots upon casting or token-based encryption, [26] introduces *strong receipt-freeness*, and [7] and [27] achieve *private eligibility verifiability*. Similar to [6] and [25], our enhancement achieves voter

---

<sup>2</sup><https://www.iacr.org/elections/>

*eligibility* by enhancing the voter registration and authentication process through the integration of biometrics. Note that the privacy of biometric attributes is protected by a cryptographic secret sharing scheme [12]. Furthermore, in accordance with [7] and [27], our enhancement supports *private eligibility verifiability*. To support *private eligibility verifiability*, we remove the link between voters and their corresponding ballot trackers on the public bulletin board. Our enhancement not only safeguards voter anonymity, but also mitigates ballot copying [13] and clash attacks [14]. Moreover, unlike [7] and [27], our enhancement mitigates the challenge of overcrowding the bulletin board. Building on the notion of *strong receipt-freeness* [26], we introduce the notion of *practical coercion-resistance*. We achieve *practical coercion-resistance* by allowing voters to update their vote and assuming that voters are not under the influence of a coercer during the entire election period.

To summarize, our enhancements surpass mere theoretical conjecture, as we have substantiated our proposal into tangible outcomes by implementing a robust proof of concept. While our literature investigation encompasses insights into usability and security enhancements related to *vote integrity* and *voter eligibility*, it is essential to acknowledge the existence of additional research that delve into alternative aspects of improving the usability and security of Helios. For instance, [19] improves vote privacy and fairness, [6] introduces distributed tallying, [28] ensures long term privacy, [29] enables a boardroom voting setting, [30] enables proxy voting, [31] introduces blind ballot copying, [32] supports quadratic voting, and [2, 24] improve usability of the interfaces. These works, while significant, fall outside our current scope.

### 3. Implementation

This section provides an overview of the principal extended and newly introduced components in Helios, based on forking the main Helios branch on GitHub<sup>3</sup>. Thereby, we describe the high-level system architecture, the integration of biometric authentication, and the implemented mobile application HelioScan. Other important adaptations of Helios, such as the structural model and databases, are documented on GitHub<sup>4</sup>.

#### 3.1. High level system architecture

To enable and facilitate improvements of Helios security and usability, it is necessary to extend and integrate new components and functionality to the base of the Helios architecture. Figure 1 illustrates the Helios architecture, with new components and functionalities added to the base system, highlighted in red boxes. Note that we have derived the current Helios architecture by thoroughly investigating the corresponding documentation and code.

#### 3.2. Biometric registration and authentication

When a user, for example an admin, a voter, or a trustee, registers for the first time in Helios, the user can register through one of the third-party authenticator options Helios implements, i.e., Google or GitHub. Thus, the user is redirected to the corresponding authenticator page and is required to authenticate using their respective Google<sup>5</sup> or GitHub<sup>6</sup> account. After successful authentication, the third-party authenticator sends Helios the required information about the user contained in an OAuth token<sup>7</sup>, such as name and email. This information is subsequently stored on the Helios database server for use in future authentication processes.

We extended the existing registration and authentication process by integrating the APIs of Regula Forensics<sup>8</sup>, hence adding facial recognition as an additional security layer. As a result, the registration

<sup>3</sup><https://github.com/benadida/helios>

<sup>4</sup>[https://github.com/Jurlind/Enhanced\\_Helios\\_QatarUniversity](https://github.com/Jurlind/Enhanced_Helios_QatarUniversity)

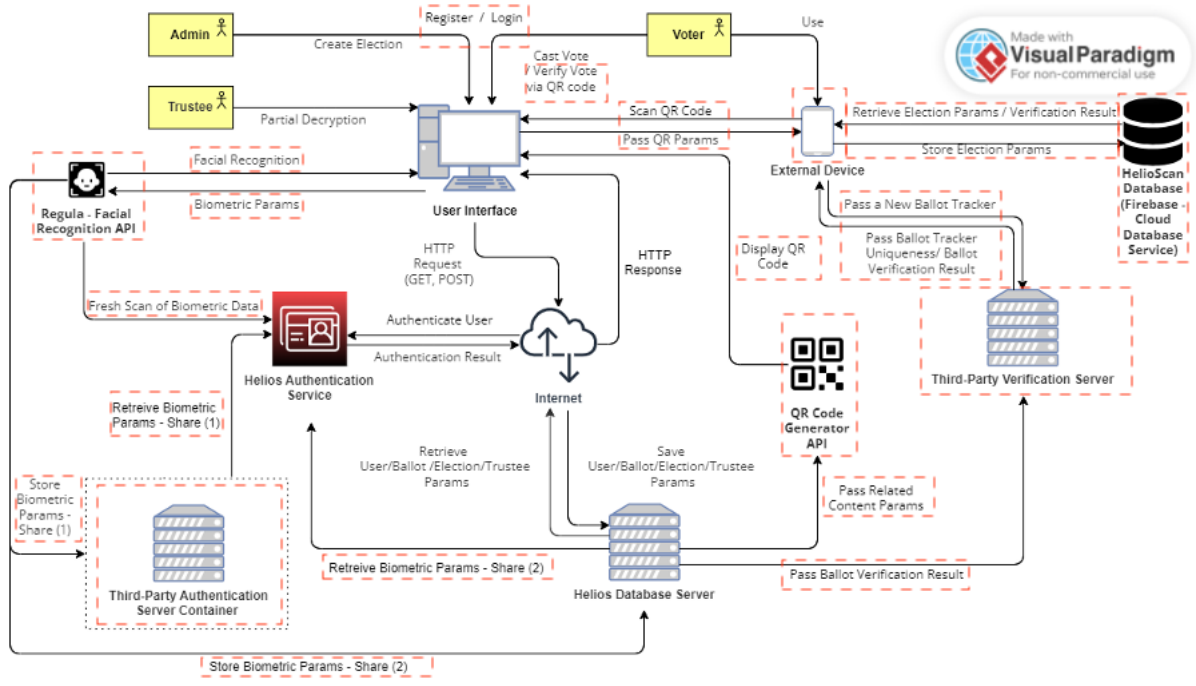
<sup>5</sup><https://www.google.com/account/about/>

<sup>6</sup><https://github.com/signup>

<sup>7</sup><https://auth0.com/docs/>

<sup>8</sup><https://docs.regulaforensics.com/develop/face-sdk/>

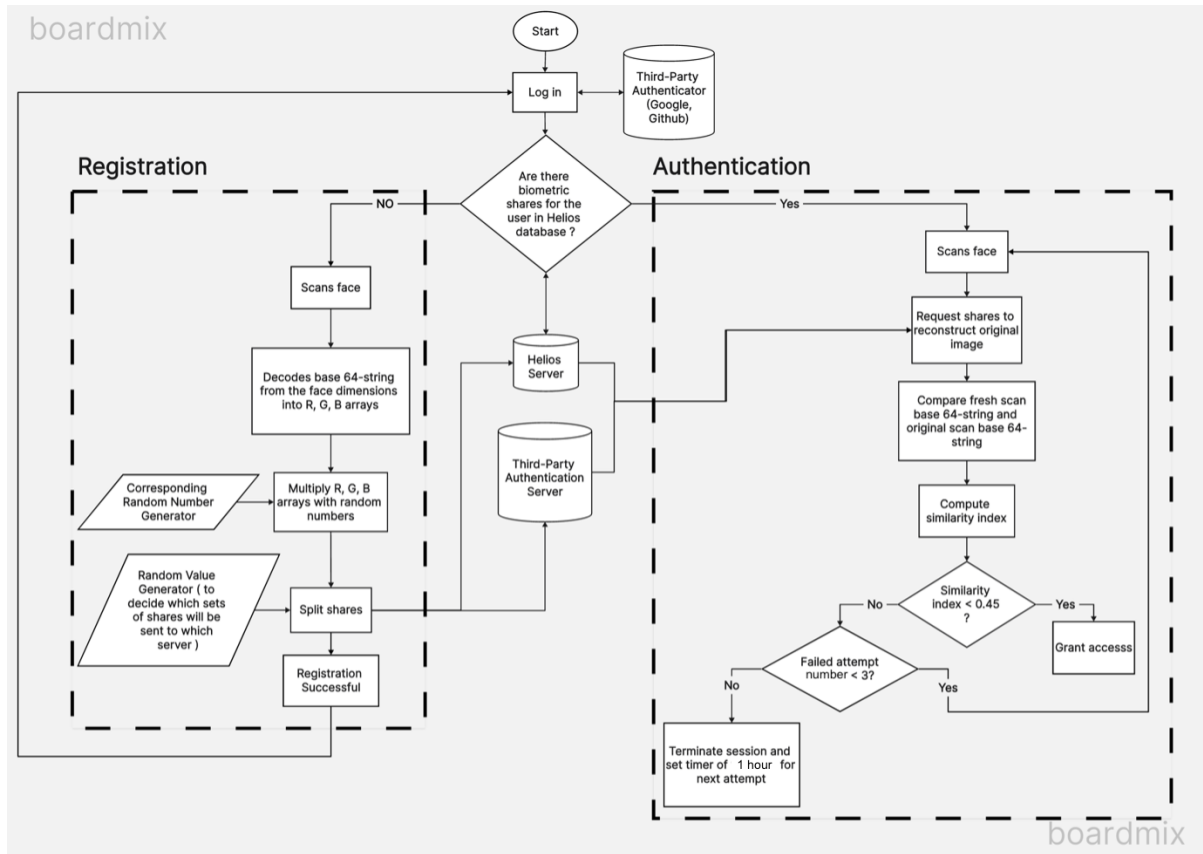




**Figure 1:** Enhanced Helios - High level system architecture.

and authentication processes for eligible voters remain the same but are now supplemented with biometric data. Although we acknowledge that the use of a proprietary solution may be considered a limitation, we have opted for this approach because of its ease of integration and the reliability it offers for biometric authentication. Specifically, the solution includes a mechanism that requires users to rotate their heads during authentication, effectively mitigating the risk of masquerading attacks, such as the use of photographs to impersonate the user. Moreover, the solution provides the option to use a self-hosted server, which presents a plausible compromise to securely handle sensitive biometric data, thus maintaining control over data storage and privacy. Given the modularity of our extensions, integrating open-source alternatives that provide the same level of reliability should be straightforward, allowing for greater flexibility and adaptability in future implementations.

Figure 2 shows the extended user registration and authentication workflow. Initially, during the registration phase, the user's face (whether admin, voter, or trustee) is captured and encoded into a base64 string. This string is divided into three arrays  $R$ ,  $G$  and  $B$ , representing the RGB color model [33]. Each array represents the pixel intensity values and has a size of  $1280 \times 720$  pixels. In addition, three random arrays of the same size are generated, denoted  $R\text{-rand}$ ,  $G\text{-rand}$ , and  $B\text{-rand}$ . Afterwards, each array is multiplied in element-wise order with its corresponding random array. For example, if the  $R$  array is  $[1, 7, 0, 2, 6]$ , and the  $R\text{-rand}$  array is  $[3, 6, 8, 2, 9]$ , their multiplication results in  $[3, 42, 0, 4, 54]$ . This process is repeated for the  $G$  and  $G\text{-rand}$ , as well as for the  $B$  and  $B\text{-rand}$  arrays. The Helios database server retains two sets of arrays: one containing two of the generated random arrays, and the other containing the multiplied arrays, denoted  $R\text{-mul}$ ,  $G\text{-mul}$ , and  $B\text{-mul}$ , that do not belong to any of the retained random arrays. The distribution and allocation mechanism that determines whether Helios retains the data  $R$ ,  $G$ , or  $B$  is determined by a random number generated ranging from 1 to 3. For example, if the random number is 2, the Helios database server receives  $R\text{-rand}$ ,  $B\text{-rand}$ , and the multiplied array  $G\text{-mul}$ . On the other hand, the external server receives the third random array, whose corresponding data is stored on the Helios database server, and the two remaining multiplied arrays  $R$ ,  $G$ ,  $B$ . Following the example above, the external server receives  $G\text{-rand}$  and the multiplied arrays  $B\text{-mul}$  and  $R\text{-mul}$ . Hence, the biometric data, specifically an image of the user's scanned face, is split into two sets of shares. One set is stored on a server hosted by a third-party, while the other is stored on the Helios database server, alongside the data from the original registration process. This ensures



**Figure 2:** Enhanced Helios - Biometric registration and authentication workflow.

that neither party has the complete information necessary to reconstruct any part of the original image, thus implementing a form of visual secret sharing [12]. It is important to emphasize that storing the biometric shares separately protects users' biometric data in the event of any accidental or malicious compromise of any of the two database servers.

Note that if the user is already registered, the third-party authentication step via Google or Github remains the same. This means that after successful authentication, at the corresponding third-party, the received OAuth token is compared with the information previously stored on the Helios database server. The difference arises when biometric authentication is used. Hence, in the authentication phase, after the user's face is scanned through the facial recognition API, the corresponding biometric data is passed to the Helios Authentication Service. The scanned data are then compared to the previously stored biometric data, which is requested and reconstructed from the shares stored in the separate databases.<sup>9</sup> This reconstruction process involves performing a division for each of the multiplied arrays  $R$ ,  $G$ ,  $B$  with their corresponding arrays  $R\text{-rand}$ ,  $G\text{-rand}$ , and  $B\text{-rand}$ . These shares are uploaded from both the Helios server and the external server. Afterwards, Helios compares the recently scanned user's face, thus the resulting base64 string, with the reconstructed base64 string for the respective user by computing a similarity index between the two strings. This similarity index is a floating-point number between 0 and 1. When this number is closer to 0, the higher the probability that the user logging in is genuine. If the similarity index is less than 0.45, access is granted. However, if the index exceeds 0.45, access is denied and the user must re-scan their face. After three failed attempts, the session is terminated and access to the account is denied for a period of 1 hour. It is important to note that the threshold of three failed authentication attempts followed by a 1 hour lockout has been set solely for the proof of concept (PoC), but can be adjusted to align with the specific needs or legal requirements of the election.

<sup>9</sup>Note that the biometric data is only fully accessible at the time of initial registration and during authentication attempts. If an attacker gains control of the systems at these points, the biometric privacy of the voter is compromised.

### 3.3. HelioScan - Automating individual verifiability

The HelioScan application is implemented and designed to offer voters a straightforward verification experience, focusing on simplicity and clarity. HelioScan uses the Flutter<sup>10</sup> framework, an open-source platform created by Google, and is built using the Dart<sup>11</sup> programming language. HelioScan can be used on the two most common mobile operating systems, namely iOS<sup>12</sup> and Android<sup>13</sup>.

Upon launching the app, voters are greeted with a welcome message and a concise guide on how to navigate the system for verification purposes, refer to figure 3 a). After familiarizing themselves with the instructions, voters can proceed by clicking the start button. The next screen presents, depicted in figure 3 b), voters with a list of elections they have participated in via the app. At this point, voters have two options: they can either select one of the listed elections to perform verification functionalities or update the registered ballot tracker, or scan a new QR code for a different election. Choosing to scan a new QR code leads the app to save the corresponding ballot tracker in the database and guide the voter to a specific welcome page for that election, outlined in figure 3 c).



**Figure 3:** HelioScan - Home, Participated Elections, and Election Welcome Page.

This is only done after verifying the uniqueness of the ballot tracker in the particular election by communicating with the third-party verification server. After its uniqueness is confirmed, the ballot tracker is also saved to the list of ballot trackers present in the third-party verification server database. Thus, if the voter's ballot tracker is confirmed to be unique, it indicates that the system is functioning correctly, and no clash attack is detected. Consequently, the server stores the ballot tracker in its database and sends a success message to HelioScan, allowing the app to save the ballot tracker in its mobile database. However, if a potential clash attack is detected, i.e., the ballot tracker is not unique, the server sends a failure message to HelioScan to alert the voter.

In the vote-casting phase, voters can verify the integrity of their ballot tracker between voting pages until successful casting. Following the instructions mentioned in the Helios interfaces, voters can scan the QR code of the ballot tracker each time it appears, from the first moment to successful casting. This process protects against manipulation of the voter's ballot tracker, as the ballot tracker is stored in the app once the voter scans it. Thus, any potential modification of the ballot tracker before casting would be detected. In addition, voters can verify the correct construction of the ballot tracker, i.e., *cast-as-intended* by scanning the QR code shown on the Helios *Spoil and Verify* page. While the current version of the app receives the verification result from Helios and therefore relies on its' computation, it is straightforward to migrate the computation process to the app itself or any existing verification application trusted by the voters, to ensure independence from the Helios server. This enables the

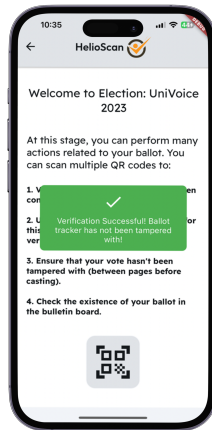
<sup>10</sup><https://flutter.dev>

<sup>11</sup><https://dart.dev>

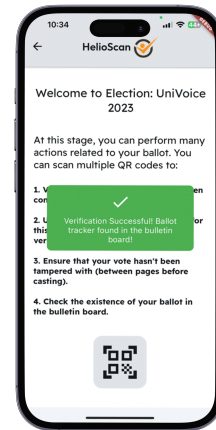
<sup>12</sup><https://www.apple.com/de/ios/ios-17/>

<sup>13</sup><https://www.android.com/>

automation of the *cast-as-intended* verification step and guarantees that the large cryptographic text displayed in Helios does indeed correspond to the voter's selection, as well as the corresponding ballot tracker matches with the one previously scanned by the voter, shown in figure 4 a).



(a) HelioScan - Verifying that the ballot tracker has not been modified.



(b) HelioScan - Verifying that the scanned ballot tracker exists.

**Figure 4:** HelioScan - Performing individual verifiability.

Next, voters can automatically perform the *stored-as-cast* verifiability step by verifying the existence of the scanned ballot tracker on the election bulletin board. Thereby, the voter navigates to the election bulletin board and scans the QR code displayed there, see figure 5. This QR code contains the ballot trackers for all cast ballots. Hence, this allows voters to automatically verify if their previously scanned ballot tracker exists, as shown in figure 4 b). While we acknowledge the inherent limitation of a single QR code in terms of the amount of data it can store, for the sake of simplicity in demonstrating our Proof of Concept (PoC) for automating the *stored-as-cast* verification process, we have opted to use a single QR code. We assume that this QR code has sufficient capacity to hold all the cast ballots. An alternative solution could involve using multiple QR codes, which would be time-sequenced, requiring the voter to scan only the QR code relevant to the specific time of vote casting. Further alternatives and considerations regarding the automation of the *stored-as-cast* verification process are discussed in section 4.



**Figure 5:** Enhanced Helios - Bulletin Board.

Finally, whenever voters revote or verify the correct construction of the ballot tracker, a new ballot tracker with a new QR code is generated. Voters simply need to scan this code again to update their record in the app, after which they can perform any of the stated functionality on the updated ballot tracker. Overall, with this approach, 3 separate QR codes are generated. The first QR code, which contains the ballot tracker, is displayed after selecting the preferred option and before casting or

verifying it. The second QR code is shown on the bulletin board above all the ballot trackers. This QR code contains all valid and invalid ballot trackers, which are retrieved from the Helios database server through a GET HTTP request. The third QR code is displayed on the *Spoil and Verify* page once a voter decides to verify (instead of casting) the ballot and is computed by getting/retrieving the verification result of the current ballot parameters.

## 4. Discussion

This section outlines the security assumptions of the enhancements and discusses alternative approaches to strengthen or implement them.

### 4.1. Security assumptions

As described in section 3 our extensions are implemented upon the underlying Helios architecture. Therefore, the core security assumptions inherent to Helios persist; for instance, the private keys of the Helios server are secure, potential vote manipulations are detected when voters verify, and trustees behave according to the protocol. Although our enhancements augment the existing security framework of Helios, such as by integrating biometric authentication, ensuring *private eligibility verifiability* and achieving *practical coercion-resistance* - the incorporation of novel components or parties introduces additional assumptions:

*Assumption 1 - Limited presence of a coercer:* We assume that coercers are not consistently present with voters throughout the duration of the election.

*Assumption 2 - Trusted third-party for biometric authentication:* We assume that any third-party engaged in the authentication process, responsible for requesting the secret shares of stored biometric data and reconstructing them for comparison with voters' current facial features, is trustworthy in safeguarding the reconstructed and scanned biometric data and accurately reporting the comparison results. Such a service could be a publicly administered platform, used by users in different contexts. Examples include government identity verification systems or social security services. Furthermore, it is important to highlight that the current Helios system relies on third-party authentication providers, such as Google or GitHub, which offer Multi-Factor Authentication (MFA), though it is not enforced within the Helios framework itself. In contrast, our solution not only enforces MFA but also offers the flexibility of using a self-hosted server for two-factor authentication, providing full control over the process - especially crucial when handling sensitive biometric data as the second authentication factor. Although both solutions still rely on third-party servers, our approach ensures these servers are independent of the Helios system and public service providers, safeguarding sensitive authentication information by preventing its storage on external servers, thus enhancing both security and privacy.

*Assumption 3 - Trusted verifiability application:* We assume that the HelioScan application, or any other application that offers similar functionality, is deemed reliable for accurately determining the uniqueness of the ballot tracker and for performing *individual verifiability*.

*Assumption 4 - Trusted third-party verification server:* We assume that any third-party involved in the *individual verification* process reliably provides the accurate data set of the cast ballots, which are stored on the Helios database server.

### 4.2. Alternative approaches

While our enhancements do not require or impose alterations to the underlying Helios architecture, there are alternative methods to implement or to strengthen these improvements.

In the context of integrating biometric authentication, an alternative approach involves entrusting a single entity with the responsibilities of facial recognition and biometric data storage. This alternative offers the primary advantage of streamlined integration into Helios, as it eliminates the need for additional implementation efforts while also enhancing the efficiency of the facial recognition process. However, this alternative introduces a single point of failure and lacks control over cryptographic

protocols, such as the implementation of secret-sharing schemes. Another approach involves implementing secure multiparty computation [34] in both the registration and authentication process. This ensures the privacy of biometric data, because the parties involved cannot learn anything more than the prescribed output. An alternative to ensure the privacy of biometric data is the implementation of biometric cryptosystems [35].

Regarding the third-party verification server, an alternative approach is to directly verify the data set of the cast ballots on the Helios database server, which functions as the bulletin board. This method improves the current implementation from both the security and performance perspectives. A similar approach can be applied to third-party verification applications, such as HelioScan. In this way, the computations to verify *cast-as-intended* and the uniqueness of the ballot tracker (*stored-as-cast*) can be migrated to the app itself or to any existing verification application trusted by voters. Although this approach improves security, it requires additional implementation or integration with other verification applications trusted by voters.

Finally, there are proposals that achieve coercion resistance in the context of internet voting, such as those by Juels et al. [36], Araújo et al. [37], and Locher et al. [38]. However, implementing these approaches in Helios would require substantial modifications to the underlying vote casting and *individual verifiability* processes. Furthermore, these proposals face several challenges regarding usability, security and trust, including unrealistic assumptions, lack of self-efficacy, limited interactive feedback, and acceptance issues [39].

## 5. Conclusion

Overall, this article contributes to the ongoing discourse on internet voting security and usability, providing practical solutions to enhance the integrity and accessibility of electoral processes. Specifically, this paper addresses important usable security shortcomings in Helios. By integrating biometric authentication into Helios and enhancing voter authentication, alongside dissociating voters' identities from their ballot trackers, we have fortified its security. Furthermore, the introduction of a QR code mechanism and the corresponding mobile application authenticator has streamlined the vote verification process, enhancing usability for voters. Although our enhancements represent significant progress in addressing the identified shortcomings, they are not without limitations. Hence, more research is essential to advance the field and ensure the trustworthiness, reliability, and acceptance of internet voting systems in democratic societies.

For future work, we plan to implement the verification computations in the mobile application and provide an independent verifiability app. Thus, we can eliminate the need for a third-party verification server, hence reducing assumptions, simplifying the system architecture, and improving performance. Furthermore, we plan to investigate, evaluate, and integrate existing solutions of secure-multiparty protocols or biometric cryptosystems to reduce trust assumptions regarding biometric authentication. Finally, we plan to conduct a user study to evaluate the effectiveness of security enhancements and mobile application interfaces, while focusing on individual verifiability. Our objective is to evaluate the alternatives for automating individual verifiability, focusing on their impact on user experience and verification effectiveness. Understanding how users interact with the system and perceive its security features is crucial to refine the user experience, increase trust, and ensure widespread acceptance.

## Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to verify grammar and spelling, paraphrase, and reword. After using this service, the authors reviewed and edited the content as needed and assume full responsibility for the content of the publication.

## References

- [1] B. Adida, Helios: Web-based Open-Audit Voting., in: USENIX security symposium, volume 17, 2008, pp. 335–348.
- [2] F. Karayumak, M. M. Olembo, M. Kauer, M. Volkamer, Usability analysis of helios—an open source verifiable remote electronic voting system, in: 2011 Electronic Voting Technology Workshop/-Workshop on Trustworthy Elections (EVT/WOTE 11), 2011.
- [3] O. Kulyk, J. Henzel, K. Renaud, M. Volkamer, Comparing “challenge-based” and “code-based” internet voting verification implementations, in: D. Lamas, F. Loizides, L. Nacke, H. Petrie, M. Winckler, P. Zaphiris (Eds.), Human-Computer Interaction – INTERACT 2019, Springer International Publishing, Cham, 2019, pp. 519–538.
- [4] S. Neumann, M. M. Olembo, K. Renaud, M. Volkamer, Helios verification: To alleviate, or to nominate: Is that the question, or shall we have both?, in: Electronic Government and the Information Systems Perspective: Third International Conference, EGOVIS 2014, Munich, Germany, September 1-3, 2014. Proceedings 3, Springer, 2014, pp. 246–260.
- [5] K. Marky, O. Kulyk, K. Renaud, M. Volkamer, What did i really vote for? on the usability of verifiable e-voting schemes, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1–13. doi:10.1145/3173574.3173750.
- [6] V. Cortier, D. Galindo, S. Glondou, M. Izabachene, Election verifiability for helios under weaker trust assumptions, in: Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19, Springer, 2014, pp. 327–344.
- [7] O. Kulyk, V. Teague, M. Volkamer, Extending helios towards private eligibility verifiability, in: E-Voting and Identity: 5th International Conference, VoteID 2015, Bern, Switzerland, September 2-4, 2015, Proceedings 5, Springer, 2015, pp. 57–73.
- [8] G. Tsoukalas, K. Papadimitriou, P. Louridas, From helios to zeus, USENIX Journal of Election Technology and Systems (JETS) 1 (2013) 1–17.
- [9] D. Gawel, M. Kosarzecki, P. L. Vora, H. Wu, F. Zagórski, Apollo—end-to-end verifiable internet voting with recovery from vote manipulation, in: Electronic Voting: First International Joint Conference, E-Vote-ID 2016, Bregenz, Austria, October 18-21, 2016, Proceedings 1, Springer, 2017, pp. 125–143.
- [10] C. Staff, Acm’s 2014 general election: please take this opportunity to vote, Commun. ACM 57 (2014) 9–17. doi:10.1145/2597769.
- [11] J. Bonneau, C. Herley, P. C. v. Oorschot, F. Stajano, The quest to replace passwords: A framework for comparative evaluation of web authentication schemes, in: 2012 IEEE Symposium on Security and Privacy, 2012, pp. 553–567. doi:10.1109/SP.2012.44.
- [12] M. Naor, A. Shamir, Visual cryptography, in: Advances in Cryptology—EUROCRYPT’94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings 13, Springer, 1995, pp. 1–12.
- [13] B. Smyth, D. Bernhard, Ballot secrecy and ballot independence coincide, in: Computer Security-ESORICS 2013: 18th European Symposium on Research in Computer Security, Egham, UK, September 9-13, 2013. Proceedings 18, Springer, 2013, pp. 463–480.
- [14] R. Kusters, T. Truderung, A. Vogt, Clash attacks on the verifiability of e-voting systems, in: 2012 IEEE Symposium on Security and Privacy, IEEE, 2012, pp. 395–409.
- [15] Federal Constitutional Court of Germany. Decisions: Order of 03 March 2009 - 2 BvC 3/07, [http://www.bundesverfassungsgericht.de/SharedDocs/Entscheidungen/EN/2009/03/cs20090303\\_2bvc000307en.html](http://www.bundesverfassungsgericht.de/SharedDocs/Entscheidungen/EN/2009/03/cs20090303_2bvc000307en.html), 2009.
- [16] V. Cortier, D. Galindo, R. Küsters, J. Müller, T. Truderung, Sok: Verifiability notions for e-voting protocols, in: 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 779–798.
- [17] O. Pereira, Internet voting with helios, Real-World Electronic Voting: Design, Analysis and Deployment 8604 (2016).



- [18] J. Benaloh, Simple verifiable elections., EVT 6 (2006) 5–5.
- [19] B. Adida, O. De Marneffe, O. Pereira, J.-J. Quisquater, et al., Electing a university president using open-audit voting: Analysis of real-world use of helios, EVT/WOTE 9 (2009).
- [20] P. Ryan, P. Rønne, V. Iovino, Selene: Voting with transparent verifiability and coercion-mitigation, IACR Cryptology ePrint Archive 2015 (2015) 1105.
- [21] D. Bernhard, O. Pereira, B. Warinschi, How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios, in: Advances in Cryptology–ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2–6, 2012. Proceedings 18, Springer, 2012, pp. 626–643.
- [22] A. Escala, S. Guasch, J. Herranz, P. Morillo, Universal cast-as-intended verifiability, in: International Conference on Financial Cryptography and Data Security, Springer, 2016, pp. 233–250.
- [23] S. Guasch, P. Morillo, How to challenge and cast your e-vote, in: International Conference on Financial Cryptography and Data Security, Springer, 2016, pp. 130–145.
- [24] F. Karayumak, M. Kauer, M. M. Olembo, T. Volk, M. Volkamer, User study of the improved helios voting system interfaces, in: 2011 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST), IEEE, 2011, pp. 37–44.
- [25] S. Srinivasan, C. Culnane, J. Heather, S. Schneider, Z. Xia, Countering ballot stuffing and incorporating eligibility verifiability in helios, in: Network and System Security: 8th International Conference, NSS 2014, Xi’an, China, October 15–17, 2014, Proceedings 8, Springer, 2014, pp. 335–348.
- [26] V. Cortier, G. Fuchsbauer, D. Galindo, Beleniosrf: A strongly receipt-free electronic voting scheme., IACR Cryptol. ePrint Arch. 2015 (2015) 629.
- [27] D. Bernhard, O. Kulyk, M. Volkamer, Security proofs for participation privacy and stronger verifiability for helios, CRISP-Center for Research in Security and Privacy, 2016.
- [28] D. Demirel, J. Van De Graaf, R. S. dos Santos Araújo, Improving helios with everlasting privacy towards the public., Evt/wote 12 (2012).
- [29] O. Kulyk, S. Neumann, M. Volkamer, C. Feier, T. Koster, Electronic voting with fully distributed trust and maximized flexibility regarding ballot design, in: 2014 6th International Conference on Electronic Voting: Verifying the Vote (EVOTE), IEEE, 2014, pp. 1–10.
- [30] O. Kulyk, K. Marky, S. Neumann, M. Volkamer, Introducing proxy voting to helios, in: 2016 11th International Conference on Availability, Reliability and Security (ARES), 2016, pp. 98–106. doi:10.1109/ARES.2016.38.
- [31] Y. Desmedt, P. Chaidos, Applying divertibility to blind ballot copying in the helios internet voting system, in: Computer Security–ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10–12, 2012. Proceedings 17, Springer, 2012, pp. 433–450.
- [32] S. Park, R. L. Rivest, Towards secure quadratic voting, Cryptology ePrint Archive, Paper 2016/400, 2016. <https://eprint.iacr.org/2016/400>.
- [33] T. Young, II. The Bakerian Lecture. On the theory of light and colours, Philosophical transactions of the Royal Society of London 92 (1802) 12–48. doi:10.1098/rstl.1802.0004.
- [34] Y. Lindell, Secure multiparty computation, Communications of the ACM 64 (2020) 86–96.
- [35] C. Rathgeb, A. Uhl, A survey on biometric cryptosystems and cancelable biometrics, EURASIP journal on information security 2011 (2011) 1–25.
- [36] A. Juels, D. Catalano, M. Jakobsson, Coercion-resistant electronic elections, in: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, 2005, pp. 61–70.
- [37] R. Araujo, S. Foulle, J. Traoré, A practical and secure coercion-resistant scheme for internet voting, in: Towards Trustworthy Elections: New Directions in Electronic Voting, Springer, 2010, pp. 330–342.
- [38] P. Locher, R. Haenni, R. E. Koenig, Coercion-resistant internet voting with everlasting privacy, in: Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, 2016, Revised Selected Papers 20, Springer, 2016, pp. 161–175.
- [39] O. Kulyk, S. Neumann, Human factors in coercion resistant internet voting—a review of existing solutions and open challenges, in: Sixth International Joint Conference on Electronic Voting (E-Vote-ID 2020), TalTech press, 2020.