E-voting system evaluation based on the Council of Europe recommendations: Helios Voting

Luis Panizo, Mila Gascó, David Y. Marcos del Blanco, José A. Hermida, Jordi Barrat and Héctor Aláiz

Abstract — Despite the claimed benefits of e-voting initiatives, wider adoption of e-voting mechanisms and implementation processes is slower than expected. Several technical, social, and cultural challenges hinder generability and applicability of e-voting. Amongst them, the evaluation and harmonization of e-voting systems, given different legal and statutory frameworks, is still an important challenge to overcome. Yet, only a few works have addressed this topic in the field.

This article aims to contribute to further understanding this unexplored topic by applying a practical evaluation framework to Helios Voting, one of the most widely used e-voting tools to date. Our framework, strongly based on the technical and security requirements issued by the Council of Europe in 2017, is a valuable source of information for election officials, researchers and voters to understand the strengths and weaknesses of Helios Voting and, as a result, to improve decision-making processes regarding the type and size of elections that can be securely handled by Helios Voting. The ultimate goal of our paper is to conceptually and practically support the gradual, secure and protocolized expansion of e-voting.

Index Terms — e-democracy, e-government, e-voting.

1 Introduction

INFORMATION and Communication Technologies (ICT) have had a huge impact in the day-to-day lives of billions of citizens in recent years. Back in the early 2000s, it was widely anticipated that ICT would also influence public elections and other democratic processes, as an integral part of what has been labeled *e-democracy*.

More than one decade after, that promise has not been realized yet. Some countries such as Estonia, Australia, Norway, Switzerland, Germany, and Canada have implemented i-voting systems for legally binding elections totaling more than 6 million votes cast. Among them, only Estonia has used i-voting in each general election and for the whole census. Nonetheless, security flaws have been reported, which might have jeopardized the elections' results [1].

Certainly, e-voting is a multifaceted discipline that needs to take into account a complex combination of technical and non-technical issues that often evolve around the topic of security:

- The need to fulfill simultaneously two antagonistic properties: integrity and privacy.
- The consideration of democracy's legitimacy as the main outcome of electoral processes, which results in the need of preventing any potential attacks and frauds that could be very difficult to revert once elections are over.
- The existence of a traditional voting system which is reasonably simple, intuitive, verifiable and functional.
- Voters' devices, of which 30-40% may be infected by malware [2].
- The network, particularly when it is related to a relevant record of attacks linked to the associated cryptographic protocols [3], [4].
- The e-voting system, which oftentimes carries vulnerabilities serious enough to put the elections in jeopardy [1], [5], [6], [7].

The attacks coming from foreign nations during, for example, the i-voting experiences in the US and Australia among others, together with the increasing alert globally speaking (i.e. [8]) show the importance of the link between tecnhnology, on one hand, and society and culture, on the other [9], [10], [11], [12]. It is also critical to assess security in electoral democratic processes taking the aforementioned link into account.

Despite the relevance of security in e-voting processes and the growing international experience [13] , the harmonization of e-voting systems, given different legal and stat-

- *Mila Gasco is with the Rockefeller College of Public Affairs & Policy and the Center for Technology in Government, University at Albany - SUNY, Albany, NY12222. E-mail: mgasco@ctg.albany.edu.*
- *Luis Panizo, David Y. Marcos del Blanco José A. Hermida and Héctor Aláiz. are with the Engineering School, University of Leon. Campus de Vegazana. Leon. 24193 Spain. E-mail: luis.panizo@unileon.es, dmarcb01@estudiantes.unileon.es, jahera@unileon.es and hector.moreton@unileon.es.*
- *Jordi Barrat is with the Public Law Department, University Rovira i Virgili, Av. Catalunya, 35 43002 Tarragona, Catalonia. E-mail: jordi.barrat@urv.cat*

utory frameworks, is still an important challenge to overcome. Yet, only a few works have addressed this topic. In 2015, the IEEE reactivated the 1622 Committee on Voting System Standards, but it only included a set of recommendations instead of specific requirements or proofs. In 2016, Neumann proposed a probabilistic framework for e-voting schemes [14]. Subsequently, Marcos et al. introduced a comprehensive methodology with technical and legal remarks as well as practical recommendations in their evaluation system [15] presented during the E-Vote-ID 2016 conference [16]. In particular, they considered the Council of Europe e-voting requirements [17], as well as a set of 41 technical and practical factors evaluated by 21 international experts in the e-voting field.

In this regard, the Council of Europe (CoE) updated in 2017 its Recommendation 2004(11) on legal, technical and operational standards for e-voting. Despite its non-binding effects, both the new document [Rec. 2017(5)] and the Guidelines that complete its content establish a legal framework that aims at serving as guidance for e-voting practicioners.

The 2017 version takes note of the evolution of the e-voting field over the last two decades. For instance, the Recommendation pays especial attention to remote voting (Internet voting), whose most important implementations took place after the approval of the first version. Accordingly, certification, verifiability and transparency issues are duly covered as concerns whose importance grew along with Internet voting deployments.

Past experience proves that both electoral authorities, e-voting companies and other players appreciate this normative compendium, but systematic and consistent research applying CoE's recommendations to e-voting real-word is still missing. That's why initiatives assessing the compliance with CoE's standards of given e-voting solutions will likely strengthen the crucial role of the Recommendation and pave the way for a real standardization / harmonizarion of e-voting platforms.

In this paper we aim at contributing to this unexplored topic by applying the aforementioned practical evaluation framework to Helios Voting [18], [19], a widely used e-voting tool developed by Harvard University researcher Ben Adida. Helios Voting is considered a benchmark for e-voting, having been used in more than 1,000 binding elections, mostly in academia/university environments, to cast over 100,000 votes. We hope that the thorough analysis and the final recommendations can become a relevant source of information for officials and researchers in order to establish a safe range of implementation for Helios Voting and ultimately contribute to a secure and protocolized expansion of e-voting sytems.

The remainder of this paper is organized as follows: section 2 introduces the most relevant related works and cryptographic foundations to this article. Section 3 includes a brief explanation of the evaluation methodology itself [15]. Its practical application to Helios Voting is detailed in section 4 and the results and limitations are exposed and analyzed in section 5. Finally, the conclusions and future works are explained in section 6. Additionally, *Appendix A* is devoted to a brief introduction of Helios Voting, with a special emphasys in its voting process and implemented cryptographic tools.

2 Related works and Cryptographic Foundations

2.1 Related Works

One of the most relevant research to date is that of Bräunlich, Grimm and Richter in 2013 [20], in which the authors presented the first interdisciplinary collaboration to transform legal requirements into technical criteria. In particular, the authors come up with thirty Technical Design Goals (TDG), built upon the KORA method (Konkretisierung Rechtlicher Inforderungen, Concretization of Legal Requirements) [21], and which had been previously used for other sectors such as mobile devices.

Building on the work by Bräunlich, Grimm and Richter, Neumann, from the Technische Universität Darmstadt in Germany, combined the previous methodology with the Common Criteria for IT-Security Evaluation [22] and defined sixteen technical requirements to link the legal criteria with Bräunlich's TDGs.

Although Neumann's research [14] contributes to building a valid framework, it still presents some limitations, namely:

On one hand, the security evaluation framework targets at schemes rather than complete systems. On the other, Neumann himself introduces an example of a structural flaw that would not be detected with his evaluation scheme: *"for instance, the Vote Forwarding Server and the Vote Storage Server of the Estonian Internet voting scheme are developed and maintained by the same vendor"* [14, p. 135].

In addition, the security evaluation is based on the assumption that voters will sufficiently utilize the verification tools provided by the system. Unfortunately, e-voting application has shown that voters tend to not verify: in one of the biggest electoral e-voting initiatives to date in New South Wales in 2015, only a 1.7% of 283.669 votes were verified [23]. Finally, Neumann's framework is based on probabilistic attack strategies (either an attacker is capable of causing certain impact or s/he is not), applying Monte-Carlo simulations [14]. It is an interesting approach indeed, although only at a scheme level and, therefore, less useful for a practical evaluation of the e-voting tool.

As a result, Neumann concludes *"we therefore recommend to incorporate the security evaluation framework into a larger decision-support system for elections officials"* [14, p. 138].

Building on Neumann's work, Marcos, et al. present a proposal of a decision-support system in the form of a practical evaluation framework [16]. It is compliant with the 2017 Council of Europe' guidelines ("Guidelines on the implementation of the provisions of Recommendation CM/Rec(2017)5 on standards for e-voting"), by the Directorate General of Democracy and Political Affairs [17] with regards to the five key principles of a democratic election (universal, free, equal, direct and secret) depicted in the same document. Section 3 presents a simplified version of the system.

2.2 Cryptographic Foundations

One of the key aspects to analyze an e-voting system is the underlying cryptographic primitives implemented. Depending on which ones are introduced, the e-voting tool could present vulnerabilities that might put the elections in jeopardy. Therefore, in order to facilitate the overall understanding of the technical aspects of the evaluation performed in section 4, we introduce a brief list of simplified cryptographic foundations or building blocks directly used by Helios Voting. They are all integral part of Helios' cryptographic implementation and therefore relevant for the system evaluation.

The subsequent definitions are based on a recent work by some of the most renowned experts in the cryptographic and e-voting fields such as Barbara Simons, Ron Rivest, Peter Ryan, Ben Adida (author of Helios Voting) or J. Alex Halderman among others[2]:

The first building block is the additively homomorphic (public key) encryption. The space of plaintexts is a group structure over a binary operation +. Then, given the ciphertexts ψ_1= *Enc(pk,* v_1*)* and ψ_2 =*Enc(pk,* v_2*)*, there is a ciphertext ψ which can be computed and corresponds to an encryption of *Enc(pk,* $v_1 + v_2$*)*. It is also required that if, at least one of ψ_1, ψ_2 is uniformly shaped over all ciphertexts, the output ψ follows the uniform distribution over all ciphertexts of $v_1 + v_2$.

It has two main direct applications in Helios:

1. Given k ciphertexts $\psi 1$,…, ψk encoding k votes v_1,…, $v_k \in \{0, 1\}$ it is possible to derive a single ciphertext ψ which encodes $T = \sum_{i=1}^{k} v_i$. In practice, T is used to derive the tally of an election if each plaintext vote v_i corresponds to the choice made by the *i*-th voter.
2. It is possible to refresh the randomness of ψ for ψ= *Enc(pk, v)* by processing ψ with *Enc(*p_k*, 0)*

The second building block is the zero knowledge proof, a protocol between two parties: the prover and the verifier with a language $L = \{x \mid \exists w : R(x,w)\}$, *R* being a polynomial-time predicate in a parameter *k* and *x* and *w* strings of length *k*. The ZKP protocol allows the prover to convince the verifier that she is in possession of a witness *w* about the fact that $x \in L$.

A variation of the ZKP is the non-interactive ZKP or NIZKP. In this protocol, the prover is able to produce a string π in one move, which convinces the verifier about the status of $x \in L$. Therefore, a NIZKP requires a public parameter *p* produced by an independent third party. In the case of Helios Voting, a NIZKP is used for the Sako-Kilian mixnet (see below) to proof that the shuffle of the ciphertexts containing the valid votes is performed correctly.

Next, given parameters *(p, q, g)* where *p* and *q* are large primes such that $q \mid p - 1$, g is a generator of the multiplicative group $\mathbb{Z}_q^*$ and a number *n* of trustees, ElGamal, the third building block, defines the following operations [24] used in Helios:

1. Distributed key generation: Each trustee $i \in n$ selects a private key share $x_i \in_R \mathbb{Z}_q^*$ and computes a public key share $h_i = g^{x_i} mod\, p$. The public key is: $h = h_1 \cdot \ldots \cdot h_n mod\, p$.
2. Encryption: Given a vote *v* and a public key *h*, select a random nonce $r \in_R \mathbb{Z}_q^*$ and derive the ciphertext $(a, b) = (g^r mod\, p, g^m \cdot\ h^r\ mod\, p)$.
3. Re-encryption: Given a ciphertext *(a, b)* of a vote and a public key *h,* select a random nonce $r' \in_R \mathbb{Z}_q^*$ and derive the re-encrypted ciphertext $(a', b') = (a \cdot g^{r'} mod\, p, b \cdot h^{r'}\ mod\, p)$.

Finally, in the Sako-Kilian mixnet, all inputs are ElGamal ciphertexts corresponding to valid Helios votes. A mix server takes *N* inputs, re-encrypts them using re-encryption factors $\{s_i\}_{i\, \in\, [1,N]}$ and permutes them according to the random permutation π_N, so that $d_i = Reenc(c_{\pi(i)}, s_i)$.

3 Evaluation Methodologies

The evaluation of e-voting systems against legal provisions has been an issue to which Bräunlich, Grimm and Richter [20] greatly contributed by presenting the first interdisciplinary collaboration refining election principles into technical design proposals.

Subsequently, Neumann [14] identified the shortcomings of Bräunlich's approach and developed a set of technical requirements to link legal criteria with Bräunlich's technical design goals.

While Neumann's work introduced an undeniable improvement, it was still a scheme evaluation tool based on probabilistic proofs and Monte-Carlo simulations rather than a practical framework to provide evaluation information for election officials.

Partially based on [14], Marcos et al. studied the shortcomings of Neumann's scheme against the Spanish Constitution [25] and the Council of Europe's Standards for Electoral Law and e-voting system certification [17], and proposed a comprehensive evaluation system [16], which included the following steps:

1. Definition of an homogeneous set of e-voting requirements based on: the KORA methodology [21], the CC and ISO 27001-IT Grundschutz guideline [22], their integration by Simic-Draws et al. [26], the Council of Europe Guidelines [15] [17] and Neumann's methodology [14].
2. Formal equivalence between point 1 (e-voting requirements) and Bräunlich [20].
3. Consultation with more than 30 international experts in e-voting (both from academia and industry) to review the methodology and add weighting factors.
4. Formal definition of the practical evaluation framework, including two sine-qua-non requirements and 41 evaluation items.

It is worth emphasizing the work by [16] aimed at linking for the first time the end to end verifiability (E2Ev) and coercion resistance (CR) to the legal requirements for any democratic election according to both the Spanish Constitution and the Council of Europe [27], [25]: *"the Parliament is, elected by universal, free, equal, direct and secret suffrage". Correspondingly, The Council of Europe* [27], *claims that: "The five key principles of electoral law are: universal, equal, free, direct and secret suffrage and they are at the root of democracy"* (article 68 of the Spanish Constitution [25]).

More precisely, Marcos et al. introduce the equivalence of the aforementioned five key principles into a formal verification of the end to end verifiability [28] + elegibility verifiability [29] for the universal, free, equal and direct properties and the coercion resistance [30] for the secret requirement.

The previous system, while sound from a legal point of view, presents similar limitations to Neumann's methodology [14] given the lack of coverage of many of the technical and practical aspects of a complete e-voting system.

As a result, a set of five requirements for e-voting systems was included partially based on the research by Popoveniuc, Benaloh, Rivest, Ryan and Volkamer [28], [31], [32], [33]. Finally, the requirements were codified, refined and itemized into 41 specific items by partially applying Zissis and Lekkas [34] and New Zealand's Department of Internal Affair's Report on e-voting [35] [1].

Once completely defined, the authors established the complete correspondence between Bräunlich's TDGs and [16], as presented in Figure 1 and Table 1:

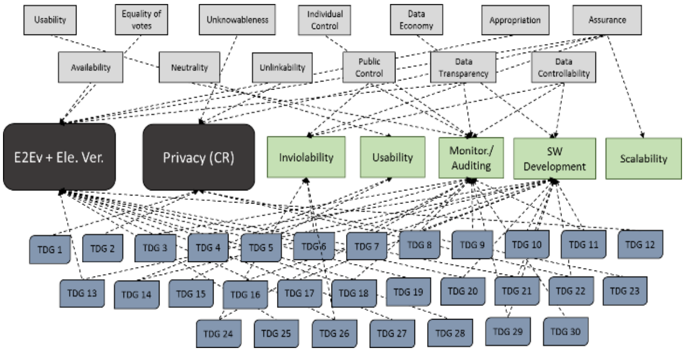

Fig. 1: Integration of Bräunlich and Marcos et al. 2016 schemes , having been previously used by [16].

TABLE 1
CORRESPONDANCE OF MARCOS ET AL. 2016 AND BRÄUNLICH SCHEMES

Requirement	Legal Criteria [19], [19]	*Technical Design Goals* [20],[14]
E2Ev + Ele. Ver.	Equality of votes, availability, appropriation, assurance, data transparency	TDG 5, TDG 12, TDG 19, TDG 20, TDG 21, TDG 25, TDG 26, TDG 27, TDG 28
Privacy (CR)	Unknowableness, assurance, unlinkability	TDG 1, TDG 2, TDG 22, TDG 23
Inviolability	Public control, data controllability, assurance	TDG 6, TDG 24, TDG 26
Usability	Usability	TDG 14, TDG 15
Monit./Audit.	Data controllability, individual control, public control, neutrality, data transparency	TDG 3, TDG 5, TDG 7, TDG 8, TDG 9, TDG 11, TDG 18, TDG 24, TDG 29, TDG 30
SW Development	Data economy, data controllability	TDG 4, TDG 8, TDG 10, TDG 11, TDG 13, TDG 14, TDG 16, TDG 17, TDG 20, TDG 21, TDG 22, TDG 29
Scalability	Assurance	

After the consultation with experts from Canada, France, Norway, Switzerland, Germany and Spain, the evaluation methodology was refined. In particular, the experts reviewed the requirements and assigned a weighting factor to each of them to fine-tune their specific importance within the practical framework. The weighting-factor allows for an intuitive and easy understanding of the evaluation methodology. Additionally, a comparative-legal approach has also been taken into account, in particular the work by Driza-Mauer and Barrat [36]. The feedback was included in the evaluation framework, as shown in Figure 2.

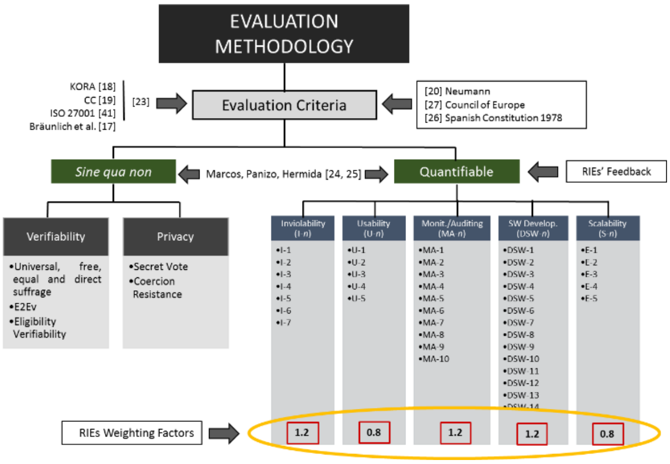

Fig. 2: Complete practical methodology for the evaluation of E-Voting Systems.

Figure 2 actually shows that there are two types of criteria and evaluation terms. First, the sine-qua-non one, for which end-to-end verifiability + elegibility verifiability and coercion resistance [29], [30], [37] represent the five mandatory principles of a democratic election (Spanish Constitution [25] and the Council of Europe [27]). Accoding to this criterium, evaluation is not in terms of a numerical value related to performance but rather as "holds" (○) or "does not hold" (×). There is a third possibility, when the property "holds under certain, plausible assumptions" (Δ).

The second criterium is a quantifiable requirement, which are evaluated from 0 to 10 with a one decimal accuracy. In or-

[1] For a complete explanation of the previous process, please refer to the original work in [15], [16].

der to calculate the numerical evaluation, each of the 41 specific, measurable items are reviewed with three possible veredicts: non-compliant (×), partially compliant (∆) and compliant (○) [2].

The final objective of this analysis is to offer a comprehensive, multi-faceted evaluation in one research article. The explanation of each of the evaluation items has been simplified due to space constraints.

4 Evaluation

In this section, we analyze the current version of Helios Voting, available at the official website [19]. More recent versions such as Belenios [38], Helios-C [39] or KTV-Helios [40] have not yet been fully deployed and therefore the practical evaluation framework cannot be entirely applied.

Helios Voting [18] is a free, open-source, web-based e-voting system. It has been used in several relevant binding elections, such as the ones held in the Catholic University of Louvain [41], the International Association of Cryptologic Research [42], and Princeton University [43]. Altogether, more than 100,000 votes have been cast with Helios Voting. It is widely considered the cornerstone in open-source e-voting and is one of the main references to develop new voting systems.

Currently, the latest version is fully available online, including a Github repository with the source code [19] as well as other technical documents and a FAQ section.

From a cryptographic standpoint, Helios exploits the additive homomorphic and distributed decryption properties of ElGamal [24] and the Sako-Killian mixnet protocol. It also uses the Chaum-Pedersen protocol [37], as a proof of decryption.

For a complete explanation of Helios Voting protocol, please refer to Appendix A.

4.1 End to end verifiability and elegibility verifiability

Currently, there does not exist a formal, universal definition for end-to-end verifiability (E2Ev). Although there are several tools for the symbolic analysis of security protocols such as ProVerif [44], AKISS [45] or APTE [46], accepting equivalence properties, they do not have a good application to e-voting systems because they all face the same unresolved challenge: associative and commutative operators are out of reach, making it impossible to analyze the following homomorphic property, as pointed out in [47]:

$$enc(pk; v_1) * enc(pk; v_2) = enc(pk; v_1 + v_2) \qquad (1)$$

As a consequence, the challenge of formally defining verifiability remains unresolved, which results in case by case analyses.

The most widely accepted definition of E2Ev is comprised of three properties [28]:

1. Cast as intended: voters can get convincing evidence that their encrypted votes accurately reflect their choices.
2. Recorded as cast: voters can check that their encrypted votes have been correctly included by finding exactly the encrypted value they cast on a public bulletin board.
3. Tallied as recorded: any member of the public can check that all the published encrypted votes are properly tallied, without knowing how individuals voted.

The first two properties are usually referred to as individual verifiability and the last one as universal verifiability.

For the cast as intented property, Helios introduces the cast-or-audit approach: the voter can audit her vote as many times as she wants, until she is convinced that Helios is trustable. Regarding recorded as cast, the voter receives a hash of her encrypted vote, which can later check on the bulletin board. Finally, for the tallied as recorded condition, ElGamal together with the Sako-Kilian mixnet are implemented. They also include a ZKP as previously explained.

Because of all of the aforementioned measures, Adida, the creator of Helios Voting, concluded that the system is end-to-end verifiable [18]. Certainly, in an ideal scenario, the system would be E2Ev if it both the bulletin board and the election authorities were acceoted as being honest. Otherwise, any of these two parties could irregularly add votes cast by invalid voters (“ballot stuffing”).

An interesting approach on verifiability in Helios, from a broader security standpoint had previously been introduced in [48], [49], [50].

Specifically, Volkamer was among the pioneers in addressing the security issue in [50]. More recently, Neumann, Noll and Volkamer herself presented a very interesting probabilistic approach on security applied to privacy [49], which is briefly introduced in the subsequent subsection 4.2 about coercion resistance, the most demanding type of privacy for e-voting systems.

With regards to security applied to verifiability in Helios, Bernhard et al. proved in [48] that the KTV-Helios scheme [40] provides verifiability if the register of eligible voters and the voting devices are trustworthy. For public binding elections, such assumptions are not to be taken carelessly.

In addition, Bernhard, Pereira, and Warinsch [51] and Adrian et al. [4] identified important flaws in the Fiat-Shamir heuristic, used in the Helios Voting scheme for the NIZKP. More specifically, Helios implements a version known as “weak Fiat-Shamir” which can the tallying procedure to run indefinitely and even tamper with the election result.

To address that vulnerability, Cortier et al. introduced a variant known as Helios-C (Helios with credentials) [39] which softens the premises to guarantee E2Ev, requesting honesty from either the bulletin board or the election authorities. Under these conditions, Helios is E2Ev and adds eligibility verifiability, needed to avoid “ballot stuffing”. Further, given the suitability of Helios Voting in student government bodies, local clubs, online groups, and other education-related organizations [18], it could be easily be accepted that there is a low risk of corrupted bulleting boards or election authorities. Similarly, addressing the flaws in the Fiat-Shamir

[2] For a complete explanation of the methodology, the detailed definition of the two sine-qua-non, the five quantifiable requirements and the 41 evaluation items, the reader can refer to the original work in [15], [16].

heuristic request a considerable investment in time and resources to be exploited. However, elections in local clubs or student government bodies are usually not challenged by high-skilled attackers.[3]

In sum, given the current use of Helios Voting, we can accept that bulleting boards and election officials are honest and, therefore, that Helios Voting is E2Ev.

Evaluation: Δ. E2Ev holds under certain, plausible assumptions. In such case, the universal, free, equal and direct properties (Council of Europe [27]) are met according to the definitions in Barrat [52], [53] .

4.2 Coercion resistance

There are three levels for privacy [30]: voter's privacy (the vote is not revealed to anyone), receipt-freeness (the voter cannot get a receipt or any proof that shows how she voted), and cercion resistance (CR) (a voter cannot cannot prove to a coercer that she has voted in a certain way, even if she is willing to). CR is therefore related to the highest degree of privacy.

According to Adida [18]: "with Helios, we do not try to solve the coercion problem" and therefore "privacy is ensured by recruiting enough trustees" (p. 1). Despite this statement, several authors have addressed ballot privacy attacks on Helios voting [37], [51], showing concern as the following examples shows.

We consider an election with $t_1 \dots t_l$ candidates and three eligible voters id_1, id_2, id_3. Let also id_3 be dishonest.

The bullentin board for the honest voters looks like:

$$id_1, ciph_1, spk_1, spk'_1 \tag{2}$$

$$id_2, ciph_2, spk_2, spk'_2 \tag{3}$$

In which, for $i \in \{1, 2\}$ *we have:*

$$ciph_i = (a_{i,1}, b_{i,1}), \dots, (a_{i,l}, b_{i,l}) \tag{4}$$

$$spk_i = (\bar{a}_{i,1}, \bar{b}_{i,1}, \bar{c}_{i,1}, \bar{s}_{i,1}, \bar{a}'_{i,1}, \bar{b}'_{i,1}, \bar{c}'_{i,1}, \bar{s}'_{i,1}), \dots (\bar{a}_{i,l}, \bar{b}_{i,l}, \bar{c}_{i,l}, \bar{s}_{i,l}, \bar{a}'_{i,l}, \bar{b}'_{i,l}, \bar{c}'_{i,l}, \bar{s}'_{i,l}) \tag{5}$$

$$spk'_i = (\bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{s}_i, \bar{a}'_i, \bar{b}'_i, \bar{c}'_i, \bar{s}'_i) \tag{6}$$

$ciph_i$ is the i-th encrypted ballot, spk_i shows that ciphertexts $(a_{i,1}, b_{i,1}), \dots, (a_{i,l}, b_{i,l})$ contain either 0 or 1. spk'_i demonstrates that $(a_{i,1} \cdot \dots \cdot a_{i,l}), (b_{i,1} \cdot \dots \cdot b_{i,l})$ contain 0 or 1.

The adversary selects a valid vote from the BBbullentin board: $id_k, ciph_k, spk_k, spk'_k$ being id_k the voter whose privacy will be compromised. The adversary submits $ciph_k, spk_k, spk'_k$ and the BB is composed as follows:

$$id_1, ciph_1, spk_1, spk'_1 \tag{7}$$

$$id_2, ciph_2, spk_2, spk'_2 \tag{8}$$

$$id_3, ciph_k, spk_k, spk'_k \tag{9}$$

For elections with a reduced number of voters, privacy could be compromised if one or several dishonest voters agreed on acting as id_3 in the previous example. Moreover, if the attacker wanted to cast an invalid vote, he could exploit malleability. Thus:

Given a valid vote v_1:

$$(a_1, b_1), \dots, (a_l, b_l), (\bar{a}_1, \bar{b}_1, \bar{c}_1, \bar{s}_1, \bar{a}'_1, \bar{b}'_1, \bar{c}'_1, \bar{s}'_1), \dots (\bar{a}_l, \bar{b}_l, \bar{c}_l, \bar{s}_l, \bar{a}'_l, \bar{b}'_l, \bar{c}'_l, \bar{s}'_l), (\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}') \tag{10}$$

Even a very simple attack implementing the following tainted vote would be accepted:

$$(a_1, b_1), \dots, (a_l, b_l), (\bar{a}_1, \bar{b}_1, \bar{c}_1, \bar{s}_1 + q, \bar{a}'_1, \bar{b}'_1, \bar{c}'_1, \bar{s}'_1 + q) \dots, (\bar{a}_l, \bar{b}_l, \bar{c}_l, \bar{s}_l + q, \bar{a}'_l, \bar{b}'_l, \bar{c}'_l, \bar{s}'_l + q), (\bar{a}, \bar{b}, \bar{c}, \bar{s}, \bar{a}', \bar{b}', \bar{c}', \bar{s}' + q) \tag{11}$$

Where q is added to the response components of the original ballot, which changes the ballot but not the vote.

In sum, Helios Voting was not originally conceptualized to address coercion resistance. Several attacks have actually shown that privacy is not guaranteed [37], [51]. Thus, Helios performance in this respect is not satisfactory and, in addition, does not comply with the recommendation number 23 of the Council of Europe on e-voting standards [17].

Nonetheless, there have been promising advances in the improvement of security related to privacy in its different degrees. Specifically, Neumann et al. introduced in [49] a proposal of security evaluation based on probabilistic attacks. While indeed a relevant step forward, it still relies on assigning subjective percentages to each attack possibility, which in practice is not an easy task. Additionally, Neumann's evaluation targets schemes rather than software systems.

Evaluation: X. Coercion resistance does not hold in Helios Voting.

4.3 Inviolability (I-*n*)

Helios allows identification through third parties (Facebook and Google) and, in doing so, it fails to comply with with I-1. Similarly, it does not include any tracking tools, offline backups, risk assessment protocols or threat modelling protocols (I-3, I-5).

In the case of I-2, there is a brief Attacks and Defenses section on the official website. It is useful. However, even considering the academic nature of Helios, it is not sufficient.

Regarding I-4, after the attack described in [54], distributed policies were implemented, although they have been proved vulnerable [47]. Finally, upon reviewing the source code, certain modularity principles are implemented (I-6). Consequently, I-2, I-4 and I-6 are partially compliant.

With regards to I-7, the open-source approach, together with Adida's eagerness to help scholars improve Helios [40] [38] [39], made the system compliant for this item.

TABLE 2
Inviolability in Helios Voting

I-n	Definition	Val.
I-1	Software and auxiliary system's protection through safe authentication protocols. Access through third-parties or vulnerable-servers not permitted.	X
I-2	Existence of action protocols in the event of compromised inviolability.	Δ
I-3	Tracking tools and offline backup copies available.	X
I-4	Distributed control in the critical nodes with division of responsibilities to minimize collusion risks.	Δ

[3] Unfortunately, the last official Helios version [51] is previous to [38], and therefore we could not take Corti-er's improvements into consideration.

I-5	**Existence of *Risk Assessment* and *Threat Modelling* protocols.**	X
I-6	Implementation of modularity principles to confine potential attacks and coding bugs.	Δ
I-7	Proper updating of items I-1…I-6	○

Evaluation: 6/10 points. Adida leaves no room for misunderstandings. Helios is a voting system suitable for minor elections in low-coercion, low-risk environments. Under these circumstances, Helios shows a fair level of inviolability overall.

4.4 Usability (U-*n*)

According to [55], [56], usability (U-1, U-3, U-4, and U-5) of Helios Voting could be improved. In their works, the authors show that 1) the terminology is a bit misleading for non-experts (i.e.: fingerprint, encrypt, check credentials etc.), 2) the voting/audit process is confusing for half of the voters, leading to a 38% of them not being able to successfully complete it, 3) the help button only opens an e-mail form, which does not help in practice, and 4) election officials need to store ElGamal's secret key, which might become a challenge for inexperienced users.

On the positive side, the vast majority of the voters (over 85%) felt very comfortable using Helios, positively assessing convenience over usability.

Overall, performance in terms of usability is below average, although current challenges may be addressed by paying closer attention to detail instead of by making big investments/changes.

TABLE 3

USABILITY IN HELIOS VOTING

U-*n*	Definition	Val
U-1	Simplicity in the authentication, voting and verification	Δ
U-2	Special attention to vulnerable groups pursuant to the Council of Europe and the United Nations' resolutions on the matter.	X
U-3	Transparency and clarity communicating the voter that the voting process has successfully ended and the vote has been received.	Δ
U-4	Privacy and integrity preference over usability in a compromise.	Δ
U-5	Intuitive and user-friendly administration interface for setting up and managing elections.	Δ

Evaluation: 4/10 points.

4.5 Monitoring/Auditing (MA-*n*)

In this article, [18] refers to audit, that is, to the fact that anyone can verify the validity of the votes, their inclusion in the bulletin board and the whole voting process until the final ballot counting. Unfortunately, there is no specific monitoring/auditing policy available in Helios, thus many of the items are not applicable, despite the Council of Europe [17]'s recommendation no. 39 directly referring to the need of a proper audit system.

MA-1, MA-4, MA-5 and MA-8 are labeled as partially compliant because, although they are not available in the standard version [57], there is a precedent of an Helios implementation where they were partially fulfilled [41] (the election of the President of the University of Louvain in 2009).

For that particular occasion, Helios' creator himself took part in the organization and deployment of the system, international experts were invited as observers, the organizers hired an external company to develop an audit program in Python, there was a service desk available prior and during the elections to address any issues, and one full day was reserved to audit the bulletin board and put comments/complains forward.

Interestingly enough, these circumstances are not reproduced in Helios' standard version [18] [4].

In sum, Helios Voting's audit policy is based on facilitating the auditing of both the individual and universal verifiability rather than on implementing a solid and independent protocol. In addition, Helios was originally conceptualized for small-scale, low-risk contexts. In such cases, although insufficient, the monitoring/auditing policy could be considered acceptable.

TABLE 4

MONITORING/AUDITING IN HELIOS VOTING

MA-*n*	Definition	Val
MA-1	External, independent and distributed.	Δ
MA-2	Existence of a MA protocol from the design phase, to assure a correct development throughout the whole lifecycle of the project.	NA
MA-3	Specific control over the *Risk Assessment* and *Thread Modelling* strategies.	NA
MA-4	Generation of periodical, tamper-proof, indelible logs; stored offline in isolated premises guarded by different personnel from other critical nodes with low collusion risk.	Δ
MA-5	Practical implementation development from census collecting to post-electoral maintenance.	Δ
MA-6	Well-documented, detailed information in the appropriate format.	Δ
MA-7	Existence of a test bench continuously running to verify that the system is working correctly.	NA
MA-8	The members of the monitoring/auditing team must be independent from the rest of authorities/administrators involved.	Δ
MA-9	Existence of an auditing protocol for previous attacks and events and for the MA protocol itself.	NA
MA-10	In the event of a successful attack, the system will give total priority to the vote/voter's privacy, even at the cost of calling off the elections.	NA

Evaluation: 4/10 points considering the original scope of Helios Voting.

4.6 Software Development (SWD-*n*)

The open-source and academic attributes of Helios Voting [38], [39], [40] may be considered an important benchmark.

Since 2008, researchers in cryptography, cybersecurity, and e-democracy have paid particularly attention to Helios Voting. As a consequence, the current software has been thoroughly reviewed by [37], [40], [51], among other, and several improvements have been proposed and implemented [39],

[4] It is worth noting that the elections in Louvain gathered more than 23,000 potential voters, directly managing over 8,000 votes.

[40] , [51].

In this respect, items SW-1, SW-4, SW-5, SW-7, SW-8, SW11, SW-12 and SW-13 are compliant.

Software testing in platforms, operational systems and browsers with a market share ≥ 1% (SWD-9) still show weaknesses as does access through third-party programs (SWD-10) and an appropriate update policy (SWD-14) (the last update took place before 2014).

Finally, the distributed approach (SW-2), usability (SW-3) and the receipt-freeness are elements that have only been developed to some extent and therefore can only be considered partly compliant.

Overall, Helios presents a satisfactory degree of software development, considering its academic origins and the limited resources. Yet, there is room for improvement in areas such as usability, distributed approach, access through third-parties and updating.

TABLE 5
SOFTWARE DEVELOPMENT IN HELIOS VOTING

SWD-*n*	Definition	Val
SWD-1	Usual software engineering requirements in terms of design, implementation and documentation.	○
SWD-2	Distributed approach, with a special focus on critical operations. No authority should have enough attributions to single-handedly modify critical parameters.	Δ
SWD-3	User-friendly approach. User's guide and administrator's guide well documented and available well in advance.	Δ
SWD-4	Existence of a secure and accessible website, with a well-documented FAQ section.	○
SWD-5	The voting options must be presented in a totally objective and unbiased way, showing no preference whatsoever.	○
SWD-6	The system must not provide the voter with enough evidence to proof her vote.	Δ
SWD-7	The system must guarantee the voter's privacy throughout the whole voting process, not being possible to rebuild the vote/voter link.	○
SWD-8	The voting process must offer the possibility to be terminated at any time, with the system not saving any information that could compromise the voter's privacy.	○
SWD-9	The SW must be tested in every platform, operational system and browser with a market share ≥ 1%.	X
SWD-10	The software must neither allow for third-party access (including social media) nor include links to programs or sites managed by servers outside the e-voting system infrastructure.	X
SWD-11	The cryptographic primitives shall be tested in advance under conditions more demanding than the ones expected during the elections in order to avoid breakdowns and foresee shortages.	○
SWD-12	Access to the source code by independent experts/researchers to look for potential bugs and reinforce security. The code developer can demand an NDA to protect its IP.	○
SWD-13	Implementation, to the extent possible, of protocolized systems and open standards to improve interoperability.	○
SWD-14	Existence of an update policy, especially against new e-voting attacks as they are discovered.	X

Evaluation: 7.5/10 points

4.7 Scalability (S-*n*)

Helios's origin and academic nature have great impact on the available resources for its development and updating, as we have just shown. They condition the ability to undertake capacity and performance tests, and attack simulations (S-1, S-2, S-3).

Regarding S-4 and S-5, the official website and documentation do not offer specific figures and/or metrics on Helios's maximum capacity. However, we can learn from previous real implementations. On one hand, the experience of IACR [42] and Princeton University [43], which used the standard version of Helios, shows that only limited human resources were needed to physically organize and control the polls, resulting in over 1,000 ballots successfully managed. On the other hand, the President election at the University of Louvain in 2009 [41] showcased two rounds of 3,000 ballots each, for a total census of around 25,000 voters. This implementation required ad-hoc infrastructure, implementation and resources.

As a result, it can be said that the maximum proved scope for Helios, without any special arrangements, is about 1,000 votes, which is a non trivial scalability limitation even considering the original scope of Helios.

Regarding the election typology, Helios' absence of available performance tests and metrics makes it only suitable for small-scale, not politically binding and low-risk elections. Its capacities and scalability do not allow the official standard version of Helios Voting to handle more complex procedures [53].

TABLE 6

SCALABILITY IN HELIOS VOTING

S-*n*	Definition	Val.
S-1	Maximum capacity tests both from a SW and a HW standpoint in environments more demanding than the actual elections to be managed.	X
S-2	Existence of specific performance tests for the most critical operations (authentication, encryption, decryption, cryptographic primitives, tallying etc.).	X
S-3	Existence of test benches more demanding than the actual elections.	X
S-4	Existence of clear indicators and metrics on the maximum size and complexity of elections which can be handled from both a SW (mathematic/cryptographic capabilities, number of voters) and an ex_SW (infrastructure, costs, logistics, second channels, HHRR etc) standpoint.	Δ
S-5	Typology of elections, which can be adequately handled by the *e*-voting system (from consultative referenda to political binding elections).	Δ

Evaluation: 4/10 points.

5 FINAL RESULTS AND DISCUSSION

The evaluation above shows that, first, under the sine-qua-non criteria, (Council of Europe [27] , Marcos et al. [16]), Helios can be considered E2Ev provided that both the bullentin board and the authorities are honest and that a potential attacker does not have enough resources to compromise the Fiat Shamir heuristic [6]. Such assumptions can only be made regarding minor elections in low-risk, low-coercion environments. Any other conditions would dangerously increase the risk of attracting attackers with enough resources to compromise the cryptographic security or entice the authorities to behave dishonestly.

In addition, coercion resistance does not hold because because the system does attempt to address the problem, as originally noted in [18], and because ballot stuffing and malleability-based attacks have been deemed feasible [37], [51]. The latter is not compliant with recommendation no. 23 of the 2017 Council of Europe guidelines for e-voting [17].

As previously explained, an e-voting system has to comply with both the E2Ev and the coercion resistance requirements in order to meet the five key principles of a legally binding, public democratic election according to the Council of Europe [27]. Hence, Helios Voting is not recommended to be used in such type of elections.

Second, the evaluation has thrown the following results: 9 compliances ○, 17 partial compliances Δ, 10 no compliances X, and 5 not-applicable.

These results show that:

- Helios Voting does not comply with the two sine-qua-non properties for e-voting systems and therefore its deployment is not recommended for legally-binding, public political elections.
- Inviolability could at the moment be enough for minor elections in low-coercion, low-risk contexts. Under such circumstances, no attacks have been reported. However, potential access through third party programs together with a lack of backup copies and risk assessment protocols advise against deploying Helios Voting in more resource-demanding elections.
- There is still room for improvement in usability and, therefore, in the simplicity and clarity of the interface as well as in the development of a version adapted to vulnerable groups according to the Council of Europe and the United Nations' resolutions on the matter [17] although research shows that more than 85% of voters felt comfortable with the interface [55], [56].
- Helios is not totally concerned about monitoring and auditing, despite the experience in some elections For example, in the election of the President of the University of Louvain in 2009, additional resources were allocated and the amount of votes managed was over 8,000 [41]. Specifically, there was an independent team of external auditors and there were periodical, tamper-proof logs generated and subsequently analyzed. In this smaller election processes, the current monitoring and auditing practices of Helios can be considered sufficient.
- With regards to software development, Helios Voting performance is solid, based on its open-source nature and a set of thorough reviews [37], [40]. Despite such efforts, additional software testing, access through third parties and a better update policy are some areas that deserve further improvement.
- Lastly, scalability has not been formally planned or tested. Empirically, Helios has a proven range of around 1,000 votes [42] in its standard version and around 8,000 ballots out of a 25,000 census in a reinforced version [41]. Any deployment surpassing those figures is highly discouraged, since it has not been tested before.

Finally, the evaluation system with the weighting factors (proportionally rounded up to 10) throws a numerical evaluation in Table 7 and Figure 3 [16]:

$$\sum_{i=1}^{n}\frac{f_1\cdot w_1+\cdots+f_n\cdot w_n}{n}\cdot\frac{n}{t}=\sum_{i=1}^{n}\frac{f_1\cdot w_1+\cdots+f_n\cdot w_n}{t} \qquad (12)$$

TABLE 7

PRACTICAL EVALUATION FRAMEWORK APPLIED TO HELIOS VOTING

Requirement	Weighting	Helios Voting
E2Ev	N.A.	Δ
CR	N.A.	**X**
Inviolability	**2.32**	6 * 2.32 = **13.92**
Usability	**1.53**	4 * 1.53 = **6.12**
Monitoring/Auditing	**2.3**	4 * 2.3 = **9.2**
SW Develop.	**2.32**	7.5 * 2.32 = **17.4**
Scalability	**1.53**	4 * 1.53 = **6.12**
TOTAL	**10**	**52.76**

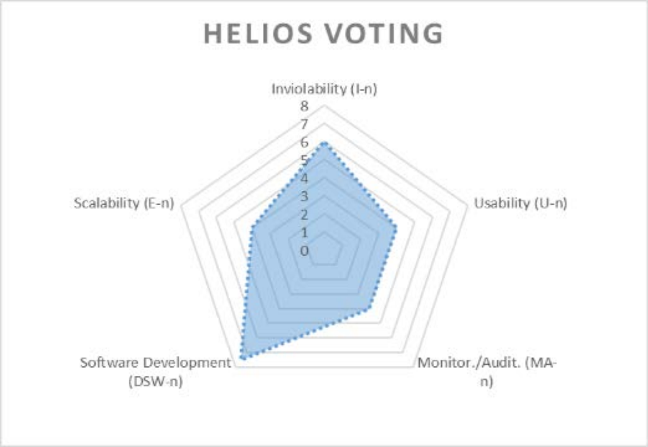

Fig. 3: Radial Analysis of Helios Voting

6 Conclusion

The interplay of social, cultural and technological developments in the success of electronic voting processes is key. In this article, we have focused on the technical aspects of e-democracy and e-voting and, in particular, in election security. It has been having an increasing role in national security, as proved by the technological-related security issues that countries such as the US, Russia, Ukraine and Australia have been recently facing.

Despite the huge difference that ICT and e-voting could contribute to make in binding elections, there is still an important lack of harmonization for e-voting requirements and evaluation methodologies.

In this article, we have aimed at applying, from a practical perspective (i.e., using one of the most relevant e-voting tools to date, Helios Voting), a novel evaluation scheme based on [16], with an emphasis on the most recent legal requirements and certification of e-voting systems by the Council of Europe [17], [27].

The ultimate goal was to provide a technical and protocolized source of information and to, therefore, contribute to the generalization of e-voting in a protocolized and safe way and to provide valuable and efficient evaluation methods for election officials.

We conclude that Helios Voting is a very useful tool in two ways. First, as a fully-operative, open source and "auditable" e-voting system, Helios Voting is a valid, almost free of charge option for minor elections in low-risk and low-coercion environments such as universities, local clubs, unions, and professional associations. Second, it is a good starting platform for e-voting researchers. In fact, there are several examples of prominent e-voting protocols based on Helios Voting in different stages of development such as Zeus [58], Belenios [38], Helios-C [39] or KTV-Helios [40].

In addition, our results have important practical implications for election officials, considering to use Helios Voting:

On one hand, development should be gradual and preceded by the required legal changes, as included in recommendations no. 27 and 28 by the Council of Europe on standards for e-voting [17]. On the other, it is wise to create an indepent and competent body to evaluate the compliance of the i-voting systems and constantly screen its performance. The implantation of e-voting solutions needs to be a decision built over firm, technical criteria and analysis together with legitimate political reasons. In that regard, the Norwegian case is a very interesting example on how to proceed [2]. It also embodies recommendation no. 37 by the Council of Europe [17].

Nevertheless, the use of Helios Voting is not always advisable given its limitations in relation to end-to-end verifiability and coercion resistance, which do not make it suitable in, for example, legally-binding public elections.

The main limitation of the current article is the fact that the methodology is applied to only one e-voting system, although a very relevant one. Conducting a comparative study could produce a more complete set of data and information, which could be used to further improve the evaluation methodology. Additionally, a bigger sample of international experts taking part in the development of the evaluation framework could have contributed to a reinforced fine-tuning. For the current version, 31 were contacted. Both limitations are currently being addressed for upcoming articles.

Future research in the field could further address some of the additional practical issues that have arisen in legally-binding deployments in Norway, Estonia, Canada, Spain, Australia, the USA, and Switzerland (e.g. cost, maintenance, non-software related requirements, surveillance, and attackers/intruders protocols), as well as their inclusion in [16]. Also, the application of the practical evaluation framework to other relevant e-voting systems (such as Scytl, Estonia, and *n*-votes) could provide supplementary useful information for researchers and elections officials on how to introduce e-voting solutions in legally-binding elections in a more secure, technically-sound and legally-compliant way within more demanding environments.

From a more technical perspective, the evaluation framework could be further developed to include other elements, such as attacks/incidence protocols, system maintenance, cost, and ex_software features (access control, logistic chain, auditing and backup protocols, etc.). This new evaluation framework could be made available though the development of a software tool.

Finally, further research could also focus on specific improvement to Helios Votinng given the deficiencies identified in the evaluation conducted.

References

[1] D. Springall *et al.*, "Security Analysis of the Estonian Internet Voting System," *Proc. 21st ACM Conf. Comput. Commun. Secur.*, no. May, p. 12, 2014.

[2] U. S. V. Foundation, "The Future of Voting," *The Future of Voting*, 2015. [Online]. Available: https://www.usvotefoundation.org/e2e-viv/summary.

[3] "The FREAK Attack," *The FREAK Attack*, 2015. [Online]. Available: https://censys.io/blog/freak.

[4] D. Adrian *et al.*, "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice," in *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 5–17.

[5] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions," *Adv. Cryptol. – EUROCRYPT 2005*, pp. 19–35, 2005.

[6] S. Goldwasser and Y. Tauman, "On the (In)security of the Fiat-Shamir Paradigm," *Focs*, 2003.

[7] D. Achenbach, C. Kempka, C. Kempka, B. Löwe, and J. Mul̈ler-Quade, "Improved Coercion-Resistant Electronic Elections through Deniable Re-Voting," *USENIX J. Elect. Technol. Syst.*, vol. 3, no. 2, pp. 26–45, 2015.

[8] "Russian Atack Report," 2017.

[9] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman, "Attacking the Washington , D . C . Internet Voting System," *System*, pp. 1–18, 2012.

[10] C. M, "Ukraine election narrowly avoided 'wanton destruction' from hackers," *Christian Science Monitor*, 2014. [Online]. Available: http://www.csmonitor.com/World/Passcode/2014/0617/Ukraine-election-narrowly-avoided-wanton-destruction-from-hackers-video.

[11] J. A. Halderman and V. Teague, "The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election," in *E-Voting and Identity: 5th International Conference, VoteID 2015, Bern, Switzerland, September 2-4, 2015, Proceedings*, R. Haenni, R. E. Koenig, and D. Wikström, Eds. Cham: Springer International Publishing, 2015, pp. 35–53.

[12] E. Nakashima, "Arizona: Russian hackers targeted Arizona election system," *The Washington Post*, 2016.

[13] I. VSSC/1622, "IEEE VSSC/1622: Common Data Format for Election Equipment," 2015. [Online]. Available: http://grouper.ieee.org/groups/1622/.

[14] S. R. Neumann, "Evaluation and Improvement of Internet Voting Schemes Based on Legally-Founded Security Requirements," Technische Universität Darmstadt.

[15] D. Y. Marcos, L. Panizo, and J. A. Hermida, "Development of a Holistic Methodology for the Evaluation of Remote Electronic Voting System," *Int. J. Complex Syst. Sci.*, vol. 6, no. 1, pp. 1–21, 2016.

[16] D. Marcos del Blanco, L. Panizo Alonso, and J. A. Hermida Alonso, "The need for Harmonization in the online voting field: Towards an European Standard for edemocracy," 2016, pp. 339–340.

[17] T. Standards, "Guidelines on the implementation of the provisions of Recommendation CM/Rec(2017)5 on standards for e-voting," *1289 th Meet. , 14 June 2017 2 . 3 Ad hoc Comm. Expert. Leg. , Oper. Tech. Stand. e- voting (CAHVE)*, no. June, pp. 1–19, 2017.

[18] B. Adida, "Helios: Web-based Open-audit Voting," in *Proceedings of the 17th Conference on Security Symposium*, 2008, pp. 335–348.

[19] "Helios Voting web." [Online]. Available: https://vote.heliosvoting.org/.

[20] R. A. Bräunlich K., Grimm R., Richter P., "Sichere Internetwahlen Ein rechtswissenschaftlich-informatisches Modell." .

[21] R. A. Hammer V., Pordesch U., *KORA (Konkretisierung Rechtlicher Inforderungen, Concretization of Legal Requirements). Betriebliche Telefon und ISDN-Anlagen rechtsgemäss gestaltet*. 1993.

[22] The Common Criteria Recognition Agreement, "Common Criteria for Information Technology Security Evaluation Part 1 : Introduction and general model July 2009 Revision 3 Final Foreword," *Nist*, vol. 49, no. July, p. 93, 2009.

[23] "Electoral Commision New South Gales," 2017. [Online]. Available: http://www.elections.nsw.gov.au/voting/ivote.

[24] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," in *Advances in Cryptology: Proceedings of CRYPTO 84*, G. R. Blakley and D. Chaum, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 10–18.

[25] "Constitución Española-Boletin oficial del Estado." pp. 101931–101941, 2011.

[26] D. Simić-Draws *et al.*, "Holistic and Law Compatible IT Security Evaluation: Integration of Common Criteria, ISO 27001/IT-Grundschutz and KORA," *Int. J. Inf. Secur. Priv.*, vol. 7, no. 3, pp. 16–35, 2013.

[27] C. of E. V. Commission, *European Stand-ards of Electoral Law in Contemporary Constitutionalism. Council of Europe Publishing*. 2005.

[28] J. D. C. Benaloh, R. Rivest, P. Y. A. Ryan, P. Stark, V. Teague, and P. Vora, "End-to-end verifiability," *arXiv e-prints*, 2014.

[29] S. Kremer, M. Ryan, and B. Smyth, "Election verifiability in electronic voting protocols," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6345 LNCS. pp. 389–404, 2010.

[30] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6000 LNCS. pp. 37–63, 2010.

[31] S. Popoveniuc, J. Kelsey, A. Regenscheid, and P. Vora, "Performance requirements for end-to-end verifiable elections," *… on Trustworthy Elections*. 2010.

[32] D. Bernhard, S. Neumann, and M. Volkamer, "Towards a Practical Cryptographic Voting Scheme Based on Malleable Proofs," in *E-Voting and Identify: 4th International Conference, Vote-ID 2013, Guildford, UK, July 17-19, 2013. Proceedings*, J. Heather, S. Schneider, and V. Teague, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 176–192.

[33] M. Volkamer and D. Hutter, "From Legal Principles to an Internet Voting System," *Proc. 1st Conf. Electron. Voting*, pp. 111–120, 2004.

[34] D. Zissis and D. Lekkas, "Design, Development, and Use of Secure Electronic Voting Systems." IGI Global, Hershey, PA, USA, pp. 1–270, 2014.

[35] T. T. Taiwhenua, "The Department of Internal Affairs - Online voting." [Online]. Available: https://www.dia.govt.nz/online-voting.

[36] A. D. Maurer and J. Barrat, *E-Voting case law: a comparative analysis*. Routledge, 2016.

[37] V. Cortier and B. Smyth, "Attacking and fixing Helios: An analysis of ballot secrecy," *J. Comput. Secur.*, vol. 21, no. 1, pp. 89–148, 2013.

[38] S. Glondu, "Belenios specification." pp. 1–8, 2013.

[39] V. Cortier, D. Galindo, S. Glondu, and M. Izabachène, "Election verifiability for helios under weaker trust assumptions," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8713 LNCS,

no. PART 2, pp. 327–344, 2014.

[40] O. Kulyk, V. Teague, and 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*, R. Haenni, R. E. Koenig, and D. Wikström, Eds. Cham: Springer International Publishing, 2015, pp. 57–73.

[41] B. Adida, O. De Marneffe, O. Pereira, and J.-J. Quisquater, "Electing a university president using open-audit voting: analysis of real-world use of Helios," *… Electron. voting …*, no. i, pp. 1–15, 2009.

[42] Stuart Haber, J. Benaloh, and S. Halevi, "The Helios e-Voting Demo for the IACR," *Helios*, pp. 1–7, 2010.

[43] "Welcome to Princeton Undergraduate Elections," *Welcome to Princeton Undergraduate Elections*, 2017. [Online]. Available: https://princeton.heliosvoting.org/.

[44] B. Blanchet, "An Automatic Security Protocol Verifier based on Resolution Theorem Proving (invited tutorial)," in *20th International Conference on Automated Deduction (CADE-20)*, 2005.

[45] R. Chadha, V. Cheval, S. Ciobaca, and S. Kremer, "Automated Verification of Equivalence Properties of Cryptographic Protocols," 2012.

[46] V. Cheval, "APTE: An Algorithm for Proving Trace Equivalence," in *Tools and Algorithms for the Construction and Analysis of Systems: 20th International Conference, TACAS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings*, E. Ábrahám and K. Havelund, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 587–592.

[47] V. Cortier, "Formal Verification of e-Voting: Solutions and Challenges," *ACM SIGLOG News*, vol. 2, no. 1, pp. 25–34, 2015.

[48] D. Bernhard, O. Kulyk, and M. Volkamer, "Security Proofs for Participation Privacy, Receipt-Freeness and Ballot Privacy for the Helios Voting Scheme," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, p. 1:1--1:10.

[49] S. Neumann, M. Noll, and M. Volkamer, "Election-Dependent Security Evaluation of Internet Voting Schemes," in *ICT Systems Security and Privacy Protection: 32nd IFIP TC 11 International Conference, SEC 2017, Rome, Italy, May 29-31, 2017, Proceedings*, S. di Vimercati and F. Martinelli, Eds. Cham: Springer International Publishing, 2017, pp. 371–382.

[50] M. Volkamer, *Evaluation of Electronic Voting. Requirements and Evaluation Procedures to Support Responsible Election Authorities*, vol. 30, no. 0. 2009.

[51] D. Bernhard, O. Pereira, and 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*, X. Wang and K. Sako, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 626–643.

[52] J. Barrat, M. Chevalier, B. Goldsmith, D. Jandura, J. Turner, and R. Sharma, "Internet voting and individual verifiability: the Norwegian return codes," *Electron. Voting*, vol. 205, pp. 274–283, 2012.

[53] J. B. i Esteve, "El secreto del voto en el sufragio por internet," *Rev. Mex. Análisis Político y Adm. Pública*, no. 2, pp. 57–72, 2012.

[54] B. Adida and O. P. de Marneffe, "Helios Voting," 2017. .

[55] C. Z. Acemyan, P. Kortum, M. D. Byrne, and D. S. Wallach, "From Error to Error: Why Voters Could not Cast a Ballot and Verify Their Vote With Helios, Pr{ê}t {à} Voter, and Scantegrity II," *USENIX J. Elect. Technol. Syst.*, vol. 3, no. 2, pp. 1–25, 2015.

[56] F. Karayumak, M. M. Olembo, M. Kauer, and M. Volkamer, "Usability Analysis of Helios: An Open Source Verifiable Remote Electronic Voting System," in *Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, 2011, p. 5.

[57] "Helios Server -GitHub," 2016. [Online]. Available: https://github.com/benadida/helios-server.

[58] G. Tsoukalas, K. Papadimitriou, P. Louridas, and P. Tsanakas, "From Helios to Zeus," *USENIX J. Elect. Technol. Syst.*, vol. 1, no. 1, pp. 1–17, 2013.

Luis Panizo Alonso received his degree in Telecommunications Engineering degree in 1980 from the Polytechnic University of Madrid and his PhD in Information Technologies (e-voting) in 2015. He has been Associate Professor at the University of Leon since 1985. He has more than 20 articles published in the Computing Engineering and e-voting fields and has been Secretary General of the Electronic Voting Observatory (OVE) from 2004-2012, having taken part in e-voting pilots in 5 countries. He has served as Vice-President of the University of Leon during 2015-2016.

Mila Gascó received her degree and M.Sc in Business Administration from ESADE Business & Law School and Polytechnic University of Catalonia in 1994. She received her PhD in Public Management from the Rovira I Virgili University in 2001. In 2002, she received the Enric Prat de la Riba Award to the Best PhD thesis. From 1995-2000 she has been Assistant Professor at the Rovira I Virgili University, from 2007-2011 Adjunct Professor at the Pompeu Fabra University and Senior Researcher at the Institute of Public Governance at ESADE Business & Law School. Currently, Mila is the Associate Research Director at the Center for Technology in Government and a Research Associate Professor in the Department of Public Administration and Policy at Rockefeller College. She has authored or co-authored over 50 research articles and 5 books on open government, e-government and smart cities and has lead over 30 projects in those areas.

David Y. Marcos del Blanco received his degree in Computer Engineering in 2005, his M.Sc. in 2006 and his PhD in cryptography in 2018 from the University of Leon in Spain. He has over 10 publications in the e-voting field and is a researcher at the University of Leon in e-democracy, cryptography and e-voting. He is Adjunct Professor at IE University in Madrid and Globis University in Tokyo. He has received the 2017 Engineer of the Year Award from the Professional Association of Computer Engineers, Castille and Leon Chapter.

José Ángel Hermida Alonso received his Mathematics Degree with Honorable Mention in 1978 from the University of Valladolid and his PhD in Applied Mathematics in 1983. From 1986 he has been Associate Professor of Algebra, becoming Full Professor in 1994. In 2003, he received a second Full Professor position in Applied Mathematics from the University of Leon and from 2008 to 2016 he has served as the President of the University of Leon. He has authored or co-authored more than 70 articles in Algebra, Applied Mathematics and Cryptography.

Jordi Barrat received a PhD in constitutional law from the University of Leon (1997) and a Law Degree from the University of Navarre (1986-1991). He serves as a professor of constitutional law at the University Rovira Virgili and had similar positions at the universities of

Navarre, Alacant and Leon. He was also Deputy of the Catalan Office for the Quality of Democracy (2013 / 2014). He completed academic stages in Peru, Mexico, Moldova, Mongolia. His research focuses the legal framework of new voting technologies and he has provided consultancies for different international organizations (e.g.: Council of Europa, European Union, OSCE/ODIHR, IDEA, IFES, OAS, A-WEB).

Héctor Alaiz Moretón received his degree in Computer Science, performing the final project at Dublin Institute of Technology, in 2003. He received his PhD in Information Technologies in 2008 (University of León). He has worked as a lecturer since 2005 at the School of Engineering at the University of León. His research interests include knowledge engineering, networks communication and security. He has several works published in international conferences, as well as books and scientific papers in peer review journals. He has been member of scientific committees in conferences. He has headed several Phd Thesis and research projects.

Appendix A: Helios Voting Process

The complete voting process implemented by Helios Voting includes the following steps:

1. The election starts with the naming of an election officer, the selection of a group of trustees, and the introduction of the list of authorized voters.
2. The Ballot Preparation System (BPS) generates the ballot as well as a distributed key pair pk, sk (public and private key respectively).
3. Each voter receives an email containing her user ID, password and election URL.
4. Upon clicking, the Javascript starts and downloads the parameters.
5. The voter selects her option and the BPS encrypts it with pk. The vote also contains a NIZKP to verify that the vote is well formed (preventing a malicious voter to introduce an integer i value instead of 1, allowing the ballot to represent i votes) because votes are not decrypted individually to be counted. Instead, the homomorphic properties of exponential ElGamal are used.
6. The software client shows the voter a hash of her encrypted vote. The voter then has two options:
    - To audit the ballot: the voter receives the nonce used to encrypt her vote. She can use it to verify that her vote has been included and that it represents her elected option. However, the audited ballot is no longer valid and the voter has to restart with the voting process. The voter can verify her vote until she is convinced that Helios is trustable.
    - To seal the vote: before sending it, and with identification purposes, the BPS will ask to provide her user ID and password.
7. The voter sends her user ID, password, encrypted ballot and ZKP to the server, which verifies that all the information is correct.
8. Once the voting phase is over, the server publishes the Bulletin Board with all the encrypted votes, together with the voter's name (or an alias in subsequent versions).
9. Each of the trustees publishes a partial decryption of the encrypted tally, together with a signature of knowledge proving the partial decryption's correct construction. Anyone can verify those proofs.
10. The election officer decrypts the tally and publishes the result. Anyone can check the decryption.

In the hypothetical case of a yes/no type of election, and three voters, Alice, Bob, and Charlite, the following would be a formal description:

1. If Alice wants to vote for option a, her vote is represented as va which is encrypted with the public key pk: $\{v_a\}_{p_k}$.
2. There is a ZKP attached to the vote in order to verify that the vote is valid; which means that either va = 0 or va = 1. The verification is critical because if it didn't exist, a malicious voter could send va = i, being i a positive or negative integer, making her vote amount for i valid votes instead of just one. The ZKP for va is identified as ZKPa.
3. Alice sends $\{v_a\}_{p_k}, ZKP_a$ to the Bulletin Board (BB). Since the BB is public, Alice can check whether her vote has arrived or not.

Similarly, for the three proposed voters, the BB would be as presented in Table 8:

TABLE 8
BULLETIN BOARD IN HELIOS VOTING

Bulletin Board	
Voter	**Vote**
Alice	$\{v_a\}_{pk}, ZKP_a$
Charlie	$\{v_c\}_{pk}, ZKP_c$
Bob	$\{v_b\}_{pk}, ZKP_b$

For the tally, the homomorphic properties of ElGamal are used [48]: the multiplication of the votes' encryption corresponds to the encryption of the addition of the votes:

$$\prod_{i=1}^{n}\{v_i\}_{pk} = \{\sum_{i=1}^{n} v_i\}_{pk} \tag{13}$$

The previous operation can be executed by any party. Once finished, the authorities only decrypt $\{\sum_{i=1}^{n} v_i\}_{pk}$ and publish the results, after which the protocol comes to an end.