Ceremonies for End-to-End Verifiable Elections

Aggelos Kiayias[1⋆], Thomas Zacharias[1⋆], and Bingsheng Zhang[2⋆]

[1] University of Edinburgh, UK
akiayias@inf.ed.ac.uk
tzachari@inf.ed.ac.uk
[2] Security Lancaster Research Centre, Lancaster University, UK
b.zhang2@lancaster.ac.uk

Abstract. State-of-the-art e-voting systems rely on voters to perform certain actions to ensure that the election authorities are not manipulating the election result. This so-called "end-to-end (E2E) verifiability" is the hallmark of current e-voting protocols; nevertheless, thorough analysis of current systems is still far from being complete.

In this work, we initiate the study of e-voting protocols as *ceremonies*. A ceremony, as introduced by Ellison [25], is an extension of the notion of a protocol that includes human participants as separate nodes of the system that should be taken into account when performing the security analysis. that centers on the two properties of end-to-end verifiability and voter privacy and allows the consideration of arbitrary behavioural distributions for the human participants.

We then analyse the Helios system as an e-voting ceremony. Security in the e-voting ceremony model requires the specification of a class of human behaviours with respect to which the security properties can be preserved. We show how end-to-end verifiability and voter privacy are sensitive to human behaviour in the protocol by characterizing the set of behaviours under which the security can be preserved and also showing explicit scenarios where it fails.

We then provide experimental evaluation with human subjects from two different sources where people used Helios: the elections of the International Association for Cryptologic Research (IACR) and a poll of senior year computer science students. We report on the auditing behaviour of the participants as we measured it and we discuss the effects on the level of certainty that can be given by each of the two electorates.

The outcome of our analysis is a negative one: the auditing behaviour of people (including cryptographers) is not sufficient to ensure the correctness of the tally with good probability in either case studied. The same holds true even for simulated data that capture the case of relatively well trained participants while, finally, the security of the ceremony can be shown but under the assumption of essentially ideally behaving human subjects. We note that while our results are stated for Helios, they automatically transfer to various other e-voting systems that, as Helios, rely on client-side encryption to encode the voter's choice.

[⋆] This research was partly supported by ERC project #259152 (CODAMODA), Horizon 2020 project #653497 (PANORAMIX), and project FINER, Greek Secretariat of Research and Technology, funded under action ARISTEIA 1.

1 Introduction

A ceremony, introduced by Ellison [25], extends the notion of a security protocol to include "human nodes" in the protocol specification together with regular computer nodes. Human nodes, are computationally limited and error-prone; they are able to interact with computer nodes via a user interface (UI) as well as communicate with each other via direct communication lines. In this model, computer nodes can be thought of as stateful and probabilistic interactive Turing machines, while human nodes, even though they are stateful, they are limited in terms of computational power and their behaviour can only be considered as a random variable following some arbitrary probability distribution over a set of "admissible behaviours" that are dictated by the UI's they are provided with. Designing and analyzing the security of ceremonies has proven to be valuable for problems that non-trivially rely on human node interaction to ensure their security properties, such as key provisioning and web authentication, see e.g., [25,34,48,10].

In this work, we initiate the study of secure *e-voting ceremonies*. An e-voting ceremony is a protocol between computer and human nodes that aims to assist a subset of the humans (the voters) to cast a ballot for a specified election race. We argue that viewing e-voting as a ceremony (i.e., a protocol with human and computer nodes) captures the security intricacies of the e-voting problem much more effectively than standard protocol based modelling as it was done so far. The reason for this, is that the properties of an election system, most importantly verifiability, rely on human participant behaviour in a highly non-trivial manner. The ability of human nodes to compromise overall security due to their negligence is well known in e-voting system design (cf. [32]) and it is high time that cryptographic models extend to incorporate formally the human participants.

The capability to perform auditing is widely accepted as the most important characteristic for modern e-voting systems. However, even widely deployed[3] systems such as Helios [1] that are touted to be verifiable via auditing still provide only unquantified guarantees of verifiability. The main reason for this is that the correctness of the election result when the election authorities are adversarial is impossible to verify unless the humans that participate in the protocol follow a suitable behaviour. This means that the voters, beyond the ballot-casting procedure, are supposed to carry out additional steps that many may find to be counterintuitive, see e.g., [46] for more discussion of this issue. This potentially leads to the defective execution of the appropriate steps that are to be carried out for verifiability to be supported and hence the verifiability of the election may collapse. Recent studies have shown that voters have rather limited participation and interest to perform the verification steps (e.g., [24] reports about 23 out of a sample of 747 people performed a verifiability check in a deployed end-to-end (E2E) verifiable system). Given that the auditing performed by the

[3] The web-site of the project reports that more than 100,000 votes have been cast with the system.

voters is critical for the integrity of the election result as a whole, it is imperative to determine the class of distributions of behaviours that are able to detect (significant) misbehaviour of the election authorities. Once this class is characterised then one may then try to influence participants to approximate the behaviour by training them.

Traditionally (cf. [11,49,30,16,13,45]), election verifiability was considered at the "individual level" (i.e., a single voter is able to verify her vote intent is properly included in the tally) and the "universal level" (i.e., the election transcript appears to be properly formed). No voter behavioural characteristics were taken into account in the security analysis and the protocols were deemed "end-to-end verifiable" as long as they satisfied merely these two features[4]. The work of [40,41,42] showed that individual verifiability and universal verifiability, even if combined, can still fail to guarantee that the election tally is correct. To mend the concept of verifiability, a "holistic" notion of global verifiability was introduced. Nevertheless, such global verifiability is unattainable without any assumption on human behaviour. Indeed, [42] establishes the verifiability of the Helios system by assuming that voters perform an unbounded number of independent coin flips — an assumption which should be at best considered of theoretical interest, since no voter using the Helios system (or any e-voting system for that matter) should be expected to actually perform ballot-casting via the employment of independent coin flips.

Beyond verifiability, an e-voting system is supposed to also satisfy privacy and other desired properties such as receipt-freeness/coercion resistance. These properties interact with verifiability in various important ways: First, without privacy it is substantially easier to achieve verifiability (this is due to the fact that verification of the recording of one's vote can be done in relatively straightforward manner assuming a public "bulletin-board" [4]). Second, receipt-freeness combined with verifiability suggests that the receipt obtained by the voter from ballot-casting can be delegated to a third-party without fear of coercion or privacy leakage. Given these reasons, a proper analysis of an e-voting system should also include the analysis of at least these properties. The fact that privacy will be entrusted to a set of "trustees" that are human participants in the e-voting system, points again to the importance of the ceremony approach for the case of privacy.

Our results. Our results are as follows.

■ We initiate the study of e-voting ceremonies, i.e., e-voting protocols that involve computer and human nodes, and enable the human participant voters to cast privately their ballots and calculate their tally. In an execution of an e-voting ceremony, human nodes follow a certain behaviour which is sampled according to some distribution over all possible admissible behaviours. No specific assumptions can be made about how human nodes behave and thus the distribution of each human node is a parameter of the security analysis. It follows that the security properties of e-voting ceremonies are conditional on vectors of proba-

[4] A notable departure from this restriction is [52], nevertheless no formal security analysis is performed for the verifiability of this system.

bility distributions of human behaviours. Such vectors are specified over sets of suitably defined deterministic finite state machines with output (transducers[5]) that determine all possible ways that each human participant may interact with the UI's of the computer nodes that are available to them.

■ Extending the work of [42,37], we provide a threat model for (end-to-end) verifiability for e-voting ceremonies. Our threat model has the following characteristics: (i) it provides a holistic approach to argue about end-to-end verifiability by casting the property as an "attack game" played between the adversary and a challenger. (ii) it provides an explicit final goal the adversary wants to achieve by introducing a metric over all possible election outcomes and stating an explicit amount of deviation that the adversary wants to achieve in this metric space. (iii) the adversary is successful provided that the election tally appears to be correct even though it deviates from the true tally according to the stated metric while the number of complaining voters in any failed ballot-casting processes is below a threshold (a ballot-casting process may fail because of adversarial interference). (iv) the resources of the adversary include the complete control of all trustees, election authorities, all voter PC's as well as a subset of the voters themselves. Regarding privacy, we extend the work of [8,37], by providing a threat model for privacy and passive coercion resistance in the sense of [2] for e-voting ceremonies.

■ We cast Helios as an e-voting ceremony: voters and trustees are the human participants of the protocol that are supposed to handle credentials and receipts as well as generate and validate ciphertexts. During ballot-casting, voters perform the Benaloh challenge process [5] and are free to choose to cast their ballot. Voters may further choose to audit their ballot in the bulletin board if they wish to. Trustees are supposed to execute deterministic steps in order to perform the public-key generation during the setup stage of the election and are able to verify their public-key in the bulletin board if they wish. The set of admissible behaviours for voters include any number of Benaloh challenges followed by casting the ciphertext and choosing whether to audit it in the bulletin board.

■ We analyse the Helios e-voting ceremony with respect to the threat-model for privacy and passive coercion and end-to-end verifiability. The behaviours of voters are an explicit component of the security analysis. Specifically, for end-to-end verifiability, we characterise the space of admissible behaviours that enable the verifiability of the election result and we prove an infeasibility and a feasibility result:

1. It is *infeasible* to detect a large deviation in the published tally of the election even if a high number of voters audit it, if (i) there is some i^* that the average voter will perform exactly i^* Benaloh audits with high enough probability compared to the tolerance level of complaints, or (ii) there is a set of indices $\mathcal{J}^*$ that if the average voter performs $j \in \mathcal{J}^*$ Benaloh audits, this can be

[5] We opt to use a finite state machine for voters in order to emphasise that voters do not perform complex calculations. Nevertheless, our model readily generalises if one is willing to assume that voters can perform more complex tasks.

used as a predictor for not auditing the bulletin board; (see Theorem 1 for the precise formulation of the infeasibility result).

2. It is *feasible* to detect a deviation in the tally if a suitable number of voters audit the election, provided that (i) for all i the probability that the adversary performs exactly i Benaloh audits is sufficiently small, and (ii) if the number j of Benaloh audits can be used as a predictor of not auditing the bulletin board, then it holds that the likelihood of j Benaloh audits is sufficiently small; (see Theorem 2 for the precise formulation of the feasibility result).

Regarding privacy, we show that assuming the trustees audit with sufficiently high probability the correct posting of the public-key information, Helios maintains privacy under the assumption that the underlying public-key encryption scheme is IND-CPA.

■ We provide an experimental evaluation from two different sources of human data where people used Helios. We report on the auditing behaviour of the participants as we measured it and we discuss the effects on the level of certainty that can be given in each of the two elections. The message from our evaluation is a negative one: The behaviour profile of people is not such that it can provide sufficient certainty on the correctness of the election result. For instance, as we show from the data collected from the elections of the directors of the International Association for Cryptologic Research (IACR), for elections in the order of hundreds (500) more than 3% of the votes could be overturned with significant probability of no detection (25%), cf. Figure 2. Based on public data on recent election results of the IACR the votes for elected candidates were sufficiently close to candidates that lost in the election and consequently, the results could have been overturned with significant probability without being detected, cf. Table 3. Our results are similarly negative in the second case study. Given our negative results for actual human data, we turn to simulated results for investigating the case when the voters are supposedly well trained with respect to election guidelines. Even for a voter behaviour distribution with supposedly relatively well trained voters our simulated experiment show that the validity of the election result is sustained with rather low confidence.

We note that even though we focused on Helios in this work, our results (including our threat-model analysis for ceremonies and associated security theorems) immediately apply to a number of other e-voting systems. Such systems (that have been identified as single-pass systems in [8]) include [20,21,22,35,51].

Related work. *Ceremony study.* In 2008, protocol 'ceremony' was introduced by Ellison [25] to expand a security protocol with out-of-band channels and the human users. Subsequently, Karlof, Tygar, and Wagner [33] formalised the 'conditioned-safe ceremony' notion, that encompasses forcing functions, defence in depth, and human tendencies. They then evaluated an e-mail web authentication ceremony with 200 participants. Later, the strengths and weaknesses of the 'ceremony' notion were examined by Radke *et al.* [47] in the context of HTTPS, EMV and Opera Mini protocols/ceremonies. In 2013, Carlos *et al.* [9,43] claimed that even though Dolev-Yao's threat model can represent the most powerful attacker in a ceremony, the attacker in this model is not realistic in certain scenarios, especially those related to human peers. They then proposed a threat model that can be adjusted according to each ceremony and consequently adapt the model and the ceremony analysis to realistic scenarios. In 2014, Hatunic-Webster *et al.* [28] proposed an Anti-Phishing Authentication Ceremony Framework for investigating phishing attacks in authentication ceremonies, which builds on the human-in-the-loop security framework of communication processing. Bella and Coles-Kemp [3] introduced a layered analysis of security ceremonies. Their work focuses on the human-computer interaction layer, which features a socio-technical protocol between a user "persona" and a computer interface. As a more related work, in 2015, Johansen and Jøsang [29] proposed a formal probabilistic model for verifying a security ceremony. In their work, the human agent interaction with the user interface are modelled as a non-deterministic process.

E-voting modelling. Conventionally, the verifiability and privacy of an e-voting system is modelled and analysed separately. In terms of the verifiability, individual verifiability [11] and universal verifiability [49,30] was introduced about 20 years ago. End-to-end verifiability in the sense of cast-as-intended, recorded-as-cast, tallied-as-recorded was introduced by [13] and [45] in 2004. The term of End-to-end verifiability/integrity also appeared in [18]. Later, Küsters *et al.* [40] formally proposed symbolic and computational definitions of verifiability. The verifiability of Helios was studied in both symbolic model [39] and computational model [50]. [41] showed that individual verifiability and universal verifiability are not sufficient to guarantee the "global" verifiability of an e-voting system and In [42], they introduced clash attacks, which break the verifiability of some variants of Helios. In terms of privacy, computational privacy was introduced by Benaloh and Fischer [17], while receipt-freeness has been first studied by Benaloh and Tuinstra [6]. Formal definitions for privacy and receipt-freeness have been proposed in the context of applied pi calculus [23] and the universal composability model [27,44]. In [41], the level of privacy of an e-voting system is measured w.r.t. to the observation power the adversary has in a protocol run. In [7], Bernhard *et al.* proposed a game-based notion of ballot privacy and study the privacy of Helios. Their definition was extended by Bernhard, Pereira and Warinschi [8] by allowing the adversary to statically corrupt election authorities. Both these definitions, although they imply a strong indistinguishability property, do not consider receipt-freeness.

Roadmap. The rest of the paper is organised as follows. In Section 2, we introduce the entities, the syntax and the security framework of an e-voting ceremony. In Section 3, we describe the Helios e-voting ceremony according to our syntax. In Section 4, we analyse the E2E verifiability of Helios ceremony. Namely, we prove (I) an infeasibility and (II) a feasibility result under specific classes of voter behaviours, and we comment on the logical tightness of the two classes. In Section 5, we prove the voter privacy/passive coercion resistance of the Helios ceremony. In Section 6, we present evaluations of our results for the E2E verifiability of Helios ceremony. Our evaluations are based on actual human data obtained by elections using Helios as well as simulated data for various sets of parameters. Finally, in the concluding Section 7, where we recall the objectives, methodology, analysis and results of this paper and discuss future work.

2 E-Voting Ceremonies

A ceremony [25] is an extension of a network protocol that involves human nodes along side computer nodes. Computer nodes will be modeled in a standard way while we will model humans as probability distributions over a support set of simple finite state machines. We base our framework for ceremonies on the e-voting system modeling from [37] suitably extending it to our setting.

2.1 The entities of the e-voting ceremony

An e-voting ceremony $\boldsymbol{\mathcal{VC}}$ is associated with three parameters set to be polynomial in the security parameter λ; the number of voters n, the number of options m and the number of trustees k. We use the notation $\mathcal{O} = \{\mathsf{opt}_1, ..., \mathsf{opt}_m\}$ for the set of options, $\mathcal{V} = \{V_1, ..., V_n\}$ for the set of voters and $\mathcal{T} = \{T_1, \ldots, T_k\}$ for the set of trustees. The allowed ways to vote is determined by the collection of subsets $\mathcal{U} \subseteq 2^{\mathcal{O}}$ an the option selection $\mathcal{U}_\ell$ of voter V_ℓ is an element in $\mathcal{U}$.

Let $\mathcal{U}^*$ be the set of vectors of option selections of arbitrary length. Let f be the *election evaluation function* from $\mathcal{U}^*$ to the set $\mathbb{Z}_+^m$ so that $f(\mathcal{U}_1, \ldots, \mathcal{U}_n)$ is equal to an m-vector whose i-th location is equal to the number of times opt_j was chosen in the option selections $\mathcal{U}_1, \ldots, \mathcal{U}_n$.

The interaction among the entities involved in an e-voting ceremony is depicted in Figure 1. The said entities comprise:

- *The human nodes* are the trustees $T_1, \ldots, T_k$, the voters $V_1, \ldots, V_n$ and the *credential distributor* (CD). The latter additional entity is responsible for issuing the credentials generated at the setup phase to the voters. Note that in practice, the CD may be an organization of more than one human nodes executing another ceremony but we do not model this as part of the e-voting ceremony. Here we make the simplifying choice of modeling CD as a single human node (that is able to identify voters using an external identification mechanism operating among humans).
- *The computer nodes* are the voting supporting devices (VSDs), the trustee supporting devices (TSDs), the auditing supporting devices (ASDs), the election authority (EA), and the bulletin board (BB).

Modelling human nodes. We model each human node as a collection of simple finite state machines that can communicate with computer nodes (via a user interface) as well as with each other via direct communication. Specifically, we consider a -potentially infinite- collection of *transducers*, i.e. finite state machines with an input and an output tape, that is additionally equipped with a communication tape.

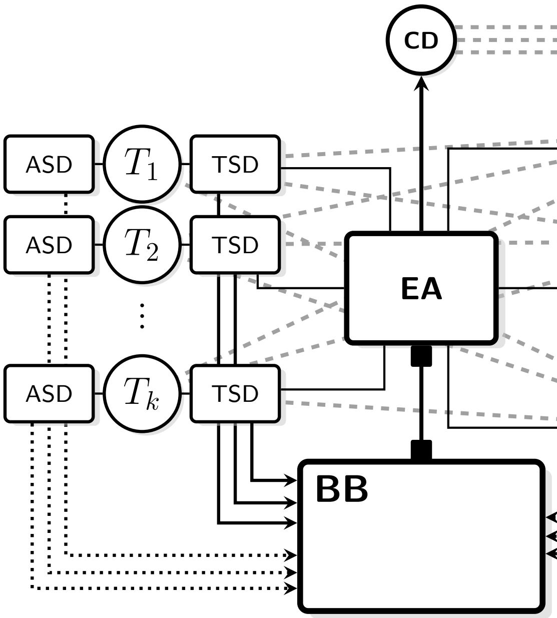

Fig. 1: The entities and the channels active in an e-voting ceremony. The human nodes and the computer nodes used are shown as circles and rectangles respectively. Each voter or trustee human node, interacts with two computer nodes (supporting devices) while the CD human node interacts with the EA. The dotted lines denote read-only access on the BB. The dotted lines denote read-only access on the BB. The grey dashed lines denote channels between human nodes.

We restrict the size of each voter transducer to depend only on the number of options m. Note that this has the implication that the voter transducer *cannot be used to perform cryptographic operations*, which require polynomial number of steps in λ. Transducers may interact with computer nodes, (supporting devices) and use them to produce ciphertexts and transmit them to other computer nodes. The transducers interact with each other via *human level communication channels* (depicted as dashed gray lines in Figure 1), where the exchanged messages are readable by humans (e.g. credentials, PINs, or short message texts but not cryptographic data).

Transducer collections corresponding to voter nodes, trustee nodes and the CD will be denoted as the sets $\mathcal{M}^V$, $\mathcal{M}^T$, and $\mathcal{M}^{\mathsf{CD}}$ respectively. We assume that all sets $\mathcal{M}^V, \mathcal{M}^T$ and $\mathcal{M}^{\mathsf{CD}}$ are polynomial time samplable, i.e., one can produce the description of a transducer from the set in polynomial-time and they have an efficient membership test.

2.2 Syntax and Semantics

In order to express the threat model for the e-voting ceremony, we need to formally describe the syntax and semantics of the procedures executed by the ceremony. We think of an e-voting ceremony $\boldsymbol{\mathcal{VC}}$ as a quintuple of algorithms and ceremonies denoted by $\langle$**Setup**, **Cast**, **Tally**, **Result**, **Verify**$\rangle$ together with the sets of transducers $\mathcal{M}^V, \mathcal{M}^T$ and $\mathcal{M}^{\mathrm{CD}}$ that express the human node operations; these are specified as follows:

The **Setup**$(1^\lambda, \mathcal{O}, \mathcal{V}, \mathcal{U}, \mathcal{T})$ *ceremony* : The setup phase is a ceremony executed by the EA, the BB, the transducers $M_{i_1}, \ldots, M_{i_n} \in \mathcal{M}^V$ that determine the behaviour of voter $V_1, \ldots, V_n$ respectively, a transducer $M^{\mathsf{CD}} \in \mathcal{M}^{\mathsf{CD}}$ describing the behaviour of CD, the transducers $M_i^T \in \mathcal{M}^T$, $i = 1, \ldots, k$ describing the behaviour of the trustees $T_1, \ldots T_k$ respectively and their TSDs. The ceremony generates $\boldsymbol{\mathcal{VC}}$'s public parameters info (which include $\mathcal{O}, \mathcal{V}, \mathcal{U}$) and the voter credentials $\mathsf{cr}_1, \ldots, \mathsf{cr}_n$. After the ceremony execution, each TSD has a private state st_i, each trustee T_i obtains a secret $\overline{s}_i$ and the CD obtains the credentials $\mathsf{cr}_1, \ldots, \mathsf{cr}_n$. In addition, the EA posts an election transcript τ initialised as info on BB. At the end of the **Setup**, the CD will provide $\mathsf{cr}_1, \ldots, \mathsf{cr}_n$ to the voters $V_1, \ldots, V_n$.

The **Cast** *ceremony* : The voting phase is a ceremony executed by the EA, the BB, a transducer $M_{i_\ell} \in \mathcal{M}^V$ that determines the behaviour of voter V_ℓ and her supporting devices VSD_ℓ, ASD_ℓ. V_ℓ executes the **Cast** ceremony according to the behaviour M_{i_ℓ} as follows: M_{i_ℓ} has input $(\mathsf{cr}_\ell, \mathcal{U}_\ell)$, where cr_ℓ is the voter's credential and $\mathcal{U}_\ell$ represents the option selection of V_ℓ. All communication between the voter V_ℓ and EA (resp. BB) happens via VSD_ℓ (resp. ASD_ℓ), where BB has input τ. Upon successful termination, M_{i_ℓ}'s output tape contains the individual audit information audit_ℓ returned by VSD_ℓ. If the termination is not successful, M_{i_ℓ}'s output tape possibly contains a special symbol '$\mathsf{Complain}$', indicating that voter V_ℓ has decided to complain about the incorrect execution of the election procedure. In any case of termination (successful or not), M_{i_ℓ}'s output tape may contain a special symbol 'Audit', indicating that V_ℓ has taken the decision to use her individual audit information audit_ℓ to perform verification at the end of the election; in this case, the individual audit information audit_ℓ will be provided as input to the ASD of V_ℓ. At the end of the ceremony, EA updates its state and BB updates the public transcript τ as necessary.

The **Tally** *ceremony* : After voting period ends, the tally phase is a ceremony executed by the EA, the BB and the trustees $M_i^T \in \mathcal{M}^T$, $i = 1, \ldots, k$ as well as their TSDs. Namely, the EA provides each trustee with the set of cast votes $\mathsf{V}_{\mathsf{tally}}$. Then, the trustees collectively compute the election result and upon successful termination and update the public transcript τ in the BB either directly or via the EA.

The **Result**(τ) *algorithm* : The election result can be computed from any party by parsing the election transcript.

The **Verify**(τ, audit) *algorithm* : The verification algorithm outputs a value in $\{0, 1\}$, where audit is a voter's individual audit information obtained after the voter's engagement in the **Cast** protocol.

The definition of correctness of an honest execution of $\boldsymbol{\mathcal{VC}}$ is straightforward and is provided in the full version of this paper [36, Definition 1].

2.3 Threat model for E2E Verifiability

In order to define the threat model for E2E verifiability we need first to determine the adversarial objective. Intuitively, the objective of the adversary is to manipulate the election result without raising suspicion amongst the participating voters. To express this formally, we have to introduce a suitable notation; given that option selections are elements of a set of m choices, we may encode them as m-bit strings, where the bit in the i-th position is 1 if and only if option P_i is selected. Further, we may aggregate the election results as the list with the number of votes each option has received, thus the output of the **Result** algorithm is a vector in $\mathbb{Z}_+^m$. In this case, a result is feasible if and only if the sum of any of its coordinates is no greater than the number of voters.

Vote extractor. Borrowing from [37], in order to express the threat model for E2E verifiability properly, we will ask for a *vote extractor* algorithm $\mathcal{E}$ (not necessarily efficient, e.g., not running in polynomial-time) that receives as input the election transcript τ and the set of individual audit information $\{\alpha_\ell\}_{\ell\in\mathcal{V}_{\mathsf{succ}}}$, where by $\mathcal{V}_{\mathsf{succ}}$, we denote the set of honest voters that voted successfully. Given such input, $\mathcal{E}$ will attempt to compute $n-|\mathcal{V}_{\mathsf{succ}}|$ vectors $\langle\mathcal{U}_\ell\rangle_{V_\ell\in\mathcal{V}\setminus\mathcal{V}_{\mathsf{succ}}}$ in $\{0,1\}^m$ which correspond to all the voters outside of $\mathcal{V}_{\mathsf{succ}}$ and can be either a option selection, if the voter has voted adversarially or a zero vector, if the voter has not voted successfully. In case $\mathcal{E}$ is incapable of presenting such selection, the symbol $\perp$ will be returned instead. The purpose of the algorithm $\mathcal{E}$ is to express the requirement that the election transcript τ that is posted by the EA in the BB at the end of the procedure contains (in potentially encoded form) a set of well-formed actual votes. Using this notion of extractor, we are capable to express the "actual" result encoded in an election transcript despite the fact that the adversary controls some voters. Note when the extractor $\mathcal{E}$ fails it means that τ is meaningless as an election transcript and thus unverifiable.

Election result deviation. Next, we want to define a measure of *deviation* from the actual election result, as such deviation is the objective of the adversary in an E2E verifiability attack. This will complete the requirements for expressing the adversarial objective in the E2E attack game. To achieve this, it is natural to equip the space of results with a *metric*. We use the metric derived by the 1-norm, $\|\cdot\|_1$ scaled to half, i.e., $\mathrm{d}_1 : \mathbb{Z}_+^m \times \mathbb{Z}_+^m \longrightarrow \mathbb{R}$ and $\mathrm{d}_1(R,R') = \frac{1}{2}\cdot\sum_{i=1}^m |R_i - R'_i|$, where R_i, R'_i is the i-th coordinate of R, R' respectively. Intuitively, moving δ votes from one option to another translates to a distance $\mathrm{d}_1(R,R')$ of exactly δ.

The E2E verifiability game. Let $\boldsymbol{\mathcal{D}} = \langle \mathbf{D}_1,\ldots,\mathbf{D}_n,\mathbf{D}_1^T,\ldots,\mathbf{D}_k^T,\mathbf{D}^{\mathsf{CD}}\rangle$ be a vector of distributions that consists of the distributions $\mathbf{D}_1,\ldots,\mathbf{D}_n$ over the collection of voter transducers $\mathcal{M}^V$, the distributions $\mathbf{D}_1^T,\ldots,\mathbf{D}_k^T$ over the collection of trustee transducers $\mathcal{M}^T$ and the distribution $\mathbf{D}^{\mathsf{CD}}$ over the collection of CD transducers $\mathcal{M}^{\mathsf{CD}}$. We define the E2E verifiability Ceremony game $G_{\mathrm{E2E}}^{\mathcal{A},\mathcal{E},\boldsymbol{\mathcal{D}},\delta,\theta,\phi}$ between the adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ w.r.t. $\boldsymbol{\mathcal{D}}$ and the vote extractor $\mathcal{E}$ which takes as input the security parameter λ, the number of voters n, the number of options m, and the number of trustees k and is parameterised by (i) the deviation amount, δ, (according to the metric $\mathrm{d}_1(\cdot,\cdot)$) that the adversary wants to achieve, (ii) the number of honest voters, θ, that terminate the **Cast** ceremony successfully and (iii) the number of honest voters, ϕ, that submit a complaint in case of unsuccessful termination during the **Cast** ceremony.

Throughout the game, the adversary fully controls the election by corrupting the EA and all the trustees $\mathcal{T} = \{T_1, \dots T_k\}$, while the CD remains honest during the setup phase. In addition, it corrupts all the voters VSDs and manages the **Cast** ceremony executions. For each voter V_ℓ, the adversary may choose to corrupt V_ℓ or to allow the challenger to play on her behalf. Note that the challenger retains the control of the ASD[6] for honest voters and samples for each honest voter a transducer from the corresponding distribution. If a voter V_ℓ is uncorrupted, the adversary provides the option selection that V_ℓ should use in the **Cast** ceremony; the challenger samples a transducer $M_{i_\ell} \xleftarrow{\mathbf{D}_\ell} \mathcal{M}^V$ from voter transducer distribution $\mathbf{D}_\ell$ and then executes the **Cast** ceremony according to M_{i_ℓ}'s description to vote the given option selection and decide whether to audit the election result at the end. The adversary finally posts the election transcript in the BB. The adversary will win the game provided that there are at least θ of honest voters that terminate the ballot-casting successfully and at most ϕ complaining honest voters, but the deviation of the tally is bigger than δ w.r.t. d_1 or the extractor fails to produce the option election of the dishonest voters. The attack game is specified in detail in Figure 2.

Definition 1. *Let $\epsilon \in [0,1]$ and $n, m, k, \delta, \theta, \phi \in \mathbb{N}$ with $\theta, \phi \leq n$. The e-voting ceremony $\boldsymbol{\mathcal{VC}}$ w.r.t. the election function f achieves* E2E verifiability with error *ϵ, transducer distribution vector $\boldsymbol{\mathcal{D}}$, a number of at least θ honest successful voters, at most ϕ honest complaining voters and tally deviation at most d if there exists a (not necessarily polynomial-time) vote extractor $\mathcal{E}$ such that for every PPT adversary $\mathcal{A}$:*

$$\Pr[G_{\mathrm{E2E}}^{\mathcal{A},\mathcal{E},\boldsymbol{\mathcal{D}},\delta,\theta,\phi}(1^\lambda, n, m, k) = 1] \leq \epsilon.$$

Remark 1 (Universal voter distribution). In some e-voting systems, the voters can be uniquely identified during the **Cast** ceremonies, e.g. the voter's real ID is used. Hence, the adversary is able to identify each voter V_ℓ and learn its profile expressed by $\mathbf{D}_\ell$. Then, the adversary may choose the best attack strategy depending on $\mathbf{D}_\ell$. Nevertheless, in case the credentials are randomly and anonymously assigned to the voters by the CD, the adversary will not be able to profile voters given his view in the ballot-casting ceremony (recall that in the E2E game the CD remains honest). Therefore, it is possible to unify the distributions to a *universal voter* distribution, denoted as $\mathbf{D}$, which reflects the profile of the "average voter." Specifically, in this case, we will have $\mathbf{D}_1 = \cdots = \mathbf{D}_n = \mathbf{D}$.

[6] In the voting phase client-side encryption systems like Helios [1], the voters' ASDs must be live for potential ballot auditing.

E2E Verifiability Ceremony Game $G_{\mathrm{E2E}}^{\mathcal{A},\mathcal{E},\boldsymbol{\mathcal{D}},\delta,\theta,\phi}(1^\lambda, n, m, k)$

- The adversary $\mathcal{A}$ chooses a list of options $\mathcal{O} = \{\mathsf{opt}_1, ..., \mathsf{opt}_m\}$, a set of voters $\mathcal{V} = \{V_1, ..., V_n\}$, a set of trustees $\mathcal{T} = \{T_1, \dots T_k\}$ and the set of allowed option selections $\mathcal{U}$. It provides Ch with the sets $\mathcal{O}, \mathcal{V}, \mathcal{T}, \mathcal{U}$. Throughout the game, the challenger $\mathcal{C}$ plays the role of the BB.
- $\mathcal{C}$ and $\mathcal{A}$ engage in the **Setup** ceremony on input $(1^\lambda, \mathcal{O}, \mathcal{V}, \mathcal{U}, \mathcal{T})$ with $\mathcal{A}$ playing the role of EA and all trustees and their associated TSDs while $\mathcal{C}$ plays the role of CD by following the transducer $M^{\mathsf{CD}} \overset{\mathbf{D}^{\mathsf{CD}}}{\longleftarrow} \mathcal{M}^{\mathsf{CD}}$. In this way $\mathcal{C}$ obtains info and the voter credentials $\mathsf{cr}_1, \dots, \mathsf{cr}_n$. . If the CD refuses to distribute the credentials to the voters, then the game terminates.
- $\mathcal{A}$ and $\mathcal{C}$ engage in an interaction where $\mathcal{A}$ schedules the **Cast** ceremonies of all voters. For each voter V_ℓ, $\mathcal{A}$ can either completely control the voter or allow $\mathcal{C}$ operate on their behalf. In the latter case. $\mathcal{A}$ provides a option selection $\mathcal{U}_\ell$ to $\mathcal{C}$ which samples a transducer $M_{i_\ell} \overset{\mathbf{D}_\ell}{\longleftarrow} \mathcal{M}^V$ and engages with the adversary $\mathcal{A}$ in the **Cast** ceremony so that $\mathcal{A}$ plays the role of VSD_ℓ and EA and $\mathcal{C}$ plays the role of V_ℓ according to transducer M_{i_ℓ} on input $(\mathsf{cr}_\ell, \mathcal{U}_\ell)$ and its associated ASD_ℓ. Provided the ceremony terminates successfully, $\mathcal{C}$ obtains the individual audit information audit_ℓ produced by M_{i_ℓ}, on behalf of V_ℓ.
- Finally, $\mathcal{A}$ posts the election transcript τ to the BB.

We define the following subsets of honest voters (i.e., those controlled by $\mathcal{C}$):

- $\mathcal{V}_{\mathsf{succ}}$ is the set of honest voters that terminated successfully.
- $\mathcal{V}_{\mathsf{comp}}$ is the set of honest voters s.t. the special symbol '$\mathsf{Complain}$' is written on the output tape of the corresponding transducer.
- $\mathcal{V}_{\mathsf{audit}}$ is the set of honest voters s.t. the special symbol 'Audit' is written on the output tape of the corresponding transducer.

The game returns a bit which is 1 if and only if the following conditions hold true:

1. $|\mathcal{V}_{\mathsf{succ}}| \geq \theta$,
2. $|\mathcal{V}_{\mathsf{comp}}| \leq \phi$, (i.e., at most ϕ honest voters complain).
3. $\forall \ell \in [n]$: if $V_\ell \in \mathcal{V}_{\mathsf{audit}}$, then $\mathbf{Verify}(\tau, \mathsf{audit}_\ell) = 1$.

and either one of the following two conditions:

4. (a) If $\bot \neq \langle \mathcal{U}_\ell \rangle_{V_\ell \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{succ}}} \leftarrow \mathcal{E}(\tau, \{\mathsf{audit}_\ell\}_{V_\ell \in \mathcal{V}_{\mathsf{succ}}})$, then

$$\boxed{\mathrm{d}_1(\mathbf{Result}(\tau), f(\langle \mathcal{U}_1, \dots, \mathcal{U}_n \rangle)) \geq \delta} \; .$$

   (b) $\bot \leftarrow \mathcal{E}(\tau, \{\mathsf{audit}_\ell\}_{V_\ell \in \mathcal{V}_{\mathsf{succ}}})$.

Fig. 2: The E2E Verifiability Ceremony Game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ w.r.t. the vote extractor $\mathcal{E}$ and the vector of transducer distributions $\boldsymbol{\mathcal{D}} = \langle \mathbf{D}_1, \dots, \mathbf{D}_n, \mathbf{D}_1^T, \dots, \mathbf{D}_k^T, \mathbf{D}^{\mathsf{CD}} \rangle$.

2.4 Threat model for Voter Privacy

The threat model of privacy concerns the actions that may be taken by the adversary to figure out the choices of the honest voters. We specify the goal of the adversary in a very general way. In particular, for an attack against privacy to succeed, we ask that there is an election result, for which the adversary is capable of distinguishing how people vote while it has access to (i) the actual individual audit information that the voters obtain after ballot-casting as well as (ii) a set of ceremony views that are consistent with all the honest voters' views in the **Cast** ceremony instances they participate.

Observe that any system that is secure against such a threat scenario possesses also "passive coercion resistance", i.e., voters cannot prove how they voted by showing the individual audit information ceremony or even presenting the view they obtain from the **Cast**. Given that in the threat model we allow the adversary to observe the view of the voter in the **Cast** ceremony, we need to allow the voter to be able to lie about her view (otherwise an attack could be trivially mounted). We stress that the simulated view of the voter in the **Cast** ceremony does not contain the view of the internals of the VSD. This means that, with respect to privacy, the adversary may not look into the internals of the VSD for the honest voters. The above is consistent, for instance, with the scenario that the voter can give to the VSD her option choice to be encoded. While the adversary will be allowed to observe a simulated view of the voter during the **Cast** ceremony, it will be denied access to the internals of the VSD during the **Cast** execution. This increases the opportunities where the voter can lie about how she executes the **Cast** ceremony.

The Voter Privacy Game. Following the same logic as in the E2E Verifiability game, we specify a vector of transducer distributions over the collection of voter transducers $\mathcal{M}^V$, trustee transducers $\mathcal{M}^T$ and CD transducers $\mathcal{M}^{\mathrm{CD}}$ denoted by $\mathcal{D} = \langle \mathbf{D}_1, \ldots, \mathbf{D}_n, \mathbf{D}_1^T, \ldots, \mathbf{D}_k^T, \mathbf{D}^{\mathrm{CD}} \rangle$. We then express the threat model as a *Voter Privacy game*, denoted by $G_{t\text{-priv}}^{\mathcal{A},\mathcal{S},\mathcal{D}}$, that is played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, that takes as input the security parameter λ, the number of voters n, the number of options m, and the number of trustees k as described in Figure 3 and returns 1 or 0 depending on whether the adversary wins. An important feature of the voter privacy game is the existence of an *efficient simulator* $\mathcal{S}$ that provides a simulated view of the voter in the **Cast** ceremony. Note that the simulator is not responsible to provide the view of the voter's supporting device (VSD). Intuitively, this simulator captures the way the voter can lie about her choice in the **Cast** ceremony in case she is coerced to present her view after she completes the ballot-casting procedure.

The attack game is parameterised by t, v. The adversary starts by selecting the voter, option and trustee identities for given parameters n, m, k and determines the allowed ways to vote. The challenger subsequently flips a coin b (that will change its behaviour during the course of the game) and will perform the **Setup** ceremony with the adversary playing the role of the EA, the CD and up to t trustees along with their associated TSDs and ASDs.

Voter Privacy Game $G_{\text{priv}}^{\mathcal{A},\mathcal{S},\boldsymbol{\mathcal{D}},t,v}(1^\lambda, n, m, k)$

- $\mathcal{A}$ on input $1^\lambda, n, m, k$, chooses a list of options $\mathcal{O} = \{\mathsf{opt}_1, ..., \mathsf{opt}_m\}$, a set of voters $\mathcal{V} = \{V_1, ..., V_n\}$, a set of trustees $\mathcal{T} = \{T_1, ..., T_k\}$ a trustee $T_h \in \mathcal{T}$ and the set of allowed option selections $\mathcal{U}$. It provides $\mathcal{C}$ with the sets $\mathcal{O}, \mathcal{V}, \mathcal{U}$ as well as the set of corrupted trustees $\mathcal{T}_{\mathsf{corr}}$.
- $\mathcal{C}$ flips a coin $b \in \{0, 1\}$ and performs the **Setup** ceremony on input $(1^\lambda, \mathcal{O}, \mathcal{V}, \mathcal{U}, \mathcal{T})$ with the adversary playing the role of the EA, CD and all trustees in $\mathcal{T}_{\mathsf{corr}}$, while $\mathcal{C}$ plays the role of all the honest trustees. The role of every honest trustee $T_h \in \mathcal{T} \setminus \mathcal{T}_{\mathsf{corr}}$ is played by $\mathcal{C}$ following the transducers $M^{T_h} \xleftarrow{\mathbf{D}^{T_h}} \mathcal{M}^T$.
- The adversary $\mathcal{A}$ and the challenger $\mathcal{C}$ engage in an interaction where $\mathcal{A}$ corrupts the EA and schedules the **Cast** ceremonies of all voters which may run concurrently. $\mathcal{A}$ also controls the ASDs of all voters. At the onset of each voter ceremony, $\mathcal{A}$ chooses whether voter V_ℓ, $\ell = 1, \ldots, n$ and its associated VSD is corrupted or not.
  - If V_ℓ and its associated VSD are corrupted, then no specific action is taken by the challenger, as the execution is internal to adversary.
  - If V_ℓ and its associated VSD are not corrupted, then $\mathcal{A}$ provides $\mathcal{C}$ with two option selections $\langle \mathcal{U}_\ell^0, \mathcal{U}_\ell^1 \rangle$. The challenger samples $M_{i_\ell} \xleftarrow{\mathbf{D}_\ell} \mathcal{M}^V$ and sets V_ℓ's input to $(\mathsf{cr}_\ell, \mathcal{U}_\ell^b)$, where cr_ℓ is the credential provided by the adversarially controlled CD. Then, $\mathcal{C}$ and $\mathcal{A}$ engage in the **Cast** ceremony with $\mathcal{C}$ controlling V_ℓ (that behaves according to M_{i_ℓ}) and her VSD, while the adversary $\mathcal{A}$ observes the network interaction. When the **Cast** ceremony terminates, the challenger $\mathcal{C}$ provides to $\mathcal{A}$: (i) the individual audit information audit_ℓ that V_ℓ obtains from the ceremony, and (ii) $\underline{\text{if } b = 0}$, the current view of the internal state of the voter V_ℓ that the challenger obtains from the **Cast** execution, or $\underline{\text{if } b = 1}$, a simulated view of the internal state of V_ℓ produced by $\mathcal{S}(\mathsf{view}_\mathcal{C})$, where $\mathsf{view}_\mathcal{C}$ is the current view of the challenger.
- $\mathcal{A}$ and $\mathcal{C}$ engage in the **Tally** ceremony with the adversary playing the role of the EA, CD and all trustees in $\mathcal{T}_{\mathsf{corr}}$, while $\mathcal{C}$ plays the role of all the honest trustees.
- Finally, $\mathcal{A}$ terminates returning a bit b^*.

Denote the set of corrupted voters as $\mathcal{V}_{\mathsf{corr}}$. The game returns a bit which is 1 if and only if the following hold true:

1. $b = b^*$ (i.e., the adversary guesses b correctly).
2. $|\mathcal{T}_{\mathsf{corr}}| \leq t$ (i.e., the number of corrupted trustees is bounded by t).
3. $|\mathcal{V}_{\mathsf{corr}}| \leq v$ (i.e., the number of corrupted voters is bounded by v).
4. $f(\langle \mathcal{U}_\ell^0 \rangle_{V_\ell \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{corr}}}) = f(\langle \mathcal{U}_\ell^1 \rangle_{V_\ell \in \mathcal{V} \setminus \mathcal{V}_{\mathsf{corr}}})$ (i.e., the election result w.r.t. the set of non-corrupted voters does not leak b).

Fig. 3: The Voter Privacy Game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ w.r.t. the view simulator $\mathcal{S}$ and the vector of transducer distributions $\boldsymbol{\mathcal{D}} = \langle \mathbf{D}_1, \ldots, \mathbf{D}_n, \mathbf{D}_1^T, \ldots, \mathbf{D}_k^T, \mathbf{D}^{\mathsf{CD}} \rangle$.

The honest trustees' behaviours will be determined by transducers selected at random by the challenger from $\mathcal{M}^T$ according to the corresponding distribution. Subsequently, the adversary will schedule all **Cast** ceremonies selecting which voters it prefers to corrupt and which ones it prefers to allow to vote honestly. The adversary is allowed to corrupt at most v voters and their VSDs. In addition, $\mathcal{A}$ is allowed to corrupt the ASDs of all voters. The voters that remain uncorrupted are operated by the challenger and they are given two option selections to vote. For each uncorrupted voter V_ℓ, the challenger first samples a transducer $M_{i_\ell} \leftarrow \mathbf{D}_\ell$ and then executes the **Cast** ceremony according to M_{i_ℓ}'s description to vote one of its two option selections based on b.

The adversary will also receive the individual audit information that is obtained by each voter as well as either (i) the actual view (if $b = 0$) or (ii) a *simulated* view, generated by $\mathcal{S}$ (if $b = 1$), of each voter during the **Cast** ceremony (this addresses the individual audit information-freeness aspect of the attack game). Upon completion of ballot-casting, the adversary will execute with the challenger the **Tally** ceremony and subsequently the adversary will attempt to guess b. The attack is successful provided that the election result is the same with respect to the two alternatives provided for each honest voter by the adversary and the adversary manages to guess the challenger's bit b correctly. The game is presented in detail in Figure 3.

Definition 2. *Let $m, n, k, t, v \in \mathbb{N}$ with $t \leq k$ and $v \leq n$. Let $\boldsymbol{\mathcal{VC}}$ be an e-voting ceremony with m options, n voters and k trustees w.r.t. the evaluation election unction f. We say that $\boldsymbol{\mathcal{VC}}$ achieves* voter privacy with error ϵ *for transducer distribution vector $\boldsymbol{\mathcal{D}}$, at most t corrupted trustees and v corrupted voters, if there is an efficient simulator $\mathcal{S}$ such that for any PPT adversary $\mathcal{A}$:*

$$\left| \Pr[G_{\text{priv}}^{\mathcal{A},\mathcal{S},\boldsymbol{\mathcal{D}},t,v}(1^\lambda, n, m, k) = 1] - \frac{1}{2} \right| \leq \epsilon ,$$

Threat model alternatives. The framework presented in this section is a first attempt to model human behaviour in the cryptographic e-voting analysis, therefore various approaches or extensions could be considered. In the full version of this paper [36, Section 2.5], we discuss on some selected possible alternatives on this subject.

3 Syntax of Helios Ceremony

In this section, we present a formal description of Helios ceremony according to the syntax provided in Subsection 2.2. For simplicity, we consider the case of *1-out-of-m elections*, where the set of allowed selections $\mathcal{U}$ is the collection of singletons, $\{\{\mathsf{opt}_1\}, \ldots, \{\mathsf{opt}_m\}\}$, from the set of options $\mathcal{O}$. Our syntax does not reflect the current implemented version of Helios, as it adapts necessary minimum modifications to make Helios secure. For instance, we ensure that each voter is given a *unique identifier* to prevent Helios from the clash attacks introduced in [42]. In addition, we consider a hash function $H(\cdot)$ that all parties have oracle access to, used for committing to election information and ballot generation, as well as the *Fiat-Shamir transformations* [26] in the NIZK proofs that the system requires. As we state below, in the generation of the NIZK proofs for ballot correctness, the unique identifier is included in the hash to prevent replaying attacks presented in [19]. Moreover, we apply strong Fiat-Shamir transformations, where the statement of the NIZK should also be included in the hash. As shown in [8], strong Fiat-Shamir based NIZKs are *simulation sound extractable*, while weak Fiat-Shamir based NIZKs make the Helios vulnerable.

Finally, we stress that we model trustees' behaviour by considering the event that the trustee will or will not the verify the correct posting of its partial public key. This is done so that we capture the possible privacy vulnerability in Helios's implementation architecure studied in [38]; that is, in the case where no honest trustee performs such verification then a malicious EA may act as man-in-the-middle and replace the trustees' partial public keys with ones it adversarially generates, thus resulting to a total break of voters' privacy.

The Helios's transducers :

We define the collections of transducers $\mathcal{M}^V, \mathcal{M}^T, \mathcal{M}^{\mathsf{CD}}$ that reflect the admissible behaviours of voters, trustees and CD respectively.

The set of admissible voter transducers is denoted by $\mathcal{M}^V := \{M_{i,c,a}\}_{i\in[0,q]}^{c,a\in\{0,1\}}$, where $q \in \mathbb{N}$; The transducer $M_{i,c,a}$ audits the ballot created by the VSD exactly i times (using its ASD) and then submits the $(i+1)$-th ballot created by the VSD; Upon successful termination, it outputs a individual audit information audit obtained from the VSD; If the termination is not successful and $c = 1$, $M_{i,c,a}$ outputs a special symbol '$\mathsf{Complain}$' to complain about its failed engagement in the **Cast** ceremony. In any case of termination, when $a = 1$, $M_{i,c,a}$ also outputs a special symbol 'Audit' and sends audit to the ASD. To guarantee termination, we limit the maximum number of ballot audits by threshold q.

The admissible trustee transducers are two and labelled as M_0^T, M_1^T (so that $\mathcal{M}^T = \{M_0^T, M_1^T\}$). At a high level, both M_0^T and M_1^T will utilise the TSD to generate a partial public/secret key pair in the **Setup** ceremony. However, only M_1^T will verify the correct posting of its partial public key in the BB, whereas M_0^T will have no other interaction with the election.

The CD is required to check the validity of the credentials $\mathsf{cr}_1, \ldots, \mathsf{cr}_n$ generated by the potentially malicious EA before distributing them. In Helios, we define the credential $\mathsf{cr}_i := (\mathrm{ID}_i, t_i)$, where ID_i is a unique voter identity and t_i is an authentication token. The credential distributor first checks for all $i, j \in [n]$: if $i \neq j$ then $\mathrm{ID}_i \neq \mathrm{ID}_j$, and halts if the verification fails. Upon success, it randomly sends each voter V_ℓ a credential though some human channels. Hence, we define the set of CD transducers as $\mathcal{M}^{\mathsf{CD}} := \{M_\sigma^{\mathsf{CD}}\}_{\sigma\in S_n}$, where S_n stands for all possible permutations $[n] \mapsto [n]$.

We define the Helios ceremony quintuple $\langle$**Setup**, **Cast**, **Tally**, **Result**, **Verify**$\rangle$, using the hash function $H(\cdot)$ as follows:

The **Setup**$(1^\lambda, \mathcal{O}, \mathcal{V}, \mathcal{U}, \mathcal{T})$ *ceremony* :

Each trustee transducer $M_{b_i}^{T_i} \in \{M_0^T, M_1^T\}$, $i = 1, \ldots, k$ sends signal to its TSD. The TSD generates a pair of threshold ElGamal partial keys $(\mathsf{pk}_i, \mathsf{sk}_i)$ and sends pk_i together with a Schnorr (strong Fiat-Shamir) NIZK proof of knowledge of sk_i to the EA. In addition, the TSD returns a trustee secret $\bar{s}_i := (H(\mathsf{pk}_i), \mathsf{sk}_i)$ to $M_{b_i}^{T_i}$. If there is a proof that EA does not verify, then EA aborts the protocol. Next, EA computes the election public key $\mathsf{pk} = \prod_{i \in [k]} \mathsf{pk}_i$. The public parameters, info, which include pk and the partial public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_k$ as well as the related NIZK proofs of knowledge are posted in the BB by the EA.

Trustee auditing step [38]: for $i = 1, \ldots, k$, if $b_i = 1$, then $M_{b_i}^{T_i}$ sends $H(\mathsf{pk}_i)$ to its ASD, and the ASD will fetch info from the BB to verify if there exists a partial public key pk_* such that its hash matches $H(\mathsf{pk}_i)$. In case this verification fails, T_i sends a message '$\mathsf{Invalid\ public\ key}$' to all the voters via the human communication channels shown in Figure 1.

Finally, the EA generates the voter credentials $\mathsf{cr}_1, \ldots, \mathsf{cr}_n$, where $\mathsf{cr}_i := (\mathrm{ID}_i, t_i)$, and t_i is a random authentication code. Then, forwards the credentials to the CD transducer M^{CD}. The CD transducer M_σ^{CD} checks the uniqueness of each ID_i and distributes them to the voter transducers $M_{i_\ell, c_\ell, a_\ell}$ for $\ell \in [n]$, according to the permutation σ over $[n]$ that specifies its behaviour.

The **Cast** *ceremony* :

For each voter V_ℓ, the corresponding transducer $M_{i_\ell, c_\ell, a_\ell}$ has a pre-defined number of i_ℓ ballot auditing steps, where $i_\ell \in [0, q]$. The input of $M_{i_\ell, c_\ell, a_\ell}$ is $(\mathsf{cr}_\ell, \mathcal{U}_\ell)$. If V_ℓ has received an '$\mathsf{Invalid\ public\ key}$' from at least one trustee, then it aborts the ceremony. If no such message was sent, then for $u \in [i_\ell]$, the following steps are executed:

1. $M_{i_\ell, c_\ell, a_\ell}$ sends $(\mathrm{ID}_\ell, \mathcal{U}_\ell)$ to its VSD, labelled as VSD_ℓ. Let opt_{j_ℓ} be the option selection of V_ℓ, i.e. $\mathcal{U}_\ell = \{\mathsf{opt}_{j_\ell}\}$.
2. For $j = 1, \ldots, m$, VSD_ℓ creates a ciphertext, $C_{\ell,j}$, that is a lifted ElGamal encryption under pk of 1, if $j = j_\ell$ (the selected option position), or 0 otherwise. In addition, it attaches a NIZK proof $\pi_{\ell,j}$ showing that $C_{\ell,j}$ is an encryption of 1 or 0. Finally, an overall NIZK proof π_ℓ is generated, showing that exactly one of these ciphertexts is an encryption of 1. These proofs are strong Fiat-Shamir transformations of *disjunctive Chaum-Pedersen (CP)* proofs [15]. To generate the CP proofs, the unique identifier ID_ℓ is included in the hash. The ballot generated is $\psi_{\ell,u} = \langle \psi_{\ell,u}^0, \psi_{\ell,u}^1 \rangle$, where $\psi_{\ell,u}^0 = \big\langle (C_{\ell,1}, \pi_{\ell,1}), \ldots, (C_{\ell,m}, \pi_{\ell,m}), \pi_\ell \big\rangle$ and $\psi_{\ell,u}^1 = H(\psi_{\ell,u}^0)$. The VSD responds to $M_{i_\ell, c_\ell, a_\ell}$ with the ballot $\psi_{\ell,u}$.
3. Then, $M_{i_\ell, c_\ell, a_\ell}$ sends a *Benaloh audit request* to VSD_ℓ. In turn, VSD_ℓ returns the randomness $r_{\ell,u}$ that was used to create the ballot $\psi_{\ell,u}$. The $M_{i_\ell, c_\ell, a_\ell}$ sends $(\mathrm{ID}_\ell, \psi_{\ell,u}, r_{\ell,u})$ to its ASD, which will audit the validity of the ballot. If the verification fails, $M_{i_\ell, c_\ell, a_\ell}$ halts. If the latter happens and $c_\ell = 1$, $M_{i_\ell, c_\ell, a_\ell}$ outputs a special symbol '$\mathsf{Complain}$', otherwise it returns no output.

After the i_ℓ-th successfully Benaloh audit, $M_{i_\ell, c_\ell, a_\ell}$ invokes VSD_ℓ to produce a new ballot ψ_ℓ as described in step 2 above; however, upon receiving ψ_ℓ, $M_{i_\ell, c_\ell, a_\ell}$

now sends cr_ℓ to VSD_ℓ, indicating it to submit the ballot to the EA. The M_{i_ℓ,c_ℓ,a_ℓ} then outputs $\mathsf{audit}_\ell := (\mathrm{ID}_\ell, \psi^1_\ell)$. If $a_\ell = 1$, M_{i_ℓ,c_ℓ,a_ℓ} also outputs a special symbol 'Audit' which indicates that it will send audit_ℓ to ASD_ℓ which will audit the BB afterwards, as specified in the **Verify** algorithm below.

When EA receives a cast vote $(\mathsf{cr}_\ell, \psi_\ell)$ from VSD_ℓ, it checks the validity of the credential cr_ℓ and that ψ_ℓ is a well-formed ballot by verifying the NIZK proofs. If the check fails, then it aborts the protocol. After voting ends, EA updates its state with the pairs $\{(\psi_\ell, \mathrm{ID}_\ell)\}_{V_\ell \in \mathcal{V}_{\mathsf{succ}}}$ of cast votes and the associated identifiers, where $\mathcal{V}_{\mathsf{succ}}$ is the set of voters that voted successfully.

The **Tally** *ceremony* :

In the **Tally** ceremony, EA sends $\{\psi_\ell\}_{V_\ell \in \mathcal{V}_{\mathsf{succ}}}$ to all trustee transducers $M^{T_i}_{b_i}$'s TSD, $i = 1, \ldots, k$. Next, the TSD of each $M^{T_i}_{b_i}$, $i = 1, \ldots, k$, performs the following computation: it constructs the product ciphertext $\mathbf{C}_j = \prod_{V_\ell \in \mathcal{V}_{\mathsf{succ}}} C_{\ell,j}$ for $j = 1, \ldots, m$. By the additive homomorphic property of (lifted) ElGamal, each $\mathbf{C}_j$ is a valid encryption of the number of votes that the option opt_j received. Then, the TSD uses sk_i to produce the partial decryption of all C_j, denoted by x^i_j, and sends it to the EA along with NIZK proofs of correct partial decryption. The latter are Fiat-Shamir transformations of CP proofs. If there is a proof that EA does not verify, then it aborts the protocol. After all trustees finish their computation, EA updates τ with $\big\{(x^i_1, \ldots, x^i_m)\big\}_{i \in [k]}$ and the NIZK proofs.

The **Result**(τ) *algorithm* :

For each option opt_j, the **Result** algorithm computes the number of votes, x_j, that opt_j has received using the partial decryptions $x^1_j, \ldots, x^k_j$. The output of the algorithm is the vector $\langle x_1, \ldots, x_m \rangle$.

The **Verify**$(\tau, \mathsf{audit}_\ell)$ *algorithm* :

The algorithm **Verify**$(\tau, \mathsf{audit}_\ell)$ outputs 1 if the following conditions hold:

1. The structure of τ and all election information is correct (using info).
2. There exists a ballot in τ, indexed by ID_ℓ, that contains the hash value ψ^1_ℓ.
3. The NIZK proofs for the correctness of all ballots in τ verify.
4. The NIZK proofs for the correctness of all trustees' partial decryptions verify.
5. For $j = 1, \ldots, m$, x_j is a decryption of $\mathbf{C}'_j$, where $\mathbf{C}'_j$ is the homomorphic ciphertext created by multiplying the respective ciphertexts in the ballots published on the BB (in an honest execution, $\mathbf{C}'_j$ should be equal to $\mathbf{C}_j$).

4 E2E Verifiability of Helios e-Voting Ceremony

In a Helios e-voting ceremony, an auditor can check the correct construction of the ballots and the valid decryption of the homomorphic tally by verifying the NIZK proofs. In our analysis, it is sufficient to require that all NIZK proofs have negligible soundness error $\epsilon(\cdot)$ in the RO model. Note that in Section 3, we explicitly modify Helios to associate ballots with the voters' identities, otherwise a clash attack [42] would break verifiability. For simplicity in presentation, we assume that the identifiers are created by the adversary, i.e. the set $\{\mathrm{ID}_\ell\}_{\ell\in[n]}$ matches the set of voters $\mathcal{V}$.

Throughout our analysis, we assume the honesty of the CD and thus the distribution of the credentials is considered to be an arbitrary permutation over $[n]$. Since there are only two admissible trustee transducers M_0^T, M_1^T, the distribution of trustee transducers $\mathbf{D}_p^T$ is set as the p-biased coin-flip below:

$$\Pr_{\mathbf{D}_p^T}[M] = \begin{cases} p, & \text{if } M = M_1^T \\ 1-p, & \text{if } M = M_0^T \end{cases} \tag{1}$$

Moreover, in the **Cast** ceremony, the ballots and individual audit information are produced before the voters show their credentials to the system. Since the CD is honest, the adversary is oblivious the the maps between the credentials to the voter transducers. The credentials are only required when the voters want to submit their ballots, hence, according to the discussion in Remark 1, we will consider only a universal voter transducer distribution $\mathbf{D}$ in the case study of Helios. Namely, $\mathbf{D}_1 = \cdots = \mathbf{D}_n = \mathbf{D}$.

4.1 Attacks on verifiability

As mentioned earlier, we have modified Helios to prevent the system from clash attacks [42]. For simplicity, we exclude all the trivial attacks that the adversary may follow, i.e. the ones that will be detected with certainty (e.g. malformed or unreadable voting interface and public information). Therefore, the meaningful (non-trivial) types of attack that an adversary may launch are the following:

■ **Collision attack**: the adversary computes two votes which hash to the same value. The collision resistance of the hash function $H(\cdot)$, prevents from these attacks except from some negligible probability ϵ' [7].

■ **Invalid vote attack**: the adversary creates a vote for some invalid plaintext, i.e. a vector that does not encode a candidate selection (e.g., multiple votes for some specific candidate). This attack can be prevented by the soundness of the NIZK proofs, except from the negligible soundness error ϵ. The NIZK verification is done via the voter's ASD.

■ **VSD attack**: the adversary creates a vote which is valid, but corresponds to different selection than the one that the voter intended. A Benaloh audit at the **Cast** ceremony step can detect such an attack with certainty, as the randomness provided by the VSD perfectly binds the plaintext with the audited ElGamal ciphertext.

■ **Replacement attack:** the adversary deletes/inserts an honest vote from/to the BB, or replaces it with some other vote of its choice, after voting has ended. Assuming no hash collisions, any such modification will be detected if the voter chooses to audit the BB via her ASD.

■ **Invalid tally decryption attack:** the adversary provides a decryption which is not the plaintext that the homomorphic tally vector encrypts. The

[7] This requires that $H(\cdot)$ has resistance to second preimage attacks.

NIZK proofs of correct decryption prevent this attack, except for a negligible soundness error ϵ.

Remark 2 (Completeness of the attack list). It can be easily shown that the above list exhausts all possible non-trivial attack strategies against Helios in our threat model. Namely, in an environment with no clash, collision and invalid encryption attacks, the set of votes is in the correct (yet unknown) one-to-one correspondence with the set of voters, and all votes reflect a valid candidate selection of the unique corresponding voter. As a result, a suitably designed vote extractor will decrypt (in super-polynomial time) and output the actual votes from the non-honest-and-successful voters, up to permutation. Consequently, if no honest vote has been modified during and after voting, and the homomorphic tally of the votes is correctly computed and decrypted, then the perfect binding of the plaintexts and ciphertexts of ElGamal implies that the decryption of the tally matches the *intended election result*.

4.2 Attacking the verifiability of Helios e-voting ceremony

As explained in the previous subsection, any attempt of collision, invalid vote and invalid tally decryption attacks has negligible probability of success for the adversary due to the collision resistance of the hash function and the soundness of the ZK proofs. Therefore, in a setting where no clash attacks are possible, the adversary's chances to break verifiability rely on combinations of VSD and Replacement attacks. The probability of these attacks being detected depends on the voter transducer distribution $\mathbf{D}$ which expresses their auditing behaviour during and after voting. In the following theorem, we prove that the verifiability of Helios is susceptible to VSD or/and Replacement attacks, when the voters sample from a class of assailable voter transducer distributions.

Theorem 1 (*Vulnerability of Helios ceremony*). *Assume an election run of Helios with n voters, m candidates and k trustees. Let $q, \delta, \theta, \phi \in \mathbb{N}$, where $0 < \theta, \phi \leq n$ and q is the maximum number of Benaloh audits. Let $\mathbf{D}$ be a (universal) voter transducer distribution s.t. for some $\kappa_1, \kappa_2, \kappa_3, \mu_1, \mu_2 \in [0, 1)$ at least one of the two following conditions holds:*

(i). *There is an $i^* \in \{0, \ldots, q\}$ that determines "*vulnerable VSD auditing behaviour*". Namely, (i.a) the probability that a voter executes at least i^* Benaloh audits is $1 - \kappa_1$ AND (i.b) the probability that a voter, given that she has executed at least i^* Benaloh audits, will cast her vote after exactly i^* Benaloh audits is $1 - \kappa_2$ AND (i.c) the probability that a voter, given that she will execute exactly i^* Benaloh audits, will not complain in case of unsuccessful audit is κ_3.*

(ii). *There is a subset $\mathcal{J}^* \subseteq \{0, \ldots, q\}$ that determines "*vulnerable BB auditing behaviour*". Namely, (ii.a) the probability that a voter executes j Benaloh audits for some $j \in \mathcal{J}^*$ is $1 - \mu_1$ AND (ii.b) for every $j \in \mathcal{J}^*$, the probability that a voter, given she has executed j Benaloh audits, will not audit the BB is at least $1 - \mu_2$.*

Let $\boldsymbol{\mathcal{D}} = \langle \mathbf{D}, \ldots, \mathbf{D}, \mathbf{D}^{T_1}, \ldots, \mathbf{D}^{T_k}, \mathbf{D}^{\mathsf{CD}} \rangle$ be a transducer distribution vector where $\mathbf{D}^{T_i} = \mathbf{D}^{T}_{p_i}$, $i = 1, \ldots, k$, is the p_i-biased coin-flip trustee transducer distribution in Eq. (1) for arbitrary $p_i \in [0,1]$ and $\mathbf{D}^{\mathsf{CD}}$ is an arbitrary CD transducer distribution. Then, there is a PPT adversary $\mathcal{A}$ that wins the E2E verifiability ceremony game $G^{\mathcal{A},\mathcal{E},\boldsymbol{\mathcal{D}},\delta,\theta,\phi}_{\mathrm{E2E}}(1^\lambda, n, m, k)$ in Figure 2 for any vote extractor $\mathcal{E}$, any $\varDelta \in [0,1)$ as follows:

- ▶ *under condition (i), provided the parameters δ, θ, ϕ satisfy:*

$$\delta \leq \big(1-\varDelta\big)^2(1-\kappa_2)(1-\kappa_1)n$$
$$\theta \leq n - (1+\varDelta)(\kappa_2 + \varDelta - \varDelta\kappa_2)(1-\kappa_1)n$$
$$\phi \geq (1+\varDelta)^2\kappa_3(\kappa_2 + \varDelta - \varDelta\kappa_2)(1-\kappa_1)n$$

  with probability of success at least $\boxed{1 - 5e^{-\kappa_3\beta_2\beta_1\frac{\varDelta^2}{3}}}$

  where $\beta_1 = (1-\varDelta)(1-\kappa_1)n$ and $\beta_2 = (\kappa_2 - \varDelta + \varDelta\kappa_2)(1-\kappa_2)$.

- ▶ *under condition (ii), provided the parameter δ satisfies $\delta \leq (1-\varDelta)(1-\mu_1)n$ with probability of success at least* $\boxed{(1 - e^{-(1-\mu_1)n\frac{\varDelta^2}{2}})(1-\mu_2)^\delta}$.

Proof. We prove the Theorem in the full version [36, Theorem 1].

4.3 End-to-end verifiability theorem Helios e-voting ceremony

In this subsection, we prove the E2E verifiability of Helios e-voting ceremony in the RO model, when the voter transducer distribution satisfies two conditions. As we will explain at length in the next subsection, these conditions are logically complementary to the ones stated in Theorem 1, as long as the complaining behaviour of the voters is balanced (i.e. the voters have 1/2 probability of complaining in case of unsuccessful termination).

Theorem 2 (***Verifiability of Helios ceremony***)**.** *Assume an election run of Helios with n voters, m candidates and k trustees. Assume that the hash function $H(\cdot)$ considered in Section 3 is a random oracle. Let $q, \delta, \theta, \phi \in \mathbb{N}$, where $0 < \theta, \phi \leq n$ and q is the maximum number of Benaloh audits. Let $\mathbf{D}$ be a (universal) transducer distribution and some $\kappa_1, \kappa_2, \kappa_3, \mu_1, \mu_2 \in [0,1)$ s.t. the two following conditions hold:*

*(i) There is an $i^* \in \{0, \ldots, q+1\}$ that guarantees "*resistance against VSD attacks*". Namely, (i.a) the probability that a voter executes at least i^* Benaloh audits is κ_1 and (i.b) for every $i \in \{0, \ldots, q\}$, if $i < i^*$, then the probability that a voter, given that she will execute at least i Benaloh audits, will cast her vote after exactly i Benaloh audits, is no more than κ_2 AND the probability that a voter, given that she will execute exactly i Benaloh audits, will complain in case of unsuccessful audit is at least $1 - \kappa_3$.*

(ii) There is a subset $\mathcal{J}^* \subseteq \{0,\ldots,q\}$ *that guarantees* "resistance against Replacement attacks"*. Namely, (ii.a) the probability that a voter executes* j *Benaloh audits for some* $j \in \mathcal{J}^*$ *is* $1-\mu_1$ *AND (ii.b) for every* $j \in \mathcal{J}^*$*, the probability that a voter, given she has executed* j *Benaloh audits, will audit the BB is at least* $1-\mu_2$*.*

Let $\boldsymbol{\mathcal{D}} = \big\langle \mathbf{D},\ldots,\mathbf{D},\mathbf{D}^{T_1},\ldots,\mathbf{D}^{T_k},\mathbf{D}^{\mathsf{CD}}\big\rangle$ *be a transducer distribution vector where* $\mathbf{D}^{T_i} = \mathbf{D}^{T}_{p_i}$*,* $i=1,\ldots,k$*, is the* p_i*-biased coin-flip trustee transducer distribution in Eq. (1) for arbitrary* $p_i \in [0,1]$ *and* $\mathbf{D}^{\mathsf{CD}}$ *is an arbitrary* CD *transducer distribution. Then, for any* $\varDelta \in [0,1)$ *for any* δ,θ*, and under the constraint*

$$\phi \leq (1-\varDelta)(1-\kappa_3)\Big(\frac{1}{(1+\varDelta)\kappa_2}-1\Big)\Big(\frac{\delta}{2}-(1+\varDelta)\kappa_1 n\Big) \;,$$

the Helios e-voting ceremony achieves E2E verifiability for $\boldsymbol{\mathcal{D}}$*, a number of* θ *honest successful voters, a number of* ϕ *honest complaining voters and tally deviation* δ *with error*

$$e^{-\min\left\{\kappa_1 n\frac{\varDelta^2}{3}\;,\;\mu_1 n\frac{\varDelta^2}{3}\;,\;\gamma(\frac{\delta}{2}-(1+\varDelta)\kappa_1 n)\frac{\varDelta^2}{3}\;,\;\mathsf{ln}\left(\frac{1}{\mu_2}\right)(\frac{\delta}{2}-(1+\varDelta)\mu_1 n)\right\}}+$$
$$+\,(\mu_1+\mu_2-\mu_1\mu_2)^{\theta}+\mathsf{negl}(\lambda)\;,$$

where $\gamma = \min\left\{\kappa_2\;,\;\frac{3}{2}(1-\kappa_3)\big(\frac{1}{(1+\varDelta)\kappa_2}-1\big)\right\}\;.$

Proof. We prove the Theorem in the full version [36, Theorem 2].

4.4 Illustrating Theorems 1 and 2

In order to provide intuition, we provide examples of assailable and resistant voter transducer distributions, in the full version [36, Subsections 4.2.1 & 4.3.1]. For every case, we illustrate our analysis via comprehensive graphs. Among other remarks, we study the role of $\varDelta$ as trade off factor between (a) optimising the bounds stated in Theorems 1 and 2, and (b) the corresponding "effectiveness zone" determined by the parameters δ,θ,ϕ (normalised by the electorate size n).

4.5 On the tightness of the conditions of Theorems 1 and 2

The conditions stated in Theorems 1 and 2 determine two classes of voter transducer distributions that correspond to vulnerable and insusceptible settings, respectively. We observe that weakening the condition (i) of Theorem 1 (resp. (i) of Theorem 2) cannot imply vulnerability (resp. security). Namely, in condition (i) of Theorem 1, if one of (1.a),(1.b) or (1.c) does not hold, then the adversary cannot be certain that it will achieve a sufficiently large deviation from VSD attacks without increasing rapidly the number of complaints. On the other hand, if condition (i.a) of Theorem 2 does not hold, then E2E verifiability cannot be preserved when (1.b) becomes a disjunction, since a high complaint rate alone is meaningless if the adversary has high success rate of VSD attacks.

Consequently, it is not possible to achieve logical (i.e. probability thresholds are considered either sufficiently ***high*** or sufficiently ***low***) tightness for interesting sets of parameters δ, θ, ϕ only by negating the conditions of each of the two theorems. However, this is possible if we assume that the voter's complaining behaviour is *balanced* by setting $\kappa_3 = 1 - \kappa_3 = 1/2$. Namely, the voters flip coins in order to decide whether they will complain in case of unsuccessful termination. Given that $\kappa_3 = 1/2$ is a "neutral" value, we have that

Condition (i) of Theorem 1 does not hold, iff condition (i) of Theorem 2 holds.
Condition (ii) of Theorem 1 does not hold, iff condition (ii) of Theorem 2 holds.

The above statement is argued in detail in the full version [36, Subsection 4.4].

5 Voter Privacy of Helios e-Voting Ceremony

In this section, we prove the voter privacy of the Helios e-voting ceremony. The proof is carried out via a reduction. Namely, we show that unless no honest trustee verifies the correct posting of their public data, if there exists a PPT adversary $\mathcal{A}$ that wins the voter privacy/PCR game for Helios with non-negligible distinguishing advantage, then there exists a PPT adversary $\mathcal{B}$ that breaks the IND-CPA security of the ElGamal encryption scheme with blackbox access to $\mathcal{A}$. Throughout the proof, we view $H(\cdot)$ as a RO.

Theorem 3 (***Voter Privacy of Helios ceremony***). *Assume an election run of Helios with n voters, m candidates and k trustees. Assume that the hash function $H(\cdot)$ considered in Section 3 is a random oracle and the underlying ElGamal encryption scheme is IND-CPA secure. Let $t, v \in \mathbb{N}$, where $t < k$ and $v < n$.*

Let $\boldsymbol{\mathcal{D}} = \big\langle \mathbf{D}, \ldots, \mathbf{D}, \mathbf{D}^{T_1}, \ldots, \mathbf{D}^{T_k}, \mathbf{D}^{\mathsf{CD}} \big\rangle$ be a transducer distribution vector where $\mathbf{D}^{T_i} = \mathbf{D}^{T}_{p_i}$, $i = 1, \ldots, k$, is the p_i-biased coin-flip trustee transducer distribution in Eq. (1) for arbitrary $p_i \in [0,1]$ and $\mathbf{D}^{\mathsf{CD}}$ is an arbitrary CD transducer distribution.

Assume that $p_1, \ldots, p_k$ are sorted in increasing order as $p_{i_1} \leq \cdots \leq p_{i_k}$. Then, Helios e-voting ceremony achieves voter privacy for $\boldsymbol{\mathcal{D}}$, at most t corrupted trustees and v corrupted voters with error

$$\frac{1}{2} \cdot \prod_{x=1}^{k-t} (1 - p_{i_x}) + \mathsf{negl}(\lambda) \ .$$

Proof. We prove the Theorem in the full version [36, Theorem 3].

6 Evaluating the E2E verifiability of an e-voting ceremony

In this section, we evaluate our results for the E2E verifiability of Helios, by instantiating the bounds in Theorems 1 and 2 for various voter transducer distributions. Our evaluations are separated into two categories: (i) evaluations that are based on actual human data that derive from elections using Helios and (ii) evaluations that are based on simulated data for various sets of parameters.

6.1 Evaluations based on human data.

Our human data are sampled from two independent surveys: the first sample is from the member elections of the Board of Directors of the International Association for Cryptographic Research (IACR); the second is a non-binding poll among the students of the Department of Informatics and Telecommunications (DI&T) of the University of Athens.

Due to space limitations, we present at length the methodology for both our surveys in the full version [36, Subsection 6.1.1]. Here, we provide the computed parameters $\kappa_1, \kappa_2, \kappa_3, \mu_1, \mu_2$ of Theorem 1 for the IACR and the DI&T surveys in Table 1. For both surveys, no complaints or audit failures were reported. Hence, due to lack of data, we choose a "neutral" value for κ_3 equal to 0.5 (see also Subsection 4.5). Note that our analysis will hold for any other value of κ_3 *not close to* 0. The case of $\kappa_3 = 0$, i.e., when the voter always complains to the authority when a Benaloh audit goes wrong, would make VSD attacks unattractive in the case that ϕ is small and would suggest that the attacker will opt for Replacement attacks, if such attacks are feasible.

The parameters $\kappa_1, \kappa_2, \kappa_3, \mu_1, \mu_2$ used in Theorem 1 express the vulnerability of Helios ceremony against verifiability attacks w.r.t. a specific voter transducer distribution. Namely, parameters $\kappa_1, \kappa_3, \mu_1$ determine the *size* of the subsets of vulnerable voters, while κ_2, μ_2 can be seen as measures of the *quality* of the VSD and Replacement attacks.

Survey	i^*	$\mathcal{J}^*$	**Parameters**				
			κ_1	κ_2	κ_3	μ_1	μ_2
IACR elections	0	$\{0\}$	0	0.315	0.5	0.315	0.084
DI&T poll	1	–	0.408	0.069	0.5	–	–

Table 1: Instantiated parameters $\kappa_1, \kappa_2, \kappa_3, \mu_1, \mu_2$ of Theorem 1 for the IACR and the DI&T surveys.

Analysis of the IACR survey

From the first row of Table 1, we read that $\mu_2 = 0.084$ which is a very small value as opposed to $\kappa_2 = 0.315$. Thus, we expect that elections where the electorate follows the voter transducer distribution of IACR elections are much more vulnerable to Replacement attacks rather than VSD attacks. Indeed, this is consistent with the analysis that we describe below.

We computed the percentage of *tally deviation/No. of voters* that the adversary can achieve when the success probability is lower bounded by 25%, 10%, 5% and 1% for various electorate scales. Specifically, we observed that the success probability bounds stated in Theorem 1 express more accurately the effectiveness of the adversarial strategy for (i) medium to large scale elections when the adversary attacks via the VSD and (ii) for small to medium scale elections when the adversary attacks via the BB. As a consequence, we present our analysis for $n = 100, 500, 1000, 2500$ and 5000 voters w.r.t. Replacement attack effectiveness and for $n = 5000, 10000$ and 50000 voters w.r.t. VSD attack effectiveness.

The data in Table 2 illustrate the power of Replacement attacks against compact bodies of voters (e.g. organizations, unions, board elections, etc.) where BB auditing is rare. We can see that in the order of hundreds, more than 5% of the votes could be swapped with significant probability of no detection. This power deteriorates rapidly as we enter the order of thousands, yet the election result could still be undermined, as deviation between 1%-2%, is possible, without the risk of *any* complaint due to unsuccessful engagement in the **Cast** ceremony (i.e. $\theta = n, \phi = 0$). Therefore, even in a setting of high complaint rate (κ_3 is close to 0), the adversary may turn into a Replacement attack strategy and still be able to alter radically the election result, as marginal differences are common in all types of elections. We stress that from published data we are aware of, there have been elections for the IACR board where the votes for winning candidates were closer than 3% to the votes of candidates that lost in the election. Therefore, if the voter distribution had been as the one derived by Table 1, and 500 members had voted, the result could have been overturned with success probability 25% even if a single complaint was considered a “stop election event” (since $\phi = 0$).

To provide more context, in Table 3, we provide the cutoff between elected and non-elected candidates for the last 11 years of IACR elections for the Board of Directors, followed by the exact success probability of a hypothetical Replacement attack strategy to overturn the election result given the actual number of cast ballots per year. We observe that the attacker success probability for many of the elections is considerable (2011, 2014, 2015, 2016), or even unacceptable (2006, 2008, 2009, 2013), at least in our estimation.

Voters	**Success probability %**			
	≥ 25	≥ 10	≥ 5	≥ 1
100	15.92	26.4	34.42	51.42
500	3.18	5.28	6.87	10.56
1000	1.59	2.64	3.42	5.28
2500	0.636	1.05	1.37	2.11
5000	0.31	0.52	0.68	1.05

Table 2: Percentage of *tally deviation/No. of voters* achieved in elections under Replacement attack strategies against electorates following the voter transducer distribution of IACR elections. The attack succeeds even when $\theta = n$ and $\phi = 0$.

Year	Participants	Cutoff %	Success probability %
2016	522	6.13	6.03
2015	437	6.87	7.35
2014	575	5.57	6.17
2013	637	2.99	19.14
2012	518	11.59	0.5
2011	621	4.03	11.35
2010	475	8.64	2.82
2009	325	4.93	24.8
2008	312	0.33	91.66
2007	–	–	–
2006	324	4.33	29.57

Table 3: Success probability of a hypothetical Replacement attack strategy against the IACR elections for the Board of Directors per election year. The success probability is computed given the number of participants and the cutoff between the last elected director and the first candidate that was not elected. The dashed line denotes the actual start of Helios use for IACR elections. Regarding the year 2007, no data were recorded in `https://www.iacr.org/elections/`.

On the other hand, the effectiveness of a VSD attack strategy against an election that follows the voter distribution in IACR elections would not have a great impact unless an unnatural number of complaints could be tolerated. Indeed, from our evaluation, it appears even for the scale of 5000, 10000 and 50000 that voters, the rate of complaints that is ignored must be close to 24%, 21% and 17% respectively, which is rather unacceptable in a real world setting. Such number of complaints would most definitely lead to a stop election event.

We conclude that the IACR voter behaviour is susceptible to Replacement attacks with significant probability of success but not VSD attacks unless there is high tolerance in voter complaints.

Analysis of the DI&T poll

Due to space limitations, we present the analysis of the DI&T poll in the full version [36, Subsection 6.1.3]. In few words, from the second row of Table 1, we read that $\kappa_2 = 0.069$ which is a very small value leading to significant VSD vulnerability.

6.2 Evaluations based on simulated data

Our human data analysis is obtained by real bodies of voters that have an imperfect voting behaviour. To understand what would be the security level of a Helios e-voting ceremony when executed by an "ideally trained" electorate, we evaluated the security of simulated elections. Namely, we computed the *detection probability* that Theorem 2 can guarantee defined as $(1 - \epsilon) \cdot 100\%$, where ϵ is the error stated in Theorem 2.

The voter distributions we considered were chosen from $\{\mathbf{D}_{p,q}\}_{p \in [0,1], q \in \mathbb{N}}$, a collection of distributions defined as follows: when behaving according to distribution $\mathbf{D}_{p,q}$, the voter flips a coin b with bias p to perform Benaloh audits when

$b = 1$, up to a maximum number of q audits. In any case of termination, she flips a coin b' with bias p to perform BB audit when $b' = 1$.

By the above description, we select as VSD resistance index $i^* = q$ and BB resistance set $\mathcal{J}^* = \{0, \ldots, q\}$. For these $i^*, \mathcal{J}^*$ we compute the parameters $\kappa_1 = \mu_1 = p^q$ and $\kappa_2 = \mu_2 = 1-p$, while we also set κ_3 to the balanced parameter $1/2$. Intuitively, this type of voter behaviour should result in a sufficient level of resistance against of VSD and Replacement attacks, if the values $1 - p$ and p^q are small enough.

As an instance of our search, we present our findings for $n = 250000$ voters for distributions $\mathbf{D}_{p,q}$, where $p = 0.25, 0.5, 0.75$ and $q = 3, 5, 8, 10$ in Table 4. In particular, we present the deviation cutoff that can be guaranteed with detection probability 90%, 99% and 99.9%, in an election where the ratio of complaining voters is no more than 0.1%. For a more detailed description of our methodology, we refer the reader to the full version [36, Section 6.2].

Distribution	**Detection Probability**		
	90%	99%	99.9%
	$\delta/n\%$	$\delta/n\%$	$\delta/n\%$
$\mathbf{D}_{0.25,3}$	6.1	7.31	8.71
$\mathbf{D}_{0.25,5}$	3.63	7.05	17.6
$\mathbf{D}_{0.5,3}$	28.99	30.17	31.12
$\mathbf{D}_{0.5,5}$	7.7	8.06	8.34
$\mathbf{D}_{0.5,8}$	1.5	1.62	1.69
$\mathbf{D}_{0.5,10}$	0.9	1.09	1.28
$\mathbf{D}_{0.75,8}$	26.2	27.71	28.35
$\mathbf{D}_{0.75,10}$	14.79	15.76	16.47

Table 4: Security w.r.t. detection probability $90\%, 99\%$ and $99,9\%$ of $\delta/n \cdot \% :=$*(tally deviation)/(No. of voters)* percentage for elections with $n = 250000$ voters and $\phi/n \leq 0,1\%$ for distributions $\mathbf{D}_{p,q}$, where $p = 0.25, 0.5, 0.75$ and $q = 3, 5, 8, 10$. The detection probability is defined as $(1 - \epsilon) \cdot 100\%$, where ϵ is the error stated in Theorem 2.

By reading the data in Table 4, we observe that the security guarantee is optimised for the fair coin flipping case $p = 0.5$. Nevertheless, even for this case, acceptable levels of security (e.g., (tally deviation)/(No. of voters) $\leq 3\%$ or error probability $\leq 1\%$) can be achieved only for relatively high values of $q \geq 8$. Besides, recall that these values are reached in the setting where a very small rate ($\leq$0.1%) of complaining voters is allowed. As a result, the auditing behaviour of the voters and the complaint tolerance must be almost ideal in order for a high level of security to be achieved.

7 Conclusion

We have introduced the concept of ceremonies to the setting of e-voting systems. Our framework enables the modelling of all human participants to an e-voting protocol as nodes in the protocol execution. Human nodes are modelled as random variables over a set of admissible protocol behaviours which are described by (finite state) transducers. Our analysis enables the exploration of feasibility and infeasibility results regarding the verifiability of the Helios system (suitably modified to be a ceremony) conditioning on general classes of possible voter behaviours. The results from our characterization are essentially tight in the sense that behaviours excluded from our security theorem are too weak/predictable to offer a reasonable level of verifiability.

Our results are only an initial step in the direction of fully incorporating human behavior and interaction within cryptographic modeling. There are many ways to extend the way human nodes are affected by the environment (e.g., taking into account the timing of other nodes) or being manipulated to perform the protocol steps in a wrong order (cf. [32]). Still, even with our limited analysis, we demonstrated that current election procedures, even those performed by cryptographers, are extremely prone to manipulation. Our positive results, albeit also modest, show that there exist behaviors that if uniformly regimented they can provide a reasonable level of e-voting security. Designing e-voting protocols for which this set of behaviors can be efficiently learnable by humans is a further interesting direction motivated by our work.

References

1. Ben Adida. Helios: Web-based open-audit voting. In *USENIX Security Symposium*, 2008.
2. Joël Alwen, Rafail Ostrovsky, Hong-Sheng Zhou, and Vassilis Zikas. Incoercible multi-party computation and universally composable receipt-free voting. In *CRYPTO*, 2015.
3. Giampaolo Bella and Lizzie Coles-Kemp. Layered analysis of security ceremonies. In *IFIP SEC*, pages 273–286, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
4. Josh Benaloh. Verifiable secret-ballot elections. Yale University Ph.D. Thesis YALEU/DCS/TR-561. New Haven, CT, 1987.
5. Josh Benaloh. Simple verifiable elections. In Dan S. Wallach and Ronald L. Rivest, editors, *EVT*. USENIX Association, 2006.
6. Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC*, 1994.
7. David Bernhard, Véronique Cortier, Olivier Pereira, Ben Smyth, and Bogdan Warinschi. Adapting Helios for provable ballot privacy. In *ESORICS*, 2011.
8. David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. In *ASIACRYPT*, 2012.
9. Marcelo Carlomagno Carlos, Jean Everson Martina, Geraint Price, and Ricardo Felipe Custódio. An updated threat model for security ceremonies. In *Proceedings of ACM SAC*, pages 1836–1843. ACM, 2013.
10. Marcelo Carlomagno Carlos and Geraint Price. Understanding the weaknesses of human-protocol interaction. In *Financial Cryptography and Data Security*, 2012.
11. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

12. David Chaum. Surevote: Technical overview. In *Proceedings of the Workshop on Trustworthy Elections*, WOTE, Aug. 2001.
13. David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, 2004.
14. David Chaum, Aleks Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan Sherman, and Poorvi Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *Security & Privacy, IEEE*, 6(3):40–46, 2008.
15. David Chaum and Torben P. Pedersen. Wallet databases with observers. In *CRYPTO*, 1992.
16. Benoît Chevallier-Mames, Pierre-Alain Fouque, David Pointcheval, Julien Stern, and Jacques Traoré. On some incompatible properties of voting schemes. In *Towards Trustworthy Elections*, 2010.
17. Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, 1985.
18. United States Election Assistance Commission. Voluntary voting systems guidelines, 2005.
19. Véronique Cortier and Ben Smyth. Attacking and fixing Helios: An analysis of ballot secrecy. *ePrint Archive*, 2010:625, 2010.
20. Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung. Multi-autority secret-ballot elections with linear work. In *EUROCRYPT*, 1996.
21. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *EUROCRYPT*, 1997.
22. Ivan Damgård, Jens Groth, and Gorm Salomonsen. The theory and implementation of an electronic voting system. In Dimitris Gritzalis, editor, *Secure Electronic Voting*, volume 7 of *Advances in Information Security*, pages 77–98. Springer, 2003.
23. Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.
24. Alex Delis, Konstantina Gavatha, Aggelos Kiayias, Charalampos Koutalakis, Elias Nikolakopoulos, Mema Roussopoulou, Georgios Sotirellis, Panos Stathopoulos, Lampros Paschos, Pavlos Vasilopoulos, Thomas Zacharias, and Bingsheng Zhang. Pressing the button for European elections 2014: Public attitudes towards verifiable e-voting in Greece. In *EVOTE*, 2014.
25. Carl M. Ellison. Ceremony design and analysis. *IACR Cryptology ePrint Archive*, 2007:399, 2007.
26. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.
27. Jens Groth. Evaluating security of voting schemes in the universal composability framework. In Markus Jakobsson, Moti Yung, and Jianying Zhou, editors, *ACNS*, volume 3089 of *Lecture Notes in Computer Science*, pages 46–60. Springer, 2004.
28. E. Hatunic-Webster, F. Mtenzi, and B. O'Shea. Model for analysing anti-phishing authentication ceremonies. In *ICITST*, pages 144–150, 2014.
29. Christian Johansen and Audun Jøsang. Probabilistic modelling of humans in security ceremonies. In *SETOP*, pages 277–292. Springer International Publishing, 2015.
30. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. *IACR Cryptology ePrint Archive*, 2002:165, 2002.
31. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In Vijay Atluri, Sabrina De Capitani di Vimercati, and Roger Dingledine, editors, *WPES*, pages 61–70. ACM, 2005.

32. Chris Karlof, Naveen Sastry, and David Wagner. Cryptographic voting protocols: A systems perspective. In *USENIX*, 2005.
33. Chris Karlof, J. D. Tygar, and David Wagner. Conditioned-safe ceremonies and a user study of an application to web authentication. In *SOUPS*, ACM International Conference Proceeding Series. ACM, 2009.
34. Chris Karlof, J. Doug Tygar, and David Wagner. Conditioned-safe ceremonies and a user study of an application to web authentication. In *NDSS*, 2009.
35. Aggelos Kiayias, Michael Korman, and David Walluck. An internet voting system supporting user privacy. In *ACSAC*, pages 165–174. IEEE Computer Society, 2006.
36. Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. Ceremonies for end-to-end verifiable elections. *IACR Cryptology ePrint Archive*, 2015:1166, 2015.
37. Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. End-to-end verifiable elections in the standard model. In *EUROCRYPT*, 2015.
38. Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. On the necessity of auditing for election privacy in e-voting systems. In *E-Democracy*, 2015.
39. Steve Kremer, Mark Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In *ESORICS*, pages 389–404, 2010.
40. Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Accountability: Definition and relationship to verifiability. *IACR Cryptology ePrint Archive*, 2010:236, 2010.
41. Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Verifiability, privacy, and coercion-resistance: New insights from a case study. In *IEEE Symposium on Security and Privacy*, pages 538–553. IEEE Computer Society, 2011.
42. Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Clash attacks on the verifiability of e-voting systems. In *IEEE Symposium on Security and Privacy*, pages 395–409. IEEE Computer Society, 2012.
43. Jean Everson Martina, Eduardo dos Santos, Marcelo Carlomagno Carlos, Geraint Price, and Ricardo Felipe Custódio. An adaptive threat model for security ceremonies. *International Journal of Information Security*, 14(2):103–121, 2015.
44. Tal Moran and Moni Naor. Receipt-free universally-verifiable voting with everlasting privacy. In *CRYPTO*, pages 373–392, 2006.
45. C. Andrew Neff. Practical high certainty intent verification for encrypted votes. Votehere, Inc. whitepaper, 2004.
46. Maina M. Olembo, Steffen Bartsch, and Melanie Volkamer. Mental models of verifiability in voting. In *VOTE-ID*, pages 142–155, 2013.
47. Kenneth Radke, Colin Boyd, Juan Gonzalez Nieto, and Margot Brereton. Ceremony analysis: Strengths and weaknesses. In *IFIP SEC*, pages 104–115, 2011.
48. Kenneth Radke, Colin Boyd, Juan Manuel González Nieto, and Margot Brereton. Ceremony analysis: Strengths and weaknesses. In *IFIP*, 2011.
49. Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In *EUROCRYPT*, 1995.
50. Ben Smyth, Steven Frink, and Michael R. Clarkson. Computational election verifiability: Definitions and an analysis of Helios and JCJ. Technical report.
51. Georgios Tsoukalas, Kostas Papadimitriou, Panos Louridas, and Panayiotis Tsanakas. From Helios to Zeus. In *EVT/WOTE*, 2013.
52. Filip Zagórski, Richard Carback, David Chaum, Jeremy Clark, Aleksander Essex, and Poorvi L. Vora. Remotegrity: Design and use of an end-to-end verifiable remote voting system. In *ACNS*, 2013.