# The DEMOS family of e-voting systems: End-to-end verifiable elections in the standard model

Thomas Zacharias**

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications

thzacharias@di.uoa.gr

**Abstract.** This PhD thesis introduces the DEMOS-A and DEMOS-2 e-voting systems that achieve *end-to-end verifiability in the standard model* for the first time. End-to-end verifiability in the standard model denotes that verification is executed without without putting trust in any administration authority and without assuming any trusted randomness setting. Prior to this thesis, all top-tier e-voting systems (e.g. SureVote, JCJ, Prêt à Voter, Helios, Scantegrity, etc.) assumed honesty of the voting clients, the random oracle model, or the existence a randomness beacon to achieve end-to-end verifiability.
In the core of DEMOS-A and DEMOS-2, is a novel mechanism that extracts the randomness required for verification from the *entropy generated by the voters*, when they engage in the voting phase. This entropy is *internal* with respect to the election environment, therefore the need for trusting an outer source of randomness is removed.
The security analysis is performed under a novel cryptographic framework that constitutes an additional contribution of this thesis. The end-to-end verifiability theorems for DEMOS-A and DEMOS-2 reveal that the security level is in high correlation with the auditing behaviour of the electorate. Motivated by this finding, this thesis extends the framework by modelling e-voting systems as *ceremonies*, inspired by the work of Ellison in 2007. As a case study of an e-voting ceremony, this thesis investigates the security of the well-known Helios e-voting system.

## 1 Introduction

Political activity in a modern democratic state comprises compositions of individual democratic procedures. At a high level, a democratic procedure consists of three well-defined concepts.

1. An *electorate* formed by the people legitimate to vote,
2. A *voting system*, which serves as means to record and evaluate the electorate's will, and

** *Dissertation Advisor*: Aggelos Kiayias, Associate Professor

3. A *verdict*, which stems from the consensus according to the evaluated electorate's will.

A reliable voting system must incorporate mechanisms for optimising accessibility of the electorate and guaranteeing integrity of the election result while protecting the voters' secrecy. If it does so, then it paves the way for building a healthy democratic society. On the other hand, due to their crucial role in democracy, voting systems have often been top priority targets for attackers that wish to tamper the election result and/or coerce voters to vote against their intention. Voting systems that allow people to sell their votes, or lack verification procedures that convince an auditor of the validity of the election result with minimum doubt, undermine the foundations of any democratic state they are deployed.

*e-Voting in democratic procedures*

In an e-voting system, election preparation, vote collection and/or tally is executed by electronic devices, partially or fully managed by human authorities. The motivation for introducing e-voting was originally three-fold; (i) facilitating the participation of social groups with considerable physical barriers, (ii) reduction of election cost, and (iii) acceleration of the election preparation, vote casting and tally phase. E-voting emerged in the 60s via punch-card systems, followed by systems based on either optical scan voting, ballot encryption, or vote-code typing. By today, e-voting systems have been used in several countries either in pilot executions (Australia, England, Ireland, Italy, Norway) or binding elections at a municipality or national level (Belgium, Brazil, Canada, Estonia, India, the Netherlands, Switzerland, USA). Nonetheless, they have been subject to often trenchant criticism, mainly due to the disquiet about potential security threats caused by the amount of power now transferred to the machines.

Based on their infrastructure, e-voting systems are classified into (i) *On-site e-voting systems*, where the election is executed in polling stations, and supervision by human authorities is similar to traditional elections, and (ii) *Remote e-voting (i-voting) systems*, where the voters submit their votes using devices (PCs, notebooks, tablets, smartphones) that have internet access.

*End-to-end verifiability and e-voting*

Besides advancing participation and reduction of election cost and time, several state-of-the-art e-voting systems [10,14,37,13,1,2,44,42] support an attractive and highly non-trivial security feature that traditional voting unavoidably misses by its nature. Namely, the voter can verify that her vote was properly cast, recorded and tallied into the election result without relying to the honesty of any of the election administrators. This strong property is named *end-to-end (E2E) verifiability* and is usually interpreted as the ability of the voter to verify that her vote was (i) cast-as-intended, (ii) recorded-as cast, and (iii) tallied-as-recorded.

Before this PhD thesis, E2E verifiability could not be justified with minimum assumptions. Under a strong cryptographic definition, E2E verifiability could provenly hold only assuming the existence of a *trusted randomness setting*

that could be either a function modelled as a random oracle [1,2,42], or some randomness beacon [10,14,37,13,44].

*Objectives and contributions of this thesis*

The main objective this thesis investigates, is the feasibility of E2E verifiability *in the standard model*, which denotes that verification is executed with assuming the existence of a trusted randomness setting. As already mentioned, until the writing of this thesis, E2E verifiability in an all-malicious setting could provenly hold only under certain setup assumption for randomness.

In order to illustrate why previous techniques did not work, we elaborate on the previous statement. By its design, Helios [1]-and other client-side encryption E2E verifiable systems as [20,31,2,42]- requires the voter to utilise a voter supporting device to prepare a ciphertext and after an indeterminate number of trials, the voter will cast the produced ciphertext. The submitted ciphertexts are to be homomorphically tallied and thus they should be accompanied by a proof of proper computation. While such proofs are easy to construct based on e.g., [19], they can be argued either (i) interactively or (ii) using a *non-interactive zero-knowledge (NIZK) proof* [6]. Interaction is insufficient in E2E verifiability setting since a corrupt election authority together with a corrupt voter may cook up a malformed proof that is indistinguishable from a proper one. As a result, the non-interactive approach is mandatory. However, NIZK proofs can be sound only under setup assumptions as a *random oracle* or a *common reference string (CRS)* [26]. If the CRS is setup by the election authority, then, in case it is malicious, it will know and exploit the trapdoor; on the other hand, the voters are not interacting with each other and hence cannot setup the CRS by employing a standard multi-party computation protocol [25,12].

On the other hand, in the case of Remotegrity/Scantegrity [13,44] -and other client-side cryptography E2E verifiable systems as [10,14,37]- the random coins need to be obtained from the randomness beacon in order to prove the result correct. It is easy to verify that the system is insecure in terms of E2E verifiability in case the randomness beacon is biased. As before, the only parties active are the election authority and the voters who cannot implement a randomness beacon that is required in the construction.

As a consequence of the aforementioned technical restrictions , the following question remained open until recently:

**Q1.** *Can the integrity of the election result be proven in the* standard model *i.e. without believing in trusted hardware, random oracles or randomness beacons?*

This PhD thesis answers this question affirmatively by introducing the *DEMOS-A* and *DEMOS-2* e-voting systems that achieve E2E verifiability in the standard model, as long as a *publicly accessible bulletin board* where the election results are posted remains consistent. Furthermore, DEMOS-A and DEMOS-2 preserve privacy given the hardness of a standard cryptographic problem (Decisional

Diffie-Hellman). The core idea for this accomplishment is a novel mechanism for extracting randomness from the entropy injected to the system by the voters' entanglement. This entropy is *internal* with respect to the election environment, a fact that removes the requirement for an external randomness source.

The two systems follow different approaches with respect to their design. In particular, DEMOS-A follows the *code-voting approach*, where the voters obtain ballots that contain independent and random encodings of the election options (typically vote-codes in one-to-one correspondence with the election options). At the voting phase, the voters cast the encodings that correspond to their intended selections in their ballots. Consequently, vote submission becomes a simple procedure which can be run by devices of minimum computational power. However, this flexibility comes with a price of high complexity at the election preparation phase from the election servers side, resulting in important scalability restrictions for DEMOS-A. To resolve this issue, this thesis introduces the DEMOS-2 e-voting system, in the spirit of the *client-side encryption*. Namely, in DEMOS-2, the overhead is distributed to the voting clients, which now must be computationally able to locally encrypt the voters' ballots, hence to perform cryptographic operations. As a result, DEMOS-A and DEMOS-2 have complementary benefits and weaknesses regarding their functionality and security, hence the choice of the most preferable system depends on the given election setting.

The second objective studied in this thesis is the effect of the human factor in the security of an E2E verifiable e-voting system. The security analysis of DEMOS-A provides evidence of a strong correlation between the active participation of honest voters in the auditing procedure and the (parameterised) level of E2E verifiability that can be guaranteed. A natural question follows from this observation:

> **Q2.** *At what extent can human behaviour, even within protocol specification, affect the security of an e-voting system?*

This PhD thesis follows a formal cryptographic direction to deal with this matter. Motivated by the *ceremony framework* introduced by Ellison [22] for the analysis of network protocols, it proposes an extension of standard e-voting security modelling, where human nodes are separated from computer nodes and are formalised as finite state machines (transducers) with limited power, hence incapable of performing cryptographic operations. As a case study of the extended ceremony framework, Helios stands out in terms of the range of possible human behaviour due to (i) the dependence of E2E verifiability on (i.a) the statistics related to the Benaloh audit rate performed by the voters and (i.b) the portion of voters that look up their votes in the bulletin board after election using their ballot trackers and (ii) the dependence of privacy on the trustees auditing the correct uploading of the public key, combined with lack of public key infrastructure (PKI) for support authentication of posted data.

In summary, the contributions of this PhD thesis comprise:

1. The introduction of a robust cryptographic framework for the security analysis of e-voting systems. The said framework captures definitions of E2E verifiability, voter privacy and *passive coercion resistance (PCR)* (often referred as receipt-freeness). The latter property denotes the inability of an e-voting system to allow the voters to prove how they voted or sell their votes, even against an adversary that observes network traffic and requests from the voter the transcript containing their personal view of interaction with the election system. The suggested framework is extended to the ceremony model, suitable for the formal study of human behaviour in an e-voting execution.
2. The presentation of two remote e-voting systems, (i) the vote-coding based DEMOS-A and (ii) the client-side encryption based DEMOS-2 that enrich both major e-voting categories with a member that achieves *E2E verifiability in the standard model* for the first time. The two systems are proven secure under the aforementioned framework and their voter privacy/passive coercion resistance holds assuming the hardness of the extensively studied Decisional Diffie-Hellman problem. These two systems give birth to the *DEMOS family of e-voting systems* sharing the attribute of E2E verifiability in the standard model.
3. A thorough analysis of the Helios e-voting system under the ceremony framework. This analysis is threefold consisting of (i) a rigorous mathematical characterisation of classes of voter behaviours that are assailable or resistant to attacks on verifiability, (ii) an evaluation of the expected E2E verifiability guarantee of Helios based on the previous theoretical context given instantiations of real world Helios applications as well as simulation data, and (iii) a presentation of a standard *man-in-the-middle* attack against Helios's privacy, in cases where election guidelines do not encourage trustees (modelled as human nodes) to verify the correct posting of the election public key in the bulletin board.

*Related work*

Up to the present moment, numerous noticeable e-voting systems have been introduced [8,18,9,23,20,10,28,30,14,31,40,13,1,17,37,24,2,42,44], adding to cryptographic literature novel directions or ameliorating existing techniques. In the following table, we depict the classification of a list of e-voting systems, according to their infrastructure and vote submission method.

| | **Client-side encryption** | **Code-voting** |
|---|---|---|
| **On-site** | [2] | [14,40,13,37] |
| **Remote** | [8,18,9,23,20,28,30,31,1,17,24,42] | [10,44] |

End-to-end verifiability in the sense of cast-as-intended, recorded-as-cast, tallied-as-recorded was an outcome of the works in [11] and [36] that introduced the generation of receipts which could be used for simple voter verification while preserving privacy. Prior definitions referring to the weaker notions of *individual* and *universal* verifiability are found in [8,38,29,32,15]. Rigorous end-to-end verifiability definitions have been proposed in [33] and [41]. Definitions of privacy

and receipt-freeness have been introduced in [18,3,16,21,27,35,34,4,5] under the cryptographic, symbolic and universal composability [7] model.

## 2 Results

In this section, we provide an overview of the components of this PhD thesis that comprise the complete presentation and analysis of DEMOS-A, i.e. the syntax, the end-to-end verifiability and voter privacy/PCR definitions, system's description, and the statement of the security theorems for DEMOS-A. Due to space limitations, we refer the reader interested in the results related to DEMOS-2 and the ceremony framework to [43, Chapter 5] and [43, Chapter 6], respectively.

### 2.1 Preliminaries

We use $\lambda$ as the security parameter and consider three additional parameters; the number of voters $n$, options $m$, and trustees $k$, all of which are thought as polynomial in $\lambda$.

For an e-voting system $\boldsymbol{\mathcal{VS}}$, we fix the set of options $\mathcal{O} = \{\mathsf{opt}_1, ..., \mathsf{opt}_m\}$. We denote by $\mathcal{U} \subseteq 2^{\mathcal{O}}$ the collection of subsets of options that the voters are allowed to choose to vote for (which may include a "blank" option too). The option selection $\mathcal{U}_\ell$ of voter $V_\ell$ 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 $\mathsf{opt}_j$ was chosen in the option selections $\mathcal{U}_1, \ldots, \mathcal{U}_n$. The entities involved in an e-voting system $\boldsymbol{\mathcal{VS}}$ are the following:

- The *election authority* $\mathsf{EA}$ that prepares all the election information.
- The *voters* $\mathcal{V} = \{V_1, \ldots, V_n\}$, possibly equipped with *voting supporting devices (VSDs)*.
- The *vote collector* $\mathsf{VC}$ that realises the digital ballot box functionality.
- The set of *trustees* $\mathcal{T} = \{T_1, \ldots T_k\}$ responsible for computing the tally and announcing the election result.
- A publicly accessible and consistent *bulletin board* $\mathsf{BB}$ where the election result and all audit information is posted.

### 2.2 Security framework

**Definition of end-to-end verifiability.** In order to define E2E verifiability formally, we introduce a suitable notation; given that option selections are elements from a set of $m$ choices, we encode them as $m$-bit strings, where the bit in the $j$-th position is 1 if and only if option $\mathsf{opt}_j$ is selected. Further, we aggregate the election results as the list with the number of votes each option has received. Thus, the **Result** algorithm outputs a vector in $\mathbb{Z}_+^m$, i.e., the range of the election evaluation function $f$.

Then, we use the metric $\mathrm{d}_1$ derived by the $\ell_1$-norm scaled to half, i.e., $\mathrm{d}_1(R, R') = \frac{1}{2} \cdot \sum_{i=1}^{n} |R_i - R'_i|$, where $R_i, R'_i$ is the $i$-th coordinate of $R, R'$ respectively, to measure the success probability of the adversary with respect to the amount of tally deviation $\delta$ and the number of voters that perform audit $\theta$. In addition, we make use of a *vote extractor* algorithm $\mathcal{E}$ (not necessarily running in polynomial-time) that extracts the non-honestly cast votes.

We define the E2E Verifiability game, $G_{\mathrm{E2E}}^{\mathcal{A},\mathcal{E},\delta,\theta}$, between the adversary $\mathcal{A}$ and a challenger $\mathsf{Ch}$ using a vote extractor $\mathcal{E}$. The game takes as input the security parameter, $\lambda$, the number of options, $m$, the number of voters, $n$, and the number of trustees $k$. The game is also is parameterised by $\delta$, which is the deviation amount (according to the metric $\mathrm{d}_1(\cdot,\cdot)$) that the adversary wants to achieve and $\theta$, the minimum number of voters that $\mathcal{A}$ must allow to vote honestly and terminate successfully.

The adversary $\mathcal{A}$ starts by selecting the voter, option, and trustee identities for given parameters $n, m, k$. It also determines the allowed ways to vote as described by the set $\mathcal{U}$. Then, $\mathcal{A}$ fully controls the election by corrupting the $\mathsf{EA}$, the $\mathsf{VC}$, all the trustees $\mathcal{T} = \{T_1, \ldots T_k\}$ and all the VSDs. In addition, it manages the **Cast** protocol executions where it assumes the role of the $\mathsf{VC}$. For each voter, $\mathcal{A}$ may choose to corrupt her or to allow the challenger to play on her behalf. In the second case, $\mathcal{A}$ provides the honest voter with the option selection that will use in the **Cast** protocol. Finally, $\mathcal{A}$ completes the election execution which results to the complete election transcript published in the BB.

The adversary will win the game provided that all $\theta$ honest voters that completed the **Cast** protocol successfully will also audit the result successfully, while either (a) the deviation of the tally is at least $\delta$ or (b) the extractor fails to produce the option selection of the dishonest voters.

**Definition 1.** *Let $\epsilon \in [0,1]$ and $m, n, k, \delta, \theta \in \mathbb{N}$ with $\delta > 0$ and $0 < \theta \leqslant n$. Let $\boldsymbol{\mathcal{VS}}$ be an e-voting system with $m$ options, $n$ voters and $k$ trustees w.r.t. the evaluation election unction $f$. We say that $\boldsymbol{\mathcal{VS}}$ achieves* E2E verifiability *with error $\epsilon$, for a number of at least $\theta$ honest successful voters and tally deviation $\delta$ if there exists a (not necessarily polynomial-time) vote-extractor $\mathcal{E}$ such that for any adversary $\mathcal{A}$*

$$\Pr\left[G_{\mathrm{E2E}}^{\mathcal{A},\mathcal{E},\delta,\theta}(1^{\lambda}, m, n, k) = 1\right] \leqslant \epsilon \,.$$

**Modelling voter privacy/PCR.** The definition of privacy concerns the actions that may be taken by the adversary in order to obtain information about the option selections of the honest voters. We specify the goal of the adversary in a very general way; for an attack to succeed, we ask that there is an election result, for which the adversary is capable of distinguishing how the honest voters have voted, while it has access to (i) the individual audit information that the voters obtained after ballot-casting as well as (ii) a set of protocol views that are consistent with all the honest voters' views in the **Cast** protocol instances they participated and the adversary has monitored.

Observe that any system secure against the aforementioned attack scenario would possess also PCR, i.e., voters cannot prove how they voted by showing

the individual audit information they obtain from the **Cast** protocol or even presenting their view in the **Cast** protocol. Given that in the privacy definition we allow the adversary to observe the view of the voter in the **Cast** protocol, we must allow the voter to be able to "lie" about her view, otherwise an attack could be trivially mounted.

In order to capture the PCR property as described above, we utilise an efficient *view simulator* $\mathcal{S}$ that provides a simulated view of the voter in the **Cast** protocol. Intuitively, $\mathcal{S}$ captures the way the voter can lie about her option selection in the **Cast** protocol in case she is coerced to present her view after she completes the ballot-casting procedure. It is imperative that the simulated view is indistinguishable from the actual view the voter obtains.

### 2.3 The DEMOS-A e-voting system

*Description overview*

In DEMOS-A, each voter may select 1 out of $m$ options and cast her vote using vote-codes listed in her ballot. Each ballot has two functionally equivalent parts (with a complete list of the $m$ options in each part), instructing the voter to pick one of the two parts at random. The cryptographic payload of DEMOS-A consists of lists of the following primitives:

**Additively homomorphic commitments:** where a value $M$ is posted in a committed form, denoted by $\mathsf{Com}(M)$, such that (i) when the opening of $\mathsf{Com}(M)$ denoted by $\tilde{M}$ is posted, then no other value than $M$ can be extracted from $\tilde{M}$ (*binding* property), (ii) $\mathsf{Com}(M)$ reveals no information about $M$ to any computationally bounded adversary (*hiding* property) and (iii) for any two values $M_1, M_2$, it holds that $\mathsf{Com}(M_1) \cdot \mathsf{Com}(M_2) = \mathsf{Com}(M_1 + M_2)$ (*additive homomorphic* property). DEMOS-A utilises ElGamal as a commitment scheme that is (i) perfectly binding and (ii) hiding, assuming that the Decisional Diffie-Hellman problem is hard for the underlying group.

**Zero knowledge proofs:** these are proofs such that (i) if the honest verifier accepts a proof, then she is assured that the statement is true (*soundness* property) and (ii) the proof reveals no other information than the truth of the statement (*zero-knowledge property*). Specifically, DEMOS-A makes uses of *three-move* zero-knowledge proofs, where the interaction is accomplished by a first move from the prover, a second move that is the verifier's challenge (the source of which is the random choice from the voters regarding the part of the ballot they chose), and a third move where the prover responds to the challenge which completes the proof.

Formally, DEMOS-A consists of five protocols/algorithms: **Setup**, **Cast**, **Tally**, **Result**, and **Verify**. We will briefly present them here, omitting many cryptographic details, for simplicity.

In the **Setup** protocol, the EA generates the initialization data for each election entity. More specifically, each randomly generated vote-code points to a cryptographic payload, consisting of additively homomorphic commitments

of the *option-encoding*, where the $i$-th option, $\mathsf{option}_i$, is encoded into $(n+1)^{i-1}$. These commitments are associated with necessary zero-knowledge proofs (prover's first move) that allow the EA to show that each commitment is valid (i.e., it commits to an option encoding) later on, without revealing its actual content. The EA then assigns each ballot with two functionally equivalent parts. When the ballot preparation is finalized, the EA distributes the ballots to the voters.

In the **Cast** protocol, the voter randomly chooses one of the two parts of the ballot to vote by submitting the vote-code corresponding to her intended option. The unused part will be kept for auditing after the election ends.

In the **Tally** protocol, the EA fetches the entire election transcript from the BB and posts additional data on the BB. In this step, the tally result is produced using the homomorphic property by "adding" all the option-encoding commitments associated with the vote-codes cast by the voters and are marked as "$\mathsf{voted}$". Note that, the result is in committed form and requires the corresponding opening to be decoded. Furthermore, the commitments that correspond to the unused parts of voter ballots are also revealed for auditing. Finally, the EA derives the challenge (second move) of the zero-knowledge proofs based on the voters' choices of used ballot parts and completes the zero-knowledge proofs that correspond to the option-encoding commitments marked as "$\mathsf{voted}$", by posting all the respective third moves of the prover.

The **Result** algorithm takes as input the entire BB information, and can be executed by anyone. For instance, if $n = 9$, the *option-encodings* of options $1, 2, 3$ are $1, 10, 100$, respectively. Suppose we got 3 votes for $\mathsf{option}_1$, 5 votes for $\mathsf{option}_3$, the sum of the *option-encodings* is $3 * 1 + 5 * 100 = 503$. By the opening of the homomorphic tally, the **Result** algorithm extracts 503 and decodes it as $(3, 0, 5)$, which represents the corresponding votes for each election option.

The **Verify** algorithm can be executed by voters and any third-party auditors. A third-party auditor is able to verify the validity of all the commitments by checking the completed zero-knowledge proofs. Besides, each voter is allowed to perform "print check" by comparing her private ballot with the information on the BB. As the number of auditing voters increases, the probability of election fraud going undetected diminishes exponentially. For example, even if only 10 voters audit, with each one having $\frac{1}{2}$ probability to detect ballot fraud, the probability of ballot fraud going undetected is only $\frac{1}{2}^{10} = 0.00097$.

*Security of DEMOS-A*

The end-to-end verifiability and voter privacy/PCR that DEMOS-A achieves are formally stated in the following theorems.

**Theorem 1.** *Assume an election run of DEMOS-A with $n$ voters, $m$ candidates and $k$ trustees. Let $q$ be the size of the group for the of the underlying commitment scheme described. Then, DEMOS-A achieves E2E verifiability information theoretically for at least $\theta$ honest successful voters and tally deviation $\delta$ with error*

$$2^{-\delta} + 2^{-\theta + \lceil n/\lfloor \log q \rfloor \rceil (\log\log m + 1)} .$$

**Theorem 2.** *Assume an election run of DEMOS-A with $n$ voters, $m$ candidates and $k$ trustees. Assume there exists a constant $c, 0 < c < 1$ such that for any $2^{\lambda^c}$-time adversary $\mathcal{A}$, the advantage of breaking the hiding property of the commitment scheme is $\mathsf{Adv}_{\mathsf{hide}}(\mathcal{A}) = \mathsf{negl}(\lambda)$. Let $t = \lambda^{c'}$ for any constant $c' < c$. Then, for any constant $m$ and $n, k$ polynomial in the security parameter $\lambda$, DEMOS-A achieves voter privacy/PCR against any adversary that corrupts at most $t$ corrupted voters.*

## 3 Conclusions

The completion of this PhD thesis concludes an extended formal cryptographic argumentation on the boundaries of optimal E2E verifiability and the relation of e-voting security with human auditing behaviour. The introduction of the DEMOS family initialised to the pair of DEMOS-A and DEMOS-2 e-voting systems answers affirmatively to question **Q1** of the introduction, promising election execution where the integrity of the result is proven under the standard model, i.e. without trusting a source of randomness. In addition, the honesty of no election administrator or voting supporting device is required.

As far as studying human behaviour is concerned, this thesis has set the necessary cryptographic background and its mathematically argued results on this matter raise intriguing issues. The security analysis of the widely used Helios e-voting system pointed out its weaknesses, in cases where human verification. Our analysis leads to a debate that, beyond its technical basis, can be viewed from a rather political and philosophical lens; if human behaviour, even within protocol specification, *can affect* the security of an e-voting system, then specifying explicitly the extent of the risks -thus answering question **Q2** of the introduction- becomes a top priority. Can these risks be mitigated by significantly better systems, or do they set a security guarantee upper bound, as price for moving responsibility directly to the voters? In order to ask for end-to-end verifiable security, is people's proper training a prerequisite? Stated abstractly,

*Is political maturity an inevitable trade-off for provenly secure direct democratic procedures?*

The robust ceremony model of this PhD thesis could be the means for translating these questions into strict mathematical language and thus provide a valuable asset for subsequent research.

## References


1. Ben Adida. Helios: Web-based open-audit voting. In *USENIX*, 2008.
2. Josh Benaloh, Michael D. Byrne, Bryce Eakin, Philip T. Kortum, Neal McBurnett, Olivier Pereira, Philip B. Stark, Dan S. Wallach, Gail Fisher, Julian Montoya, Michelle Parker, and Michael Winn. STAR-vote: A secure, transparent, auditable, and reliable voting system. In *EVT/WOTE*, 2013.

3. Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC*, 1994.
4. David Bernhard, Véronique Cortier, Olivier Pereira, Ben Smyth, and Bogdan Warinschi. Adapting helios for provable ballot privacy. In *ESORICS*, 2011.
5. 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.
6. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In Simon [39], pages 103–112.
7. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 136–145. IEEE Computer Society, 2001.
8. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.
9. David Chaum. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In *EUROCRYPT*, 1988.
10. David Chaum. Surevote: Technical overview. In *Proceedings of the Workshop on Trustworthy Elections*, WOTE, 2001.
11. David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, 2004.
12. David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In Simon [39], pages 11–19.
13. David Chaum, Aleksander Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan T. Sherman, and Poorvi L. Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security & Privacy*, 6(3):40–46, 2008.
14. David Chaum, Peter Y. A. Ryan, and Steve A. Schneider. A practical voter-verifiable election scheme. In *ESORICS*, 2005.
15. 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.
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. Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy*, 2008.
18. Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, 1985.
19. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, 1994.
20. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *EUROCRYPT*, 1997.
21. Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.
22. Carl M. Ellison. Ceremony design and analysis. *IACR Cryptology ePrint Archive*, 2007:399, 2007.
23. Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *AUSCRYPT*, 1992.
24. Kristian Gjøsteen. Analysis of an internet voting protocol. *IACR Cryptology ePrint Archive*, 2010:380, 2010.

25. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC*, 1987.
26. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.
27. Jens Groth. Evaluating security of voting schemes in the universal composability framework. In *ACNS'04*, pages 46–60, 2004.
28. Engelbert Hubbers, Bart Jacobs, and Wolter Pieters. RIES - internet voting in action. In *COMPSAC*, 2005.
29. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. *IACR Cryptology ePrint Archive*, 2002:165, 2002.
30. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *WPES*, 2005.
31. Aggelos Kiayias, Michael Korman, and David Walluck. An internet voting system supporting user privacy. In *ACSAC*, 2006.
32. Steve Kremer, Mark Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In *ESORICS*, 2010.
33. Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Accountability: definition and relationship to verifiability. In *CCS*, 2010.
34. Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Verifiability, privacy, and coercion-resistance: New insights from a case study. In *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*, pages 538–553. IEEE Computer Society, 2011.
35. Tal Moran and Moni Naor. Receipt-free universally-verifiable voting with everlasting privacy. In *CRYPTO*.
36. C. Andrew Neff. Practical high certainty intent verification for encrypted votes. Votehere, Inc. whitepaper, 2004.
37. Stefan Popoveniuc and Benjamin Hosp. An introduction to PunchScan. In *WOTE*, 2010.
38. Kazue Sako and Joe Kilian. Receipt-free mix-type voting scheme - A practical solution to the implementation of a voting booth. In *EUROCRYPT*, 1995.
39. Janos Simon, editor. *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*. ACM, 1988.
40. Warren D. Smith. Three voting protocols: Threeballot, vav, and twin. In *USENIX/ACCURATE*, 2007.
41. Ben Smyth, Steven Frink, and Michael R. Clarkson. Computational election verifiability: Definitions and an analysis of helios and JCJ. *IACR Cryptology ePrint Archive*, 2015:233, 2015.
42. Georgios Tsoukalas, Kostas Papadimitriou, Panos Louridas, and Panayiotis Tsanakas. From Helios to Zeus. In *EVT/WOTE*, 2013.
43. Thomas Zacharias. The DEMOS family of e-voting systems: End-to-end vevrifiable elections in the standard model. PhD thesis, National and Kapodistrian University of Athens, July 2016.
44. 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.