Noname manuscript No.
(will be inserted by the editor)

SecIVo: A Quantitative Security Evaluation Framework for Internet Voting Schemes

Stephan Neumann · Melanie Volkamer · Jurlind Budurushi · Marco Prandini

Received: date / Accepted: date

Abstract Voting over the Internet is subject to a number of security requirements. Each voting scheme has its own bespoke set of assumptions to ensure these security requirements. The criticality of these assumptions depends on the election setting (*e.g.* how trustworthy the voting servers or the voting devices are). The consequence of this is that the security of different Internet voting schemes cannot easily be compared. We have addressed this shortcoming by developing SecIVo, a quantitative security evaluation framework for Internet voting schemes. On the basis of uniform adversarial capabilities, the framework provides two specification languages, namely qualitative security models and election settings. Upon system analysis, system analysts feed the framework with qualitative security models composed of adversarial capabilities. On the other side, election officials specify their election setting in terms of –amongst others– expected adversarial capabilities. The framework evaluates the qualitative security models within the given election setting and returns

This project (HA project no. 435/14-25) is funded in the framework of Hessen ModellProjekte, financed with funds of LOEWE Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence). Furthermore, the first author is partially funded by CASED project ComVote.

Stephan Neumann
Technische Universtität Darmstadt
E-mail: stephan.neumann@secuso.org

Melanie Volkamer
Technische Universtität Darmstadt; Karlstad University
E-mail: melanie.volkamer@secuso.org

Jurlind Budurushi
Technische Universtität Darmstadt
E-mail: jurlind.budurushi@secuso.org

Marco Prandini
Università di Bologna
E-mail: marco.prandini@unibo.it

satisfaction degrees for a set of security requirements. We apply SecIVo to quantitatively evaluate Helios and Remotegrity within three election settings. It turns out that there is no scheme which outperforms the other scheme in all settings. Consequently, selecting the most appropriate scheme from a security perspective depends on the environment into which the scheme is to be embedded.

Keywords Internet Voting · Security Evaluation · Security Requirements

1 Introduction

The need for secure elections is paramount, especially when the election process incorporates notoriously vulnerable components such as the Internet and desktop or mobile computers. Consequently, the implementation of Internet voting is subject to rigorous security requirements such as vote secrecy and vote integrity. Given the fact that (some of the) security requirements seem to contradict each other, the numerous Internet voting schemes proposed, *e.g.* [1, 12, 18, 19, 30, 37, 63], can only implement these requirements by making certain assumptions about the operational environment. These assumptions can be highly diverse. For example, the JCJ/Civitas [19, 37] builds vote secrecy (also referred to as vote privacy [26, 63]) upon the assumption that the device over which a voter casts her vote is trustworthy. Pretty Good Democracy [63] – as a classical code voting scheme – enforces vote secrecy in the presence of malicious voting devices, yet in order to ensure vote secrecy , the scheme assumes that the voter can cast her vote without adversarial influence. The diversity of security requirements and assumptions implies that the overall security of Internet voting schemes cannot easily be quantified. As a consequence, decision makers in charge of selecting the most appropriate scheme for their concrete application environment cannot make an informed decision about which scheme to select.

Contribution: We construct SecIVo, a quantitative security evaluation framework for Internet voting schemes. The framework provides two specification languages on the basis of uniform and sufficiently abstract adversarial capabilities. The language of *qualitative security models* enables system analysts to specify the security of Internet voting schemes in an election-independent manner. The language of *election settings* allows election officials to specify their election environment in terms of expected adversaries and the number of expected voters. Ultimately, the framework evaluates given qualitative security models within a given election setting by the application of a risk-based approach and Monte-Carlo simulations. As result, the framework returns *satisfaction degrees* for a set of security requirements. To that end, the constructed framework represents a complement to established protocol analysis techniques, as for instance symbolic protocol analysis [4, 7, 8, 60] or cryptographic protocol analysis [14, 69]. We apply SecIVo to quantitatively evaluate the security of Helios [1] and Remotegrity [70] within three election settings.

2 SecIVo from Bird's Eye View

The quantitative security evaluation framework SecIVo is composed of several building blocks which are abstractly described in the remainder of this section. An overview of SecIVo and its interfaces is provided in Fig. 1.

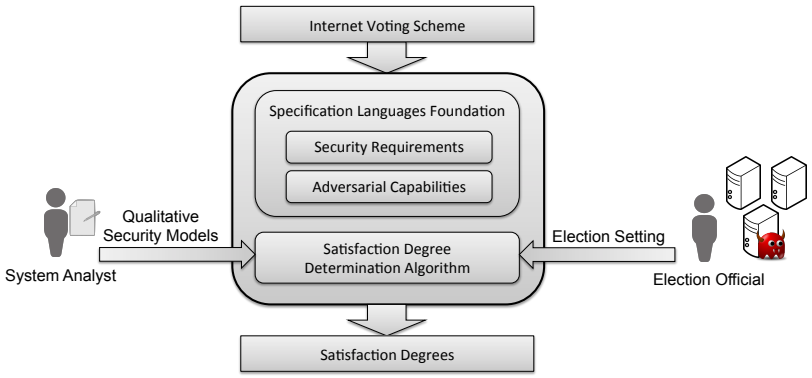


Fig. 1: The quantitative security evaluation framework SecIVo and its building blocks.

Internet voting schemes address a diversity of security requirements. Therefore, in the first place, a comprehensive set of security requirements has to be determined. Assessing the enforcement of security requirements for diverse Internet voting schemes depends on the election setting. The setting encodes –amongst others– information about the expected adversaries in a scheme-independent manner. The block *Specification Languages Foundation* (Section 3) is dedicated to determining security requirements and uniform adversarial capabilities. The block *Qualitative Security Models* (Section 5) provides a specification language to system analysts which they use to capture the qualitative security of Internet voting schemes. The resulting qualitative security models indicate which type of adversary can cause which impact to the different security requirements. The block *Election Setting* (Section 6) provides a specification language to election officials which they use to capture – amongst others – the expected adversary. The block *Satisfaction Degree Determination Algorithm* (Section 7) defines an algorithm that evaluates the qualitative security models of an Internet voting scheme within the specified election setting. The output of this algorithm are satisfaction degrees for all security requirements.

3 Security Requirements and Adversarial Capabilities

The basis of the constructed framework are a set of security requirements for Internet voting schemes and a set of uniform adversarial capabilities. These

capabilities are used to describe adversaries against which schemes shall ensure the security requirements.

3.1 Security Requirements

In preparation for this work, the proceedings of major security conferences, and electronic voting conferences and workshops were reviewed. A broad set of security requirements was determined. The requirements are provided in alphabetical order together with example sources.

- *Accountability* [44]: The voting scheme ensures that in the case of verification failures, the responsible entity can be held accountable for the failure.
- *Coercion-resistance* [16, 26, 64]: The voting scheme offers mechanisms to protect voters from coercers forcing voters to cast a vote in a specific way.
- *Eligibility* [16, 64]: The voting scheme ensures that only eligible voters' votes are included once in the election result.
- *Eligibility Verifiability* [23, 40]: The voting scheme offers any observer the possibility to verify that only eligible voters' votes are included once in the election result.
- *Fairness* [16, 64]: The voting scheme does not reveal any eligible voter's intention before the end of the election.
- *Individual Verifiability* [16, 40, 64]: The voting scheme offers each eligible voter the possibility to verify that her intention has been correctly stored for tabulation.
- *Long-term vote secrecy* [51]: The voting scheme does not provide more evidence about an eligible voter's intention than the election result does, even against computationally unrestricted adversaries.
- *Receipt-freeness* [16,26,64]: The voting scheme does not provide any receipt enabling the voter to prove her vote.
- *Robustness* [64]: The voting scheme returns an election result.
- *Universal Verifiability* [40, 64]: The voting scheme offers any observer the possibility to verify that the stored ballots have been correctly tallied.
- *Vote Integrity* [16,64]: The voting scheme ensures that each vote is correctly included in the election result.
- *Vote Secrecy* [16,26,64]: The voting scheme does not provide more evidence about an eligible voter's intention than the election result does.

Vote integrity and eligibility overlap to some extent. If an adversary is able to overwrite a voter's vote, the adversary essentially violates both vote integrity and eligibility. We resolve this overlapping by interpreting the altering of a vote with the purpose of casting a different vote, an integrity violation, and the injection of additional illegitimate votes, an eligibility violation.

3.2 Adversarial Capabilities

Several works have addressed the derivation of adversarial capabilities, either with regard to security protocols [27], security ceremonies [15], or specifically addressing Internet voting [46, 53]. We build upon these works to define a set of uniform and adequately abstract adversarial capabilities for our framework. We classify these capabilities into the classes *corruption capabilities*, *channel capabilities*, and *computational capabilities*. Note that in the following paragraphs, variables are indicated by [*`X`*].

Corruption Capabilities: The security of Internet voting might be threatened by corrupt authorities carrying the election duties, be it either in terms of administrators, hardware, or software components. We distinguish between authorities that are not in direct contact with voters (*backend authorities*) and those that are in direct contact with voters (*online authorities*). It shall be emphasized that backend authorities are not necessarily disconnected from the Internet, but rather are not directly accessible by the general public. We propose this distinction because of the difference in attack strategies required to compromise these authorities. While online authorities are generally threatened by external entities, such as malicious voters or hackers, the compromise of backend authorities in the most general case requires the collaboration of malicious insiders.

ABE: The adversary can corrupt a [*`backend authority`*].
AON: The adversary can corrupt an [*`online authority`*].

On the voter-side, another crucial component is the device used to cast a vote. This device might be under adversarial control.

VD: The adversary can corrupt a [*`voting device`*].

The security of the voting schemes does, however, not only depend on the trustworthiness of certain authorities or devices. Rather, these schemes' security relies on the human-computer interaction. Voters might be interested in or coerced into deviating from their original intention. We distinguish between the capabilities that the adversary might receive objects or data from voters (*voter output*), *e.g.* vote receipts, and that the adversary might provide voters with objects or data (*voter input*), *e.g.* instructions to cast a vote in a unique and identifiable manner.

VO: The adversary can receive objects/data from a [*`voter`*].
VI: The adversary can send objects/data to a [*`voter`*].

Channel Capabilities: In accordance to the widely established Dolev-Yao model [27], we assume channels between the voting devices and authorities or between authorities to be public. Carlos *et al.* [15] propose a refined Dolev-Yao

model incorporating human-device communication channels, addressing so-called security ceremonies. However, Carlos *et al.* argue that assuming these new communication channels to be completely public might be too pessimistic. We follow that argumentation and define one adversarial capability indicating that the channel between a voter and her device(s) might be controlled by the adversary.

CH: The adversary can control a [*`communication channel`*] between a voter and her voting device(s).

Computational Capabilities: A number or scientific works consider adversaries capable of obtaining (practically) unlimited computational resources, *e.g.* [51]. This is captured by the following capability.

CR: The adversary is computationally unrestricted.

In the following, we refer to the set

$$C = \{ABE, AON, VO, VI, VD, CH, CR\}$$

as *abstract capabilities* and to the set

$$C^S = \{ABE_1, \ldots, ABE_{n_1}, \ldots, CH_1, \ldots, CH_{n_6}, CR\}$$

as *instantiated capabilities* of scheme S if S captures n_1 backend authorities, $\ldots$, and n_6 voters.

3.3 Revision of Security Requirements

The lists of security requirements compiled in Section 3.1 and adversarial capabilities compiled in Section 3.2 result in a certain redundancy. To counter this artifact, we follow Delaune, Kremer, and Ryan [25], Küsters and Truderung [41] and Smyth [66] and consider receipt-freeness and coercion-resistance as special cases of vote secrecy. To that end, receipt-freeness ensures vote secrecy in the presence of an adversary receiving objects/data from the voters (refer to capabilities ***VO***). Coercion-resistance ensures vote secrecy in the presence of an adversary receiving and sending objects/data from/to the voters (refer to capabilities ***VO*** and ***VI***). Additionally, following from its definition, long-term vote secrecy ensures vote secrecy in the presence of computationally unrestricted adversaries (refer to capability ***CR***).

Furthermore, we revise the definitions of vote integrity and eligibility by incorporating their verifiability counterparts. If vote integrity, respectively eligibility, is verifiable, then vote integrity, respectively eligibility, is enforced against arbitrary adversarial capabilities. If vote integrity, respectively eligibility, is not verifiable, then vote integrity, respectively eligibility, is enforced against conspiracies which are not capable of altering votes in an undetectable manner.

In the remainder of this work, security is assessed in terms of the security requirements *eligibility*, *fairness*, *robustness*, *vote integrity*, and *vote secrecy*.

4 Running Example – Simple Internet Voting Scheme

We consider a simple Internet voting scheme as running example for the constructed framework. The sequence diagram of the example scheme is provided in Fig. 2.

To initiate the voting process on her voting device, a voter establishes an authenticated and encrypted connection towards the registration server *RS* (online authority). The voter authenticates herself towards *RS*. *RS* verifies the voter's eligibility and additionally consults the validation server *VS* (backend authority). *VS* verifies the voter's eligibility one more time and generates a credential upon successful eligibility check. *VS* forwards that credential to the ballot box server *BBS* (online authority) and *RS*, which in turn forwards the credential to the voter. In order to cast her vote, the voter consults the election website (hosted by *BBS*) in an authenticated and encrypted manner. She subsequently casts her vote together with her credential. *BBS* verifies the validity of the cast vote by checking whether the credential has been generated by *VS* and has not yet been used to cast a vote. Upon success, *BBS* stores the vote for the later vote tallying. After all votes have been cast, *BBS* sums up all received votes and announces the election result. It shall be emphasized that the scheme does neither defend security requirements against malicious voting devices nor does the scheme provide verifiability mechanisms to maximize vote integrity.

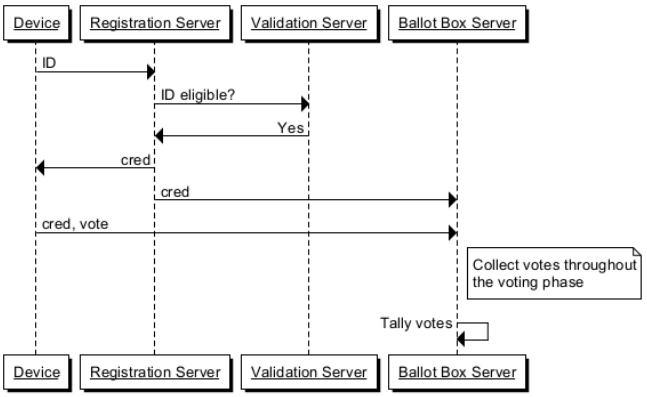


Fig. 2: Sequence diagram of the example scheme.

5 Language for the Specification of Qualitative Security Models

On the basis of security requirements and uniform adversarial capabilities, system analysts determine qualitative security models of Internet voting schemes.

The goal of this section is the definition of a specification language for system analysts.

We first specify qualitative adversary models, the foundation of qualitative security models. In the second part, we define qualitative security models of Internet voting schemes.

5.1 Composing Adversarial Capabilities to Qualitative Adversary Models

The foundation of adversaries against Internet voting schemes are adversarial capabilities. However, adversaries must generally possess a number of these capabilities to violate security requirements, *e.g.* several servers must be compromised. We define a *qualitative adversary model* as follows:

Definition 1 (Qualitative Adversary Model) *Suppose a voting scheme S with the set of instantiated capabilities C^S. We say that $\mathcal{A}_i^S$ is a qualitative adversary model, or simply adversary, against scheme S if $\mathcal{A}_i^S \subseteq C^S$.*

Example Scheme: For the sake of brevity, consider for the moment only the capabilities AON and ABE. RS and BBS are accessible by anybody; hence, these servers are online servers (online authorities). In spite of the fact that VS interacts with RS and BBS throughout the voting phase, RS does not directly interact with the general public. Hence, aligned with the capability specification in Section 3.2, VS is therefore considered as a backend server (backend authority). To compromise RS and BBS, the adversary consequently needs the capabilities AON_{RS} and AON_{BBS}. To compromise VS, the adversary needs the capability ABE_{VS}. Hence, within the given Internet voting scheme, any set of capabilities $\mathcal{A}_i^S$ such that $\mathcal{A}_i^S \subseteq \{AON_{RS}, AON_{BBS}, ABE_{VS}\}$ is a qualitative adversary model of the given scheme S.

5.2 Composing Adversarial Capabilities to Qualitative Security Models

In analogy to the definition of qualitative adversary models, we define qualitative security models within this subsection. Therefore, we first introduce the minimal cut sets notation. Thereafter, we discuss the impact that adversaries can cause on security requirements and ultimately define qualitative security models of Internet voting schemes.

5.2.1 Minimal Cut Sets

To specify qualitative security models, we pick up the concept of *minimal cut sets* [48]. Cut sets are a standard concept in reliability and availability theory [2, 5, 36]. Fuqua [31] precisely describes cut sets as "*... any basic event or combination of basic events whose occurrence will cause the top event to*

occur." A cut set is *minimal*, if none of its subsets is a cut set. A violation of a security requirement (refer to Section 3.3) is the top event, while the possession of instantiated adversarial capabilities (refer to Section 3.2) corresponds to basic events.

Example Scheme: We stick to the restriction and consider only the capabilities AON and ABE. Throughout the voting phase, RS and VS learn the relation between the voter's identity and her voting credential. Furthermore, BBS learns the relation between voting credentials and the votes cast with those credentials. Hence, the malicious collaboration between RS and BBS or between VS and BBS results in a violation of vote secrecy. None of these servers might, however, violate vote secrecy individually. Consequently, the sets

$$\{AON_{RS}, AON_{BBS}\}, \{ABE_{VS}, AON_{BBS}\}, \{AON_{RS}, ABE_{VS}, AON_{BBS}\}$$

are cut sets. However, only the sets

$$\{AON_{RS}, AON_{BBS}\}, \{ABE_{VS}, AON_{BBS}\}$$

are minimal as both sets are a subset of the cut set

$$\{AON_{RS}, ABE_{VS}, AON_{BBS}\}.$$

Notational Conventions: For the sake of higher readability, we introduce two notational conventions: First, as part of the scheme description, we make clear which capability is needed to compromise/control/influence which part of the scheme. Therefore, we omit the capability, but rather provide the component. For instance, instead of AON_{RS} and ABE_{BBS}, we simply write RS and BBS. Second, we rewrite lists of minimal cut sets in *disjunctive normal form*. Hence, rather than writing

$$\{RS, BBS\}, \{VS, BBS\},$$

we write

$$(RS \wedge BBS) \vee (VS \wedge BBS).$$

5.2.2 Adversarial Impact on Security Requirements

The violation of security requirements cannot be related to the presence of a unique adversary, but in fact different adversaries might cause different impact levels to security requirements. Consider for instance the security requirement vote integrity. Rather than assigning one precise successful adversary to that requirement, it is intuitive to indicate which adversaries might undetectably violate the integrity of *one*, of *two*, ..., of *all* cast votes. For example, an adversary controlling one voting device can manipulate one voter's vote, while an adversary controlling ten voting devices can manipulate ten voters' vote; an adversary controlling the ballot box server can even be in possession of an

attack strategy manipulating all cast votes. Furthermore, there might be attacks that work up to a certain impact level, but could not cause the maximum impact. One typical attack of this form is a clash attack [43], because clash attacks can only target two votes that are equal while their equality cannot be known in advance[1].

Hence, for each specific impact level, a qualitative security model is specified. Note that the number of impact levels depends on the number of eligible voters n_{el} and the number of expected voters n_{ex} and can therefore not be known throughout the determination of qualitative security models. Hence, qualitative security models are specified in a generic manner. Generally, attack strategies are successful up to a certain extent. For instance, the corruption of central servers would often result in the violation of a security requirement for all expected voters. In that case, the respective attack strategies are incorporated into all instantiated security models up to impact level n_{ex}. The corruption of one voting device might generally only violate a security requirement for one voter. Hence, once the numbers n_{el} and n_{ex} are known, the impact levels can be instantiated and so can the abstract qualitative security models.

With regard to different security requirements, the impact on these requirements might slightly differ. Vote secrecy, vote integrity, and fairness are only defined for voters that actually cast a vote. Hence the maximum impact of these requirements is n_{ex}. Eligibility relates to those voters that abstain from the election. Hence, an adversary causes maximum impact on eligibility if he is able to cast illegitimate votes for all $n_{el} - n_{ex}$ abstaining voters. Ultimately, with regard to robustness, only two impact levels are considered, *i.e.* an adversary that is capable of preventing the election result from being computed (impact n_{ex}) or an adversary that is not capable (impact 0). In the remainder of this work, we denote the maximum impact generically by n as an abstraction of n_{ex} or $n_{el} - n_{ex}$ respectively.

5.2.3 Definition of Qualitative Security Models

After the definition of qualitative adversary models and the discussion of adversarial impact of security requirements, we are able to define qualitative security models.

Definition 2 (Qualitative Security Model) *Let a voting scheme S with the set of instantiated capabilities C^S be given. We say that*

$$\mathcal{M}^S_{r,l} = ((c^{S,r,l}_{1,1} \wedge \cdots \wedge c^{S,r,l}_{1,m_1}) \vee \cdots \vee (c^{S,r,l}_{k,1} \wedge \cdots \wedge c^{S,r,l}_{k,m_k})) \text{ with } c^{S,r,l}_{i,j} \in C^S$$

is a qualitative security model of S with regard to security requirement r and impact level l if there exists a set of adversaries $\{\mathcal{A}^S_1, \ldots, \mathcal{A}^S_k\}$ where $\mathcal{A}^S_i$ is specified by capabilities $\{c^{S,r,l}_{i,1}, \ldots, c^{S,r,l}_{i,m_i}\}$, such that

[1] Knowing in advance to the election which voters will cast identical votes is at least very hard.

1. *the capabilities of all adversaries* $\mathcal{A} \in \{\mathcal{A}_1^S, \ldots, \mathcal{A}_k^S\}$ *suffice to cause impact* l *on* r*, and*
2. *for all adversaries* $\mathcal{A} \in \{\mathcal{A}_1^S, \ldots, \mathcal{A}_k^S\}$*, there is no adversary* $\mathcal{A}' \subset \mathcal{A}$ *such that the capabilities of* $\mathcal{A}'$ *suffice to cause impact* l *on* r*, and*
3. *for all adversaries* $\mathcal{A}'$*, of which the capabilities suffice to cause impact* l *on* r*, there is an adversary* $\mathcal{A} \in \{\mathcal{A}_1^S, \ldots, \mathcal{A}_k^S\}$ *such that* $\mathcal{A} \subseteq \mathcal{A}'$*.*

Within the herein constructed framework, we do not address the question whether specific adversarial capabilities suffice to cause specific impact on a specific requirement. We rather make the general assumption that all combinations of adversarial capabilities that suffice to cause specific impact on a specific requirement are known to system analysts. However, orthogonal to our work, several research approaches address this question rigorously. At least two major approaches can be distinguished, namely symbolic protocol analysis [4, 7, 8, 60], *e.g.* model checking [4] or inductive theorem proving [60], and cryptographic protocol analysis [14,69]. Both approaches generally build upon *one* concrete adversary model and are used to prove or disprove a protocol's security under that specific model. To that end, these approaches complement the concept of the herein specified qualitative security models.

Example Scheme: We relax the previous restriction and, in addition to the capabilities ABE and AON, consider that an adversary might gain the capability VD (corruption of voting devices). Hence, to cause impact 1 the adversary might either compromise any single voting device, or any two voting devices, or ... or all voting devices. Furthermore the adversary might compromise either the registration server and the ballot box server or the validation server and the ballot box server. To cause impact 2, the adversary might either compromise any two voting devices, or any three voting devices, or ... or all voting devices. Furthermore the adversary might compromise either the registration server and the ballot box server or the validation server and the ballot box server. Generically, to cause impact i, the adversary might either compromise any i voting devices, or any $i+1$ voting devices, or ... or all voting devices. Furthermore, the adversary might compromise either the registration server and the ballot box server or the validation server and the ballot box server. The resulting qualitative security models of the example scheme are provided in Table 1.

Requirement	Qualitative Security Models	Impact
Vote Secrecy	$(VD_1 \vee VD_2 \vee \cdots \vee VD_n) \vee$	
	$((VD_1 \wedge VD_2) \vee (VD_1 \wedge VD_3) \vee \cdots \vee (VD_{n-1} \wedge VD_n)) \vee$	
	$(RS \wedge BBS) \vee (VS \wedge BBS)$	1
	...	...
	$\bigvee_{I \subseteq \{1,\ldots,n\}, \lvert I \rvert \geq l} (\bigwedge_{i \in I} VD_i) \vee (RS \wedge BBS) \vee (VS \wedge BBS)$	$1 \leq l \leq n$

Table 1: Qualitative security models of the example scheme for vote secrecy.

6 Language for the Specification of Election Settings

Adversarial capabilities are not only the underpinning of qualitative security models, but furthermore they form the basis for the specification of election settings. The goal of this section is the definition of a specification language for election officials.

The quantitative evaluation of qualitative security models could be conducted in a simple manner if election officials could precisely assign probabilities to the presence of adversarial capabilities (refer to Section 3.2). However, election officials might provide these probabilities with some uncertainty due to the lack of available knowledge regarding capabilities. Furthermore, because of the potential complexity of qualitative security models, their quantitative evaluation might be significantly impacted by minor changes in capability probabilities. We take account of this and incorporate Monte-Carlo simulations [10, 62] into the quantification process. Rather than precise capability probabilities, we require the election official to provide probability distributions for abstract adversarial capabilities.

Additionally, the election official specifies the number of *eligible voters* n_{el} and estimates the *number of expected voters* n_{ex}. These numbers are needed to instantiate all possible impact levels. Eventually, election settings are defined as follows:

Definition 3 (Election Setting) *Given the set of abstract capabilities C, the number of eligible voters n_{el}, the number of expected voters n_{ex}, and probability distributions $d^{c_1}, \ldots, d^{c_{|C|}}$ for all capabilities $c_i \in C$, we say that the tuple*

$$\mathcal{E} = (d^{c_1}, \ldots, d^{c_{|C|}}, n_{el}, n_{ex})$$

is an election setting.

Example Election Setting: We consider the following election setting, where $U(a, b)$ denotes the uniform distribution with support $[a, b]$:

$$\begin{aligned}
E = (&d^{ABE} = U(0.0001, 0.0005); && \text{Distribution for capability ABE}\\
&d^{AON} = U(0.001, 0.005); && \text{Distribution for capability AON}\\
&d^{VO} = U(0.01, 0.05); && \text{Distribution for capability VO}\\
&d^{VI} = U(0.01, 0.05); && \text{Distribution for capability VI}\\
&d^{VD} = U(0.01, 0.05); && \text{Distribution for capability VD}\\
&d^{CH} = U(0.01, 0.05); && \text{Distribution for capability CH}\\
&d^{CR} = U(0, 0); && \text{Distribution for capability CR}\\
&2,000) && \text{Number of eligible voters}\\
&1,000) && \text{Number of expected voters}
\end{aligned}$$

7 Determination of Satisfaction Degrees in Election Settings

The core of the framework is the algorithm for the quantitative evaluation of qualitative security models within specific election settings. In the first part, qualitative security models are transformed into probability formulas. Thereafter, we show how Monte-Carlo simulations support the quantitative evaluation of qualitative security models. Finally, the algorithm for the quantitative evaluation of qualitative security models with respect to specific election settings is introduced.

7.1 Transforming Qualitative Security Models into Probability Formulas

Before their actual evaluation, the probabilities of the event that an adversary violates qualitative security models needs to be determined. Therefore, standard probability theory is applied. Recall the structure of qualitative security models:

$$\mathcal{M}^S_{r,l} = ((c^{S,r,l}_{1,1} \wedge \cdots \wedge c^{S,r,l}_{1,m_1}) \vee \cdots \vee (c^{S,r,l}_{n,1} \wedge \cdots \wedge c^{S,r,l}_{k,m_k})) \text{ with } c^{S,r,l}_{i,j} \in C^S$$

By $cc^{S,r,l}_i = (c^{S,r,l}_{n,1} \wedge \cdots \wedge c^{S,r,l}_{i,m_i})$ we denote the i-th minimal cut set of scheme S with regard to requirement r and impact level l. Note that the minimal cut sets $cc^{S,r,l}_1, \ldots, cc^{S,r,l}_k$ might be overlapping, *i.e.* they share specific instantiated capabilities, for instance if the corruption of the same server appears in both sets. We define ec as the event that the adversary gains capability $c \in C$, $ec^{S,r,l}_{i,j}$ as the event that the adversary gains capability $c^{S,r,l}_{i,j}$ and $ecc^{S,r,l}_i$ as the event that the adversary satisfies the minimal cut set $cc^{S,r,l}_i$.

Suppose that probabilities for all abstract capabilities (refer to Section 3.2) be given by $P(ec)$ with $c \in C$. All instantiated capabilities of c inherit the probability $P(ec)$ and are assumed to be independent from each other. Hence, for any two instantiated capabilities $c^{S,r,l}_{i,j}$ and $c^{S,r,l}_{x,y}$ that stem from the same abstract capability c, it holds $P(ec^{S,r,l}_{i,j}) = P(ec^{S,r,l}_{x,y}) = P(ec)$. Then one can compute the probability that an adversary satisfies $ecc^{S,r,l}_i$ as:

$$P(ecc^{S,r,l}_i) = P(ec^{S,r,l}_{i,1}) \cdot P(ec^{S,r,l}_{i,2}) \cdot \cdots \cdot P(ec^{S,r,l}_{i,m_i})$$

Ultimately, we are interested in the probability that an adversary might cause impact l on requirement r in scheme S, *i.e.* the probability $P(\bigcup_{i=1}^{k} ecc^{S,r,l}_i)$. Different minimal cut sets of the same qualitative security model might be overlapping, *i.e.* the same instantiated capability is required to satisfy these minimal cut sets. The *inclusion-exclusion principle* [38, 68] provides a means to calculate the probability that at least one of several overlapping events happens. Consequently, to calculate the probability $P(\bigcup_{i=1}^{k} ecc^{S,r,l}_i)$, the application of the inclusion-exclusion principle leads to the following probability:

$$P(\bigcup_{i=1}^{k} ecc^{S,r,l}_i) = \sum_{j=1}^{k} \left((-1)^{j-1} \sum_{I \subset \{1,\ldots,k\}, |I|=j} P(\bigcap_{i \in I} ecc^{S,r,l}_i) \right)$$

If none of the minimal cut sets overlap, then one can apply De Morgan's Law and compute $P(\bigcup_{i=1}^{k} ecc_i^{S,r,l})$ by the complementary events of $ecc_i^{S,r,l}$. Hence, the resulting probability formulas for non-overlapping minimal cut sets is:

$$P(\bigcup_{i=1}^{k} ecc_i^{S,r,l}) = 1 - ((1 - P(ecc_1^{S,r,l}) \cdot (1 - P(ecc_2^{S,r,l})) \cdot \dots \cdot (1 - P(ecc_k^{S,r,l})))$$

In addition to the inclusion-exclusion principle, minimal cut sets of the form "*at least* x events" that stem from the same abstract capability (*e.g.* in the case of voting devices or threshold cryptography), the *cumulative binomial probability* computation is applied. Finally, the resulting probability formulas build the foundation for quantitative security evaluation. Note that system analysts might provide probability formulas directly rather than qualitative security models. However, the transformation of qualitative security models would require the system analyst to consider the overlappings of different attack strategies and the mathematical modelling of those overlappings. To lower the system analyst's burden, the transformation is incorporated into the framework's quantification process.

Example Scheme: We first consider the probability of the event that either the registration server *and* the ballot box server *or* the validation server *and* the ballot box server are compromised (event A). Therefore, we apply the inclusion-exclusion principle:

$$P(A) = P(RS) \cdot P(US) + P(VS) \cdot P(US) - P(RS) \cdot P(VS) \cdot P(US)$$

Furthermore, we consider the probability of the event that at least l voting devices are compromised (event B):

$$P(B) = 1 - \left(\sum_{i=0}^{l-1} \binom{n}{i} P(VD)^i \cdot (1 - P(VD)^{n-i}) \right)$$

Given the independence of events A and B (no overlappings), we can compute the probability that vote secrecy of at least l votes is violated as follows:

$$P(A \cup B) = 1 - ((1 - P(A)) \cdot (1 - P(B)))$$

7.2 Incorporating Monte-Carlo Simulations into the Quantitative Evaluation of Qualitative Security Models

Recall that election officials assign probability distributions rather than precise probabilities to adversarial capabilities. Following the Monte-Carlo approach, see for instance [10, 45, 62], the given distributions (refer to Section 6) are sampled and the qualitative security models are evaluated with those random samples. The evaluation of qualitative security models on the basis of

probability distributions is ultimately assigned to the empirical mean of the qualitative security model evaluations with random samples. The *number of Monte-Carlo simulations* indicates how often the qualitative security models are evaluated with independent random samples. Additionally, a confidence interval is calculated. Surrounding the empirical mean, the confidence interval indicates the stability of the empirical mean. Mathematically, the confidence interval contains the statistical mean of infinite many simulated qualitative security evaluations with a certain confidence. The *confidence value* indicates the certainty with which the statistical mean is within the obtained confidence interval. As rule of thumb, one can say that the larger the number of Monte-Carlo simulations and the smaller the confidence value, the smaller the resulting confidence interval. Consequently, for the application of Monte-Carlo simulations, the number of Monte-Carlo simulations m to be run and a confidence value z, also referred to as z-score [28], have to be specified. With regard to the number of Monte-Carlo simulations, we follow the recommendations by Mundform *et al.* [52] and set $m = 10,000$. Additionally, we set the confidence value to $z = 2$, thereby obtaining a certainty of $\approx 95.5\%$ that the statistical mean lies within the confidence interval generated around the empirical mean.

7.3 Determining Satisfaction Degrees from Qualitative Security Models and Election Settings

The evaluation of qualitative security models within election settings is built upon standard risk theory (refer for instance to the NIST 800-30 Risk Management Guide for Information Technology Systems [67]). To determine the satisfaction degree of an Internet voting scheme S with qualitative security models $\mathcal{M}^S_{r,l}$ (refer to Sections 2 and 7.1) with regard to a security requirement $r \in R$ (refer to Section 3.1) within a specified election setting $\mathcal{E} = (d^{ABE}, d^{AON}, d^{VO}, d^{VI}, d^{VI}, d^{VD}, d^{CH}, n_{el}, n_{ex})$ (refer to Section 6), the following algorithm is defined:

1. Based on the number of eligible voters n_{el} and the number of expected voters n_{ex} (refer to Section 6), the number of impact levels is instantiated and probability formulas accordingly (refer to Section 7.1). Consequently, n (depending on the security requirement under investigation either n_{ex} or $n_{el} - n_{ex}$) impact levels are assigned to n probability formulas. The probability formula for causing impact i against vote secrecy within the example scheme is given in Section 7.1.
2. The following steps are conducted m times (number Monte-Carlo simulations):
   (a) For each adversarial capability $c \in C$ (refer to Section 3.2), an estimator of the probability $P(ec)$ is sampled according to the probability distribution d^c in $\mathcal{E}$ (refer to Section 6). For the example election setting this could lead to the following probability samples: $d^{ABE}_1 = 0.000232, d^{AON}_1 = 0.004283, d^{VO}_1 = 0.02482, d^{VO}_1 = 0.03993, d^{VD}_1 = 0.04832, d^{CH}_1 = 0.04813$.

(b) For each impact level generated in step 1, the probability formula (refer to Section 7.1) of the qualitative security model is evaluated based on the samples generated in step 2.a). We provide an excerpt of this step for the example scheme:

Impact	Probability (Qualitative Security Models)
1	1
⋮	⋮
39	0.9063074207
⋮	⋮
1000	0.00001933348919

(c) For each impact value, a risk value is calculated by multiplying the normalized impact with the evaluated probability formula of the respective qualitative security model (result of step 2.b)). We provide an excerpt of this step for the example scheme:

Impact	Probability (Qualitative Security Models)	Risk
1	1	0.001
⋮	⋮	⋮
39	0.9063074207	0.03534598941
⋮	⋮	⋮
1000	0.00001933348919	0.00001933348919

(d) The largest value of step 2.c) is identified. The value is assigned to x_i in the i-th Monte Carlo simulation. In the given election setting, the largest risk value appears at impact level 39 and equals 0.03534598941.

3. The following statistics are calculated with the result $x = (x_1, \ldots, x_m)$ of process step 2):

(a) By the application of standard statistics, the empirical mean is calculated:

$$\overline{x}_m = \frac{1}{m}(x_1 + \cdots + x_m)$$

The satisfaction degree is calculated:

$$sd = 1 - \overline{x}_m$$

The satisfaction degree of the example scheme after $10,000$ Monte-Carlo simulations equals 0.9786.

(b) By the application of standard statistics, the empirical standard deviation is calculated:

$$s_m = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x}_m)^2}$$

The empirical standard deviation of the example scheme after $10,000$ Monte-Carlo simulations equals 0.0091.

(c) Based on the confidence value (refer to Section 7.2), the confidence interval for sample model mean is calculated (refer to [28] for further details):

$$CI = \left[1 - (\overline{x}_m + z\frac{s_m}{\sqrt{m}}), 1 - (\overline{x}_m - z\frac{s_m}{\sqrt{m}})\right]$$

The confidence interval of the example scheme after $10,000$ Monte-Carlo simulations is:

$$[0.9783, 0.9787]$$

In the remainder of this work, the satisfaction degree of requirement r resulting from the quantification of scheme S and environment $\mathcal{E}$ is denoted as $SD^r_{S,\mathcal{E}}$, the confidence interval $CI^r_{S,\mathcal{E}}$ respectively.

Limitations of SecIVo: The evaluation of security requirements, more precisely eligibility and vote integrity relies on the assumption that voters verify everything that can be verified, independent of the required capabilities and motivation. Currently, the probabilities of gaining different adversarial capabilities are assumed to be independent. Yet, the possibility of gaining one adversarial capability might influence the probability of gaining another capability. For instance, an adversary developing a strategy to break into one voting device might apply the same strategy to another voting device thereby raising the probability of successful corruption. The constructed framework can be adapted in the future by incorporating conditional probabilities, *e.g.* by means of a Bayesian network. Furthermore, the capabilities are deliberately abstract to capture a large amount of specific assumptions. Yet, the degree of abstraction might be adapted with regard to officials' expertise. For instance, the difference between authorities might not only regard their online/backend status but there might be further indicators influencing the corruption probability, such as the number of administrators responsible for the specific server. While the framework allows election officials to specify expected adversaries as arbitrary probability distributions, the current implementation builds upon uniform distributions.

8 Proof of Concept Application: Quantitative Security Evaluation of Helios and Remotegrity

After its construction, we substantiate the framework's applicability and value by a proof of concept application. We build the proof of concept application upon Helios [1,35] with homomorphic tallying as applied for the IACR elections and Remotegrity [70] as applied for the Takoma Park Municipality elections in 2011. We selected these schemes because they have found their way into the practical application and simultaneously have been a target of interest for scientific research.

In the first part of this section, we provide information about which versions of the schemes have been considered and provide the corresponding minimal cut sets. Thereafter, we quantitatively evaluate the security of the schemes on the basis of three election settings and discuss the findings.

8.1 Qualitative Security Models of Helios and Remotegrity

In preparation for this work, we reviewed published articles, documents (see for instance [9, 17, 21, 22, 24, 32–34, 43, 61, 70]) and if possible involved persons have been consulted. In the remainder of this section we restrict our focus to the specification of qualitative security models. For further information about the individual schemes, we refer the reader to the herein cited documents.

The Remotegrity Scheme: Remotegrity provides verifiability against the misbehavior of any central component with regard to vote integrity. Any violation attempt by the four trustees (T_1, T_2, T_3, T_4) (backend authorities) is either detected by the verifying voter, or by the observing public either throughout the pre-voting (ballot auditing) or post-voting phase (opening of commitments). Eligibility might be violated by malicious or coerced voters forwarding their voting material. Yet, under the assumption that the electoral register is linked to the voting and authorization IDs, trustees cannot add ineligible voters. Assuming that at least some abstaining voters check whether their ID appears on the bulletin board as votes, trustees or the offline signing server (*OSS*) (backend authority) cannot vote for abstaining voters because of the dispute-freeness measure provided by Remotegrity due to use of scratch fields. Throughout the voting phase, trustees or the *OSS* might relate confirmation codes to respective candidates, thereby obtaining intermediate results and violating fairness. Also voters forwarding their election material might violate fairness as confirmation codes are published. Voters forwarding their election material might also violate vote secrecy. Furthermore, a conspiracy of at least two out four trustees might maliciously collaborate to relate received voting codes to the voter who obtained this code to vote, thereby violating vote secrecy. Robustness of the entire voting procedure might be violated if at least two out of four election officials do not provide their shares to reconstruct the seed.

The Helios Scheme: Helios provides voters with the possibility to verify generated ballots in cut-and-choose manner and additionally to verify that cast votes have been stored in an unaltered manner on the voting server (VS) (backend authority). Assuming that the voter verifies these two facts with a second device as implemented by Neumann *et al.* [54], votes might be altered if both devices maliciously collaborate. Alternatively, if the VS issues malicious applications and the voter's verification device is compromised, votes might be altered undetectably. Up to a certain number of votes, the VS might launch clash attacks [43] on voters with identical vote selection by providing, identical voting materials and malicious voting applications thereby violating vote integrity. We assume that the clash attack might be mounted up to one hundredth of all cast votes. Voters might forward their voting credentials thereby violating eligibility. A malicious voting device as well as a malicious voting application (provided by the malicious VS) might forward plaintext votes, thereby violating fairness. If the adversary observes the channel between voter and her voting device, fairness is broken. The trustees in collaboration might violate fairness by opening encrypted votes. The qualitative security model with regard to vote secrecy essentially corresponds to the fairness model. However, as opposed to the fairness case, trustees are not able to relate decrypted votes to the voter who cast it. Eventually, due to the lack of threshold cryptography[2] either trustee might prevent the election result from being computed by not providing their respective private key.

The qualitative security models of both Internet voting schemes are provided in Tables 2 and 3.

Requirement	Qualitative Security Models	Impact
Vote Integrity	$\emptyset$	$0 \leq l \leq 1$
Eligibility	$\left(\bigvee_{I \subseteq \{1,\ldots,n\}, \lvert I \rvert \geq l} (\bigwedge_{i \in I} VO_i)\right)$	$0 \leq l \leq 1$
Fairness	$\left(\bigvee_{I \subseteq \{1,\ldots,n\}, \lvert I \rvert \geq l} (\bigwedge_{i \in I} VO_i)\right) \vee OSS \vee$ (2 out of $\{T_1, \ldots, T_4\}$))	$0 \leq l \leq 1$
Vote Secrecy	$\left(\bigvee_{I \subseteq \{1,\ldots,n\}, \lvert I \rvert \geq l} (\bigwedge_{i \in I} VO_i)\right) \vee$ (2 out of $\{T_1, \ldots, T_4\}$)	$0 \leq l \leq 1$
Robustness	(2 out of $\{T_1, \ldots, T_4\}$)	1

Table 2: Qualitative security models of the Remotegrity voting scheme as applied for the Takoma Park Municipality elections in 2011.

8.2 Election Settings under Investigation

For the proof of concept application of the framework, three election settings are specified. Within all election settings, we set $n_{el} = 2,000$ as the *number of*

[2] The IACR deploys an n-out-of-n secret sharing scheme [24].

Requirement	Qualitative Security Models	Impact
Vote Integrity	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i^1 \wedge VD_i^2)\right) \vee VS$	$0 \leq l \leq \frac{1}{100}$
	$\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i^1 \wedge VD_i^2) \vee$ $\left(\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i^2)\right) \wedge VS\right)$	$\frac{1}{100} < l \leq 1$
Eligibility	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VO_i)\right)$	$0 \leq l \leq 1$
Fairness	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i^1)\right) \vee VS \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} CH_i)\right) \vee (T_1 \wedge T_2 \wedge T_3)$	$0 \leq l \leq 1$
Vote Secrecy	$\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} VD_i^1)\right) \vee VS \vee$ $\left(\bigvee_{I\subseteq\{1,\ldots,n\},\lvert I\rvert\geq l}(\bigwedge_{i\in I} CH_i)\right)$	$0 \leq l \leq 1$
Robustness	$T_1 \vee T_2 \vee T_3$	1

Table 3: Qualitative security models of the Helios voting scheme as applied for the IACR elections.

eligible voters and $n_{ex} = 1,000$ as the *number of expected voters*. Furthermore, on the basis of the four probability distributions

$$f = U(0.1, 0.2), \quad g = U(0.01, 0.02),$$

$$h = (0.001, 0.002), \quad i = (0.0001, 0.0002),$$

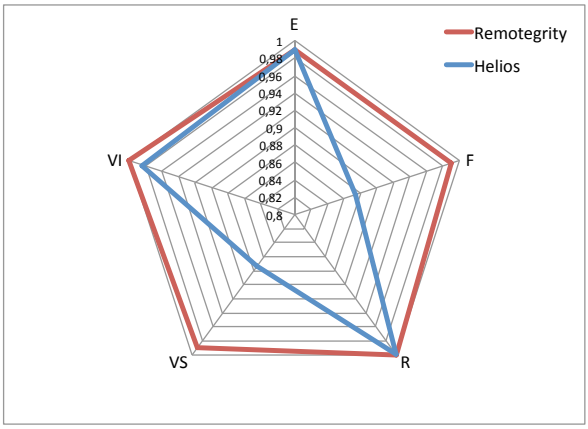
the probabilistic adversary models within the three election settings are provided in Table 4. Election setting 1 defines an adversary that is relatively strong with regard to corrupting voting devices. In contrast, election setting 2 captures an adversary that is relatively strong with regard to voter interaction, *i.e.* either obtaining objects or data from the voter or providing the voter with objects or data. Finally, the adversary defined in election setting 3 is relatively capable of controlling backend authorities. We do not consider computationally unrestricted adversaries.

El. Setting	VD	AON	ABE	VO	VI	CH	CR
E_1	f	h	i	g	g	g	0
E_2	g	h	i	f	f	g	0
E_3	h	h	g	g	g	h	0

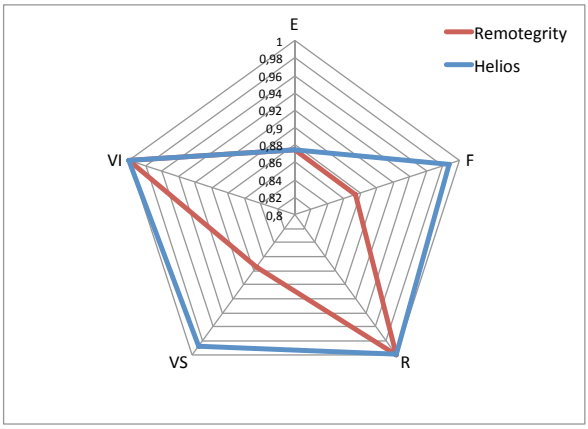
Table 4: On the basis of four probability distributions, three probabilistic adversary models are defined and assigned to election settings.

8.3 Satisfaction Degrees of the Internet Voting Schemes

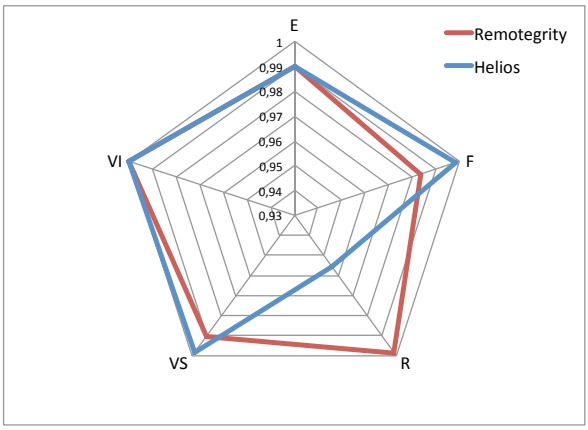
An overview about the resulting satisfaction degrees of both schemes in the three election settings is provided in Fig. 3. Due to the lack of space, we omit



(a) Election setting 1.



(b) Election setting 2.



(c) Election setting 3.

Fig. 3: Satisfaction degrees of Helios and Remotegrity in different election settings with regard to the requirements eligibility (E), fairness (F), robustness (R), vote secrecy (VS), and vote integrity (VI).

the precise confidence intervals. Yet, we emphasize that apart from the cases in which qualitative security models are equal in terms of needed capabilities, there is no overlap of confidence intervals, *i.e.* if the empirical satisfaction degree is larger in one scheme than in the other, then the statistical satisfaction degree is with a certainty of $\approx 95.5\%$ (confidence value $z = 2$) also larger. Consequently, the relative performance of both schemes is stable with regard to all security requirements in all election settings.

The figures show that Remotegrity outperforms Helios in the case of high voting device corruption, particularly with regard to vote secrecy and fairness. The rationale behind this is Remotegrity's code-based characteristic which essentially turns threats caused by malicious voting devices ineffective.

On the other side, summing up over all satisfaction degrees Remotegrity performs significantly worse than Helios when it comes to the threat of coerced or malicious voters. This quantitative shortcoming is explained by the fact that Remotegrity's Takoma park implementation allows voters to forward their election material. The material can be used by an adversary to match the published confirmation code against voter's code sheet; by doing so the adversary is able of violating both fairness and vote secrecy.

Ultimately, in the presence of an adversary that shows its strength with regard to backend authority corruption, it turns out that none of the schemes outperforms the other. Rather, selecting the most appropriate scheme might require an *a posteriori* decision, *e.g.* on the basis of the relative importance of different security requirements. While Remotegrity provides lower satisfaction degree with regard to vote secrecy and fairness, Remotegrity as applied for the Takoma Park election results in significantly higher robustness guarantees than Helios; this stems from the fact that threshold cryptography has been applied for the Takoma Park election, while individual trustees might sabotage the tallying process of the Helios scheme as applied for the IACR elections.

9 Related Work

Several works have addressed the assessment of risks for electronic voting schemes [6, 13, 39, 47, 55, 57–59] by deriving threats trees for electronic voting schemes. Comprehensive threat trees for electronic voting (or Internet voting) schemes are of great value for the deduction of adversaries violating security requirements, yet, the fine-grained threats considered in these works require decision makers to assign probabilities to specific threats. First, reviewing threat trees for Internet voting scheme poses a significant burden on election officials, e.g. [29] provides a 18-page threat tree for Internet voting. Second, threats might be strongly correlating, in which case, quantification becomes cumbersome, e.g. the threats *create undervote* and *delete races* (see [29], page 254). Coney et *al.* [20] are motivated by the lack of a taxonomy of electronic voting systems satisfying pre-defined requirements. The authors provide an interesting sketch of measuring privacy in voting systems by the voting systems' deviation from providing perfect privacy (this mainly corresponds to our

secrecy definition). The authors define this deviation by quantifying the maximum reduction of the uncertainty about a voter's vote. Due to the 2-page restrictions of their work, it only touches on privacy rather than any other security requirement.

Madan *et al.* [50] and in similar vein Almasizadeh *et al.* [3] develop semi-Markov chains for the quantification of security. As indicated by the authors, their focus is on the analysis methodology, not on transition probability assignment. Due to the high complexity of the model, we are critical about the fact that those parameters might be assigned by decision makers. Biondi and Legay [11] provide a Markov-chain based quantification approach for vote secrecy (the authors use the term anonymity). As opposed to this work, the authors focus on the information leakage through publicly available election data, namely the election result. Ouchani *et al.* [56] quantify attack patterns of the Common Attack Pattern Enumeration and Classification catalogue. As opposed to our work, the probability assignment is not tailored to take decision makers' expertise into consideration, which is crucial to Internet voting. Luna *et al.* [49] develop a quantitative threat modeling with particular focus on privacy-by-design requirement. Their approach derives from the Microsoft threat modeling approach STRIDE and threat-risk ranking approach DREAD as well as established privacy protection goals. With the consideration of fine-grained threats, the approach is not tailored towards non-security experts such as election officials. Similar to the present work, Schryen *et al.* [65] develop a quantitative trust metric upon qualitative propositional logic. The quantification builds on standard probability theory. As opposed to the present work, the authors do neither model the severity of threats to the system's security nor do they include uncertainty measures into their specification. Consequently, referring to the present work, their trust metric falls short in terms of expressiveness.

Küsters *et al.* [42, 43] provide a formal framework for measuring the level of verifiability, privacy, coercion-resistance, and accountability of voting protocols. The framework measures (by means of a so called δ) the adversary's chance of achieving her goal, *e.g.* making a verifier accept an incorrect election result (*verifiability*) or distinguishing between the fact whether an observed voter casts a vote for one candidate or another candidate (*privacy*). The measurement depends on a number of factors, such as the set of honest authorities, the number of honest voters, the number of voting options, and probability distributions for these voting options. In other words, the framework precisely measures to what extent specific adversarial capabilities (given in terms of dishonest authorities and voters) suffice to cause specific impact on a specific requirement. In spite of its contribution, the framework does not provide an interface to election officials and does not incorporate election settings (*e.g.* by means of probabilistic adversaries). Hence, the framework does not directly support election officials in evaluating a scheme's adequacy within concrete election settings. To that end, both works turn out to have complementary goals. Despite this difference, both works address quantitative security from

different directions and can therefore benefit from each other. We consequently foresee an integration of both approaches as future work.

10 Conclusion and Future Work

This work has addressed the research challenge of quantifying the security of Internet voting schemes within specific election settings. Therefore, the quantitative security evaluation framework SecIVo has been developed. SecIVo provides a set of security requirements and adversarial capabilities. By its conceptual nature, SecIVo builds the interface between system analysts and election officials. On the one side, system analysts deduce qualitative security models of Internet voting schemes on the basis of security requirements and adversarial capabilities. On the other side, election officials specify their election settings by means of the number of eligible voters and the number of expected voters, as well as probabilistic adversary models. On the foundation of qualitative security models and a specified election setting, SecIVo determines the satisfaction degrees of the scheme under investigation by the application of risk theory and Monte-Carlo simulations. As such, SecIVo forms the foundation for election officials to build their decision about the selection of an Internet voting scheme upon.

The applicability of SecIVo has been substantiated by a proof of concept application. Therefore, on the basis of a literature review, qualitative security models of Helios and Remotegrity have been determined and evaluated within three election settings. The findings indicate that neither scheme conceptually dominates the other, but rather the adequacy of both schemes depends on the election setting under consideration. While Remotegrity plays its strength when it comes to compromised voting devices, Helios might be the more appropriate choice whenever adversaries provoke direct interaction with voters.

For the future, we see several research directions: SecIVo builds upon a set of widely established security requirements. Yet, there are specific security requirements, which are beyond the current scope. For instance, there might be cases in which the fact whether a voter participated in the election or not might be private (aka. anonymity). In fact, considering requirements beyond security, such as usability and understandability, requires the consolidation of different metrics and scales. We intend to address these challenges in future work. Furthermore, we plan to encourage system analysts to deduce qualitative security models for a range of Internet voting schemes such as JCJ / Civitas [19,37], Pretty Good Democracy [63], Pretty Understandable Democracy [12]. Additionally, we intend to investigate whether the quantification approach by Küsters *et al.* can be combined with our construction to obtain the best of both approaches.

Acknowledgement. The authors would like to thank the reviewers for their constructive recommendations, which helped to improve this work significantly.

References

1. Adida, B.: Helios: Web-based open-audit voting. In: USENIX Security Symposium, pp. 335–348 (2008)
2. Allan, R., Billinton, R., de Oliveira, M.F.: An efficient algorithm for deducing the minimal cuts and reliability indices of a general network configuration. Reliability, IEEE Transactions on **25**(4), 226–233 (1976)
3. Almasizadeh, J., Azgomi, M.A.: Intrusion process modeling for security quantification. In: Availability, Reliability and Security (ARES), 2009 Fourth International Conference on, pp. 114–121. IEEE (2009)
4. Armando, A., Compagna, L.: Satmc: A sat-based model checker for security protocols. In: Logics in Artificial Intelligence, pp. 730–733. Springer (2004)
5. Aven, T.: Reliability/availability evaluations of coherent systems based on minimal cut sets. Reliability Engineering **13**(2), 93–104 (1985)
6. Bannister, F., Connolly, R.: A risk assessment framework for electronic voting. International Journal of Technology, Policy and Management **7**(2), 190–208 (2007)
7. Basin, D., Mödersheim, S., Vigano, L.: Ofmc: A symbolic model checker for security protocols. International Journal of Information Security **4**(3), 181–208 (2005)
8. Bella, G., Paulson, L.C., Massacci, F.: The verification of an industrial payment protocol: The set purchase phase. In: Proceedings of the 9th ACM conference on Computer and communications security, pp. 12–20. ACM (2002)
9. Benaloh, J., Quisquater, J.J., Vaudenay, S.: IACR 2010 election report (2010)
10. Binder, K.: Introduction: Theory and technical aspects of Monte Carlo simulations. Springer (1986)
11. Biondi, F., Legay, A.: Quantitative anonymity evaluation of voting protocols. In: Software Engineering and Formal Methods, Lecture Notes in Computer Science, pp. 335–349. Springer International Publishing (2015)
12. Budurushi, J., Neumann, S., Olembo, M.M., Volkamer, M.: Pretty understandable democracy-a secure and understandable internet voting scheme. In: Availability, Reliability and Security (ARES), 2013 Eighth International Conference on, pp. 198–207. IEEE (2013)
13. Buldas, A., Mägi, T.: Practical security analysis of e-voting systems. In: Advances in Information and Computer Security, pp. 320–335. Springer (2007)
14. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Advances in CryptologyEurocrypt 2003, pp. 255–271. Springer (2003)
15. Carlos, M.C., Martina, J.E., Price, G., Custódio, R.F.: An updated threat model for security ceremonies. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 1836–1843. ACM (2013)
16. Cetinkaya, O.: Analysis of security requirements for cryptographic voting protocols. In: Availability, Reliability and Security (ARES), 2008 Third International Conference on, pp. 1451–1456. IEEE (2008)
17. Chaum, D., Carback, R., Clark, J., Essex, A., Popoveniuc, S., Rivest, R.L., Ryan, P.Y., Shen, E., Sherman, A.T.: Scantegrity ii: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. EVT **8**, 1–13 (2008)
18. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM **24**(2), 84–90 (1981)
19. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: A secure voting system. Tech. rep., Cornell University (2007)
20. Coney, L., Hall, J.L., Vora, P.L., Wagner, D.: Towards a privacy measurement criterion for voting systems. In: Proceedings of the 2005 national conference on Digital government research, pp. 287–288. Digital Government Society of North America (2005)
21. Cortier, V., Galindo, D., Glondu, S., Izabachène, M.: Election verifiability for helios under weaker trust assumptions. In: Computer Security-ESORICS 2014, pp. 327–344. Springer (2014)
22. Cortier, V., Galindo, D., Glondu, S., Izabachne, M.: A generic construction for voting correctness at minimum cost - application to helios. IACR Cryptology ePrint Archive **2013**, 177 (2013)

23. Cortier, V., Smyth, B.: Attacking and fixing helios: An analysis of ballot secrecy. Journal of Computer Security **21**(1), 89–148 (2013)
24. Cuvelier, E., Pereira, O., Peters, T.: Election verifiability or ballot privacy: Do we need to choose? In: Computer Security–ESORICS 2013, pp. 481–498. Springer (2013)
25. Delaune, S., Kremer, S., Ryan, M.: Coercion-resistance and receipt-freeness in electronic voting. In: Computer Security Foundations Workshop, 2006. 19th IEEE, pp. 12–pp. IEEE (2006)
26. Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols. Journal of Computer Security **17**, 435–487 (2009). DOI 10.3233/JCS-2009-0340
27. Dolev, D., Yao, A.C.: On the security of public key protocols. Information Theory, IEEE Transactions on **29**(2), 198–208 (1983)
28. Driels, M.R., Shin, Y.S.: Determining the number of iterations for monte carlo simulations of weapon effectiveness. Tech. rep., DTIC Document (2004)
29. EAC Advisory Board and Standards Board: Threat trees and matrices and threat instance risk analyzer (tira) (2009)
30. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Advances in CryptologyAUSCRYPT'92, pp. 244–251. Springer (1993)
31. Fuqua, N.: Reliability engineering for electronic design, vol. 34. CRC Press (1987)
32. Haber, S., Benaloh, J., Halevi, S.: The helios e-voting demo for the IACR. International Association for Cryptologic Research. http://www. iacr. org/elections/eVoting/helios-Demo. pdf (2010)
33. IACR: IACR 2010 Election (2010). URL https://vote.heliosvoting.org/helios/elections/85db4808-cc46-11df-a972-12313f025959/view
34. IACR: A short explanation of helios for cryptographers (2010). URL http://www.iacr.org/elections/2010/HeliosForCryptographers.html
35. IACR: About the Helios System (2016). URL http://www.iacr.org/elections/eVoting/about-helios.html
36. Iida, Y., Wakabayashi, H.: An approximation method of terminal reliability of road network using partial minimal path and cut sets. In: Transport Policy, Management & Technology Towards 2001: Selected Proceedings of the Fifth World Conference on Transport Research, vol. 4 (1989)
37. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Proceedings of the 2005 ACM workshop on Privacy in the electronic society, pp. 61–70. ACM (2005)
38. Kahn, J., Linial, N., Samorodnitsky, A.: Inclusion-exclusion: Exact and approximate. Combinatorica **16**(4), 465–477 (1996)
39. Kim, H.M., Nevo, S.: Development and application of a framework for evaluating multi-mode voting risks. Internet Research **18**(1), 121–135 (2008)
40. Kremer, S., Ryan, M., Smyth, B.: Election verifiability in electronic voting protocols. In: ESORICS, *Lecture Notes in Computer Science*, vol. 6345, pp. 389–404. Springer (2010)
41. Küsters, R., Truderung, T.: An epistemic approach to coercion-resistance for electronic voting protocols. In: Security and Privacy (SP), 2009 30th IEEE Symposium on, pp. 251–266. IEEE (2009)
42. Küsters, R., Truderung, T., Vogt, A.: Verifiability, privacy, and coercion-resistance: New insights from a case study. In: Security and Privacy (SP), 2011 IEEE Symposium on, pp. 538–553. IEEE (2011)
43. Küsters, R., Truderung, T., Vogt, A.: Clash attacks on the verifiability of e-voting systems. In: Security and Privacy (SP), 2012 33rd IEEE Symposium on, pp. 395–409. IEEE (2012)
44. Ksters, R., Truderung, T., Vogt, A.: Accountability: definition and relationship to verifiability. In: Proceedings of the 17th ACM conference on Computer and communications security, pp. 526–535. ACM (2010)
45. Landau, D.P., Binder, K.: A guide to Monte Carlo simulations in statistical physics. Cambridge university press (2014)
46. Langer, L.: Privacy and verifiability in electronic voting. Ph.D. thesis, TU Darmstadt (2010)

47. Lauer, T.W.: The risk of e-voting. Electronic Journal of e-Government **2**, 177–186 (2004)
48. Lee, W.S., Grosh, D.L., Tillman, F.A., Lie, C.H.: Fault tree analysis, methods, and applications: A review. Reliability, IEEE Transactions on **34**(3), 194–203 (1985)
49. Luna, J., Suri, N., Krontiris, I.: Privacy-by-design based on quantitative threat modeling. In: Risk and Security of Internet and Systems (CRiSIS), 2012 Seventh International Conference on, pp. 1–8. IEEE (2012)
50. Madan, B.B., Goševa-Popstojanova, K., Vaidyanathan, K., Trivedi, K.S.: A method for modeling and quantifying the security attributes of intrusion tolerant systems. Performance Evaluation **56**(1), 167–186 (2004)
51. Moran, T., Naor, M.: Receipt-free universally-verifiable voting with everlasting privacy. In: Advances in Cryptology-CRYPTO 2006, pp. 373–392. Springer (2006)
52. Mundform, D.J., Schaffer, J., Kim, M.J., Shaw, D., Thongteeraparp, A., Supawan, P.: Number of replications required in monte carlo simulation studies: a synthesis of four studies. Journal of Modern Applied Statistical Methods **10**(1), 4 (2011)
53. Neumann, S., Budurushi, J., Volkamer, M.: Analysis of Security and Cryptographic Approaches to Provide Secret and Verifiable Electronic Voting, chap. 2, pp. 27–61. Design, Development, and Use of Secure Electronic Voting Systems. IGI Global (2014)
54. Neumann, S., Olembo, M.M., Renaud, K., Volkamer, M.: Helios verification: To alleviate, or to nominate: Is that the question, or shall we have both? In: Electronic Government and the Information Systems Perspective, pp. 246–260. Springer (2014)
55. Nevo, S., Kim, H.: How to compare and analyse risks of internet voting versus other modes of voting. Electronic Government, an International Journal **3**(1), 105–112 (2006)
56. Ouchani, S., Jarraya, Y., Mohamed, O.A.: Model-based systems security quantification. In: Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on, pp. 142–149. IEEE (2011)
57. Pardue, H., Landry, J., Yasinsac, A.: A risk assessment model for voting systems using threat trees and monte carlo simulation. In: Requirements Engineering for e-Voting Systems (RE-VOTE), 2009 First International Workshop on, pp. 55–60. IEEE (2010)
58. Pardue, H., Landry, J.P., Yasinsac, A.: E-voting risk assessment: A threat tree for direct recording electronic systems. International Journal of Information Security and Privacy (IJISP) **5**(3), 19–35 (2011)
59. Pardue, H., Yasinsac, A., Landry, J.: Towards internet voting security: A threat tree for risk assessment. In: Risks and Security of Internet and Systems (CRiSIS), 2010 Fifth International Conference on, pp. 1–7. IEEE (2010)
60. Paulson, L.C.: Proving properties of security protocols by induction. In: Computer Security Foundations Workshop, 1997. Proceedings., 10th, pp. 70–83. IEEE (1997)
61. Pereira, O.: personal communication (2014)
62. Rubinstein, R.Y., Kroese, D.P.: Simulation and the Monte Carlo method, vol. 707. John Wiley & Sons (2011)
63. Ryan, P.Y., Teague, V.: Pretty good democracy. In: Security Protocols XVII, pp. 111–130. Springer (2013)
64. Sampigethaya, K., Poovendran, R.: A framework and taxonomy for comparison of electronic voting schemes. Computers & Security **25**(2), 137–153 (2006)
65. Schryen, G., Volkamer, M., Ries, S., Habib, S.M.: A formal approach towards measuring trust in distributed systems. In: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 1739–1745. ACM (2011)
66. Smyth, B.: Replay attacks that violate ballot secrecy in Helios. Tech. rep., Cryptology ePrint Archive (2012)
67. Stoneburner, G., Goguen, A., Feringa, A.: Risk management guide for information technology systems. Tech. rep., National Institute of Standards and Technology Special Publication 800-30 (2002)
68. Vaurio, J.K.: An implicit method for incorporating common-cause failures in system analysis. Reliability, IEEE Transactions on **47**(2), 173–180 (1998)
69. Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In: Advances in Cryptology-CRYPTO 2009, pp. 619–636. Springer (2009)
70. Zagórski, F., Carback, R.T., Chaum, D., Clark, J., Essex, A., Vora, P.L.: Remotegrity: Design and use of an end-to-end verifiable remote voting system. In: Applied Cryptography and Network Security, pp. 441–457. Springer Berlin Heidelberg (2013)