Understanding the Development Trends of Electronic Voting Systems

Ali Fawzi Najm Al-Shammari1,2
1Faculty of Comp. Science
Free University of Bolzano
Bolzano, Italy
Email: alifawznajm.alshammari@stud-inf.unibz.it

Adolfo Villafiorita, Komminist Weldemariam
2Center for Information Technology,
Fondazione Bruno Kessler (FBK-irst)
Trento, Italy
{alshammar,adolfo,sisai}@fbk.eu

Abstract—During the past few years a huge interests in e-voting has occurred, which resulted in a significant funding for e-voting research and development (R&D) projects with the aim of developing trustworthy e-voting systems. Although it is good that major e-voting development trends are capitalizing on previous R&D efforts in other domains, it is difficult to say that the necessary R&D efforts has been utilized to realize the vision and principles of e-voting. Today, we now have a much better understanding of the issues that pose the development of a trustworthy e-voting system. In this paper, we categorize and analyze the most important research and development trends in e-voting systems. We identify the lessons learned from these projects and analyze whether the existing R&D projects meet the identified flaws in past. Finally, based on these results the way to the future of e-voting R&D is postulated.

Keywords—e-voting, requirements, security, evaluation and certification, formal analysis, process models, vote verification,

I. WHY THE INTEREST IN E-VOTING SYSTEMS?

The simple answer to this question is, because these systems offer the possibility of out-performing traditional voting in many ways. These include accessibility and prevention of voter’s mistakes — e.g., preventing overvotes and feedback on undervotes until before the voter confirms to the final decision; audio interfaces to let visually impaired voter without assistance; the ballot management is vastly simplified using some kinds of memory cards instead of paper; moreover, direct-recording electronic (DRE) voting machines can save the costs associated with producing and securing paper ballots [1].

Not surprisingly, as the technology for e-voting changes along with an evolution of regulatory environment, many open questions emerge. Namely, the use of computer systems for voting process has introduced serious risks in security, trust and reliability concerns about voting machines and the voting process itself [2], [3]. At the same time, there is a huge demand for applying appropriate engineering techniques to help the correct construction of trustworthy e-voting systems to mitigate some of the consequences and associated risks with them. So far, however, it does not seem that known software or security engineering techniques are explored in the way needed for the correct construction of such systems [4]. Also, works to rigorously define e-voting properties, attack models and languages for describing the counter-measurements are unsatisfactory [5].

In fact a range of development trends that span from modeling and formal analysis, trusted computing to hardware separation techniques have been explored and proposed. Several evaluation standards, requirements development strategies, as well as works that evaluate the security of deployed e-voting systems have been used. Works that evaluate the security of deployed e-voting system are discussed in, e.g., [6], [4], [7]. Yet, even a casual analysis of social acceptance suggested that we have a long way to go in terms of providing a mean through which we convince how individual vote is actually recorded and stored. These all show that there are several progresses which have been made (also on going) to understand and support the development of e-voting systems. These progresses can fall on understanding the three-dimensional gap: the technological gap — that is, between hardware and software, the socio-technical gap — that is, between social and computer policies, and the social gap — that is, between social policies and human behavior, as noted prior in [8]. This underlines the fact that e-voting systems are not just to ensure technical security, but also must ensure all the non-technical (procedurally and/or environmentally guaranteed) aspects of the system and the processes themselves.

However, there is no detailed classification to understand the common characteristics and limitations of the existing works. Without having such a comprehensive study, it is hard for developers and engineers to choose appropriate development and engineering practices. Thus, this paper synthesizes the current situation with regard to the state of the art in e-voting R&D practices. We characterize where we are, what has been done right, what has not worked well, and what we believe are the key directions for future research in the area. Finally, based on our investigation of the current challenges, we indicate some major directions to be taken for the future development of trustworthy e-voting systems.

II. CLASSIFICATION OF THE DEVELOPMENT TRENDS

Although the fundamental principles of voting are similar, the design and implementation with respect to the type of e-voting systems is quite different. For instance, the
requirements, design and implementation of a remote on-
line voting system is different than stand-alone DRE system.
However, what is common to all e-voting systems is that
after the ballot is cast votes are directly stored internally.

Table I shows our categorization of the current R&D
trends in the development of e-voting systems and/or ma-
chines into five areas. The following sections discuss the
most important progresses that are made along the above
listed areas.

III. PROGRESS ON DEVELOPMENT ASPECTS

In this section, we focus on the process and development
aspect of the existing works. More specifically, we discuss
works that focus on requirement engineering, business pro-
cess and implementation of e-voting systems.

A. Requirements Engineering and BPR

An essential activity to ensure an e-voting system behaves
correctly is laying down what behaving correctly means for
that system. Along this line, several countries (e.g., U.S.A
[35], Council of Europe [36]) around the world as well as
several researchers —e.g., [10], [11], [12], [13], [9]—
have defined requirements for voting systems or machines.
These documents can be classified according to the context
in which the requirements have been developed, the type
of e-voting system addressed, the categories in which the
requirements are classified, and the level of detail for the
requirements.

In case of stand-alone e-voting machines, the require-
ments are distinguished among organizational, certification,
and technical requirements. The organizational ones mainly
define how to deliver e-voting machines for an election, how
a user guide must look like, and how to check whether
an e-voting machine is the one that has been evaluated
and approved. The technical requirements define both the
technical assembly and the actual functionality of the sys-
tem. Furthermore, the evaluation or certification of stand
alone e-voting machines is typically constrained by a certain
regulation. This regulation defines the responsibilities for
(re)evaluation, certification, and revocation of the system and
its usage scenario. However, it does not define the evaluation
methodology itself, such as the evaluation techniques in use,
the evaluation depth, or the underlaying trust model. For
the remote e-voting machines, in contrast, the recom-
mandation of the Council of Europe document [36] provided a
comprehensive list of requirements. This document distin-
guishes among the legal standards, procedural safeguards,
operational standards, and technical requirements. However,
the way these documents describe (security) requirements is
fairly hard to understand.

The authors in [12], [9] attempted to narrow (some of)
the above difficulties by providing structured (software)
ingineering techniques. More specifically, the author in [12]
presented an approach for analyzing the root causes of
conflicts in requirements and organizational or procedural
barriers for running elections. Their approach uses top-down
and bottom-up strategies to develop requirements for critical
election. The top-down strategy is aimed at developing a
set of requirements from an existing catalogue, whereas
the bottom-up focuses on the development of a new set
of requirements from scratch which are then categorized
according to several abstract concepts. The author in [9]
was able to deduce technical requirements from correspond-
ing election laws and/or regulations by using mathematical
oriented methodology. The methodology was also used to
cross-check such requirements with existing catalogues and
with possible threats. This can further allow software engi-
neers and developers to easily understand how their system
meets these requirements.

With respect to BPR, the major works focus on under-

<table>
<thead>
<tr>
<th>Work</th>
<th>Brief description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Engineering, e.g., [9], [10], [11], [12], [13]</td>
<td>Contribute to the definition, development and structuring of requirements for e-voting systems with a clear separation between functional and non-functional requirements (e.g., audits log features) as well as the specifications for various hardware components’ requirements.</td>
</tr>
<tr>
<td>Business Process (Re)engineering, e.g., [14], [15], [16], [17], [18]</td>
<td>Understand the effective implementation of e-voting procedures, namely by using BPR to understand what changes could be introduced to the conventional voting procedures to allow a safe and secure transition to electronic elections.</td>
</tr>
<tr>
<td>Design &amp; Implementation, e.g., [19], [20], [21], [22], [23], [24]</td>
<td>Design of cryptographic schemes, protocols, and/or techniques to improve the design of voting systems or machines, as well as the actual implementation of the voting systems themselves.</td>
</tr>
<tr>
<td>Security Evaluation, e.g., [4], [6], [7], [25], [26], [27]</td>
<td>Combine different security engineering techniques to test and/or analyze the security posture of e-voting systems. The works in this area mainly assessed the security of e-voting systems used in real-world elections, and identified procedures that may eliminate or mitigate discovered issues.</td>
</tr>
<tr>
<td>Formal Methods, e.g., [28], [29], [30], [31], [32]</td>
<td>Apply formal specification and verification techniques (such as theorem provers and model checkers) to analyze the security of deployed e-voting systems, thereby ensuring the correctness of the voting process, and the underlying infrastructure mathematically.</td>
</tr>
<tr>
<td>Vote Verification Methods, e.g., [33], [34]</td>
<td>Apply software and/or hardware techniques attempting to ensure that cast votes match with voters’ selections.</td>
</tr>
</tbody>
</table>
standing the business context and model of the electoral system prior to introduce electronic solutions. The first research on this topic was actually discussed in [14]. In this work, the authors used BPR to evaluate the introduction of an e-voting system in UK. They evaluated the underlying procedures with respect to possible risks through procedural security analysis [37], [38], also by highlighting (some of) the security implications of the administrative workflow (like [15]) in e-voting. However, their approach lack a machinery to systematically model and analysis the procedural alternatives nor provide precise notions for the redesigning activities. To complement this, Mattioli [16] developed a generic methodology based on UML. The methodology is further refined and supported with a tool in [39], to facilitate the modeling, analyzing and structuring of electoral procedures as business process models. The methodology and the tool have been effectively used in the development of ProVotE e-voting system [19].

An approach to reason on security properties of the to-be (i.e., the software delivery) models in order to formally evaluate procedural alternatives in e-voting systems is presented in [40]. Here, the authors aimed at understanding the problematic trust/delegation relationships and eventually finding ways to adopt a solution to detect security properties violations at design time. Finally, a methodology based on formal notion specifically to support the procedural security analysis is discussed in [17].

B. Design and Implementation

Various cryptographic protocols have been developed aimed at ensuring a voter can be certain about her/his vote has been recorded correctly and accurately (voter verifiability), no voter can prove to anyone else how s/he voted (receipt freeness), and an independent body can verify that the recorded votes exactly match with the published tally after the election. For example, among the others we mention PunchScan [41], Scantegrity [42] and Prêt à Voter [24] schemes which are based on pure cryptographic protocols. Due to space limitation, we omit the technical details of these schemes.

Recently, the use of specialized techniques —such as, pre-rendering user interface and hardware separation techniques— borrowed from other domain have been proposed to the development of e-voting systems. These approaches attempt to achieve a certain level trustworthy during the designing and implementation of voting systems. For instance, Paul and Tanenbaum [43] implemented a DRE-based voting system by exploring the TPM infrastructures (e.g., PKI, hardware protection of cryptographic keys). The authors presented a scheme that improves registration integrity, and introduced a design that prioritizes election integrity. Their voting system has a nine-step as a whole, which takes place from an election’s inception to its final conclusion. Here it should be noted that TPMs are attractive for use in voting machines mainly for the hardware protection of cryptographic keys.

In another view, in [44] an approach that combines two techniques with a particular focus on designing voting machines for verification is proposed. The first focus is on creating a trustworthy vote confirmation process for which the author proposed an architecture that splits the vote confirmation code into separate modules whose integrity is protected using hardware isolation techniques. The author then proposed a hardware resets technique that restores the state of modules components to a consistent initial value between consecutive voters, thereby eliminating the risk of privacy breaches and ensuring that all voters are treated equally by the systems.

ProVotE is a DRE-base e-voting system with a unique characteristics from the development and experimentation viewpoints [19]. Among them, for instance, the use of a model-driven integrated development based on UML; the incorporation of formal specification and verification using the state of the art model checker; the introduction of signaling system to display the status of the machine to poll workers such that the poll workers can check if the voter has completed voting process correctly or withdrew from voting; the choice of Java and Linux operating system (customized, with specific functionality); the incorporation of feedback gathered from various experiments and users experiences on the machine usages and performance of the machine— to the development of the machine.

Helios is the first web-based cryptographic e-voting system [58]. It has a single trusted component, the Helios server, and uses a public bulletin board (BB). In registration phase, voters obtain their password by email. Helios separates the ballot preparation from ballot casting. Using the system anyone can generate and audit ballots as voters are authenticated only at ballot casting time. After the ballot has been generated, the voting system commits to the encrypted vote by displaying a hash of the ciphertext. Next, the correct preparation of the ballot can either be audited, or the ballot can be cast after the voter has been authenticated. If the voter chooses to audit the ballot, the ciphertext and the randomness used for encryption is displayed, which allows for checking that the vote was correctly transformed into the ballot. Once the ballot is cast the voter obtains a hash of his/her encrypted vote, which is also posted on the BB next to the voter name. In tallying phase the ballots are shuffled and decrypted, providing proofs of correctness for both steps.

IV. SECURITY, RELIABILITY AND TRUST

This section discusses works which attempt to improve the security, reliability and trust-worthiness of voting system by analyzing already deployed systems and/or by proposing a new techniques to ensure vote integrity.
A. Security Assessment

With respect to the security assessment, existing works can be grouped into two complementary areas: security evaluation and the usage of formal methods to systematically assist the security analysis activity.

1) Security Evaluation: As said earlier, some e-voting systems currently deployed for elections have recently undergone a thorough and independent scrutiny to evaluate their security and quality, as demonstrated by various academic researches [26], [46], [47], [6], [4], [7]. These works assess both the hardware and software of different forms of e-voting machines (e.g., Diebold/Premier, ES&S, Inter-Civic, Indian EVM). The studies identified serious design and implementation flaws that are notable for their level of egregiousness. More specifically, these analyses have showed that the current e-voting systems are vulnerable to very serious attacks. The authors in [48] also analyzed the security of Scantegrity II [42] based on a rigorous cryptographic definition of coercion-resistance in order to provide an optimal level of coercion-resistance. The results of all these analyzes suggested a drastic change in the way in which e-voting systems are designed, developed, and tested. Moreover, some of these the developed methodologies and toolsets (e.g., [4]) to support the analysis process, and the experiences gained provide useful lessons that can be adapted for similar projects.

On top of the above technical security evaluations, works such as [9], [25] focus on the definition of methodologies for the evaluation and certification of e-voting systems. Some works are also on going with respect to the development of common standards in order to provide a basis for standardized evaluations with comparable results. The Protection Profile [49] of the Common Criteria [50] is one of these efforts. At the time of writing this paper, the developed Protection Profile focuses more on remote electronic voting (e.g., the Polya system [51]) and on digital election pen [9].

Schmidt et al. [25] also discuss an evaluation and certification approach for Voting Service Providers (VSP), by combining the evaluation of remote e-voting system and operational environment. The VSP is required to provide a security concept in which it demonstrates satisfaction of the security requirements defined in the legal regulation. The authors suggested the incorporation of existing evaluation methodologies to facilitate the evaluation and certification process.

2) Formal Analysis and Verification for e-voting: The use of formal methods such as model checking and theorem provers can provide higher assurances for the design and implementation of e-voting systems. More specifically, formal techniques allow designers to prove, test, or otherwise examine interesting properties of a complex process whose behavior is specified abstractly, and then interactively refine the behavioral specification to be as close to an implementation as appropriate for a given assurance level. The current trends in this area can be grouped into three closely related verification focuses: verifying cryptographic protocols, e.g., [28], [29], system behavior, e.g., [30], [31], [32], and procedures, e.g., [17], [40].

It is clear that the use of formal techniques for the verification of e-voting system models can help to address problems associated with assurance against critical properties, and can improve the trustworthiness of the final systems. The current trends can be grouped into three closely related verification focuses: verifying cryptographic protocols, e.g., [28], [29], system behavior, e.g., [30], [31], [32], and procedures, e.g., [17], [40].

With respect to the first (i.e., verifying cryptographic protocols), the authors in [28] particularly present a framework for formal specification and verification of e-voting protocols. These properties are vote-privacy, receipt-freeness, and coercion-resistance. Their work mainly focused on formally verifying the correctness of protocols with respect to these properties. The authors used applied π-calculus to formalize these properties as observational equivalence, after being formalized the voting protocol as a set of processes using the same machinery.

With respect to the behavioral specification and verification, an approach for the design and analysis of an e-voting machine based on a combination of formal verification and systematic testing is presented in [19], [31]. More specifically, the authors in [31] verify the correctness of each of the individual component of voting machine, as well as verifying some crucial correctness properties of the composition. This work is particularly targeted to the following verification goals: ensuring that each individual component of the voting machine and their composition should meet the specification of the individual components and their composition respectively; voting machine should be structured to enable sound systematic system testing; ensuring that the voting machine must behave and store votes according to the voters selection when configured with a particular election definition file. For each module, they construct a formal specification that fully characterizes the intended behavior of that component. A number of properties related to the structural and functional aspects that the machine should satisfy are identified and specified. Similarly, in [19], the authors demonstrates the integration of formal methods in the development process of the voting system they developed in Italy by thoroughly analyzing the behavior of the system core functionality.

The above two usage scenarios of formal techniques do not focus on the aspects related to procedures in their modeling and analysis. The authors in [17] complemented the works discussed above by widening their analysis scope with procedures analysis. An approach to reason on security properties of the to-be models (which are derived from as-is model) in order to evaluate procedural alternatives in e-voting systems is presented [40]. Additionally, the authors in
Votegrity, Scantegrity, Prêt à Voter, and Votescript+ to ensure that the votes stored in the e-voting machine are calculated and will be compared with that of the integrity register records of the votes stored in the DRE record corresponding the vote. At the end of the election, memory by calculating and storing also an integrity register the VVM encrypts and stores the casted vote in its internal button on the device panel. Once the voter has confirmed, by reading on VVM screen and pressing an appropriate button on the device panel. Once the voter has confirmed, by pressing the confirmation button on DRE panel after having heard the generated message. In contrast, using VVPAT when a voter selects or cancels a candidate for a particular contest, an appropriate indication is printed on the VVPAT record [8], [53]. If the voter selects a candidate, the VVPAT record (i.e., paper audit trial) is marked as “Selected” and scrolled out of sight; otherwise, it is marked as “Canceled” and scrolled out of sight. The VVPAT is connected to the DRE and the paper record is viewable by the voter. The trails (i.e., the voter’s choices) are often under a transparent cover so that they cannot be modified other than through the normal voting process. The voter is eventually given the opportunity to review his ballot on both VVPAT window and on DRE screen, and if the voter commits to it (confirms it), it is recorded to local storage. In the VVATT method, all voice messages (confirmed or not) will be recorded in voice-operated tape recorder (VOR) to be used for auditing purposes. The verification process continues in this way for all qualified voters in both VVATT and VVPAT.

2) End-to-End Vote Verification: As noted by Ron Rivest during the NIST Workshop [54], end-to-end systems should employ cryptographic methods to craft receipts that allow voters to verify that their votes were not maliciously modified, without revealing which candidates were voted for. In line with this, a number of e-voting systems “claim-to-allow” end-to-end vote verification have been proposed. Among them, we mention Scantegrity, Prêt à Voter, VoteHere, Votegrity, and Votescript+. Scantegrity and Prêt à Voter are described in detail in several papers: more information, including all system description and requirements, can be found in [42], [23], [55], [24], [56]. The Votescript+ system allows independent end-to-end vote verification by enabling a voter to have encrypted copy of the vote stored in smartcard. After tallying phase, in Votescript+ system, a voter can verify his/her vote by going to dedicated place equipped with verification machines. The voter can insert the smartcard into the verification machine in order to communicate with a tally board for verification.

The other methods, instead, attempt to ensure vote verifi-
Table II
SUMMARY OF VOTE VERIFICATION METHODS.

<table>
<thead>
<tr>
<th>E-voting technology</th>
<th>Verification mechanisms used</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRE</td>
<td>by reading the vote selection on the verification device screen (e.g., Pnyx.DRE); by hearing the vote selection on voice message via headset (e.g., VV AATT); reading the printed audit trails (e.g., VVPAT) &amp; by receiving encrypted paper receipt (e.g., VoteHere &amp; Votegrity)</td>
</tr>
<tr>
<td>Optical Scan</td>
<td>Encrypted paper receipt (e.g., Scantegrity &amp; Pr ´et ´a)</td>
</tr>
<tr>
<td>Polling Station Remote Voting</td>
<td>Smart-card (e.g., Votescript+)</td>
</tr>
</tbody>
</table>

Table III
FEATURES OF VOTE VERIFICATION METHODS AT ELECTIONS PHASES

<table>
<thead>
<tr>
<th></th>
<th>At Elections</th>
<th>After Election</th>
<th>After Tallying</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>voter (vote recorded correctly)</td>
<td>election staff and observer (precinct auditing)</td>
<td>election staff (Auditing)</td>
</tr>
<tr>
<td>Pnyx.DRE</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>VV AATT</td>
<td>X</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>VVPAT</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VoteHere</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Votegrity</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Scantegrity</td>
<td>X</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>Votescript+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Prêt a Voter</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

cation by providing (encrypted) receipt. For instance, the vote verification methods implemented in VoteHere [33], [57] and Votegrity [21] systems allow an independent end-to-end vote verification with receipt. In case of VoteHere, a sentinel verification device is connected to DRE machine for printing the encrypted receipt for voter. Before casting, the device will calculate the code describing the voter selection and print it as a paper receipt. A corresponding code will be displayed in screen. The voter can then check the code printed as receipt that should match with the one displayed on the DRE screen, and can cast or cancel the vote accordingly. After announcing election result, a list of all codes will be announced in public website, so voters can check if their votes have been counted correctly.

In contrast, a small printer attached to Votegrity DRE prints the receipt in Votegrity system. The printer prints a double side two-layer transparent receipt in a way that when those layers combined then visualizing receipt contents is possible while each layer alone does not reveal receipt content. At this stage, the voter should select one side of and confirm the selection. After confirming the selections, the printer will print extra inch to selected layer that contains the final signature and serial code. This print is a final receipt for the vote such that the voter can use it after elections to verify his/her vote by entering the serial code to public website and check the vote has been counted as intended.

V. WHERE WE SHOULD BE HEADING

The first part of this section discusses (some) the issues in e-voting development with respect to the observations of the previous sections. In the second part of the section, we discuss the way forward to address some of these issues.

A. Common Challenges in eVoting Development

The problems as we see them with the state of the art in e-voting systems development include the following:

i) Lack of Tool-support. Existing requirement structuring methodologies mostly provide high-level principles and recommendations for e-voting systems development. A lack of tool to support the management and structuring of the requirements further creates difficulty in realizing the link between legal regulations and technical requirements, as well as in realizing requirements during development process.

ii) Less Engineering and Closed Development Practices. Existing e-voting systems are directed almost exclusively at large scale known engineering disciplines —e.g., failure to follow standard software and security engineering practices— and consequently a niche too small to drive significant adoption of technologies into our democracy. No currently deployed e-voting systems are freely available (closed development and often one vendor lock-in see iv). Thus, even the security experts who might be able to exploit the power of current e-voting systems are left using burdensome message passing systems like too classified military systems.

Moreover, while we are sure that security assessment is where the majority of the research effort will be spent, the research community do not directly involve in developing the system itself. Instead, they are active in attacking the system to find security vulnerabilities and we believe it is not the most important challenge that needs to be overcome. The state of e-voting research (and also the technology itself) has not progressed to the point where it could be used effectively to address a wide variety of real problems we faced in the
domain, such as making e-voting systems a fundamental system service that can be used with acceptable security and privacy level. In particular, the lack of open source e-voting systems has precluded this kind of acceptance by the research and citizens that e-voting could have been received. Finally, sharing and studying experimental data about the e-voting machines’ security, performance and their evolution with respect to the social and technical aspects are still relatively poor. This limits their usage in a larger scope as noted in [58].

With respect to vote verification methods analyzed, we observed the following common challenges.

iii) Exposure to Attacks. As we somehow anecdotally demonstrated in the previous section, various systems in use have vulnerabilities and weaknesses. Although procedures and adequate procedural controls are a way to mitigate and, in some cases, maybe eliminate such vulnerabilities, it is still true that, given sufficient resources it would be possible for determined attackers to compromise the results of an election. What is even worse, in some cases, the attack could remain undetected or, even, undetectable. Finally, reliance of a single solution opens up the opportunity for exploiting certain attacks on a large scale (think, e.g., spreading a virus).

iv) Vendor lock-in. Current solutions for e-voting do not interoperate. Procurement of e-voting solutions, thus, is often an “all-or-nothing” deal. That is, either all the systems to run an election are gotten from a single vendor or it will be very difficult, e.g., to integrate and tabulate data coming from different systems. Although reliance on a single vendor has its own advantages (e.g. simplified service and maintenance, procurement of “turn-key” solutions, and, of course, no data integration issues), it also prone to different critiques. Among them, for instance, the possibility of exploiting vulnerabilities on a large scale and, secondly, access to data and transparency.

v) Open Verifiability. Elections and (many systems for) e-elections rely on key stakeholders in order to ensure fairness and correctness of the overall process. Practical difficulties in running systematic audits make the reliance of such stakeholders even more important. This, however, does not have to be necessarily like that. The adoption of electronic systems in the polling stations, in fact, opens up the possibility for systematic and automated analyses of the data, e.g., to detect anomalies and attacks attempted during the electoral phases. What is more important, given the appropriate conditions, these analyses could be performed not only by the “traditional” stakeholders (e.g., Governments and accredited NGO’s), but by all actors with sufficient interest and skills to do so. This, in a sense, could be compared to “crowd-sourcing” the verification process. However, this is not yet possible, in practice, for various reasons, among which: the lack of an agreed-upon standard on what it has and it legally can be logged by systems; the lack of a technical standard defining how this information is to be stored and made available.

B. Possible Directions

This section discusses the possible directions that we believe needs to be considered in order to improve the current R&D trends in e-voting.

First of all, a tool-supported requirement engineering strategy is essential for managing and structuring requirements. This is because of the way different standards and recommendations which describe requirements is hard to understand and sometime contain contradicting or conflicting requirements. As noted earlier, works such as [9], [13], [45] present approaches to structure and highlight (some of) these standards in order to make them appropriate for the development of requirements for e-voting systems.

More specifically, according to [9] these requirements catalogues corresponds to standardized, consistent, and exhaustive list of requirements. This will allow having requirements that do not only describe properties that the system should meet, but also specify the corresponding laws or regulations for the evaluation of the systems themselves. Some preliminary results indicated that it is indeed possible to devise a methodology that comprises of cross checks existing catalogues, election principles, and possible threats based on the systematic formal analysis results. This would allow software engineers and developers to easily understand how their system meets these requirements. And, thereby it is possible to propose a formal evaluation and certification procedure by complementing the CC evaluation methodology.

Secondly, we need to understand the business model of the electoral process prior to introduce e-voting solutions. The obvious reason is that this helps to understand and discuss the possible risks that can be resulted by the introduction of new system. To do so, first we need to model and then analyze the resulting business models with respect to high-level reengineering goals.

The business layer reasoning can give information about the context of the business architecture (as-is) and the software delivery (to-be) prior to the subsequent development activities. Business process tools also help assessing the effectiveness of the processes as experienced and evaluated by the citizens outside the development and support organizations. The reason for this is that, unlike from firm processes, the electoral processes are tightly bounded by legal framework and are usually more regulated than business processes. Therefore, designing a methodology and tool is more of a necessity than a choice for reengineering activities to build trust among users. Some approaches such as the one given in [39], [40], [59] can be a starting point to exploit for reengineering process of e-voting projects.
Third, we need to understand the possible attack models in e-voting. Such attack model should allow to define the sequence of actions that can potentially lead to a particular attack under consideration. In principle, we can categorize attacks that target e-voting systems (elections in general) as: detectable vs undetectable, recoverable vs unrecoverable, and preventable vs detectable. In the first case, some attacks are undetectable no matter what practices are used while others are detectable in principle but are unlikely to be detected by the routine practices currently in place. In the second case, if an attack is detected, there is an easy way to recover while other attacks can be detected but there may be no good recovery strategy. In the third case, there is a trade-off between different strategies for dealing with attacks. One strategy is to design mechanisms to prevent the attack entirely, closing the vulnerability and rendering attack impossible.

Thus, further studies on attack surfaces and dimensions will allow us to effectively characterize the effects of attacks and also will provide a clear picture for the definition of a set of generic library of attack models corresponding to threat-actions. In line with this, some preliminary results have already been obtained to model and analyze attacks. This indicates that it is indeed possible to devise an effective technique capable of allowing for systematic injection of threats into the nominal voting process specification.

Fourth, the development of an e-voting system should be done more rigorously by integrating formal methods for reasons mentioned previously and elsewhere. Since the technique has been recognized as powerful and effective, drawing straight connection with this can help making better the current development of e-voting systems. As noted recently in [60], however, it is not yet clear how best to carry out such techniques to leverage the compositional nature of the problem, and manage the complexity of the task.

Within an initial investigation of requirements catalogues, it is possible to taxonomically classify those requirements that are of interest for formal verification, by evaluating their relevance with respect to relevant e-voting systems. This is because that once requirements’ are structured according to a given taxonomy it will be much easy to correspond to the type of properties we wish to verify (such as, like safety related, security related, and procedural or organizational related). Furthermore, this helps understanding which requirements can be verified formally when performing the actual analysis. Knowing this in advance reduces the effort of unnecessary verification process.

Finally, vote verification methods should be improved to allow public participation. This requires understanding the minimum data set that can be passed to public for open verification while keeping the secrecy and privacy of vote. For this purpose—in response to the challenges related to vote verification methods discussed previously—we are currently investigating to design a new open standard vote verification protocol. Specifically, we propose a new independent vote verification technique through an open standard protocol for a class of e-voting system (the DRE) to address some of the challenges identified previously.

We do so first, we define a robust verification process that can easily be configured for different elections and devices. Secondly, we define standards between e-voting and the verification process. Finally, we define a specific communication protocol for some steps of the voting process. These steps also require the use of formal techniques for modeling and analyzing the security and reliability properties of the proposed approach. Such open standard vote verification would able to make e-voting system more transparent to voter, third party and electoral staff while keeping privacy requirements, as well as would allow to capture vote threats and challenges during and after elections.

VI. Conclusion

There are several progresses which have been made in understanding and supporting the correct development of e-voting systems. However, the lack of their detailed classification has a little or no information to understand the pros and cons of these approaches, and thus little or no direction on choosing appropriate development techniques.

In this paper, we classified the most important R&D activities for the development e-voting systems in five major areas, according to what we believe to be their major contributions to the development of e-voting systems: understanding (the risks posed by the introduction of e-voting systems in the polling stations), requirements (developing requirements for e-voting), design and implementation (designing voting schemes, protocols, and/or techniques), security assessment and formal methods (using techniques and tools to analyze the security of existing systems), by giving lower-level and higher-level assurances). We also examined the weaknesses of existing vote verification methods from three points. The first is related to the scope in supporting verifiability — namely how robust they are in allowing verifiability for any interested party. The second is related to how these methods attempt to address or vulnerable to threat actions and verification challenges. Finally, we asked ourselves the following question: do the existing methods can easily be configured for different elections and devices? Since there is no common standard for verification in all existing methods, it makes such configuration of vote verification process difficult.

Finally, based on our investigation of the current challenges, we indicated some major directions to be taken for the specification and development of a trustworthy e-voting system. We believe this study contributes to the work done by designers, developers, certification authorities, as well as technical election officials.
REFERENCES


