A Survey: Electronic Voting Development and Trends

Komminist Weldemariam and Adolfo Villafiorita

Fondazione Bruno Kessler,
Center for Scientific and Technological Research (FBK-IRST)
via Sommarive 18
I-38050 Trento, Italy
(sisai, adolfo.villafiorita)@fbk.eu

Abstract: Any practitioner working on electronic voting (e-voting) seems to have different opinions on the main issues that seem to affect the area. On the one hand–given the criticality and the risk e-voting systems potentially pose to the democratic process–e-voting systems are permanently under a magnifying glass that amplifies any glitch, be it significant or not. On the other hand, given the interest e-voting raises within the general public, there seems to be a tendency to generalize and oversimplify. This tendency leads to attributing specific problems to all systems, regardless of context, situation, and actual systems used. Additionally, scarce know-how about the electoral context often contributes to make matters even more confused. This is not to say all e-voting systems show the security and reliability characteristics that are necessary for a system of such a criticality. On the contrary, a lot of work still has to be done. Starting from previous experiences and from a large-scale experiment we conducted in Italy, this paper provides some direction, issues, and trends in e-voting. Getting a clearer view of the research activities in the area, highlighting both positive and negative results, and emphasizing some trends could help, in our opinion, to draw a neater line between opinion and facts, and contribute to the construction of a next generation of e-voting machines to be safely and more confidently employed for elections.

1 Introduction

The advantages that e-voting systems can bring cannot be achieved without an observable cost (e.g., risks). One of which is opening up security vulnerabilities to attackers [Mer01, GGR07, BBC+08, BBC+10]. In that respect, recently we have seen that most currently deployed e-voting systems share critical failures in their design and implementation, which render their technical and procedural controls insufficient to guarantee trustworthy voting [LKK+03, KSRW04]. The lack of trust can also render even more secure and more reliable e-voting systems completely useless.

Clearly, the abundance of security threats in e-voting systems and their increasing popularity make a strong case for the need to propose new designs, protocols/schemes, techniques and tools for their design, development as well as their security assessment. The application and use of known techniques such as business process modeling and formal techniques and tools in voting, in general and in the development of an e-voting solution in particular, however are very limited and unsatisfactory. Additionally, work to rigorously define e-voting properties and attack models and languages to describe the counter-measurements is still more preliminary.

Although some progress has been made in understanding and supporting the better development of e-voting systems, e.g., [MN03, XM05b, XM07, WVM07, VWT09, DKR09], there is no classification to understand the common characteristics, objectives, and limitations of these approaches. Thus the lack of a comprehensive comparative study provides little or no direction on choosing the appropriate development techniques for particular needs.

In this paper, we classify the most important development approaches for e-voting systems and compare them with respect to motivations, methods, and logic. More specifically, we have classified them in four major categories, 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), IMPLEMENTATION (designing voting schemes, protocols, and/or techniques), and ASSURANCE (using techniques and tools to analyze the security of existing systems, by giving lower-level and higher-level assurances). We hope the work contributes to the work done by designers, developers, certification authorities, as well as technical election officials.

The paper is organized as follows. In Section 2 we review the use of (business) process modeling and redesigning to understand the context and risks caused by the introduction of electronic solutions in the polling stations. In Section 3, we briefly survey the progress made in developing requirements for e-voting systems. We continue, in Section 4, by briefly surveying progress made in designing and implementing voting schemes. In Section 5, we focus on the application of formal methods and techniques and tools to assess the security of e-voting systems. We conclude, in Section 6, by presenting some conclusive considerations and viewpoints.

2 Understanding Risks

Understanding the “context” of elections is very important prior to introducing e-voting solutions. The obvious reason is that this helps to understand and discuss the possible risks that can result through the introduction of a new system. Previous work in this area focused on the understanding, representation, and effective implementation of e-voting procedures. That is, using business process reengineering (BPR) to understand what changes could be introduced to the conventional voting procedures to allow a safe and secure transition to electronic elections.

The BPR concept pertains to the redesign in the context of existing business rules, such that the introduction of e-voting solution can be evaluated. As it is critical to define roles and responsibilities within the e-voting process which could furnish a better understanding of who is responsible for doing what during the different process stages to produce election results, it is also equally important to provide systematic methodology to deduce what could go wrong during this procedural rich workflow, instead of detecting the weaknesses well after attacks have already been taken.

As far as we are aware, the first use of BPR to evaluate the transition to e-voting is that proposed by Xenakis and Macintosh in [XM05b, XM07]. The authors investigated the need for business process reengineering to be applied to electoral process in order to propose a possible transition to an e-voting system. Risks and difficulties while introducing e-voting solutions are discussed, in more detail, in [XM04a, XM04b]. Furthermore, the same authors in [XM04c, XM05a] discussed the need for procedural security in electronic elections and provided various examples of procedural risks which occurred during trials in the UK. The approach can obviously highlight some of the security implications of the administrative workflow in e-voting, such as those discussed in [LKK+03]. However, these approaches do not provide techniques to systematically model and analyze procedural alternatives for better electronic solutions. Additionally, they do not provide ways to analyze the security aspect of these procedures. In other words, a systematic analysis of procedures is absent.

In references [Mat06, WVM07], the authors developed a UML-based methodology for modeling and analyzing electoral processes. The methodology is supported by a tool named VLPM [CMV09] that helps in the modeling, analysis and structuring of electoral procedures as business process models. Beyond that, the VLPM tool helps to assist a lawmaker to link laws with the process models, and a process engineer to analyze the effects of the changes due to the introduction of a new law (or law modification) on the models to maintain the "*synchronization*" of laws with models, as the same time by fostering collaboration between them, i.e. lawmaker and the process analyst. The methodology and the tool have been demonstrated for the development of an e-voting system named ProVotE [VWT09]. 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 [BDF+09]. In particular, using Datalog and the underlying analysis tool the authors expressed and analyzed security concerns, such as delegation of responsibility among untrusted parties and trust conflict. The aim is that of understanding problematic trust/delegation relationships and eventually finding ways to adopt a solution to the detected security properties violations.

3 Developing Requirements for E-voting

There are various international documents such as the recommendations from the European Union (EU) Venice Commission [Cou04] and the U.S. Federal Election Commission (FEC) Voting Systems Standard (VSS) [Fed02, Fed05], which describe a set of principles for voting systems. These documents mainly specify principles about the behaviors of each component of a voting system that should be respected, as well as the related procedures. The FEC-VSS, for instance, provides details about the standards to be used for performance and tests of voting machines. It also describes non-functional requirements (e.g., audits log features) and specifications for various hardware components. However, these kinds of requirements often make the development and implementation of the actual system difficult. Moreover, the way these documents describe (security) requirements is hard to understand, and sometimes they contain contradicting/conflicting requirements —specifically, the conflict between the requirements for secrecy and accuracy. If the e-voting system needs to be developed in a safe and secure way, there must be an appropriate requirements definition. We have surveyed dozen works in this area. Because of the limited space, however, we are able to present but a few of those that we think are the most important and complete.

Reference [Mer01] presents a thorough discussion on three gaps that must be comprehended prior to developing (security) requirements for e-voting systems. These gaps are 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. The same author also coined the term audit trails, which is often used in DRE machines. Namely, the type of DRE equipped with printed audit trails is often called DRE-VVPAT. That is, a touch-screen-based machine that produces a printout of each vote, verified directly by the voter, to maintain a physical and verifiable record of the votes cast. Thus an essential activity to ensure e-voting system behaves correctly is to lay down what behaving correctly means for that system. This cannot be achieved without a proper engineering approach, such as requirements engineering techniques.

The author in [McG08] presented an approach to address the mentioned problems by proposing a methodological approach for analyzing the root causes of the conflicts, organizational barriers (or procedural barriers), and requirements of a critical election. The approach comprises of two strategies for the development of requirements, namely, top-down and bottom-up. The first one is aimed at developing a set of requirements from an existing catalogue. The latter, instead is aimed at developing a new catalogue.

Subsequent to [McG08], Volkamer has provided, "*a standardized, consistent, and exhaustive list of requirements for e-voting systems*" [Vol09]. Specifically, these requirements are mostly for standalone DRE and remote e-voting systems. Such requirements not only describe requirements that the system should meet, but also specify the corresponding laws or regulations for the evaluation of the systems themselves. The author developed a methodology for the requirement development process. The results of the methodology include system requirements (divided into functional, security, and usability requirements), organizational requirements, and assurance requirements for both stand-alone DRE voting machines and remote e-voting systems. Furthermore, the methodology comprises of crosschecks, existing catalogues, election principles, and the possible threats. This could allow software engineers and developers to easily understand how their systems meet these requirements. Following that, the author proposed an evaluation and certification procedure mostly for remote voting systems by complementing the Common Criteria common evaluation methodology and also developing a protection profile for remote voting.

In reference [WMV09], the authors showed the management and structuring of requirements using finite state machines (FSMs). That is, by defining relationships between requirements and system architecture based on FSMs. More specifically, the methodology they followed allowed them to understand the election processes, identify constraints, and distinguish both common and event specific requirements from various requirements sources, e.g. from those mentioned above. These are then refined into fine-grained requirements using FSMs. The decomposition from high-level to low-level requirements and the logical dependencies among them have been demonstrated. Additionally, the separation between generic and election or configuration specific requirements is concrete and detailed enough to function as a general reference schema that could be adopted by other solutions. In other words, this approach is fairly general to be used for other e-voting systems and, possibly, to provide a roadmap —rough and draft as it might be— for bridging the gap between higher-level principles and lower level system specifications.

4 Designing Voting Schemes and/or Protocols

Prior works with respect to this area focused on the design of cryptographic schemes, protocols, and/or techniques to improve the design of voting machines. The ultimate goals of these approaches include ensuring a voter can be certain that 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 match exactly with the published tally after the election [Ive91, CFSY95, Cha04]. What is most common to all these approaches is that they rely on the underlying crypto- graphic principles to various degrees of complexity.

PunchScan [CPS+07, ECCP07] is a cryptographic voting system that is easy to use by the voter as well as by election officials, while at the same time providing a transparent and reliable process. It also provides public verifiability, election integrity and enhanced voter privacy. Scantegrity [CEC+08, CCC+09] is a successor of PunchScan that meets industrial standard by providing end-to-end verifiability of the integrity of critical steps in the voting process and election results. Prêt á Voter (verifiable electronic elections) [RBH+09] is a type of electronic voting system that uses paper based ballot forms that are converted to encrypted receipts to provide security and "auditability", at the same time remaining coercion resistant and easy to use. The Scratch & Vote is another cryptographic voting method proposed in [Adi06]. It provides public election "auditability" using simple, immediately deployable technology. The method combines a variety of existing cryptographic voting ideas such as homomorphic encryption —e.g., which allows votes to be tallied without decrypting individual votes, the cut-and-choose at the precinct approach, and so on. Additionally, works like [FOO93, BT94, RRN01, SCM08] attempt to provide (maximum) secrecy and/or anonymity for the vote and voter.

We cannot, however, say that cryptographic schemes and/or protocols address the current situation in the democratic process for several reasons. For example, the protocols that have been proposed so far do not yet overcome all of the barriers to their use in critical elections [McG08]; although DRE machines are very popular in public elections in some U.S. states, the applicability and scope of the proposed schemes are very limited in these machines. Moreover, as noted in [KSW05], some cryptographic protocols have some security holes, such that sensitive information about the election can be leaked in one way or another. Therefore, we must analyze their security by considering the system in its entirety since these protocols are only one part of a larger system composed of voting machines, software design and implementations, and complex election procedures [KSW05].

In reference [Sas07], the author presents the concept of "*designing voting machines for verification,*" aimed at providing techniques to help vendors, independent testing agencies, and others verify the critical security properties of DRE voting machines. The basis idea of the approach consists of two interesting techniques. The first focuses on creating a trustworthy vote confirmation process, where the author proposed an architecture that splits the vote confirmation code into separate modules whose integrity are protected using hardware isolation techniques. The second focuses on helping to ensure a very important property in voting, that is, "*None of a voter's interactions with the voting machine, including the final ballot, can affect any subsequent voter's sessions.*" In order to do that, the author used a hardware resets technique that restores the state of modules components to a consistent initial value between consecutive voters. With this, it could be possible to eliminate the risk of privacy breaches and ensure that all voters are treated equally by the systems.

Other works, such as [SKW06, Yee07] apply techniques used in other domains —like pre-rendering user interface and hardware separation— to build higher assurance with accessible, verifiable and secure e-voting systems. The design of a trustworthy DRE-based voting system by exploring the TPM (Trusted Platform Module) infrastructures (e.g., PKI, hardware protection of cryptographic keys) is presented in [PT09]. Additionally, the authors present a scheme that improves registration integrity, and introduces a design that prioritizes election integrity. Their voting system has nine steps as a whole, from an election's inception to its final conclusion.

5 Providing Assurances

With respect to the assurance of e-voting systems, existing works focus on two main areas to assess the security of e-voting systems. While the first one focuses on providing lower-level assurances, the other focuses on providing higher-level assurances; both use powerful techniques and tools.

5.1 Applying formal methods to e-voting

The use of formal methods in the specification and verification of e-voting systems is relatively new. Existing works in this area present formal specification and verification of an e-voting system at different levels of abstraction. These works aim to demonstrate how feasible the formal verification of voting machine logic, thereby providing a higher level of assurance about the security of the system. In this area the trends focus on three closely related aspects, mainly according to the aim of the verification. These are verifying cryptographic protocols, system behavior, and procedures.

The references [DKR09, KR05] present a framework for formal specification and verification of three privacy-type e-voting protocol properties. These properties are vote-privacy, receipt-freeness, and coercion-resistance. The authors used applied π-calculus [AF01] to formalize these properties as observational equivalence, after formalizing the voting protocol as a set of processes using the same machinery. In [CFM+08], the authors used a CCS (Calculus of Communicating Systems)-like process algebra with cryptographic primitives to specify and analyze some properties of the e-voting system they built. More specifically, they presented a small mobile implementation of an e-voting system named M-SEAS (Mobile Secure E-voting Applet System) and used formal verification technique to validate the security properties of the system.

The authors in [VWT09] demonstrate the integration of formal methods in the development process of a voting system. In particular, the authors specified the behaviors of voting control logic using a UML finite state machine and developed a tool named FSMC+[1] that automatically generates NuSMV [CCG+02] code corresponding to the specified FSM (this helped the requirements discussed in [VWT09]). Then they performed the verification using the NuSMV model checker. The results of the model checker, presented in the form of counter-measurement, are then analyzed. This enabled the authors to incorporate the analysis results of the verification into the actual development process of the core application.

In references [WKV09, WKV10], the authors show how formal methods can be used to reverse synthesize existing e-voting systems (named ES&S voting systems). They used the ASTRAL language to specify the ES&S voting process and used the PVS analysis tool. A number of critical security requirements that the machines should respect have been specified and analyzed against the specification. Subsequently, the authors specified known attacks against the system (as demonstrated in [MBV07]) using the same machinery and extended the original specifications, and then preformed the analysis on the extended model with the same set of critical security requirements that the original specifications should respect. The two main lessons drawn from their work are: formal methods help gain a better understanding of the security "boundaries" of e-voting systems, and the role that open specifications play in the development of more secure e-voting systems.

The reference [SJSW09] presents an approach for designing and analyzing of an e-voting machine based on a combination of formal verification and systematic testing. They formally verify the correctness of each of the individual components of the voting machine, as well as verify some of the crucial correctness properties of their composition. Their work is 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

[1] http://ict4g.fbk.eu/fsmcp/last/

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. They used Verilog [TM91] for the implementation of their specification and SMV[2] analysis tool and "satisfiability" solving (especially, the SMT solver) to verify that their Verilog implementation meets the specifications.

Finally, in reference [WV08], the authors proposed an approach to formally analyze procedures. Namely, they proposed a methodology based on the NuSMV [CCG+02] machine to analyze procedures systematically.

5.2 Assessing exiting e-voting systems

Some e-voting systems currently deployed in elections have recently undergone a thorough and independent scrutiny to evaluate their security and quality. This is because, in recent years, the DRE machines raised serious security concerns. These machines make the election process less verifiable and greatly expand the aspects of an election for which voters must rely solely on trust. Security vulnerabilities have been reported in each aspect of security—that is, technological, socio-technical, and social aspects, as noted prior in [Mer01]. These vulnerabilities have been systematically investigated and proved by various academic studies. This creates an enigma in the trustworthiness of the machine and the voting process as well.

In line with this, we mention the following academic researches [Jon03, KSRW04, GGR07, BBC+08, ASH+08]. These works assess both hardware and software of different forms of e-voting machines (e.g., Diebold/Premier, ES&S, InterCivic), mostly used in some U.S. states. The studies identified serious design and implementation flaws, which 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. In addition, they have produced a catalogue of vulnerabilities and possible attacks. Some analyses also suggested a drastic change in the way in which e-voting systems are designed, developed, and tested (e.g., by identifying procedures to eliminate or mitigate the discovered issues, by developing a precise methodology and toolsets for the assessment). The assessment methodology presented in [BBC+08, MBV07] is particularly astonishing as it provides various insights on each individual and in-depth step of the analysis. The software testing community can use it for the evaluation of other complex-security critical systems and evaluation.

[2] http://www.kenmcmil.com/

6 Discussions and Conclusion

There are a number of established approaches for modeling, specifying, and verifying a system satisfies a set of properties. One important contributor to the security of any system is the way in which the software is designed and developed. Standards for software engineering developed over the last forty plus years require that a system undergo a rigorous process of requirements definition, structured design and review, and careful programming and testing [Som95]. Like proper engineering leads to cars of higher quality, so too does better software engineering lead to more secure, robust software computer systems. Systems that are designed without this kind of careful design and implementation are almost certain to have flaws and security issues.

BPR techniques help to understand, model, and analyze the high-level context of the electoral processes. This provides information about the context of the business architecture (*as-is*) and software delivery (*to-be*) prior to the subsequent development activities for the introduction of an e-voting solution. It also helps in assessing the effectiveness of the processes as experienced and evaluated by the citizens outside the development and support organizations. However, it is not always possible to transform a business solution into an e-voting solution [AO05]. This is because, unlike business processes, the electoral processes are tightly bounded by legal frameworks and are usually more regulated than business processes. Thus, we need a proper methodology and tools that abet such reengineering activities. However, some approaches such as the one given in [CMV09] can be a starting point to extend and reuse in the reengineering process of e-voting projects.

The use of formal methods has been shown to improve the security and quality of complex systems. These approaches 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. In practice, moreover, the technique has been recognized as a powerful and effective mechanism for improving the security and quality of complex systems (e.g., in avionics). Thus, drawing a direct connection to this can help to improve the current development trends of e-voting machines.

Moreover, the studies of experimental data about the e-voting machines' security, performance and their evolution with respect to the social and technical aspects are still unsatisfactory. This limits their use on a larger scale. For example, data sets based on observing security threats to voters' anonymity by following standard procedures that illustrate each machine's behavior during elections can help raise the transparency in elections using electronic devices and increase the confidence of voters in the democratic system. Data sets related to the process of setting up experiments, running an election, and performing security evaluations across various voting machines (e.g., as in Diebold and ES&S) provide information about the behavior of machines under malicious circumstances, whether they are designed carefully or not, and provide recommendations that need to be considered for design alternatives.

Developing and deploying e-voting systems in a safe and secure manner requires ensuring the technical and procedural levels of assurance with respect to social and regulatory frameworks. In this paper, we have presented techniques mainly in three areas (namely, BPR, formal methods, and security) and showed how these techniques are effectively exercised for correct design and implementation of e-voting systems. Therefore, the success of the next generation of e-voting machines depends upon being able to capitalize one the lessons learned from different disciplines. The work we have presented in this paper is one way in which we can get a better understanding of the strengths and the weaknesses of existing techniques and thus lay the foundations for engineering, designing, implementing, as well as deploying a new generation of more secure and robust technologies for polling stations.

Bibliography

[Adi06] Adida, Ben 2006. Advances in cryptographic voting systems. PhD diss., Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology..

[AF01] Abadi, Martín, and Cédric Fournet. 2001. Mobile values, new names, and secure communication. *SIGPLAN Not* 36(3):104–115. New York, NY, USA: ACM.

[AO05] Alpar, Paul, and Sebastian Olbrich. 2005. Legal requirements and modelling of processes in e-government. *Electronic journal of e-government*, 3.

[ASH+ 08] Ansari, Nirwan, Pitipatana Sakarindr, Ehsan Haghani, Chao Zhang, Aridaman K. Jain, and Yun Q. Shi. 2008. Evaluating electronic voting systems equipped with voter-verified paper records. *IEEE Security and Privacy* 6(3):30–39: IEEE Computer Society.

[BBC+ 08] Balzarotti, Davide, Greg Banks, Marco Cova, Viktoria Felmetsger, Richard Kemmerer, William Robertson, Fredrik Valeur, and Giovanni Vigna. 2008. Are your votes really counted?: Testing the security of real-world electronic voting systems. In *ISSTA '08: Proceedings of the 2008 international symposium on software testing and analysis*, 237–248. New York, NY, USA: ACM.

[BBC+10] Balzarotti, D., G. Banks, M. Cova, V. Felmetsger, R. Kemmerer, W. Robertson, F. Valeur, and G. Vigna. 2010. An experience in testing the security of real-world electronic voting systems. *IEEE transactions on software engineering*.

[BDF+09] Bryl, Volha, Fabiano Dalpiaz, Roberta Ferrario, Andrea Mattioli, and Adolfo Villafiorita. 2009. Evaluating procedural alternatives: A case study in e-voting. *EG* 6(2):213– 231.

[BT94] Benaloh, Josh, and Dwight Tuinstra. 1994. Receipt-free secret-ballot elections (extended abstract). In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on theory of computing*, 544–553. New York, NY, USA: ACM.

[CCC+ 09] Chaum, D., R.T. Carback, J. Clark, A. Essex, S. Popoveniuc, R.L. Rivest, P. Ryan, E. Shen, A.T. Sherman, and P.L Vora. 2009. Scantegrity II: End-to-end verifiability by voters of optical scan elections through confirmation codes. *IEEE transactions on information forensics and security* 4(4):611–627.

[CCG+02] Cimatti, Alessandro, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. 2002. NuSMV 2: An open source tool for symbolic model checking. In *Computer aided verification, lecture notes in computer science*, 241–268. Berlin / Heidelberg: Springer.

[CEC+ 08] Chaum, David, Aleksander Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan T. Sherman, and Poorvi L. Vora. 2008. Scantegrity: end-to-end voter-verifiable optical-scan voting. *IEEE Security & Privacy*, 6(3):40–46: IEEE Computer Society.

[CFM+08] Campanelli, Stefano, Alessandro Falleni, Fabio Martinelli, Marinella Petrocchi, and Anna Vaccarelli. 2008. Mobile implementation and formal verification of an e-voting system. In *Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services,* Washington, DC, USA: IEEE Computer Society.

[CFSY95] Cramer, Ronald J.F., Matthew Franklin, L. A.M. Schoenmakers, and Moti Yung. 1995. Multi-authority secret-ballot elections with linear work. Technical report, CWI (Centre for Mathematics and Computer Science).

[Cha04] Chaum, David. 2004. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy* 2:38–47: IEEE Computer Society

[CMV09] Ciaghi, Aaron, Andrea Mattioli, and Adolfo Villafiorita. 2009. VLPM: a tool to support BPR in public administration. In *Proceedings of the Third International Conference on Digital Society (ICDS2009)*, 289–293: IEEE Computer Society.

[Cou04] Council of Europe. 2004. Recommendation on legal, operational and technical standards for e-voting. Council of Europe, September. [Available online at https://wcd.coe.int/ViewDoc.jsp?id=778189]

[CPS+ 07] Carback, Richard T., Stefan Popoveniuc, Alan T. Sherman, and David Chaum. 2007. Punchscan with independent ballot sheets: Simplifying ballot printing and distribution with independently selected ballot halves. In *Proceedings of the 2007 IAVoSS workshop on trustworthy elections (WOTE 2007)*. [Available online at http://punchscan.org/papers/ibs_carback.pdf]

[DKR09] Delaune, Stéphanie, Steve Kremer, and Mark Ryan. 2009. Verifying privacy-type properties of electronic voting protocols. *J. Computer Security* 17(4):435–487.

[ECCP07] Essex, Aleks, Jeremy Clark, Richard Carback, and Stefan Popoveniuc. 2007 Punchscan in practice: An E2E election case study. In *Proceedings of the 2007 IAVoSS Workshop on trustworthy elections (WOTE 2007)*, held in conjunction with 7th workshop on Privacy Enhancing Technologies, Ottawa, Canada.

[Fed02] Federal Election Commission. 2002. Voting system standards. USA: United States Election Assistance Commission, http://www.eac.gov/.

[Fed05] Federal Election Commission. 2005 Voluntary voting system guidelines (VVSG). USA: United States Election Assistance Commission, http://www.eac.gov/.

[FOO93] Fujioka, Atsushi, Tatsuaki Okamoto, and Kazuo Ohta. 1993.A practical secret voting scheme for large scale elections. In *ASIACRYPT '92: Proceedings of the workshop on the fheory and application of cryptographic techniques*, 244-251. London, UK, 1993: Springer-Verlag.

[GGR07] Gardner, Ryan, Sujata Garera, and Aviel Rubin. 2007. On the difficulty of validating voting machine software with software. In *EVT'07: Proceedings of the USENIX/accurate electronic voting technology on USENIX/accurate electronic voting technology workshop B*erkeley, CA, USA: USENIX Association.

[Ive91] Iversen, Kenneth R. 1991. A cryptographic scheme for computerized elections. In *CRYPTO '91: Proceedings of the 11th annual international cryptology conference on advances in cryptology,* 405–419. London, UK: Springer-Verlag.

[Jon03] Jones, Douglas W. 2003. The evaluation of voting technology, chapter 1. In Advances in Information Security, 3–16. Ed. Dimitrius Gritzalis: Kluwer Academic Publisher

[KR05] Kremer, Steve, and Mark D Ryan. 2005. Analysis of an electronic voting protocol in the applied pi-calculus. In *Proceedings of the 14th European symposium on programming (ESOP'05), lecture notes in computer science*, 186–200. Edinburgh, U.K.: Springer.

[KSRW04] Kohno, Tadayoshi, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. 2004. Analysis of an electronic voting system. IEEE Symposium on security and privacy, 0:27: IEEE Computer Society

[KSW05] Karlof, Chris, Naveen Sastry, and David Wagner. 2005. Cryptographic voting protocols: a systems perspective. In *Proceedings of the 14th conference on USENIX security symposium* Berkeley, CA, USA: USENIX Association.

[LKK+ 03] Lambrinoudakis, Costas, Spyros Kokolakis, Maria Karyda, Vasilis Tsoumas, Dimitris Gritzalis, and Sokratis Katsikas. 2003. Electronic voting systems: Security implications of the administrative workflow. In *Proceedings of the 14th international workshop on database and expert systems applications*, 467, Washington, DC, USA: IEEE Computer Society.

[Mat06] Mattioli, Andrea 2005-2006. Analisi dei processi in ambito di voto elettronico per le elezioni in Provincia di Trento. Master’s thesis, University of Trento.

[MBV07] McDaniel, P., M. Blaze, and G. Vigna. 2007. EVEREST: Evaluation and validation of election-related equipment, standards and testing. Ohio Secretary of State’s EVEREST project report. [available online at www.cs.ucsb.edu/~vigna/publications/2007_mcdaniel_blaze_vigna_voting.pdf]

[McG08] McGaley, Margaret. 2008. E-voting: An immature technology in a critical context. PhD diss., Department of Computer Science, National University of Ireland, Maynooth.

[Mer01] Mercuri, Rebecca T. 2001. Electronic vote tabulation checks and balances. PhD diss., University of Pennsylvania.

[MN03] Mercuri, Rebecca T., and Peter G. Neimann. 2003. Verification for electronic balloting systems, chapter 3. In *Advances in information security,* 31-42: Kluwer Academic Publishers.

[PT09] Paul, Nathanael, and Andrew S. Tanenbaum. 2009. The design of a trustworthy voting system. *Annual computer security applications conference (ACSAC)*, 507-517: IEEE Computer Society.

[RBH+ 09] Ryan, P.Y.A., D. Bismark, J. Heather, S. Schneider, and Zhe Xia. 2009. Prêt ǻ Voter: A voter-verifiable voting system. *IEEE transactions on information forensics and security* 4(4).

[RRN01] Ray, Indrajit, Indrakshi Ray, and Natarajan Narasimhamurthi. 2001. An anonymous electronic voting protocol for voting over the internet. In *WECWIS ’01: Proceedings of the third international workshop on advanced issues of e-commerce and web-based information systems*, 188. Washington, DC, USA: IEEE Computer Society.

[Sas07] Sastry, Naveen K. 2007. Verifying security properties in electronic voting machines. PhD diss., EECS Department, University of California, Berkeley.

[SCM08] Santin, Altair O., Regivaldo G. Costa, and Carlos A. Maziero. 2008. A three-ballot-based secure electronic voting system. *IEEE Security and Privacy* 6(3):14–21: IEEE Computer Society

[SJSW09] Sturton, Cynthia, Susmit Jha, Sanjit A. Seshia, and David Wagner. 2009. On voting machine design for verification and testability. ACM conference on computer and communications security (CCS’09), Chicago, Illinois, USA, November 9-13 , 463-476: ACM

[SKW06] Sastry, Naveen, Tadayoshi Kohno, and David Wagner. 2006. Designing voting machines for verification. In *Proceedings of the 15th conference on USENIX security symposium,* volume 15. Berkeley, CA, USA: USENIX Association.

[Som95] Sommerville, Lan. 1995. Software engineering (5th ed.). Addison Wesley Lonman Publishing Co., Inc. Redwood City, CA, USA.

[TM91] Thomas, Donald E., and Philip R. Moorby. 1991. The VERILOG hardware description language. Norwell, MA, USA: Kluwer Academic Publishers.

[Vol09] Volkamer, Melanie. 2009. Evaluation of electronic voting: requirements and evaluation procedures to support responsible election authorities. Springer Publishing Company, Incorporated: Springer.

[VWT09] Villafiorita, Adolfo, Komminist Weldemariam, and Roberto Tiella. 2009. Development, formal verification, and evaluation of an e-voting system with VVPAT. *IEEE transaction on information forensics and security* 4(4) 651--661.

[WKV09] Weldemariam, Komminist, Richard A. Kemmerer, and Adolfo Villafiorita. 2009. Formal analysis of attacks for e-voting system. In *CRiSIS ’09: Fourth international conference on risks and security of internet and systems*: IEEE.

[WKV10] Weldemariam, Komminist, Richard A. Kemmerer, and Adolfo Villafiorita. 2010. Formal specification and analysis of an e-voting system. In *The 5th international conference on availability, reliability and security (ARES 2010)*: IEEE Computer Society.

[WMV09] Weldemariam, Komminist, Andrea Mattioli, and Adolfo Villafiorita. 2009. Managing requirements for e-voting systems: Issues and approaches motivated by a case study. In *Proceedings of the first international workshop on requirements engineering for e-voting systems*: IEEE Computer Society.

[WV08] Weldemariam, Komminist, and Adolfo Villafiorita. 2008. Modeling and analysis of procedural security in (e)voting: The Trentino's approach and experiences. *In Proceedings of the conference on Electronic voting technology (EVT)*. Berkeley, CA, USA: USENIX Association.

[WVM07] Weldemariam, Komminist, Adolfo Villafiorita, and Andrea Mattioli. 2007. Assessing procedural risks and threats in e-voting: Challenges and an approach. In *VOTE-ID, lecture notes in computer science*, 38–49: Springer.

[XM04a] Xenakis, Alexandros, and Ann Macintosh. 2004. G2G collaboration to support the deployment of e-voting in the UK: A discussion paper. In *EGOV, lecture notes in computer science*, 240–245: Springer.

[XM04b] Xenakis, Alexandros, and Ann Macintosh. 2004. Levels of difficulty in introducing e-voting. In *EGOV*, 116–121: Springer.

[XM04c] Xenakis, Alexandros, and Ann Macintosh. 2004. Procedural security analysis of electronic voting. In *ICEC '04: Proceedings of the 6th international conference on electronic commerce*, 541–546. New York, NY, USA: ACM Press.

[XM05a] Xenakis, Alexandros, and Ann Macintosh. 2005. Procedural security and social acceptance in e-voting. In *HICSS '05: Proceedings of the 38th annual Hawaii international conference on system sciences (HICSS'05) - Track 5,* 118.1. Washington, DC, USA: IEEE Computer Society.

[XM05b] Xenakis, Alexandros, and Ann Macintosh. 2005. Using business process re-engineering (BPR) for the effective administration of electronic voting. *The electronic journal of e-government* 3(2) 91-98. [available online at www.ejeg.com]

[XM07] Xenakis, Alexandros, and Ann Macintosh. 2007. A methodology for the redesign of the electoral process to an e-electoral process. *International journal electronic governance* 1:4 –16.

[Yee07] Yee, Ka-Ping. 2007. Extending prerendered-interface voting software to support accessibility and other ballot features. In *EVT'07: Proceedings of the USENIX workshop on accurate electronic voting technology*, 5. Berkeley, CA, USA: USENIX Association.