Towards an Open Standard Vote Verification Framework in Electronic Voting Systems

Ali Fawzi Najm Al-Shammari\(^1,2\)
\(^1\)Faculty of Comp. Science
Free University of Boziano
Boziano, Italy
Email: alifawzinajm.alshammari@stud-inf.unibz.it
\(^2\)Center for Information Technology,
Fondazione Bruno Kessler (FBK-irst)
Trento, Italy
{alshammar,adoflo,sisai}@fbk.eu

Abstract—Vote verification allows voters or other election participating entities to verify that votes are correctly captured, stored and counted. To facilitate the vote verification process, a number of verification techniques (either physical or digital) have been developed to provide an evidence to voters and other participating entities for the assurance of the integrity of election result. However, we observed that these techniques have a number of limitations among which, the fact that the existing techniques do not fully comply with verification requirements (e.g., public verifiability). They implement limited prevention mechanisms from known attacks and they are not based on interoperable components and processes (typically, vendor lock-in). In order to address these issues, we propose a new method for vote verification based on open standards which allows interested parties or organizations to participate in vote verification process so as to enhance transparency and capture vote threats and challenges during and after elections.

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

I. INTRODUCTION

A vote may face different levels of threats during its lifecycle. Such threats can span from technical failures (e.g., the presence of software bug or machine error, malicious software in e-voting machine or in tallying server and hacking attempt if the system is networked) and procedural and/or environmental failures (e.g., the possibility of corrupted electoral and dishonest behavior of voters). The overall effect of these could compromise the integrity of the final election outcome.

Vote verification is one of the ways to ensure that cast ballots match voters selections [1], [2], [3]. For a voter, the ultimate goal of the vote verification is a system where any voter without special training can easily convince himself/herself that the counted ballot indeed reflects the actual selection. It can be used at different phases of an election with the aim of satisfying different levels and scopes of verifiability requirements. The level could be either individual or universal verifiability, spanning across election phases. Similarly, vote verification scope can vary from being a polling place vote verification to being an end-to-end vote verification. The polling place vote verification supports vote verification during election to give an evidence for the voter. End-to-end vote verification, instead supports vote verification starting from vote casting (during election) to verifying the integrity and correctness of the final election outcome.

There have been several studies on e-voting system design and implementation (e.g., see [4], [5]). The major research and development in this area focus on the design and implementation of vote verification methods that can be used during and/or after closing elections (e.g., during vote transmission, vote tallying). Most of these efforts aim at providing mainly proprietary solutions. Some of these methods have been applied in real-life e-voting systems. For example, most of the Direct Recording Electronic (DRE) voting machines that are currently deployed in polling places are equipped with verification devices in order to provide an evidence about votes cast, such as by providing paper audit trails [6] and audio messages [7].

We examine the weaknesses of these methods from three perspectives. 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 how they are vulnerable to threats and verification challenges. Finally, we asked ourselves the following question: can the existing methods be easily configured for different elections and devices? Since there is no common standard for verification in any existing methods, configuration of vote verification process is particularly difficult.

We present a new vote verification framework based on an open standard protocol to address some of the challenges we identified. Our specific goal is to develop an independent verification technique for a class of e-voting system. We do so first, by defining 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. We envisage a solution that is able to make e-voting system
more transparent to voters, third party organizations and to electoral offices. The verification method we devise also allows to capture vote threats and challenges during and after elections. To validate our research hypothesis, we use the ES&S e-voting system as a case study with respect to the EVEREST report [8].

The paper is structured as follows. The next section discusses the context of the work and presents the analysis of vote verification techniques. In Section III, we give a short overview of DRE-based e-voting to characterize the context of our work. Section IV discusses the proposed open standard framework. Section V discusses an example taken from the ES&S e-voting system for validating our hypothesis. Finally, in Section VI, we present our conclusion and future work.

II. STUDY AND ANALYSIS OF VERIFICATION TECHNIQUES

A. Synthesis of Vote Verification Techniques

Vote verification techniques vary according to the context of the e-voting technology used, the levels of verifiability (individual and universal), the mechanism they implement (e.g., by voice message, digital records or physical receipt), and the scope (e.g., polling place vote verification, end-to-end vote verification). In the table, the individual type allows voters to verify votes either during vote casting within the scope of polling places only or within the scope starting from vote cast to the announcing of election result (end-to-end). The universal type allows the public to verify cast votes either in polling places or end-to-end. The following discussion is based on the scope only, unless mentioned differently.

Polling place technologies. The verification technologies used at the polling place allow a voter to verify that his/her vote has been captured and/or cast as intended at runtime. Among the existing techniques that are used at polling place, we mention Pnyx.DRE [2], [1], VVPAT [6], [9] and VV AATT [2].

The Pnyx.DRE allows vote verification through a dedicated hardware device named Vote Verification Module (VVM) [2], [1]. The VVM is bidirectionally connected to Pnyx.DRE in order to allow each voter to verify and confirm his/her vote before the final confirmation. Namely, before a voter casts, the DRE machine sends a corresponding information to VVM, where the voter has an option to confirm or reject his/her selection by reading on the VVM screen and pressing an appropriate button on the device panel. Once the voter has confirmed, the VVM encrypts and stores the cast vote in its internal memory by calculating and storing also an integrity register record corresponding the vote. At the end of the election, the integrity register records of the votes stored in the DRE machine are calculated and compared with the records in VVM to ensure that the votes stored in the e-voting machine are indeed identical with those from the verification module. If any inconsistency is detected, then the electoral authority (poll officers) will decrypt votes stored in the VVM and compare with votes stored in DRE. This way the polling officers and candidates’ observers are able to verify the integrity of election results at precinct place.

Voter Verified Audio Audit Transcript Trail (VVAATT) and Voter Verifiable Paper Audit Trail (VVPAQT) employ the same concept for allowing vote verification but provide different types of evidences. Unlike the VVM in Pnyx.DRE, they offer the possibility for the voter to verify the start, each selection or cancel, and end information during the voting process. More specifically, using VV AATT a voter can verify his/her selection or cancel by hearing a corresponding voice message generated by the DRE machine through a headset. The vote is considered cast only if the voter has confirmed it by pressing the confirmation button on DRE panel. 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 [6], [9]. 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 VV AATT 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 VV AATT and VVPAT.

End-to-End Vote Verification. In line with End-to-End vote verification, a number of e-voting systems which ”claim-to-support” end-to-end vote verification have been proposed, aimed to allow a voter to verify that his/her vote has been tallied as intended. At attempt to implement end-to-end vote verification can be found in Scantegrity, Prêt á Voter, VoteHere, Votegrity, and Votescr ipt+.

Scantegrity and Prêt á Voter are described in detail in several papers [10], [11], [12], [13], [14], including all system description and requirements.

The Votescr ipt+ system allows an independent end-to-end vote verification by enabling a voter to have an encrypted copy of the vote stored in a smart-card. After the tallying phase, in Votescr ipt+ system, a voter can verify his/her vote by going to a dedicated place equipped with verification machines. The voter can insert the smart-card into the verification machine in order to communicate with a tally board for verification.

The other methods (i.e., except Votescr ipt), instead at-
tempt to ensure vote verification by providing (encrypted) receipt. For instance, the vote verification methods implemented in VoteHere [1], [15] and Votegrit [16] allow for an independent end-to-end vote verification with receipt. In case of VoteHere, a sentinel verification device is connected to a DRE machine for printing the encrypted receipt for the 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 on 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 on a public website, so that voters can check if their votes have been counted correctly.

In contrast, the receipt in the Votegrit system is printed by a small printer attached to Votegrit DRE. The printer prints a double side two-layer transparent receipt in a way that the two layers are combined to visualize the contents of the receipt, while each layer alone does not reveal any content. At this stage, the voter should select one side and confirm the selection. After confirming the selection, the printer will print an extra inch on the selected layer that contains the final signature and serial code. This print is a final receipt for the vote that the voter can use after elections to verify his/her vote by entering the serial code on public website.

B. (Some) Issues in e-voting Techniques

In our previous work [17], we presented some of the limitations of the above techniques. In what follows, we characterize these limitations by focusing on three main problems that we believe to be of general interest.

Issue 1: 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 as shown in [18]).

Issue 2: 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 obtained from a single vendor or it will 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.

Issue 3: 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 on 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. (See, for instance, [19] for an example of systematic auditing of the log of e-voting machines).

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.

III. CONTEXT: VOTE VERIFICATION IN DRE-BASED SYSTEM

In what follows, we give an overview of procedures and general architecture for DRE-based e-voting system to define the context of our proposed approach (see Section IV).

A typical layout of DRE-based e-voting information flows is shown in Figure 1. The figure depicts the main components and the distribution of voting and verification activities in two places: precincts and central election office. In precinct, the DRE machine is configured for vote casting and storing. The voting machine captures voter’s selection through two common types of I/O devices: touch-screen and navigation keypad. Modern DRE machines are also equipped with optional I/O devices (e.g., headsets) to voters with disabilities (e.g., disabled, visually impaired). Different types of verification mechanisms (either physical or digital) can be attached to the DRE machine in order to provide some sort of corrective feedback to voters using an output device (e.g., printer, headset or digital display).

Verification Modules (VM) can be connected to the DRE machine either by unidirectional or bidirectional connection depending on the design of verification technique. For instance, the communication from the DRE to VM (the solid arrow in Figure 1) is required to send mandatory data and information for verification purpose. In some systems (e.g.,
Pynx.DRE [1]), the VM can also send back feedback to DRE machine (e.g., confirmation signal) as shown in dotted arrow in the figure.

At the central election office, there are tallying and auditing modules which are responsible for collecting sensitive data coming from different precincts, tallying and verifying election result. The tally and audit modules could be an ordinary computer equipped with memory card reader and configured with special election management software used to perform the tallying or auditing process.

IV. OPEN STANDARD VOTE VERIFICATION FRAMEWORK

The principal idea behind our approach is to design a holistic vote verification framework for a specific class of e-voting system (DRE with VVPAT), with the goals of allowing interested entities to participate in monitoring and detecting threats that can compromise vote integrity. To do so, we combine the idea of common and interoperability standards for the design of an open standard verification protocol with three distinct features: independent verification, third party verification and standardization. More specifically, we define an open standard for the verification protocol that would allow different vendors to produce different components —i.e., voting machine and verification component. These components can be used by more than one entity (e.g., electoral staff and public observers) to enable independent verification. Moreover, the specifications of the verification standard components will be opened to any interested party allowing them to build independent verification component to be used during election, consequently, making the level of trust more open and distribute the verification process across different entities, thereby achieving public verifiability.

A. Open Standard and Framework Model

Our approach follows the layout depicted in Figure 1, where the DRE machine is connected to one or more verification modules (VMs) through a communication protocol. The communication protocol allows a real time message to be transferred from the DRE to VMs by ensuring also a clear communication path internally and externally, and define the accepted processes for approval and dissemination of outgoing (from DRE) and incoming information (to VMs). The protocol message contains voter actions (i.e., selection, cancellation or confirmation message), control or signal message and a header information. The control and header parts of the communication protocol will ensure the consistency of messaging (e.g., by using Error Correction Code (ECC) for error detection and correction) and quality assurance.

To make the protocol message standard and comprehensible we need a common standard data format for the representation and specification of the data format (e.g., election events, ballot structure, initial configuration) in DRE and VMs. For this purpose, we adopt and extend EML (Election Markup Language) schemas [20] —an XML based schemas defined to support interoperability among e-voting component. The voting machine and verification components will be initialized with data following the standard structure defined by the EML schemas. The initial data configuration should contain election event information (e.g., event name,
date, precinct locations) and blank ballot (e.g., ballot identifier, candidates list, referendum options). Figure 2 shows an example describing part of the EML election event schema. This schema defines the structure of XMLs file that carry election event information — e.g., election event must have EventIdentifier and one or more Election elements, and may have Description, Language and EventDate elements.

The VM interprets the received message according to the type of the encoded message (a voter action or control message) and the policy specification in communication protocol, and matches it with the information stored internally, also according to each policy in each layer. Figure 3 shows the framework model for the proposed open standard whose construction is based on the communication protocol specifications and open EML schemas.

For better abstraction, we represent the model in a way similar to communication networks stack model. It consists of four layers. Each layer is connected to the next layer down in hierarchy directly with appropriate interface while connected to peer layer in remote device logically through peer communication protocol. A standard supporting interoperability between different independent components requires defining the format of data to be exchanged, the communication protocol to manage data transfer and the physical interface that allows the components to be connected together.

In the figure, the application layer refers either to the e-voting or vote verification application. In both layers our initial choice of the implementation standard for the verification information to be exchanged is based on the extended EML data format. Notice that no prior work used EML for the purpose of verification, thus we may require to define new tags and data structures. The transport layer includes the main communication protocol responsible for generating messages according to the defined message format. This protocol also controls the message flow and manages error detection and correction. While the link layer is responsible for delivering the data from transport layer data to physical communication media, the physical layer allows an external VM to be connected to the DRE securely.

Additionally, the above components of the framework should only allow a unidirectional communication from DRE to VM. This is intended to prevent the verification component from loading malicious code to the DRE through
verification interface. Allowing the DRE to send data to VM may possibly attract potential attacks which aim to eavesdrop the communication channel for revealing some critical information (e.g., voters order). Thus, our design of the communication channel should be immune enough against eavesdropping during communication. Additionally, the data travelled from DRE to VM should be digitally signed before sending to VM making difficult for the attacker. In order to reduce the possibility of collaboration between attacker and single VM to perform attack to election process, our design should allow multiple VM to be attached to single DRE machine.

A simplified operational framework of the proposed approach is shown in Figure 4. The flow of data between the verification and e-voting components is based on a standard data format, communication protocol and interface as defined in the layer model.

![Figure 4. The operational framework of the proposed approach.](image)

On election day and before starting election, an authorized third party can connect the VM with the DRE machine, where the DRE is also connected to a special printer whose purpose is that of printing audit trails. After each time the poll worker activates the voting machine for an eligible voter, a voter starts voting process by directly interacting with the DRE screen and a corresponding information is shown on the VM screen. This process continues until the voter arrives to the final voting procedure, where s/he can review the selections and confirm by stating that both the DRE and VM screens are identical. After this, the voting machine will send the final selections to the printer. If the printed paper audit trail is the same as the one shown on DRE screen, the voter can cast the final vote. The vote cast will be stored both in DRE and in VM. A confirmation code will be printed at the end of the paper audit trail along with some additional information, e.g., summary information and a barcode encoding the vote. The paper trail is then cut mechanically and stored in a physical box for auditing purpose. After closing the election, two precinct tallying reports —from the DRE and from the VM— will be generated. A polling place electoral official will announce the two results and third party can verify their integrity. If any mismatch is detected, then the paper audit trail along with the VM will be used for auditing. At the end of the day, third parties (or the public) will have a copy of the verified report of polling place, which in turn can enable them to verify the final tally result after being published by election officials.

V. DISCUSSION WITH EXAMPLE

To validate our approach and to derive the necessary requirements for the implementation, we studied the ES&S iVotronic DRE e-voting system [21], [22] with respect to the EVEREST report [8]. The reason for this choice is the system is widely deployed in some states of U.S.A; its security assessment (see in [8], [18], [4], [5]) has shown that the technical, procedural or organizational implementation of the system lacks the necessary controls for vote verifications; the discovered weaknesses can be a nice use case to demonstrate how the open standards we propose can help prevent such kinds of known attacks. In fact, the open standards and protocols have to implement two potentially conflicting goals. On the one hand, they need to preserve the fundamental requirements; on the other, the information exchange should be sufficient to detect attacks. We skip the explanation of the components of ES&S due to space limitation. See in [21], [22] for the explanation with respect to this work and in [8] for a more detailed and complete view.

We use the changing the vote for a careful voter attack (taken from the EVEREST report) to illustrate the effectiveness of our proposed approach. This attack assumes the voter carefully cast, checks the screen and printout, and confirms. However, s/he is not familiar with all the details of how their votes are printed on the paper audit trail (called the Real-Time Audit Log, RTAL). The attacker does not intercept the normal voting process until after the cast ballot and confirmation screens have been shown to the voter. At this point, the attacker changes the voter electronic ballot, and the RTAL prints the modified selection. The RTAL immediately prints the summary information along with the barcode.

With the introduction of the VM, whenever the voter makes a selection, both DRE and VM show the selection accordingly and independently. After vote casting, the DRE, VM and RTAL store the cast vote. In this scenario, for a successful attack in the presence of the VM, the attacker needs to compromise all the components. That is, besides the normal attach actions without the VM, the attacker needs to compromise also the VM in a way letting the VM
to display the correct vote but store the modified one. In this case, two possible scenarios could happen: the VM shows an incorrect vote and hence the voter can notice the discrepancy. Alternatively, the VM shows and stores the correct vote so the discrepancy will be shown after closing the election when the final reports are compared. As noted before, our open standard protocol implementation of the VM also allows multiple VMs to be attached to a single ES&S machine. The fact that the attacker needs to compromise all the components (i.e., the DRE and more than one VMs) reduces the possibility for the attack to be succeed.

Our preliminary investigation of the proposed approach on ES&S voting system showed that most of the possible attacks can be reduced or removed. More specifically, since the VM reproduces the verification process by displaying the vote casting activities onto the VM screen, it reduces the possibility of some threat actions to remain undetected (e.g., screen calibration attack). In fact, this depends on the behavior of the voter. Namely, if the voter is inattentive or fleeing, such redundant verification will have less contribution with respect to the on-the-fly detection and feedback mechanism which we envisage to achieve. However, we believe that the feedback mechanism of the VM and a properly designed verification interface can potentially reduce the possibility of being exploited these kinds of voters behavior.

The other advantage the VM can provide for election authorities is a backup support since the VM stores each verified and confirmed copy of cast vote securely. This is important particularly if the ES&S machine encounters fatal errors internally (e.g., in the cast votes records) due to threat actions (e.g., clearing terminal attack) or malfunctioning of the machine itself.

A. Summary of Expected Contribution

It should now be clear how the proposed approach can help addressing the three issues we have introduced in Section II-B. Let us briefly recap why in the following paragraphs.

Issue 1: Exposure to Attacks. By allowing different (verification) components to be connected to the voting devices, it becomes possible to enhance current solutions and, second, to create a variety of solutions which makes systematic and large-scale attacks more difficult to carry out.

Issue 2: Vendor lock-in. Standards and interoperability are two key enablers to reduce reliance on a single vendor. Integration becomes possible both “horizontally” (e.g. by connecting a verification device of vendor X to the voting device of vendor Y) and “vertically” (e.g., by choosing the solution of vendor “X” for voting and that of vendor “Y” for tabulating data).

Issue 3: Open Verifiability. This is probably the simplest of the points to demonstrate, as open standards are the basis to enable forms of systematic auditing of the data.

VI. Conclusions

In this paper, we presented the idea of an independent verification technique for DRE-based voting systems. We are doing this by defining a robust verification process that can easily be configured for different elections and devices, a standard between the two processes (between the e-voting and verification processes) and a specific communication protocol for some steps of the voting.

Addressing the above challenges will allow to provide election participants (election officials, voter and public observers) with a proof about the integrity of election result, and also allowing to capture vote threats and challenges during and after elections.

Furthermore, the idea of making verification method open source and available for public use will enable different vendors to produce different verification components (i.e., prevent putting trust in single vendor provider of election components) and will increase election transparency among public (i.e., reduce required trust on election internal actors).

An important point to mention is the side effects of the proposed verification method. Namely, adding a new verification component can increase the complexity of the election process and open a new door for attackers to launch more sophisticated and undetected attacks.

Given the above and other possible risks that would face the design and implementation of the proposed approach, a technique like business process and procedural or organization security evaluation should be applied to better understand the business context, business model, and further requirements for the proposed solution prior to introduce the new method. With respect to this, works like [23], [24] can be used to evaluate the underlying procedures with respect to possible risks through procedural security analysis by considering the organizational security aspects, also by highlighting (some of) the security implications of the administrative workflow in the proposed methods.

Finally, after completing the design and implementation of the open standard protocol, we work on building a prototype for standard verification management system and verification model prior to conduct a control experiment for gathering additional requirements such as usability requirements.

REFERENCES


of Vote Verification Technologies: Findings and Experiences from the Maryland Study,” in *EVT*, 2006.


