How a Frog could end the Security Wars and revive the eVote Industry

Ingo Boltz

Information Technology Consultant

Member of the Scientific Committee at the
Observatorio del Voto Electrónico, Leon University, Spain

Wednesday, September 29, 2010

Contents

Executive Summary

The paper attempts to show how the consequent implementation of the AMVA/FROG electronic voting architecture could end the current conflict over electronic voting security and create an innovative eVote industry that focuses on producing feature rich, lightly secured ballot generator modules. An open source development effort maintained by an independent foundation, run by the academic/security community would create a standardized, radically simplified and rigorously secured vote casting module. In conjunction both would deliver feature rich, secure, cost-efficient electronic voting.

The implosion of a revolution

After its early years of successful growth, electronic voting has started to lose momentum. When ten years ago election administrators all over the world welcomed the new technology with an almost religious fervor, the year 2010 shows a radically different scenario. Cancelled eVote projects abound in Europe[1]. In Latin America, Paraguay, who had been using Brazilian eVote machines, has returned to paper voting. In the USA, one of the pioneers of electronic voting in the world, thousands of electronic voting machines are now sitting in warehouses, decertified by authorities and thus banned from use. Asia is not immune -- India is experiencing increasing opposition to its EVM machines; the Philippines have seen strong pressure against the introduction of DREs

Vendors have found themselves at the center of a storm of protests over lacking security and transparency of their wares, unleashed by computer scientists at academia and magnified by citizen activists all over the world. That wave of protest is threatening to devour their industry.

Protests can be divided into two flavors. There are the absolutists that maintain that that introducing any kind of computer into the process of voting is always non-transparent, insecure and undemocratic, and thus to be avoided under any circumstances. More moderate critics don’t discard the possibility that secure and transparent forms of electronic voting exist, but they equally question the current state of security and software engineering as manifested in eVote machines and software on the market.

Whether this sorry state of things had to come about inevitably because of the challenges of translating the particularities of a secret paper ballot into computer technology[2] or whether in the rush to conquer the new market vendors simply sacrificed design quality for speed of product development, the fact remains that many systems on the market today suffer from significant, demonstrated[3] security weaknesses.

[1] Ireland, after buying thousands of Dutch DRE electronic voting machines in 2004, decided last minute to leave them in their warehouses, and continue to use paper. The Netherlands, having used electronic voting since the early nineties, completely abolished the technology after turbulent events in 2006 and returned to paper voting. Germany’s constitutional court, after a lawsuit brought by activists, all but banned the use of electronic voting machines in that country.

[2] Voting by secret ballot poses a particular paradigm. In areas like banking personalized, independent transaction records function as safety nets. If a process or software fails, records can be consulted to reconstruct a transaction. Voters in contrast must not be able to obtain such a record of their transaction, lest they could sell their vote or be intimidated to vote a certain way; nor could the election administrator be allowed to keep records on who had voted how. Thus, while in other areas records are available to detect and correct errors, voting systems must do without them. Electronic vote processing is like aerial acrobatics without the safety net. While records make it possible to detect and correct an erroneous electronic bank transfer, their absence makes it impossible to detect or correct a wrongly stored electronic vote. Errors can only be rectified by manually recounting all ballots, or repeating the entire election process. The same paradigm applies to votes stored wrongly intentionally – electronic fraud. Such fraud, once perpetrated, is likewise hard to detect, and can only be rectified by recount or revote.

[3] See for example California Top-to-bottom-reviews

The security wars in the USA

Especially in the US, as critics of the robustness and security of voting machines became more vocal, antagonism almost immediately became the modus operandi. Election administrators dismissed their criticism as theoretical, concerns of egg head outsiders who “didn’t really know anything about what matters when organizing elections.” Vendors saw a threat to their new business and reacted by trying to stifle criticism in the courts. Critics took their concerns public, starting the “US election security wars” whose bitter enmity persists until today.

As the opponents of electronic voting progressed in advancing their arguments, the blossoming US market for electronic voting equipment turned into a very difficult one. Vendors have been facing increasingly demanding audit and certification requirements that are different across different countries, or even within them (different rules for different states in the US, for example). The cost of getting certified must often be borne by the vendor; getting certified in enough markets to gain economies of scale is becoming increasingly expensive. Requirements to have machines re-certified after even minimal software changes anywhere in the system are slowing down product cycles and make it harder to respond to customer needs. And as eVoting has become almost a “dirty word” being active in the market is even becoming a danger to companies’ reputation.

The consequences of this changed environment are already apparent. Companies that had other business lines to fall back on have left the eVote market by selling[4] or writing off and closing[5] their eVote product lines in order to avoid “image contamination” of their core business. The rest of the industry has been experiencing rapid consolidation, with only a few big players remaining of what was already quite a small field[6].

Fundamentally, the industry remains dysfunctional. Vendors, from a defensive position but still with an election systems market looking for solutions, keep launching new products, which are subsequently taken apart by the security community and, more often than not, disqualified with relish.

Vendors accuse the security community to be criticizing from a safe distance yet not walking in their shoes; after all, they are not producing any equipment that could serve as an alternative. They say it’s easy to reject what they are doing without having anything else to offer; “simply use paper” as advanced by the absolutists is not a solution for many election administrations, especially those facing complicated election systems that make fully manual counts within reasonable time near impossible.

The computer scientists and activists, in turn, accuse the eVote industry of being incapable to learn from its mistakes, and continuing to turn out badly designed systems. They maintain that it is not their job to develop products; that they are in academia and teaching, not in the business of selling software and hardware, and that it is the job of industry to respond to their criticism.

That said, there have been vendor-independent attempts to make voting machines more trustworthy by developing all their source code as open source, community maintained and open for anybody’s

[4] For example ATM maker Diebold’s sale of its eVote line
[5] SDU in the Netherlands
[6] Recent purchases of Premier and ES&S by Dominion

inspection. Examples include the Open Source Digital Voting Foundation (OSDV) and the Open Voting Consortium (OVE), both based in California.

Open Source Software (OSS) projects are competing often successfully with commercial software (e.g. the Mozilla Foundation's Firefox browser, Linux …) so perhaps an eVoting OSS project could succeed. Maintaining the large customer service and support structure that would be required to serve a large number of electoral clients is no easy feat, but there are successful companies (e.g. Red Hat with its Linux distribution) that thrive economically providing professional support for open source software. Such models may be applicable to the eVote market.

Unfortunately, designing and mass-producing electronic voting *hardware* is an altogether different business. The capital and scale needed to operate such production are significant. Recent efforts of the open source voting movement have thus focused on using COTS hardware such as iPads and office printers, instead of custom-designed electronic voting hardware. The future will tell whether such OS/COTS compound systems will be able to replace custom hardware.

With open source efforts are still in pilot stage, election administrations worldwide would benefit from a transparent, innovative and competitive yet profitable eVote system vendor market, with a sustainable business model supporting many strong players.

However, current development seems to point in rather the opposite direction: rapid industry consolidation, financial squeeze, and an ever more heated security war. The people who know best how to design secure systems are not collaborating with the people who are doing the building and selling.

> **The Dutch Case:** After voting for many years electronically on classical black-box style DRE machines, all municipalities in the Netherlands were in 2007 obliged by the federal election administration to return to manually counting paper ballots. Since then, poll workers have been up in arms in many parts of the country. After previously simply "e-counting" at 9pm on election day at the press of a button, they now have to spend hours sorting, stacking, counting, and often re-counting various times, until the early morning hours, to reconcile human errors.
>
> The municipalities have organized work shift systems to take pressure off the poll workers, yet hand-over between shifts remains an issue, as is the legal responsibility for hiccups at the polling station between different teams. Blind citizens' communities are protesting as their capacity to vote unaided has been removed again. Paper is being billed as "a return to the middle ages" and the municipalities are pressuring the federal administration openly for a return of eVoting – secure or not.
>
> The security community might be celebrating its victory and shout from the high horse: "We won the war, it's the way it is, get used to it" but is that wise? Interestingly, in the middle of the Dutch struggle, the manufacturer of the old scrapped voting machines has started to market a new touch screen eVote machine, more specifically en Electronic Ballot Printer called TK10. Unfortunately it has again been designed without consultation of the security community and destroys the security of its modular architecture through a design fault[7]. And yet, municipalities are actively lobbying the federal election administration to permit this new "secure" solution so as to be able to "go back to eVoting" as soon as possible and forget this regrettable medieval interlude. Without collaboration and new approaches, voters in the Netherlands may yet again get an electronic voting system that is insecure by design, and the whole circle of confrontation may start over again. If there is no solution to the security wars, such outcomes may result in other countries, as well.

If a universal and permanent "return to paper" is not going to happen, how is it possible to create a secure eVoting ecosystem, in which many commercial players compete in an economically sustainable

[7] See below for discussion

market, differentiating themselves with product design innovation? Can we create an environment where the security wars are a thing of the past?

Relieving commercial vendors of the need to think "secure"

Judging by the eVote industry's track record on security-conscious software and robust hardware design[8] one could suspect that, squeezed between security demands on the one side, and pressure to create innovative industrial product design, the industry has often faltered on both fronts.

So why not liberate it from one of the two obligations altogether? Why not create an ecosystem where vendors ***don't need*** to produce machines that are secure, because it will not matter? An ecosystem where they can focus on providing rich end user features, housed in physically robust products that will withstand stress and are dead-easy to use? ***A system in which they can stop thinking about certification schemes and activists on their case, and instead focus on building the iPods of eVoting?***

A FROG to the rescue

A solution to our dilemma has been in existence since the beginning of the decade – and it's a FROG.

Current standard DRE design is monolithic – one box does it all. But each feature that makes a DRE more attractive to election administrators is also making its software more complex. And for software, more complexity means less security, as larger amounts of more complex source code are much harder to audit for both errors and malignous functionality. Consequently the relationship between features and security is inversely proportional. In a monolithic design it is impossible to escape that paradox: you either have a simple machine that can be secured reasonably well, or you have a feature rich machine that can't.

In 2001 Bruck, Jefferson and Rivest[9] proposed a new ***modular*** voting architecture (AMVA), in which they divided the voting process in different tasks. The "vote generator" task would be performed in a vote generator module, separately from "vote casting", which would be performed in a vote casting module. Perhaps a tad eccentrically, they used the term FROG as a stand-in for various possible vote storage media used to transport the vote from one module to the other.

The main objective of AMVA architecture is to – by design – minimize the amount of source code that needs to be trusted in a voting machine[10]. By dividing a voting system as proposed, its source code is also divided. The majority of it (which is providing all those shiny features) is located in the vote generator module. This module helps the voter to record his choices on a human readable ballot.

---

[8] For example http://www.wired.com/threatlevel/2008/07/ny-50-percent-o/

[9] Bruck, S., Jefferson, D. and Rivest, R., "A Modular Voting Architecture ('Frog Voting')", in TowardsTrustworthy Elections: New Directions in Electronic Voting, Chaum, Jakobsson, Rivest, Ryan, Beneloh, Kutylowski, Adida (Eds.), Springer Lecture Notes in Computer Science (6000), Feb. 2010

[10] It is important to note that the emphasis is not on reducing the amount of source code in general. Features such as disabled access and sophisticated user interfaces, ballot rotation or write-in candidate voting do require programming; minimizing the amount of source code of a system in general will always potentially limit features. Yet such features are what make electronic voting so attractive in the first place.

The voter can ***easily verify*** whether his will has been adequately recorded on that ballot; if it hasn't, the process stops right there, the voter complains to the poll workers, and the error is verified and investigated if reproducible.

Because of that "human control" the ***security requirements*** for that vote generator module are ***minimal***. Errors or manipulation would be instantly detected by the voter; hence the module doesn't really need security auditing[11].

With the lions part of the source code located "before" human verification and thus largely security irrelevant, only the remaining ***much smaller part*** that takes care of the remainder of the voting process really matters. The vote casting module is the spot at which a human readable and thus transparent and trustworthy vote record is converted into a non-human (but machine) readable vote record. It is also here where the vote must pass into anonymity; it must be impossible to link the vote to the voter once it has been cast. This also means that the voter cannot retrieve his or her vote anymore to verify it has been cast correctly. The voter must be able to ***trust*** that the casting module has correctly recorded and made anonymous the human-readable vote record he/she entrusted to it.

It is at this crucial module that maximum vigilance must be directed to avoid both error and manipulation. By focusing on auditing only the small vote casting module, while safely excluding the majority of the software in the vote generator module from scrutiny, effective auditing even under time and resource pressure finally becomes viable.

Software certification, and rigid sets of specifications and requirements, could be ***limited to the casting module***. The casting module would be uncompromisingly simple, performing its basic task with minimum resources.

The FROG – the human-readable medium of vote data transport between the modules –must conform to two requirements: its format must be ***easily readable for a voter***, so as to quickly verify that the generator module has stored the vote properly; it also must be ***easily and reliably machine-readable*** by the casting module. Importantly, it does ***<u>not</u>*** have to look like a standard paper ballot; presenting election choices adequately to the voter is the business of the generator module's user interface.

The format of the FROG would be part of the casting module's design process and its specification. This means placing the burden of adapting to that format on the vote generator module, which is not fettered by a simplicity mandate and may employ different approaches to produce the needed format.

> ***Note***: AMVA/FROG architecture is focused on precinct electronic voting. It does not provide answers to security challenges in back-end vote aggregation/tallying, nor is it applicable to remote/internet voting.

Optical Scanners, Ballot Marking Devices and Electronic Ballot Printers

Optical scanners (precinct and central high speed), as well as Ballot marking devices (BMDs) (e.g. ES&S/Dominion AutoMark and Sequoia ImageCast series) and Electronic Ballot Printers (EBPs) (e.g.

[11] It would, of course, be advisable to verify that it works robustly functionally, so as to avoid breakdowns on election day and long voter queues while technicians try to fix it.

LibertyMark/ LibertyProof, Avante VoteTrakker and Dutch NEDAP's TK10), are already "froggy" in architecture. So is AMVA/FROG old news?

Unfortunately, neither scanners nor BMD/EBPs are designed to translate their modular architecture into increased security.

Scanners are modular, but they are essentially a "fat" vote casting module. Without a vote generator module, they provide none of the accessibility benefits, user guidance, multi-lingual capacities, etc. that are core benefits of electronic voting. Yet at the same time, standard scanners are not simple enough in their design to be easily securable; security reviews of scanners have found numerous vulnerabilities[12]. So in a way, optical scanners are "the worst of both worlds" – hard to secure ***and*** poor on features.

BMDs are sold primarily as "accessibility devices", to be used only for disabled voters in polling stations where all able voters use standard op-scanners. Hence the vote casting module here is simply a standard optical scanner, with complexity challenging security as just discussed. Additionally, in a BMD setup, features that the scanner has do pointlessly ***duplicate*** functionality of the BMD. None of the BMDs works in combination with a ***specially designed, radically simplified*** vote casting module as proposed above.

Furthermore, since the AutoMark is intended only for disabled voters, it has not been optimized for speed and throughput as only a few disabled voters will use it during election day. If it were to be used for regular voters, speed increases would be needed.

The EBPs all ***do*** use specially designed vote casting modules. However, all current designs rely on a ***barcode*** on the printed ballot (FROG) they produce to read the information into the casting module. However, a barcode is not human-readable. Therefore, the voter cannot – at least without additional technical aids -- verify the information that is ***actually read by the casting module***; the voter must trust the vote generator module that the barcode contains the same information as the clear text on the printed ballot. Thus the generator module must be also subjected to stringent security measures, destroying the main advantage of FROG architecture.

The true engineering challenge thus lies in designing a casting module simple enough to be transparent and securable, with a FROG that is easily human readable yet suited to be read automatically with a very low error rate by the casting module.

Work on radically simple implementations of optical scanning (for a marked-ballot-type FROG) is for example being done by the Open Source Digital Voting Foundation, yet a coordinated effort of the academic security community, organized under a structure similar to, say, the Mozilla Foundation, could massively speed up this process.

[12] E.g. California top to bottom review, ES&S M100, https://www.sos.ca.gov/voting-systems/vendors/ess/unity-3011-red-team.pdf; many other examples there

Divided responsibility, shared results

In a “froggy” eVote world, vendors are left to focus their innovative efforts on the vote generator module. They can develop generator modules suited to many different markets, unfettered by certification processes and government usage authorization – the marketplace can be the judge of quality, just as it is with other industrial products. Furthermore, it will be completely acceptable for vendors to maintain their intellectual property rights, and trade secrets, on a vote generator module. There is no conflict of interest here between commercial sustainability and security.

In turn, the security community gets to be in charge of the casting module. A standard casting module, with corresponding FROG, would be developed by a joint group of academics and election administrators. It would be designed to be radically simple, thus more easily secured. The casting module’s software would be developed as open source and available for anyone to verify. Production of the hardware could be outsourced though a transparent public bidding process to hardware makers in the private sector, following tight specifications and using rigid QA and acceptance procedures. Because the same casting module would be used in conjunction with different generator modules, large scale of production would drive costs of the module down, reducing cost of ownership by election administrators.

Such a divided ecosystem would allow specialized eVote vendors to apply their experience in elections to make money, while satisfying the demands of the security community at a reasonable cost.

The lower security requirements for the generator module would eventually allow using standard computers to run vote generator software, such as for example a school’s computer lab PCs. Especially for developing countries with their limited resources, using existing hardware without compromising security would be a beneficial concept.

A “froggy” eVoting world could be a more transparent, trustworthy and secure one. It could end conflict between vendors, academics and activists, and provide voters with technology both convenient and trusted.