CORE

Computing Research & Education

Submission to the Victorian Parliamentary Inquiry into the Conduct of the 2010 Victorian State Election

11th May 2011

CORE is an association of university departments of computer science in Australia and New Zealand. The author, Dr Vanessa Teague, is an honorary fellow in the department of computer science and software engineering at the University of Melbourne, with a research emphasis on secure electronic voting. This report has been endorsed by the president of CORE.

I was appointed by the VEC as a "Technical Observer" for the 2010 election. I was given a protocol-level description of the system, and allowed to observe many parts of the electronic voting process, including an audit of the kiosks and the decryption of the votes on election night. Apart from this cover page, this submission is identical to the report I submitted to the Victorian Electoral Commission, which they published on their website: http://www.vec.vic.gov.au/files/EAV-CORE-Report.pdf The auditor's report and the vendor's response to my report are also available, at http://www.vec.vic.gov.au/files/EAV-BMM-Report.pdf and http://www.vec.vic.gov.au/files/EAV-Scytl-CORE-Report.pdf respectively.

The main recommendations of this submission are:

- Any electronic voting system should provide at least the same security, privacy and transparency as the system it replaces.
- If electronic voting is extended to voters who can read their own printout, then it should provide a printout for them to verify (a voter-verifiable paper trail), or some other form of direct verification.
- The auditor's report should be public, and the source code should be available to a much wider group of experts for analysis.
- There should be an Australia-wide set of standards for electronic voting systems, and it should include points (1) and (2).
- Requirements for openness should be part of the initial tender and contract. The more openness, the more assurance that the system behaves as expected.
- The system should be redesigned so that no single component can compromise the integrity or privacy of the votes.

I would be very happy to discuss electronic voting further with the committee or the VEC. It is easiest to reach me by email at vjteague@unimelb.edu.au. My phone number is 8344 1274.

1. Project Description

I was appointed by the VEC as a “Technical Observer” for the 2010 election. I was given a protocol-level description of the system, and allowed to observe many parts of the electronic voting process, including an audit of the kiosks and the decryption of the votes on election night. I was able engage in a series of conversations with the VEC and the vendors, and also to suggest questions to the auditors and the penetration testers. As far as I know this was the first instance of a formal “Technical Observer” position for an Australian election, and I hope the idea will be retained and extended. I would like to thank the VEC for their openness during this project, and Scytl for their effort in producing and discussing a protocol-level description of their system.

My aims in analyzing the system were:

- To find errors leading to security vulnerabilities, so they could be fixed before deployment. There was one notable instance of this, which was communicated to the vendors and patched.
- To clarify the assumptions under which the system is guaranteed to get the correct answer. This is described under “Integrity and Correctness” below.
- To examine what degree of privacy the system provides. This is also described below.

I found one possible vulnerability to external attack, an opportunity for someone with control of the Collector Server to spoof one of the voting telephones and hence inject votes into the tally. This was communicated to the VEC and the penetration testers, then to the vendors, who patched it. I regard this as a good outcome, indeed perhaps an advertisement for openness and its positive effect on security.

2. The system: summary

Overall the system uses reasonable cryptography for protecting the system against external attack. However, it does not use all feasible techniques for guaranteeing integrity or privacy against malicious insiders, relying instead on VEC's procedural and perimeter security at several points. Compared to the postal voting system, with envelopes opened in the presence of scrutineers, it is probably harder for an outsider to intercept an electronic ballot than a postal vote, but it is easier for a small group of insiders to manipulate the outcome or discover an individual's vote. Specific issues and suggested improvements are discussed in Section 5.

3. Main Recommendations

CORE has made several submissions on electronic voting to the Victorian parliament. Our main theme is:

Any electronic voting system should provide at least the same security, privacy and transparency as the system it replaces.

Reiterating our three main recommendations made after the 2006 election:

1. **If electronic voting is extended to voters who can read their own printout, then it should provide a printout for them to verify (a voter-verifiable paper trail), or some other form of direct verification.**
2. **The auditor's report should be public, and the source code should be available to a much wider group of experts for analysis.**
3. **There should be an Australia-wide set of standards for electronic voting systems, and it should include points (1) and (2).**

The 2010 electronic voting trial did extend to voters other than the visually impaired, but it did not include a human-readable printout. The "verification value" that it produced did not actually prove that the vote had been recorded or tallied correctly. Subsequent models should provide a human-readable printout, so that voters who are able to read can verify that their vote was recorded as they intended. This applies to all computerised voting systems, whether the interface is by telephone or computer. Printing out a real ballot that could then be placed in a ballot box would be a simple way to achieve this.

The next section addresses the degree of openness of the source code and the auditor's report.

4. The process

I received a protocol-level description of the system after signing a confidentiality agreement with the VEC. This followed several rounds of negotiation during which we discussed my signing a Non-Disclosure Agreement with Scytl (which I was reluctant to do), and Scytl releasing source code to me (which they were reluctant to do without an NDA).

It is challenging to balance the legitimate wish of software vendors to protect their intellectual property with the demands of transparency and scrutiny necessary for a voting system. Everyone agrees that a very high degree of transparency should be required for electronic voting systems. The controversy is over exactly how much, and in particular over whether the source code for the system should always be available for public scrutiny. Many computer scientists believe this should be a requirement, though some would say the requirement may be relaxed if the system provides genuine verifiability such as a human-readable printout.

Recommendation : Requirements for openness should be part of the initial tender and contract. The more openness, the more assurance that the system behaves as expected.

For example, from my protocol-level information I could identify some issues concerning vote privacy, but there may be other issues that were not detectable without source code.

The Council of Europe's recently published "Guidelines on the transparency of e-elections" include the statement that "Access to documentation including minutes, certification, testing and audit reports as well as detailed system's documentation explaining in details the operation of the system, is essential for domestic and international observers." These recommendations are (perhaps deliberately) vague on whether the "details [of] the operation of the system" include source code or not, but clearly they imply a very high degree of openness, probably exceeding what was provided by the VEC and Scytl in this trial.

Leaving aside the controversy over whether it is *essential* to have public source code, it is certainly *feasible* to do so. The ACT has run electronic voting on an open-source system for many years (the EVACS system from Software Improvements [1]), and the Norwegian government recently instituted a project for Internet voting, for which open-source software was a requirement of the tender (and Scytl was the winning bidder [2]). Also, if the VEC decided to produce its own system rather than purchasing one from a private vendor, it would then be free to publicise it openly.

Openness about the system is a vital part of transparency, but it provides only partial assurance because it is infeasible to guarantee that the system so carefully analysed is the one that actually runs on the computers on voting day. That is why a human-readable paper record remains crucial for providing evidence of the correct outcome.

5. The system: detailed assessment

In this section the technical details of the system's privacy and integrity are explained in detail.

Privacy

1. The system shuffles votes only within one kiosk, *i.e.* one voting machine. This could mean that an individual's vote is anonymised only within a very small set. This is particularly relevant since kiosks can accept votes from any division in the state, so each voter is anonymised only among the set of people who voted at the same kiosk for the same division, which may be a very small set indeed. An obvious improvement would be to shuffle over all kiosks. Also, the problem would be ameliorated if more votes were cast per kiosk. (This issue has been discussed before in the context of reporting of votes from e-centres, for example in the Parliament of Victoria's most recent report on voting contres [3]. The point is that this information is available on the back office machine, and could potentially be accessed, even if it is not officially published.)

   This problem would be greatly reduced if computerised votes were simply printed out and placed into ballot boxes along with other early votes.

2. The "back office" is actually both a decryption and a mix server in one. This is rather like opening the outer and inner envelopes of a postal vote in one step, although someone who wanted to violate privacy in this way would still have to discover which encrypted vote belonged to which voter, for which they would need access to the kiosk and observation of who voted there. Modern cryptographic techniques exist for separating

the shuffling and decryption of votes, so that linking a voter to their vote requires access to at least two separate computers. These techniques should be employed in future versions.

3. Without the source code, I was not able to establish whether someone with access to the kiosk after voting could discover which votes were cast in order. Because the votes were encrypted using symmetric-key cryptography it would have been difficult to erase these values, hence possible for someone to decrypt them after polling. This would not be a problem if they were stored in a randomised order, but could compromise privacy if the vote order was not properly randomized.
4. Without the source code, I was not able to investigate the generation of randomness for cryptographic operations. I was surprised by the auditor's assessment that although the key generation was weak, this was not a practical vulnerability. In general predictable keys can make it much easier for outside attackers to decrypt messages.

Verifiability

The kiosk prints out a "verification value" which is described in the marketing literature as giving voters the opportunity to check that their vote was recorded and counted correctly. In fact it provides no evidence of either of these. It would be more accurate to call it a *tracking number* that allows voters to *query* the system on whether it recorded their vote. If there has been an inadvertent error then this may detect it, but if the system is malfunctioning due to an attack, then the attacker could easily subvert the "verification system" as well. Hence a verification failure proves that there has been some error, but a successful verification does not prove that the vote was recorded or tallied correctly.

The kiosk runs two modules, a "Voting Terminal" which takes the user's input and records the vote, and a "Verification Module" which asks the voter to confirm that the vote was recorded correctly. If the voter accepts the vote, control returns to the Voting Terminal, which prints out the "verification value." Since both modules make complete logs of all votes, it seems that misrecording a vote would require both modules to be compromised, but in fact this is not the case. Substitution of the Voting Terminal module would suffice, as described in the following attack. The point here is not to argue that this attack is more feasible than simply compromising the whole machine, but simply to show that the presence of duplicate logs does not necessarily imply that both modules must be compromised in order to change the votes without detection.

The compromised Voting Terminal module goes through the normal vote reading process with the voter, but then instead of passing control to the Verification Module, it simulates the Verification Module to the voter, asking them to accept and confirm their vote. Then it prints out a receipt, reusing a value from a voter whose vote was recorded. Then it logs everything as if the voter quit, including sending a message to the Verification Module as if that voter quit, and *neither module records the vote. The voter walks away with a "receipt," believing that they have voted successfully, and they can "verify" that their receipt number was included in the final tally, but actually their vote was not recorded. The cheating Voting Terminal would have to re-use someone else's value, which in principle is detectable, but voters are unlikely to detect that the same value is used twice. This attack could be used differentially to attack a particular candidate or party, because the cheating Voting Terminal learns the vote before it commits to carrying out the attack.*

I understand that Scytl's product generally runs the Voting Terminal and the Verification Module on separate pieces of hardware. Clearly the above attack does not apply in that case. It is outside the scope of this report to consider the assumptions under which Scytl's verification is valid when run on its usual hardware arrangement.

Integrity and Correctness 1 - Trusted Components

In secure systems of any kind, it is better to avoid placing trust in any single part of the system. For example, our system for paper-based voting is carefully designed so that no single person is completely trusted. The presence of scrutineers observing the count, the public location of the ballot box, *etc.*, all make it very difficult for a single person to violate another's privacy or to compromise the integrity of the vote. An important exception is when one person relies on another to write their vote for them because they are unable to do so alone - then the writer must be trusted to keep it secret, and sometimes trusted to write it correctly because the voter is unable to verify this. This situation is undesirable, and indeed removing it is one of the prime motivators for electronic voting. Nevertheless, the electronic voting system itself has three trusted components:

1. The kiosk is trusted. It can misrecord a vote undetectably. This problem would be solved by making it produce a human-readable paper printout.

2. The back-office is trusted. It could break voter privacy as described above. The extent to which it can change the votes depends upon which additional checks are performed. In its basic version, it performs both the decryption and the test of whether the decryption was performed correctly. It would be better if it exported the proof so that an independent machine could check its correctness.

3. The Secure Voice Interface is trusted. It can undetectably modify a telephone vote that it relays. This problem would be solved by directing the telephone interface to a human-readable paper printout.

The system should be redesigned so that no single component can compromise the integrity or privacy of the votes.

Integrity and Correctness 2 – Testing whether the correct software is running on the machine

The VEC implemented a process of random audits of voting machines, in which the software auditors (BMM) tested whether the software CD in the machine matched the code they had audited. This is a good check against unintentional installation and configuration errors, and against some unsophisticated attacks at the polling place, but it does not really prove that the program being executed is the one that was audited. For example, if the kiosk had been maliciously reconfigured to ignore the LiveCD and boot something else from the hard drive, then this would not be detected by this process, which verified only the LiveCD and the correctness of certain other files. The problem of testing what program is actually running on a computer is known to be impossible even in theory. Furthermore, this sort of testing, although useful, does not really solve the problem of there being a small number of trusted insiders. It is not clear to me whether all of the stakeholders, including candidates and parties, would necessarily be willing to accept the assurance provided by this system when the success of the test was not really demonstrable to candidate-appointed scrutineers. The scrutineers (including myself) can watch the test being done, but they really still have to trust that those performing it are doing so correctly.

Again this level of assurance is considerably less than what would be provided by a voter-verifiable printout.

Conclusions

I thank the VEC and Scytl for their openness during this trial. I believe that public confidence in the system would be improved by further openness. The system should be designed to provide a human-readable printout of the vote, and to avoid trusted components.

References

[1] Elections ACT *http://www.elections.act.gov.au/elections/evacs.html*

[2] Norwegian Ministry of Local Government and Regional Development e-vote 2011 project. *http://www.regjeringen.no/en/dep/krd/prosjekter/e-vote-2011-project.html?id=597658*

[3] Parliament of Victoria. Report on the Inquiry into the Functions and Administration of Voting Centres, p.69. June 2010. *http://www.parliament.vic.gov.au/emc/inquiries/article/957*