Answers to questions on notice

Vanessa Teague

November 6, 2017
[redacted]

Minimal standards for local government elections, whether conducted by NSWEC or not

I am not an expert on paper-only electoral processes, so I will answer this question as it applies to elections in which the count is computerised.

For those elections that are conducted electronically, I share the concerns expressed by some members of the committee about elections conducted either by the council itself or by private corporations. The incentive for developing a reputation for probity and accuracy is not as strong as it is for the Electoral Commission. Nevertheless I think the legal requirements should be the same, as described in our first submission:

- Full preference data should be posted online, as NSWEC does, to allow independent checking of the count. It is not enough to publish only the distribution of preferences.
- There should be a random audit of the paper records in the presence of scrutineers, to check that the published preferences match the paper votes.
- The system's source code should be made openly available to public scrutiny.

Open source code – specific examples of errors that have been found and fixed

First let me be clear that I am not claiming that all errors will be identified if election source code is made open. There are many examples of errors in open source code that were not detected, such as the Heartbleed vulnerability and a bug in the Norwegian e-voting system that undermined vote privacy. The point is that opening the code makes errors more likely to be detected, not certain to be so. Nor am I suggesting that opening the code removes the need for NSWEC to pay for diligent expert analysis – they should do both.

In most countries the counting algorithm is so simple that coding errors are unknown. The only examples of counting code errors I know are Australian:

- A team at ANU conducted a formal analysis[1] of the ACT counting code, which had been published online. They found three bugs, at least one of which was acknowledged and fixed.
- Arugably our results on error finding in NSWEC code are a good justification of source code openness. We were able to identify them because NSWEC is very open about the vote data. When we first identified an error in the 2012 count we suspected there were more, because our probabilistic results diverged slightly from those of NSWEC. If we'd had the code then, we would probably have been able to find the next two errors before the 2016 local government elections. As it was, we had to wait for more data, which only came after the election had run again on incorrect software.
- The Victorian Electoral Commission also publishes its counting code online, along with the code it uses for the randomised ballot draw. I do not know of any errors in either.

1 http://users.cecs.anu.edu.au/~rpg/EVoting/evote_revacs.html

Openly available source code was a requirement for the Norwegian Internet voting system, has been adopted for the Estonian Internet voting system, and is a federally-mandated requirement for all e-voting systems in Switzerland.

Security concerns are often cited as a reason to keep the source code secret, but this is complete nonsense. Insecurities in the system remain exploitable regardless of whether the source code is available for public scrutiny – we have found a number of security holes in the iVote system despite its source code being unavailable. So making a system's details open won't make it perfectly secure, but keeping them secret certainly won't either. This is summarised in a blog post I coauthored:

https://lawfareblog.com/open-source-software-wont-ensure-election-security

There have been numerous examples of security analyses of the source code of e-voting systems that have led to positive change, though not as directly as a particular issue being found and patched.

- Wallach and Rubin wrote a security analysis[2] of some Diebold source code they found on an open server, where it had probably been left by accident. Although the company never acknowledged the problems (in fact, they tried to sue the researchers), this analysis was a significant step in the wide adoption of voter-verifiable paper records.
- Springall *et al*. wrote a security analysis[3] of the Estonian Internet voting system, based on the source code that had then recently been made available. Although the authorities did not accept all their recommendations, they did respond to information about server-side vulnerabilities by attempting to introduce end-to-end verifiability.

Expert analysis under NDA

I understand that in the afternoon of the hearings there was a discussion about having the source code analysed by paid experts working under a non disclosure agreement (NDA), rather than making the code openly available. This is the way NSWEC currently operates – the quality of the analysis is lacking in the extreme. CSC security had apparently certified the 2015 iVote run without noticing a serious security problem (though as far as I know the security certification itself is not publicly available). Someone presumably certified the 2017 WA run, without noticing that the DDoS protections hadn't been properly configured. The counting software we have found numerous errors in had been certified, and indeed was re-certified after the first error we found. I do not know whether it was re-re-certified after the third error we found, nor do I know whether the companies charge for the second or third certification of perfect correctness of the same piece of software.

I think it is a good thing for the NSWEC to engage experts to examine the code with care. However, it should be an addition to open public scrutiny, not an excuse for refusing to publish the code. For people who genuinely work in the field, signing an NDA creates significant complications around intellectual property and publication. If the code was open, the experts engaged by NSWEC could speak freely, and would know that their analysis could in turn be scrutinised and questioned. The result would be much better analysis, leading to improved systems.

One more suggestion for the Legislative Council count referendum

I suggest incorporating the referendum into the 2019 state election. In that count, open the source code to demonstrate fair randomness, as described in our submission.

2 https://www.cs.cornell.edu/people/egs/cornellonly/syslunch/fall03/voting.pdf
3 https://jhalderm.com/pub/papers/ivoting-ccs14.pdf