This letter was sent to the NSWEC in response to a request for comments in August 2013. We received a form letter in reply. We believe the NSWEC made some minor protocol modifications, noted below, but the main issues concerning verifiability and privacy still stand at the time of publishing (March 2015). The text has been annotated to take account of protocol updates since the original letter was sent.

Response to "iVote Strategy for the NSW State General Election 2015"

Dr. Vanessa Teague (University of Melbourne, vjteague@unimelb.edu.au),[1]

Prof. Peter Y A Ryan (University of Luxembourg),

Prof. Joseph R Kiniry (Technical University of Denmark, kiniry@acm.org)

Comment [VT1]: (March 2015) Joseph Kiniry has since moved to a private company.

We have read the "iVote strategy for the NSW State General Election 2015" and produced a brief analysis based on our expertise in electronic voting security, privacy and verifiability and the design, construction, validation, and verification of electronic voting systems.

We find no convincing argument that the proposed draft design either protects vote privacy or guarantees election integrity under reasonable assumptions. The draft design is very vague, which makes detailed analysis difficult, so we have concentrated on fundamental design decisions. A more careful analysis of the design's security properties would be possible only with more detailed documentation. Nevertheless, we hope this analysis will be helpful to the NSWEC in making an informed decision on whether to continue with iVote in its proposed new form.

The proposed architecture of the new iVote has two main parts. The part administered by the NSWEC registers voters and a separate part which receives votes. An "independently hosted and managed" system provides a "verification server" for checking the NSWEC process. Voters log in to the "verification server", which shows them in human-readable form how they voted, asks them to confirm this, and checks that the same votes appear later in the process. The most important properties of this system design concern integrity and privacy.

Integrity

The crucial integrity claim from the iVote Strategy document is: *"This checking process can only be subverted if both systems are compromised in a harmonised way and the tampering was not evident in the logs."* (p.6)

The claim is saying that it would require only two compromised machines (the vote encoder and the "verification server") to modify votes. We disagree with both the utility and the correctness of this claim.

First observe that, even if this claim were accurate, it would fall far short of either the observability of ordinary paper-based counting or the verifiability of end-to-end verifiable electronic systems such as vVote, in which the proof of correct vote handling is publicly observable rather than being attested to by a single auditor. Even postal voting, which is far from perfectly secure, is not susceptible to complete system compromise from only two unobservable points.

Furthermore, based on the draft design document, the claim appears to be incorrect. A compromise of *either* the NSWEC system or the "verification server" could allow vote tampering in some cases:

[1] The first two authors also work on the vVote project at the Victorian Electoral Commission; the third is in discussions about reviewing some of the vVote code. This analysis of iVote is our own and is not in any way associated with the VEC.

- The “verification process” may allow voters to check that votes have not been deleted, but seems to provide no protection against the Ballot Controller System stuffing the ballot with votes that do not originate from real voters.
- The “verification server” is described as checking the votes that are output near the end of the electronic process, not those that actually go into the count. There is no mention of anyone checking that the votes that the “verification server” has “verified” are those that actually go into the count. Without such a check, this gap would be an opportunity for manipulation.
- The process for deleting votes is not precisely specified. Although the “receipt numbers” are intended to help identify fraudulent deletions, it is unclear how either their uniqueness, or their correspondence to vote deletion requests, is verified.

Privacy

The Draft design claims that it requires two compromises to break privacy. *“The underlying design objective for iVote is to ensure voter secrecy. Two or more systems would need to be breached for this to be compromised.“* (p.6) We disagree with this claim as well, in both its merit and its correctness.

First observe that plaintext vote checking, in which voters can log in and see a human-readable version of their vote, is a significant departure from other Australian precedents for vote privacy. Although postal voting privacy is certainly imperfect, we know of no other voting system that provides specific support for voters to prove how they voted after casting it. Any remote voting system is susceptible to some coercion, but this design actually facilitates and simplifies coercion compared with postal voting. Voters will be able to log in after voting and show someone else how they voted. They will also be able to transfer their voting credentials after the fact to someone who can check how they voted.

Second, based on the draft design document, the claim appears to be incorrect. A compromise of *either* the vote encoder or the “verification server” would suffice to link a voter’s identity to their plaintext vote in many cases.

- Both the vote encoder and the “verification server” communicate about the vote in plaintext (unlike alternative designs such as Helios, in which the voter encrypts their vote on their computer before sending to the server). Hence vote privacy relies entirely on the assumption that it is impossible to identify an individual when they visit the server. In general this assumption is incorrect, though it depends on which computer the voter is using. A voter who deliberately borrowed someone else’s computer or used one in a public space would be relatively hard to trace; a voter who used their own smartphone or PC, and who was not careful about privacy settings, could in many cases be identified. This privacy violation would be exacerbated by attackers’ linking with other sources of information such as ISP information, IP geolocation, and much more.
- Any attacker who compromises the “verification server” discovers all the votes. The protection of the vote with a 10-digit (32 bit) key is easily reversed with a brute-force attack. The attacker would then need to find a way to link the iVote ID to the voter’s identity.

Overall the iVote strategy document has very weak objectives for both vote privacy and verifiability, but even these objectives do not seem to be met by the draft design. The draft design is so vague that it is impossible to rule out other serious issues that are not yet evident. Only with more detailed documentation would a more thorough analysis be possible.

Comment [VT2]: (March 2015) The main significant improvement of iVote since 2013 has been the implementation of encryption at the client side. Hence this criticism no longer applies to the improved version of the voting system, though it does still apply to the “verification” system. Note also that the polling-place version of iVote requires voters to register and then vote from the same machine, which raises many of the same privacy concerns.

Comment [VT3]: (March 2015) More recent documents use 12-digit (40 bit) Receipt Numbers as keys, which are also easily opened by brute-force. A “Random Extension” of the key is also mentioned in the newer versions of the description, without an explicit length.