Legislative Assembly for the
Australian Capital Territory
Standing Committee on the Integrity Commission and Statutory Office Holders

Submission Cover Sheet

Inquiry into the operation of the 2024 ACT Election and Electoral Act 1992

Submission number: 020

Submitter: Chris Culnane, Andrew Conway, Vanessa Teague, Ty Wilson-Brown

Date authorised for publication: 4 November 2025

Submission to the Inquiry into the operation of the 2024 ACT Election and Electoral Act 1992

Chris Culnane Andrew Conway

Vanessa Teague Ty Wilson-Brown

September 28, 2025

1 The state of ACT Election conduct

Paperless electronic voting machines continue to expose ACT election outcomes to the risk of alteration through software error, hardware failure, electronic attacks or other deliberate fraud. The absence of a voter-verifiable paper record means that ACT election outcomes are *unavailable to meaningful scrutiny*, and are not accompanied by *any evidence that the announced result is accurate.*

Inadequate randomisation of published votes has exposed ACT voters' choices to any attacker who knows the order in which votes were cast. (For example, from surveillance cameras around a polling place, or by accessing electoral roll mark-off times.)

Insecure connections between voting clients and servers in polling places meant that any attacker with private network access could easily observe or modify votes.

Biased selection of the starting (party) column in audio voting may have given an advantage to some parties in the 2020 election. We were unable to confirm if this impacted non-audio electronic voting methods.

Audits performed to certify the ACT's electronic voting and counting systems continue to miss significant coding errors, both within and outside the auditing firm's areas of expertise.

1.1 Our recommendations

We have made the same recommendations to the relevant Committee consistently for several election cycles.

We recommend that ACT Electoral law be amended, *and the Electoral Commission funded* to ensure:

1. that in order to have some chance of detecting the most serious errors and vulnerabilities, electronic voting code and system documentation be made openly available for public inspection, at least six months before the election, including:
   (a) e-voting code,
   (b) paper ballot scanning code,
   (c) counting code,
   (d) electoral roll mark-off code, (due to its involvement in privacy issues in prior elections),
   (e) system requirements documentation,
   (f) system design documentation,
   (g) system test plans and test results,

(h) system accuracy, integrity, and privacy audits, and

(i) any relevant changes to the interpretation of electoral legislation by the Electoral Commission;

2. that different auditing firms are used, and rotated regularly, as is standard software engineering practice, and

3. that all system modifications, audits, and declarations be completed before candidate nomination closes, with any changed code, documentation, and legislative interpretations publicly released;

4. that the pollsite e-voting system have a voter-verifiable paper record, so that an immutable record of the vote can be verified by the voter independently of the software;

5. that paper ballot scanning checks should manually check swapped (but valid) preference orders, and missed final preferences, in the same way as invalid preference orders;

6. that when the electronic preferences are published, there should be a thorough, public, statistical audit of the paper ballots, whether filled in by hand or printed by eVACS;

7. that the number of audited paper ballots should be substantially increased, so that the residual error rate could not have affected the outcome of any recent election (to a 99% or greater confidence level);

8. that all manually corrected and informal paper ballots should be audited in the same way as the sample from the entire set of paper ballots;

9. that the paper ballot scanning audit report should include raw scanning error rates, and the correction rate at each manual checking stage; and

10. that Internet voting be permanently discontinued, due to the high levels of risk involved in current Internet voting technology.

"Openly available" means without a confidentiality deed.

1.2 Some progress, but not enough

Although there has not been significant legislative change, the ACT Electoral Commission has voluntarily addressed some of the recommendations above. In 2024, code for the eVACS e-voting and e-counting systems was made openly available (Recommendation 1), though documentation was limited and code for the scanning and roll mark-off code was not released. We were able to obtain some of this documentation via FOI for the 2020 election, and for the 2024 election, but with substantially increased redactions[1]. An improved statistical audit was conducted (Recommendation 6), which for the first time in the ACT examined actual paper ballots rather than only their images ([1], p.73). Internet voting was temporarily discontinued for 2024 (Recommendation 10). Although this is progress, *there is still an obligation for legislative reform*, because a future ACT Electoral Commission could simply change their mind on any of these processes. There is also a need for appropriate funding to support these changes.

The voter-verifiable paper record (Recommendation 4) has not been actioned. This means that ACT election outcomes continue to rely on software and hardware components that may fail undetectably, producing election outcomes that do not match voter choices. The improved audit of ballot papers does not address the integrity of ballots cast without paper.

2 eVACS security, privacy, and bias problems 2020-24

In March 2024, the ACT Electoral Commission released source code for the eVACS voting and counting modules. The program is *insecure by design against external attackers, malicious insiders or errors, and is not suitable for use in an election.*

We also identified specific serious security problems, related to privacy, integrity, and vote bias. A detailed report is available separately [2]. The three main findings were:

[1] A full list of our FOIs is published at: https://www.righttoknow.org.au/body/act_electoral_commission

- Due to the use of a weak pseudo-random number generator, the pseudo-random value (called `pindex`), attached to each published vote and which is used to shuffle the votes, can be used to recover the voting order. An attacker who learns individual voting order can therefore learn some individuals' votes.

- Due to a vulnerability in an underlying library, the secure network connection between the voting client and voting server was not being correctly validated. This left the connection open to attack by network intruders, who potentially could collect or modify votes.

- Due to a miscalculation during party column randomisation, some parties were shown first up to 1.6% more than other parties, in telephone voting and audio voting at polling booths in 2020.

The process of discovering these vulnerabilities, and the subsequent disclosure process, raised a number of broader security concerns, which will be detailed below. A more detailed technical discussion of the vulnerabilities is provided in our online paper [2]; a shorter high-level summary is included below.

2.1 Weak Pseudo-Random Number Generator

The eVACS system used a weak, non-cryptographically secure, Pseudo-Random Number Generator (PRNG). As a result it was possible to perform a brute-force (exhaustive) search of all possible seeds to recover the seeds used, and as a result recover the pseudo-random output from the PRNG. This output was used to set the `pindex` value, which in-turn was used to shuffle the votes. By recovering the PRNG output we were able to recover the order of the submission of eVACS 2020 votes and therefore invert the shuffling. The published eVACS votes from 2020 thus expose individual votes to any attacker who knew the order in which they were cast.

Some significant sources of voter order are:

- observing who is voting next to you at a polling booth, or who voted first or last in a day,

- surveillance footage around the entrance or approaches to a polling place, particularly when there are small numbers of voters, (these cameras are often deliberately or accidentally made openly available over the internet), or

- the electoral roll mark off times recorded by the networked roll software.

While the Electoral Commission makes its best efforts to secure this sensitive data, sometimes mistakes are made. Outside (or inside) attackers regularly obtain access to government data when they shouldn't, even from supposedly secure agencies.

The underlying issue is described in CVE-2024-41708.

2.2 Network Security

Polling place voting booth clients used a software library that did not perform TLS certificate validation correctly - the basis on which the connection is made secure. In the absence of such checks the client could not be certain it was talking to the correct server. As a result, network intruders could easily collect or modify votes. The underlying issue is described in CVE-2024-37015.

Additionally, the same Weak PRNG that eVACS was using was also used by the AdaCore library and as a result a second vulnerability was disclosed in relation to their usage of the weak PRNG. These security issues in the library could have also impacted eVACS, in addition to eVACS' own use of the weak PRNG.

2.3 Starting Party Column Bias

During electronic voting, the initially selected column on the ballot is randomised. Unfortunately, in the 2020 election, this calculation was biased towards the leftmost parties on the original ballot.

Each column represents a political party, or ungrouped independent candidates. The order of the party columns on all ballot papers is chosen randomly, as part of the ballot draw. The order of candidates within a column is also rotated between different ballot papers, using the Robson Rotation. These processes are used for both paper and electronic voting. We are not aware of any bias in these processes.

However, electronic voting *also randomises the initially selected column of candidates*. This randomisation is used during audio voting with physical keypads at polling booths, and over the phone. We were unable to confirm if it also happens for touch screen voting.

The exact bias depends on the number of columns. For 2020, the resulting biases were[2]:

- 8 columns on the ballot:
  - **Brindabella** no bias
- 11 columns on the ballot:
  - **Ginninderra** bias of 1.2% towards 3 unelected parties
- 9 columns on the ballot:
  - **Kurrajong** bias of 1.6% towards Greens, Liberals, and 2 unelected parties
  - **Murrumbidgee** bias of 1.6% towards Greens and 3 unelected parties
  - **Yerrabi** bias of 1.6% towards Greens, Liberals, and 2 unelected parties

Further details of this analysis, and our calculations, are available in our paper [2].

This is not a large source of bias, and may not impact a large portion of voters. However it is entirely unnecessary, and the margins in ACT elections are very small, so it is difficult to tell what impact this had.

We believe this issue was fixed before the 2024 election, but we have not reviewed the code in detail.

2.4 Responsible Disclosure, Timeline, and Support Contracts

We disclosed both network security issues to Adacore, and both were addressed in time for the 2024 election.

However, the responsible disclosure policy from AdaCore provides AdaCore with up to 90 days before we would have been permitted to disclose the vulnerabilities publicly and by extension to the Electoral Commission or Software Improvements. As it turned out, AdaCore resolved the issues quicker than that, and went public with the vulnerabilities and fixes in 57 days for the certificate validation issue, and 36 days for the use of the Weak PRNG.

AdaCore clients, those with a support contract, were informed before the public notification and provided with a description of the vulnerability and any possible workarounds - prior to a fix being available. That notification was sent to clients after 29 days for the certificate validation and 21 days for the Weak PRNG, both significantly sooner than when fixes were available and the existence of the vulnerabilities was made public.

We were not at liberty to disclose the AdaCore issues directly to the Electoral Commission or Software Improvements, as the bugs were not directly in their code, and we had a responsible disclosure agreement with AdaCore.

However, we could disclose the usage of the weak PRNG within eVACS itself, which we did on the 26th June. They then discussed this vulnerability internally[3]. This date was *after AdaCore had notified their clients with support contracts about the issue with the certificate validation.* During liaising with the Electoral Commission and Software Improvements we inquired as to whether they were aware and had taken action on the notification, *without* disclosing what was in the notification. (Our names were listed in the advisory, so the Electoral Commission or Software Improvements would have known we had made the disclosure to AdaCore). However, it became apparent that no notification had been received because Software Improvements did not have a support contract with AdaCore, and had not had one since it had lapsed in 2021.

This presented a significant problem, as we could not predict when a fix would be available and therefore when the vulnerability would be made public, beyond the 90 day deadline. If the full 90

[2] https://www.elections.act.gov.au/elections_and_voting/past_act_legislative_assembly_elections/2020-election/2020-election-results/electorate-results

[3] Obtained via FOI: https://www.righttoknow.org.au/request/12283/response/39502/attach/7/EACTFOI2024110402Document2.PDF.pdf, full request: https://www.righttoknow.org.au/request/paper_vote_scanning_accuracy_in_2#incoming-39505

days were used, the public disclosure would have taken place on the 12th August for the certificate validation issue, and the 23rd September for the Weak PRNG in AdaCore issue, both of which would have been **very close to the election date** to be requiring code and dependency updates, new builds, testing, and auditing.

To reduce this risk, we requested either AdaCore contact the Electoral Commission/Software Improvements, or permit us to do so. AdaCore graciously agreed to provide the disclosure information and engage with the Electoral Commission/Software Improvements prior to their public disclosure, even in the absence of a support contract. One of us attended their meeting[4]. Elections ACT also corresponded with AdaCore via email on multiple occasions, to obtain detailed technical advice[5]. It should be noted that AdaCore were under no obligation to provide this free support, particularly at short notice. Had a support contract been in place, Elections ACT would not have had to depend on the goodwill of AdaCore.

We were fortunate that the 2024 source code was not under a Non-Disclosure Agreement (NDA) with Elections ACT. If we were under an NDA, we wouldn't have been able to disclose to AdaCore that Elections ACT was impacted. And we already couldn't disclose the AdaCore vulnerabilities to Elections ACT, due to our responsible disclosure agreement with them. So our only recourse would have been to encourage Software Improvements to urgently obtain a support contract, so they could obtain this information from AdaCore themselves. This could have been a difficult ask, without providing any specific details of the vulnerabilities.

2.4.1 Absence of a Support Contract

The absence of a support contract raises a number of questions:

- What version of the AdaCore toolset, library and runtime was going to be used for the election in 2024?
- Had Software Improvements transitioned to the open source version or were they still using an out of date GNAT Pro version from 2021?
- In the absence of a support contract, how were vulnerabilities and bugs that had been found in the AdaCore runtime and libraries - beyond ours - being addressed - if they even were?
- Was a Support Contract obtained to cover the 2024 election? Will it be maintained in the future?
- Was Software Improvements equipped with sufficient staff and knowledge to address the issues both within eVACS, and to apply the fixes associated with the updated AdaCore libraries?

It is extraordinary to think that eVACS could be run in an election without a support contract with AdaCore, and therefore without the most up-to-date information/workarounds/fixes for the runtime and libraries it so heavily depends on.

2.5 Inadequate Testing And Auditing

The vulnerabilities in question were obvious from a quick, high-level inspection. Two of the authors of this submission independently determined there was a problem with the TLS (secure connection) within hours of seeing the code. What took time was finding the root cause and determining whether the vulnerability was a configuration issue or an inherent problem (it was the latter).

The reason we could so easily determine there was a problem was by inspecting the TLS certificate that was included in the source code release, an extract of which is shown in Figure 1.

There are two values of note, firstly, the **security certificate expired in 2022**, second, the Common Name (CN) field is a name not a domain name (URL) or IP address[6]. The first issue is the clearest sign of a problem. Expired certificates should not verify. If source code is being packaged with

[4]We later obtained a summary via FOI: https://www.righttoknow.org.au/request/12283/response/39502/attach/6/EACTFOI2024110402Document1.PDF.pdf

[5]Including 8 July 2024: https://www.righttoknow.org.au/request/12283/response/39502/attach/9/EACTFOI2024110402Document5.PDF.pdf, and 31 July 2024: https://www.righttoknow.org.au/request/12283/response/39502/attach/8/EACTFOI2024110402Document3.PDF.pdf

[6]There are also no extensions present that contain any domain or server address information.

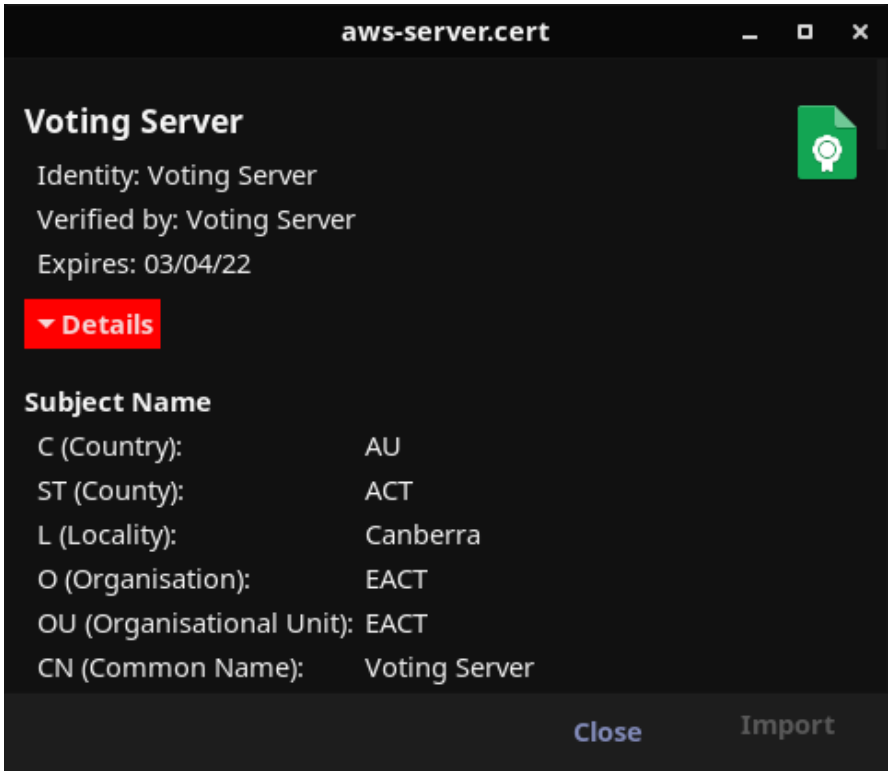

Figure 1: eVACS Server TLS Certificate Extract

an expired certificate, even if only during testing or auditing, and that code is running successfully, it indicates the certificate validation is not working (or not being tested in any way).

The second issue indicates the domain name validation - a crucial part of TLS - is not active, since it requires some form of domain name (URL) or IP address to validate against. Whilst there can be cases for not needing it, for example, where certificate pinning has been used, i.e., only one certificate is considered acceptable, a secure implementation of TLS should perform such validation by default, indicating the configuration would need to have been customised to disable it - potentially introducing errors, particularly when taken in conjunction with the expired certificate.

Such insights were gained immediately on inspection. These should have been picked up during testing, and most definitely should have been spotted during any competent security audit. Despite extensive provider assessments, security plans, and risk management plans[7], these basic security vulnerabilities were present in the April 2024 version of eVACS, and (we assume) the version used in the 2020 election.

2.5.1 bmm Audit

bmm testlabs were commissioned to perform a full source code review of eVACS[8]. bmm testlabs asserts in their report that they reviewed the codebase to "confirm the absence of any malicious or unintentional elements that could potentially impact the integrity of election results".

Biased randomness calculations are is a well-known issue in converting a larger random number to a smaller one. An auditor with extensive experience in auditing gaming machines (and their probability distributions) should be aware of this kind of bias, and have experience detecting and correcting it.

Separately, the TLS Certificate file *./ssl/aws-server.cert* falls within the project scope and therefore should have been "thoroughly reviewed". The file itself is listed as part of the Voting Module File

[7]Obtained via FOI for the 2024 election: https://www.righttoknow.org.au/request/security_design_documents_for_el#incoming-35661, and 2020 election: https://www.righttoknow.org.au/request/vote_secrecy_in_2020_election#incoming-19178

[8]An archived copy of the report is available at: https://web.archive.org/web/20240711100324/https://www.elections.act.gov.au/__data/assets/pdf_file/0020/2510192/eVACS-code-certification-62024.PDF

13.2.2. Voting Server Project

File Name	SHA256 File Hash
./bin/aws.pem	955b410fc3b860401c75633e65f9434528bc67ac7f1b3a6c9a7b898ab90994df
./ssl/aws-server.cert	56f1b16621ed23ddef35d31449939b03decd8f35b70433497619f09497289c16
./ssl/aws-server.key	981cb5ba55562a3157ac1e857bbd373a2905a3b089d45066d91ef62095b3c2da
./ssl/aws.pem	fa0b4dbc9c55434ae5f6d14f2450856d386fe0c39f02ecb86fd8b5dc7c2ca43b
./ssl/evacs.rand	4eb84efe420193dae39aa55d4006c7d14c350981fc616781989f868256041abf
./ssl/ssl-readme.txt	ef4692d06a02dd37b93188108defd03f003659531181b45e202089953d342a86

Figure 2: bmm testlabs - listing and hash of expired TLS certificate

Identification table along with a SHA256 hash of the file contents, see Figure 2

The SHA256 hash allows us to verify that the file that was made available for public inspection was the same as the one bmm testlabs reviewed. Indeed, the file we downloaded has a SHA256 fingerprint of:

```
56f1b16621ed23ddef35d31449939b03decd8f35b70433497619f09497289c16
```

identical to the fingerprint of the file reviewed by bmm testlabs and shown in Figure 2. As a result we can be certain that bmm testlabs received an expired TLS certificate as part of their review - their review took place in 2024 and the certificate expired in 2022.

It is inexcusable for a review to fail to notice that the TLS certificate has expired. If the source code received by bmm testlabs was supposed to be a working copy of eVACS, the expired certificate would immediately point to a problem with the TLS configuration or implementation. Yet no mention of it is made in the bmm testlabs report, and the source code is deemed compliant.

The audit also fails to verify that a good source of randomness is used by eVACS. eVACS used the core PRNG provided within Ada, however, that PRNG is not cryptographically secure. A competent audit would verify that the PRNG is considered to be cryptographically secure. In the absence of such information in the documentation the audit should inspect the source code, where it is available, and verify the PRNG used. If bmm testlabs had done so they would have found the following (as we did):

```
-- Note:  the implementation used in this package is a version of the
-- Mersenne Twister.  See s-rannum.adb for details and references.
-- It is suitable for simulations, but should not be used as a cryptographic
-- pseudo-random source.
```

which explicitly states it should not be used as cryptographic (secure) PRNG.
As such, we would conclude that the audit was deficient for the following reasons:

- It should have been apparent from the specification that a cryptographically secure PRNG was required for *pindex*. The privacy of the vote was dependent on the shuffling (randomisation) of the ballots. As such, the source of randomness should have been checked, and verified that it was from a recognised, secure CSPRNG.
- If the TLS certificate had been inspected, it would be obvious that there was a problem with the TLS implementation, since the certificate should not have worked. The fact this was not flagged indicates a failure to either understand the contents of the TLS certificate, or failure to actually inspect its contents.
- The column randomisation calculation had a well-known coding pattern that causes biased outputs, which should have been detected and corrected.

2.6 Deficiencies in eVACS

When the Ada version of eVACS was first implemented, the importance of using a CSPRNG for *pindex*, should have been apparent. The use of a non-CSPRNG constitutes a defect within the original

implementation. Whilst the auditing should have picked it up it, it should not have been implemented in that manner in the first place.

In a similar regard, whilst the use of the AdaCore AWS library to provide the TLS functionality ultimately resulted in the vulnerability in that library being present in eVACS, even the most basic of testing during development should have picked up the issue. When implementing TLS, it is imperative it is tested with both valid and invalid connections to ensure it is working as intended. Clearly no such testing took place, and as a result eVACS was released with a defective TLS implementation.

More broadly, the acceptance testing performed by the Electoral Commission should also have verified that the TLS connection was working as intended, and that it would not accept invalid connections. The overarching issue is that none of the parties involved did what they should have.

- Software Improvements did not do the most basic of testing, and did not select a suitably secure CSPRNG for *pindex*
- The Electoral Commission's acceptance testing did not check that the TLS functionality actually worked securely, as intended
- bmm testlabs did not correctly audit for unbiased randomisation, check TLS Certificate was secure, or verify a secure source of randomness for *pindex*

It should also be noted that these defects are not isolated to the 2024 version. The design documents for the 2020 eVACS included column randomisation.[9] We know the weak PRNG was used to generate the *pindex* in 2020, as we were able to recover the vote orders using the published data. Since the TLS vulnerability was in the underlying AWS library and we assume the same library was used in 2020, we can reasonably assume the defects were present before 2024.

This raises serious questions about the adequacy of all phases of eVACS's production: the management, development and auditing all saw serious errors. These defects were not complicated nor nuanced—they were basic failures that should not have happened, and should have been detected at the review, testing, and audit stages.

Of perhaps greatest concern is that **none of the events above have been detailed in the ACT Electoral Commission's report to the Legislative Assembly**. A defect that biases ballots (however small), or allows the recovery of vote order in the previous election, or a critical security vulnerability in the secure communications between client and server, would seem sufficiently serious to warrant mentioning. Furthermore, the situation could have been far worse had AdaCore not graciously permitted early reporting to Software Improvements and the Electoral Commission, without which the time to fix the issues and apply the patches would have been significantly shorter. AdaCore did this in spite of Software Improvements allowing their support contract to lapse.

It is also important to note that these vulnerabilities were only detected and corrected before the election, *because the source code was made publicly available*. Unlike a network attacker, who could have immediately discovered the insecure connections, we relied on source code to discover these issues. Whilst these vulnerabilities should have been picked up long before we reviewed the source code, this shows the critical importance of allowing experts to independently and freely review source code for critical pieces of democratic infrastructure.

3 Notes about the paper ballot audit

We are pleased that the Electoral Commission has adopted our 2020 recommendation of an independent paper ballot audit. For many election cycles, the Electoral Commission has conducted a multi-stage ballot paper scanning and manual correction process *based on scanned images*:[10]

- Ballot papers are scanned and their images processed by automated Optical Character Recognition. Then individual preference number images (without the surrounding ballot paper) are manually checked by Elections ACT staff.

[9] Obtained via FOI: https://www.righttoknow.org.au/request/vote_secrecy_in_2020_election#incoming-19178

[10] Ballot Paper Scanning Assurance Plan, obtained via FOI: https://www.righttoknow.org.au/request/12282/response/39505/attach/9/EACTFOI2024110401Document03.PDF.pdf

- When the system detects an invalid (non-sequential) preference order in a scanned ballot image, it sends that ballot to a ballot image checking process.
- Electoral Commission staff correct the scanned ballot image preferences, in the presence of candidate scrutineers. If there is a disagreement or uncertainty, the ballot image is sent on to further image checking and correction stages.

In 2024, Elections ACT added an independent audit of a random sample of paper ballots. This is a tremendous improvement, because it allows for the detection of discrepancies between the paper ballot and the scanned image. Previously, it was typically the scanned images were checked (which excluded any scanning defects from those checks).[11]

The paper audit is fundamentally the right approach, with a few issues that should be looked at:

- The margins of elimination with elected candidates are frequently small enough that *the sample rate is not good enough to always detect a change in the winner*, e.g. in Brindabella swapping two preferences on 42 ballots was enough in 2020 to change who was elected (84 in 2024). The auditing system was designed to demonstrate with high confidence that **the correct candidates were elected** (an ambitious and laudable goal)[12]. But the number of paper ballots in 2024 was larger than the expected maximum, so the actual sample rate did not provide the desired assurances. To demonstrate with high confidence that all candidates were correctly elected, we recommend an increased sample rate, based on the largest number of paper ballots, and smallest elimination margins for elected candidates, over multiple recent elections. (Unfortunately, the number of paper ballots, elected candidates, and their elimination margins in the current election aren't known until after all paper ballots have been received, scanned, and checked.)
- Some kinds of *incorrectly scanned paper ballot images are not detected* and manually checked by Electoral Commission staff, due to the design of the current scrutiny process. The way the scrutiny process is designed, scanned ballots are immediately accepted if scanning errors result in a valid preference order (and the individual digit images are accepted by an operator as a "good enough" match). This includes swapped preferences, and missed highest preference numbers. However, the audit process is designed to catch these kinds of errors, by checking that each scanned preference matched the original paper ballot, and there are no missing preferences. We are pleased the Electoral Commission has adopted our previous recommendation for the audit process, and encourage them to extend it to the scrutiny process as well.

The most significant limitation of the paper ballot audit is, obviously, that it can apply only to those votes that were expressed by voters on paper. The eVACS electronic ballots, in which the voter never sees reliable evidence of the contents of their vote outside the computer screen, are unavailable for any meaningful audit. Unfortunately, unlike paper ballots, it is possible to tracelessly show one set of preferences to the voter on the eVACS screen, but record another on disk. This can happen due to software or hardware defects, or deliberate internal or external source code or compiled software binary modifications.

4 Internet voting

Secure Internet voting remains an unsolved problem. The problem is not (only) the kind of obvious failures that occurred in NSW in 2021, when the iVote system's unscheduled downtime disenfranchised thousands of voters. That is not the worst thing that can happen—the worst thing that can happen is that everything seems to be fine, but the election is subject to undetectable electoral errors or fraud.

The opportunity to cast a remote paperless vote has all the disadvantages of eVACS's lack of independent voter-verifiable records, plus a much wider class of possible attack vectors, available to anyone on the Internet.

[11] Clarification email for 2020 election FOI request: https://www.righttoknow.org.au/request/paper_vote_scanning_accuracy_in#incoming-20380

[12] Sample size for a Ballot Paper Scanning Assurance Plan for ACT Legislative assembly elections, obtained via FOI: https://www.righttoknow.org.au/request/12282/response/39505/attach/6/EACTFOI2024110401Document10.PDF.pdf

5 Conclusion

Recommendation 4—attaching a voter-verifiable paper record to the direct recording electronic voting system—remains a critically important requirement for the integrity of ACT elections. Current paperless assurance methods, based on trusting software and hardware, have been repeatedly shown to be vulnerable in the ACT and other jurisdictions. They cannot reasonably be expected to maintain public trust or reliable election counts.

References

[1] ACT Electoral Commission. Report on the ACT Legislative Assembly election 2024, 2025.

[2] Chris Culnane, Andrew Conway, Vanessa Teague, and Ty Wilson-Brown. Security analysis of the australian capital territory's evacs 2020/2024 paperless direct recording electronic voting system. *arXiv preprint arXiv:2409.13570*, 2024. https://arxiv.org/abs/2409.13570.