Dear Chairwoman McCormick,

In the summer of last year, our team of election security researchers at the University of Michigan and Auburn University discovered a privacy flaw that affects several voting machines produced by Dominion Voting Systems, which we named DVSorder. We specifically discovered that when these systems are used to produce ballot-level election records, such as CVRs or ballot image files, they reveal information about the order in which ballots were cast on election day. This flaw poses a significant risk to voter privacy, but went undetected when the devices were repeatedly certified under the VVSG 1.0 guidelines.

We believe that VVSG 2.0 would similarly fail to catch this vulnerability with the standards in their current form. In light of this, we propose an addition to the VVSG 2.0 Test Assertions that requires VSTLs to perform an explicit check that no similar flaws are present in future machines seeking certification.

DVSorder arises from a piece of metadata attached to each ballot, which is called its *record ID*. Each ballot is assigned such an ID when it is cast through Dominion-brand voting machines. These IDs are meant to be randomly generated so that they cannot be tied to the voter who cast the underlying ballot. Our team discovered, however, that the randomization algorithm used by Dominion ICP and ICE ballot scanners is based on a linear congruential generator (LCG), a flawed type of random number generator which has been known to be unsuitable for security purposes since the 1970s[1]. This means the IDs assigned to ballots are entirely predictable, and that with appropriate records containing these IDs (e.g., CVRs), any member of the public can derive the order in which every ballot was cast.

This is not an isolated flaw. In 2003, researchers found a similar random number generator error in the Diebold AccuVote-TS system[2]. Broken RNGs have since been discovered in Hart InterCivic, Sequoia, and ES&S-manufactured election machines[3][4].

---

[1] Reeds, James. "'Cracking' a Random Number Generator." *Cryptologia*, vol. 1, no. 1, 1977, pp. 20–26., https://doi.org/10.1080/0161-117791832760.
[2] Kohno, T., et al. "Analysis of an Electronic Voting System." *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, 2004, https://doi.org/10.1109/secpri.2004.1301313.
[3] Inguva, S., et al. "Source Code Review of the Hart InterCivic Voting System." Part of the California Secretary of State's "Top-to-Bottom" Voting Systems Review, 2007, https://votingsystems.cdn.sos.ca.gov/oversight/ttbr/Hart-source-public.pdf
[4] Blaze, M., et al. "ES&S specific weaknesses and their implications." *EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing,* 2007, https://www.eac.gov/sites/default/files/document_library/files/EVEREST.pdf

These flaws have occurred in the United States[5] and abroad[6].

There are two specific provisions in VVSG 2.0 that we identify as being relevant to this class of privacy flaw.

First, VVSG 2.0 clause 10.2.2-B states the following:

> *The voting system must not contain data or metadata associated with the CVR and ballot image files that can be used to determine the order in which ballots votes* [sic] *are cast.*

Second, VVSG 2.0 clause 10.2.2-E states the following:

> *Randomly generated identifiers used for audits must use random bit generators specified in the latest revision of NIST SP 800-90 series on random bit generators.*

The DVSorder flaw — as well as the broader class of similar vulnerabilities — clearly violate both of these standards. The generator used to produce record IDs for Dominion introduces metadata that can perfectly restore ballot order, and an obfuscated LCG of this nature is certainly not specified or standardized by NIST. Unfortunately, however, the latest version of the VVSG 2.0 test assertions contains *no explicit checks* that enforce the aforementioned clauses. We worry that, as a consequence, both voting machine manufacturers and accredited testing laboratories might overlook these requirements when designing and certifying systems.

We note that *neither of these requirements* can be suitably tested without source code access. To see why, one need only note that encrypted information is — by design — indistinguishable from random by anyone who does not know the secret key. This means a black box voting machine could trivially assign metadata to candidates which encrypts the order of casts. That is, the first candidate could be assigned Enc(1) as her ID, the second could be assigned Enc(2), and so on. To any observer who did not know the vendor's secret key, these IDs would be indistinguishable from random. To the vendor, however, the IDs would perfectly reveal the order in which ballots were cast. Thus, the only way to validate clauses 10.2.2-B and E are satisfied is for a VSTL to examine the voting system source code.

---

[5] Blaze, M., et al. "Source Code Review of the Sequoia Voting System." Part of the California Secretary of State's "Top-to-Bottom" Voting Systems Review, 2007, https://votingsystems.cdn.sos.ca.gov/oversight/ttbr/sequoia-source-public-jul26.pdf

[6] Aranha, D. F., et al. "Software vulnerabilities in the Brazilian voting machine." *Design, Development, and Use of Secure Electronic Voting Systems,* pp. 149-175. DOI: 10.4018/978-1-4666-5820-2

We encourage implementation of specific language which would require VSTLs to examine voting machine source code and determine whether the random number generators used to create ballot metadata and facilitate audits comply with the stated standards. Specifically, we propose the following language to add to the VVSG 2.0 Test Assertions:

> ***10.2.2 - Identification in vote records***
>
> ***TA1022E-1:*** *The voting system MUST produce all randomly generated identifiers as output from random bit generators specified in the latest revision of* NIST SP 800-90 *series on random bit generators.*
>
> > ***TA1022E-1-1:*** *The system MAY re-generate randomly produced identifiers to ensure each ballot or ballot page is assigned a unique identifier as specified in requirement* 9.4-C - Unique ballot identifiers.
>
> ***TA1022E-2:*** *The voting system MUST use some unpredictable entropy to seed the random bit generator which is used to produce randomly generated identifiers.*
>
> > ***TA1022E-2-1:*** *The voting system MUST NOT use some fixed value to seed the random bit generator.*
> >
> > ***TA1022E-2-2:*** *The voting system MUST NOT use a time value to seed the random bit generator.*
>
> ***TA1022E-3:*** *The voting system MUST NOT store or report the value used to seed the random bit generator.*

This language would ensure that VSTLs evaluate whether voting systems satisfy the VVSG 2.0 requirements concerning randomly generated identifiers. It would also ensure the detection of the class of vulnerabilities described above, and help to protect voter privacy in future elections.

We also propose revisions to an existing test assertion:

> ***TA94C-1:*** *The voting system MUST EITHER have the capability of preserving the ballot scanning order or MUST be capable of affixing a unique ballot identifier such as scanner ID, batch ID, or ballot card number.*

The capability to preserve ballot scanning order should only be used if ballots have been thoroughly shuffled *before* they are scanned, which is—to our knowledge—only practiced today in a central-count context. The functionality thus should *not* be accepted in a precinct-count context, where preserving ballot order allows deanonymization of voters. Specifying this in a sub-assertion would be a valuable step towards protecting voter privacy.

There also exists a drafting issue in this assertion. The scanner ID, batch ID, and ballot card number fields function as a unique identifier only when taken together, so the final instance of the word “or” should be replaced by the word “and.” This would bring the test assertion into line with the language in the underlying VVSG 2.0 requirement.

We hope that the Commission will take these comments into consideration when evaluating and updating the text of the VVSG 2.0 test assertions. Our lab is available as a resource to you in this process, and we would be happy to collaborate further to improve testing practices.

Sincerely,

Braden Crimmins, University of Michigan
Dhanya Narayanan, University of Michigan
J. Alex Halderman, University of Michigan
Andrew Springall, Auburn University