DVSorder: Ballot Randomization Flaws Threaten Voter Privacy

Braden L. Crimmins and Dhanya Y. Narayanan, *University of Michigan;*
Drew Springall, *Auburn University;* J. Alex Halderman, *University of Michigan*

https://www.usenix.org/conference/usenixsecurity24/presentation/crimmins

This paper is included in the Proceedings of the
33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

DVSorder: Ballot Randomization Flaws Threaten Voter Privacy

Braden L. Crimmins
University of Michigan

Dhanya Y. Narayanan
University of Michigan

Drew Springall
Auburn University

J. Alex Halderman
University of Michigan

Abstract

A trend towards publishing ballot-by-ballot election results has created new risks to voter privacy due to inadequate protections by election technology. These risks are manifested by a vulnerability we discovered in precinct-based ballot scanners made by Dominion Voting Systems, which are used in parts of 21 states and Canada. In a variety of scenarios, the flaw—which we call DVSorder—would allow attackers to link individuals with their votes and compromise ballot secrecy. The root cause is that the scanners assign pseudorandom ballot identifiers using a linear congruential generator, an approach known since the 1970s to be insecure. Dominion attempted to obfuscate the generator's output, but we show that it can be broken using only pen and paper to reveal the order in which all ballots were cast. Unlike past ballot randomization flaws, which typically required insider access to exploit or access to proprietary software to discover, DVSorder can be discovered and exploited using only public information.

In addition, the election sector's response to our findings provides a case study highlighting gaps in regulations and vulnerability management within this area of critical infrastructure. Although Dominion released a software update in response to DVSorder, some localities have continued to publish vulnerable data due to inadequate information sharing and mitigation planning, and at least one state has deferred addressing the flaw until after the 2024 presidential election, more than two years following our disclosure.

1 Introduction

The secret ballot is a crucial election security mechanism that helps ensure people can vote their conscience without being bribed or coerced. Its protections, which are guaranteed in the constitutions of 44 states [21], have taken on increased importance amidst an alarming rise in voter intimidation [23].

Modern election technology creates new challenges for ballot secrecy, because computerized voting systems are capable of generating ballot-by-ballot results, i.e., data that describes the selections on each individual ballot. Recently, some election officials have started to publish such data in the belief that it bolsters voter confidence, and several states have passed [22] or are considering [2] laws that mandate its publication. Many more localities have released ballot-level data in response to a wave of lawsuits [37] and Freedom of Information Act requests [24] following the 2020 presidential election.

To prevent voters from being associated with their ballots, ballot-level data needs to be anonymized, but deployed election technology has repeatedly failed to do this securely. In 2004, the first academic study of an electronic voting machine found that it used a broken random number generator to shuffle ballot-level data [34], and in the two decades since, researchers have identified similar flaws that could compromise voter privacy in a variety of other election systems (see Section 7). Despite this concerning pattern, there has been little effort to systematically address this class of vulnerabilities, perhaps because access to ballot-level data was historically limited to election insiders. Yet the current trend towards publishing ballot-level data has transformed the nature of the threat: now anyone can potentially exploit such flaws.

To understand how this threat affects contemporary voting technology, we examined ballot-level data from recent U.S. elections. Using only public information, we uncovered a serious privacy flaw in products from Dominion Voting Systems, the country's second-largest voting equipment maker. Due to a flaw in the ballot randomization algorithm used by Dominion precinct-based ballot scanners, anyone can unshuffle ballot-level data produced from affected machines and determine the order in which the ballots were cast. We named the vulnerability DVSorder ("*devious order*"), and it was assigned CVE-2022-48506. Unlike most prior ballot privacy flaws, DVSorder can be identified and exploited by the public—it does not require malicious actions by insiders, compromise of or access to voting machines, or participation by voters themselves—and exploitation cannot be detected by election officials. At the time of discovery, parts of 21 states and Canada used the affected systems [60], and at least 11 states had released vulnerable data that was available online [44, 51, 64].

We reported the vulnerability to Dominion, which created updated software in response. Unfortunately, such a patch can only protect voters in future elections, and vulnerable data that has already been published will remain exploitable.

Since the design, testing, and operation of election technology is a heavily regulated area of critical infrastructure, one might expect flaws like DVSorder to be caught and corrected quickly. Instead, the vulnerability persisted for years, across at least 11 versions of three Dominion products, which were repeatedly certified by multiple regulators at the federal and state levels. This points to gaps in the testing and certification processes used for election equipment, especially with respect to voter privacy. We discuss these issues in Section 8.

Evidence of other kinds of gaps emerged as we carried out disclosure, which provided a rare test case for the election sector's security information sharing and vulnerability management practices (Section 6). Neither Dominion nor the responsible federal agencies effectively notified the local jurisdictions that operate the equipment, leading such major municipalities as San Francisco to continue releasing vulnerable data well after we made our findings public. Once notified, some jurisdictions promptly patched or otherwise addressed the problem, but others have not. One state—Georgia—has announced, citing logistical difficulties, that it will defer mitigation until after the 2024 presidential election, leaving voters at risk more than two years after we disclosed the flaw [61]. These challenges underscore the need for further maturation of security processes throughout the election ecosystem.

Additional information about DVSorder, as well as a data sanitization tool and other recommended mitigations for election officials, can be found online at DVSorder.org.

2 Background

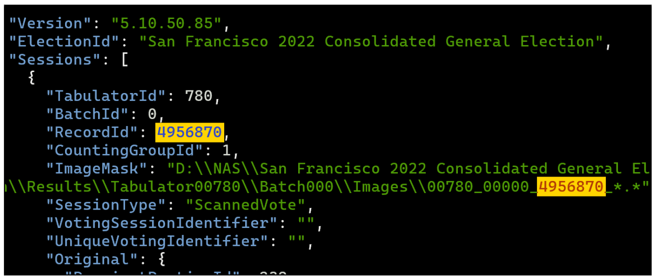
Many modern voting systems produce ballot-level data. In well implemented systems, this data is safe for public release. When systems do not include appropriate safeguards, however, such data can reveal how individuals voted.

2.1 Ballot-Level Data

Modern ballot scanners produce two main types of ballot-level data that are intended to be public facing. The first, *cast vote records* (CVRs), are the choices selected on an individual ballot along with metadata about the ballot. This metadata identifies the precinct and the tabulator that processed the ballot and includes an ID that uniquely identifies the ballot among those in the election. Dominion systems can export CVRs in CSV or JSON format, as shown in Figures 1a and 1b.

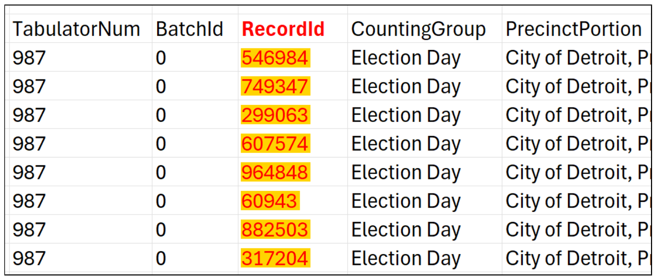
Scanners also produce a second form of ballot-level data, *ballot images*, which are digital scans of each ballot. These image files are typically associated with the same ID as the corresponding CVRs to allow cross comparison. Figure 1c shows an example of a set of ballot images from a Dominion system.

A growing number of jurisdictions publish one or both forms of ballot-level data, sometimes even making them available in interactive forms, such as the online tool in Figure 1d. Although these digital artifacts have limited value for verifying the integrity of an election outcome in the face of electronic tampering [6] (which would require inspection of the original physical ballots, e.g., in a risk-limiting audit), many public officials consider releasing them to be a beneficial "open government" practice that enhances public trust.



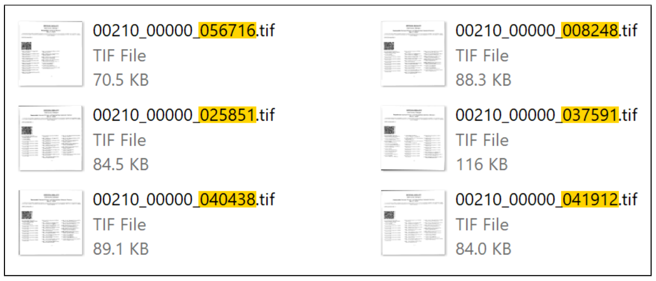
```
"Version": "5.10.50.85",
"ElectionId": "San Francisco 2022 Consolidated General Election",
"Sessions": [
  {
    "TabulatorId": 780,
    "BatchId": 0,
    "RecordId": 4956870,
    "CountingGroupId": 1,
    "ImageMask": "D:\\NAS\\San Francisco 2022 Consolidated General El
\\Results\\Tabulator00780\\Batch000\\Images\\00780_00000_4956870_*.*"
    "SessionType": "ScannedVote",
    "VotingSessionIdentifier": "",
    "UniqueVotingIdentifier": "",
    "Original": {
```

(a) JSON-format cast vote records



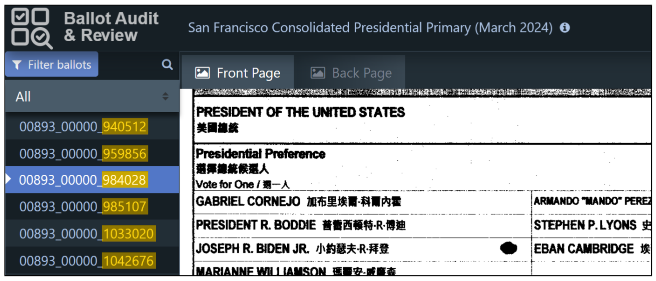
TabulatorNum	BatchId	RecordId	CountingGroup	PrecinctPortion
987	0	546984	Election Day	City of Detroit, P
987	0	749347	Election Day	City of Detroit, P
987	0	299063	Election Day	City of Detroit, P
987	0	607574	Election Day	City of Detroit, P
987	0	964848	Election Day	City of Detroit, P
987	0	60943	Election Day	City of Detroit, P
987	0	882503	Election Day	City of Detroit, P
987	0	317204	Election Day	City of Detroit, P

(b) CSV-format cast vote records



00210_00000_056716.tif TIF File 70.5 KB	00210_00000_008248.tif TIF File 88.3 KB
00210_00000_025851.tif TIF File 84.5 KB	00210_00000_037591.tif TIF File 116 KB
00210_00000_040438.tif TIF File 89.1 KB	00210_00000_041912.tif TIF File 84.0 KB

(c) Ballot image TIFF files


Ballot Audit & Review
San Francisco Consolidated Presidential Primary (March 2024)
Filter ballots
All
Front Page
Back Page

00893_00000_940512
00893_00000_959856
00893_00000_984028
00893_00000_985107
00893_00000_1033020
00893_00000_1042676

PRESIDENT OF THE UNITED STATES
美國總統

Presidential Preference
選擇總統候選人
Vote for One / 選一人

GABRIEL CORNEJO 加布里埃爾·科爾內霍	ARMANDO "MANDO" PEREZ
PRESIDENT R. BODDIE 普雷西頓特·R·博迪	STEPHEN P. LYONS
JOSEPH R. BIDEN JR. 小約瑟夫·R·拜登 ●	EBAN CAMBRIDGE
MARIANNE WILLIAMSON 瑪麗安·威廉森	

(d) Ballot images in online viewer

Figure 1: Ballot-level data from Dominion systems. Jurisdictions increasingly publish ballot-by-ballot election data in the form of cast vote records (descriptions of the votes recorded from each ballot) or ballot images (scans of each ballot). The most common formats of this data for Dominion systems are shown here. Each associates votes with the ballot's record ID (*highlighted*), the source of the DVSorder vulnerability.

2.2 Dominion Voting Systems

Dominion has been the subject of prominent conspiracy theories concerning fraud in the 2020 U.S. presidential election that have no evidentiary basis [1, 27]. However, researchers have recently shown that some of the company's products really do suffer from serious security problems [18, 27, 29].

Dominion produces a line of equipment it calls Democracy Suite, which includes election management system (EMS) software and both batch-fed and hand-fed ballot scanners, among other components. The company offers three models of hand-fed scanners: the ImageCast Precinct (ICP), ImageCast Precinct 2 (ICP2), and ImageCast Evolution (ICE). These machines are typically used as *precinct tabulators*: that is, they are deployed at polling places, and voters feed in their own marked ballots, which are counted and deposited into an integrated ballot box. When voting is complete, workers use removable memory cards to transfer ballot-level data to a central EMS computer, which aggregates results from tabulators throughout the jurisdiction to generate the official totals.

In order to track data about voted ballots, Democracy Suite uses three fields: *tabulator ID*, *batch ID*, and *record ID*.[1] The combination of these IDs uniquely identifies a ballot within an election, and they are used together to label CVRs and ballot image files, as seen in Figure 1.

Dominion tabulators assign a numeric record ID to every ballot sheet as it is scanned. Typically, these consist of six digits, which are chosen to be unique within a batch. When ballots run across multiple sheets of paper, higher-order digits are added to indicate the sheet number, so that records of a given sheet sort together. For present purposes, we can ignore the sheet number and consider the record IDs mod 1,000,000.

To protect privacy, the record IDs must not reveal the order in which the ballots were cast. Otherwise, for instance, anyone in the polling place could note that someone was the *n*th voter to use the scanner and later identify their ballot in the data. (We discuss variations of this threat, including attacks that can be performed retrospectively, in Section 4.) To address this, Dominion's hand-fed scanners assign the six low-order digits using a random number generator,[2] which the company asserted in contracts with state governments "ensur[es] the ballot images are de-coupled from voter order" [54, p. 187].

3 The DVSorder Vulnerability

... The moral of this story is that *random numbers should not be generated with a method chosen at random.*

— Donald E. Knuth, 1969 [33, p. 5]

We determined that Dominion ICP, ICP2, and ICE scanners draw their ballot record IDs sequentially from a fixed cycle of possible values that is shared across all machines of the same model. The only unpredictable element is the starting

[1]For a precinct tabulator, all voted ballots are usually assigned batch ID 0.

[2]In contrast, Dominion's batch-fed scanners, which are typically used to count mail-in ballots, assign record IDs sequentially. This is safe under the assumption that mail-in ballots will be adequately shuffled before scanning.

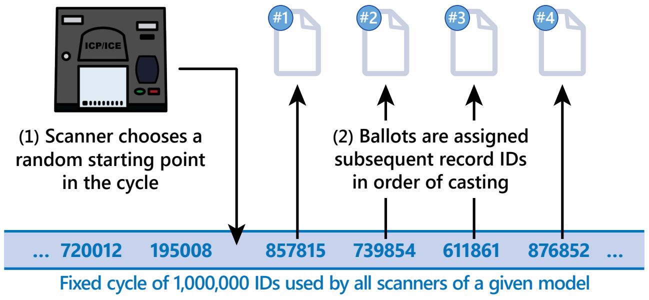

ICP/ICE
#1
#2
#3
#4
(1) Scanner chooses a random starting point in the cycle
(2) Ballots are assigned subsequent record IDs in order of casting
... 720012 195008 857815 739854 611861 876852 ...
Fixed cycle of 1,000,000 IDs used by all scanners of a given model

Figure 2: Flawed ballot ID generation. Dominion scanners assign record IDs to ballots as they are cast by following a fixed cycle of one million values. The only unpredictable feature is the starting point.

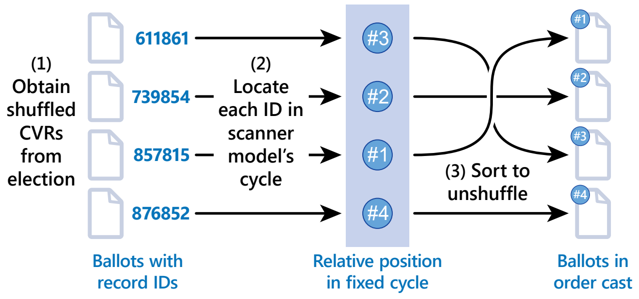

(1) Obtain shuffled CVRs from election
611861
739854
857815
876852
(2) Locate each ID in scanner model's cycle
#3
#2
#1
#4
(3) Sort to unshuffle
#1
#2
#3
#4
Ballots with record IDs
Relative position in fixed cycle
Ballots in order cast

Figure 3: Ballot order can be reconstructed from public data. Record IDs from ballot images or CVRs can be matched to their positions in the fixed cycle. The ballots in a batch will map to consecutive positions, which correspond to the order in which they were cast.

point in the cycle, which each scanner chooses at the start of the election (see Figure 2). Using only publicly available information, we managed to reconstruct the complete cycles for all three scanner models, which turn out to be the obfuscated output of a linear congruential generator. With knowledge of the cycles, anyone can perfectly determine the order the ballots were cast from their record IDs, as shown in Figure 3.

The remainder of this section details the process we used to reconstruct the cycles. It is instructive, because it reveals how readily Dominion's privacy protections can be defeated.

3.1 Reversing the ICP Generator

Our starting point was an unusual public dataset that contained a copy of the memory cards from a jurisdiction's ICP scanners. Raw data from these cards is not typically published, since it is unsafe to do so, but in 2021 card data from a single county was openly distributed by a third party, and we obtained a copy.

When the Dominion system exports ballot-level data for public consumption, it presents the ballots from each batch in a shuffled order to conceal the order of casting, but we found that the raw data from the memory cards contained CVRs in voted order. We determined this by comparing the sequence of ballot styles from the CVRs to the sequence of styles that were cast during the election, as recorded chronologically in log files stored on the cards. These matched exactly.

Using the data from one card, we extracted approximately 500 record IDs in voted order. This relatively small sequence was sufficient for us to reverse the full cycle of one million IDs shared across all ICP tabulators. Of course, any election worker with access to the equipment could obtain an ordered ID sequence even more easily by simply scanning a deck of numbered ballots and inspecting the resulting ballot images.

However an initial sample of the generator's output is obtained, the function is weak enough that it can be reversed using only pen and paper through the process described below.

Identifying the Sequence as Nonrandom Closely inspect the following sequence of record IDs (in column-major order):

850934	262494	959507	765067
437941	043481	331519	147059
614932	900792	516506	804866
079948	225788	173513	721853
486435	597795	388000	696860

There are obvious patterns in this sequence. For instance, the digits in the ten-thousands place (*blue*) follow a cycle with period 10. Inspecting a longer sequence would show that the digits in the hundreds place (*red*) follow a cycle with period 100. These cycles, combined with the output domain, suggested that a linear congruential generator (LCG) with modulus 1,000,000 might be in use.

An LCG is defined by the recurrence relation [46]:

$$x_{n+1} = a \cdot x_n + b \mod m.$$

To use an LCG in such an application would be a mistake. Researchers identified patterns in LCG output as early as 1968 [38] and used these patterns to "crack" such generators in cryptographic contexts as early as 1977 [50]. It has been known since 1982 that the outputs of an LCG with unknown parameters can be efficiently predicted with knowledge of just a few consecutive elements from the sequence [46].

If the ICP used a simple LCG with modulus 1,000,000, we would expect the least-significant digit to exhibit a cycle with period at most 10, the second-least-significant digit to exhibit a cycle with period at most 100, and so on. This is because the last decimal digit of the term x_{n+1} may only be affected by the last decimal digit in a, x_n, and b when m is a multiple of 10. Since a and b are constants, the last digit of x_n is the only piece of state affecting the last digit of x_{n+1}. There are only ten possible values for the state, which means the last digit can have a period of at most ten. This logic can be generalized to the other digits in the generator's output.

Identifying the First Obfuscation That cycles of lengths 10 and 100 appear, but in different digit positions than expected, suggests that the order of the digits is being permuted to obfuscate the generator's output. To test this theory, we can set aside the other digits and restore the two digits with known periods to their suspected positions in the underlying state.

We now see the following consecutive terms:

95	46	55	06
93	44	53	03
91	70	51	80
97	72	57	82
48	79	08	89

These terms are not consistent with an LCG, since the numbers remain close together for several sequential terms then jump significantly in a repeated pattern. This suggests that another digit-level obfuscation may be present.

Identifying the Second Obfuscation We intuit that the second obfuscation may be a substitution cipher. To determine the rule of substitution, we observe that the digit now in the tens place changes once for every three or four changes in the ones place. This pattern matches what we would expect when counting by threes. We hypothesize that each term of underlying state differs from the state of the previous term by three.

Next, we observe the four sequential terms 95, 93, 91, 97. Note that the digit in the tens place does not change. This is true when counting by threes only if the numbers in the ones place are $0, 3, 6$, and 9. This suggests that $5, 3, 1$ and 7 correspond to $0, 3, 6$, and 9 respectively. We can extend this pattern further and conclude that the sequence's next term must have a 2 in the ones place, followed by a 5, and so on. This allows us to derive a substitution rule covering each digit.

Completing the Reversal Applying the substitution and permutation rules to the original IDs yields the following terms for the underlying state, where the digits in parenthesis are in uncertain order:

(2138) 70	(4478) 85	(7719) 00	(9059) 15
(8986) 73	(1326) 88	(3667) 03	(6907) 18
(5834) 76	(7174) 91	(0515) 06	(2855) 21
(1782) 79	(4022) 94	(6363) 09	(9603) 24
(8530) 82	(0970) 97	(3211) 12	(5551) 27

This is consistent with our intuition that the underlying state in the two least-significant digits is counting by threes, and confirms that we are on the right track. We now must determine which digit is the third-least-significant of the underlying state. We observe that the least-significant digit of the original record IDs, after applying the substitution rule, switches from pattern $8, 6, 4, 2, 0$ to pattern $9, 7, 5, 3, 1$ when the two least significant digits of the state roll over from 97 to 00; in other words, it happens precisely when we would "carry the one." This suggests that the least-significant digit of the ID is the third-least-significant digit of the underlying state, and that our state increments by $b \equiv 803 \bmod 1000$ at each iteration.

We now know the substitution rule and the permutation of the three least-significant digits of the underlying state, as well as the three least-significant digits of the incrementing constant b. We also know that the multiplicative constant satisfies $a \equiv 1 \bmod 1000$, and we hypothesize that it is simply 1.

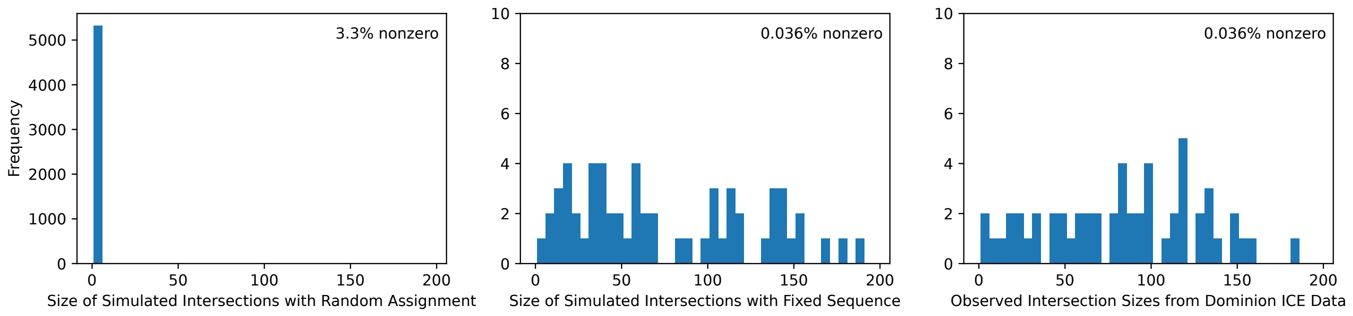


Figure 4: Statistical detection of the flawed RNG. Sets of record IDs chosen by different scanners sometimes overlap. For an election with 571 scanners, we show representative simulations of the intersection sizes obtained if IDs are assigned randomly (*left*) or by picking random starting places in a fixed cycle (*middle*). The latter closely matches the observed distribution in public data from ICE scanners in San Francisco (*right*).

To determine the permutation and incrementation constant of the rest of the state, we refer to the following three terms:

$$(583)\ 476 \qquad (178)\ 279 \qquad (853)\ 082$$

To identify which digit is the next-least-significant in the underlying state, we observe the change in value for each digit between sequential terms. We observe that the first digit increases under the modulus by 6, then by 7. The second digit increases by 9, then by 8. The third digit increases by 5, and again by 5. We note that we would expect to "carry a one" from the less-significant digits whose order we know for both incrementations, so the change in the digit's value should be consistent between the three terms. Thus, the third digit of the unordered tuple must be the third digit of the state, and we must be incrementing by 4803 at each iteration.

To determine the order of the two final digits, we observe that we would expect to carry a one from the digit we just placed on the second incrementation of the state, but not on the first. This is consistent with the change in value for the first digit of our unordered pair, but not of the second. Thus, we conclude that the two remaining digits are swapped.

The ICP Generator The reasoning above lets us derive the complete record ID generation algorithm used by the ICP. Starting from an unknown initial seed, for each subsequent ballot, the ICP iterates the state of the following LCG:

$$x_{n+1} = x_n + 864803 \mod 1000000.$$

It then transforms the LCG's output with a substitution cipher

$$[0,1,2,3,4,5,6,7,8,9] \longrightarrow [5,0,8,3,2,6,1,9,4,7]$$

and permutes the digits such that 123456 becomes 261534. Thus obfuscated, the value becomes the ballot's record ID. This generator has a period of one million and seems to have been used with identical constants across all ICP scanners.

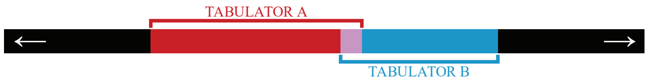
In effect, all ICPs followed the fixed cycle of one million record IDs produced by this function. Each chose an apparently random starting point, then assigned sequential IDs from the cycle to ballots in the order they were cast.

3.2 Reversing the ICE and ICP2 Generator

Dominion's other precinct-based scanners, the ICE and ICP2, do not follow the record ID cycle derived above. We initially had no data known to be in voted order for these models, so we applied a different strategy to determine the generator.

Record ID Intersections First, we determined that these other scanner models use a similar ID generation algorithm to the ICP by comparing the record IDs used in different batches of ballots in public data. If the machines drew IDs uniformly at random, we would expect the magnitude of the intersections between batches to be governed by the birthday paradox. This means we would expect a relatively large number of non-zero intersections with a small number of shared IDs in each. If the machines used a fixed cycle of IDs, however, we would expect non-zero intersections that were rarer but larger. Public CVR data matched the fixed-cycle hypothesis (see Figure 4).

The specific intersections between sets of IDs from different tabulators place constraints on the cycle. When two batches intersect, the IDs in the intersection must be consecutive, and so must the IDs in the union, as illustrated below:



By sourcing public data from many elections and analyzing these intersections, we could potentially reconstruct the complete cycle. The problem of reconstructing a full sequence based on many unordered subsequences is closely related to genome reconstruction algorithms in the field of computational biology (see, e.g., [45]) and would be interesting to pursue, but we applied a simpler method, described below.

Determining Parameters Since the ICE appeared to follow a fixed cycle like the ICP, we conjectured that their RNGs used the same algorithm with different parameters (i.e., permutation and substitution rules and constants a and b). To constrain the space of possibilities, we analyzed public data from San Francisco and identified batches from two ICEs that shared four record IDs in common. By the analysis above, these IDs must appear consecutively (in some order) in the ICE cycle. We exhaustively tested for parameters that would cause the generator to satisfy this condition, yielding two equivalent generators producing such a cycle and two more that followed the same cycle in the opposite direction. Appendix A explains why such equivalent generators exist.

One of these generators used parameters nearly identical to the ICP's. The only difference was the permutation, which, in the new generator, transformed 123456 to 342615. The constants for the LCG and substitution cipher were unchanged.

Such similarity suggested that this generator followed the cycle in the correct direction. For confirmation, we used scanner log files from an ICE published by San Francisco. The logs allowed us to identify the indices at which overvoted ballots were cast, and we verified that the ballot images the generator placed in these positions were indeed overvoted.

Finally, we located public data produced by ICP2 tabulators and confirmed that the record IDs in each batch were consecutive under the ICE's cycle. This indicated that the same generator was shared by both models.

Code To aid experimenters, the following compact Python program generates the full cycles for all three scanner models:

```
generate_cycle = lambda b,s,p:[sum(s[b*n//
   10**(6-p[i])%10]*10**(5-i) for i in range(6))
   for n in range(10**6)]
b,s = 864803, [5,0,8,3,2,6,1,9,4,7]
icp = generate_cycle(b, s, [2,6,1,5,3,4])
ice = generate_cycle(b, s, [3,4,2,6,1,5]); icp2=ice
```

3.3 Validation and Scope

To further confirm that the generators we derived were correct, and to determine the scope of the vulnerability, we attempted to locate and analyze public ballot-level data from as many jurisdictions as possible that used the affected scanners.

It is straightforward to check if a dataset uses the generators. If it does, we expect the record IDs within each batch of ballots to be consecutive (or nearly so) under one of the cycles. This was the case for every ICP, ICP2, and ICE dataset we encountered. Note that the machines skip ID 0, and on rare occasions one or a few IDs may be missing from otherwise-consecutive batches, which likely correspond to provisional ballots that were scanned but removed from the tally. We created a test tool (available at DVSorder.org) that imports CVRs or ballot images and reports whether they are vulnerable.

Ultimately, we found vulnerable data published online from parts of 11 states [44, 51, 64]: Alaska, Arizona, California, Georgia, Iowa, Michigan, New Jersey, New Mexico, Ohio, Tennessee, and Wisconsin. Twenty-one states (shown in Figure 5), Puerto Rico, and Canada used the vulnerable scanner models in at least some localities as of November 2022 [60].

The datasets we confirmed to be vulnerable were produced in jurisdictions running 11 versions of Democracy Suite: 5.2, 5.4-NM, 5.5, 5.5-A, 5.5-B, 5.5-C, 5.5-D, 5.7-A, 5.10, 5.10A, and 5.15. As of June 2024, we have yet to encounter any dataset with record IDs produced by ICP, ICP2, or ICE scanners that is not affected by the vulnerability.

4 Risks and Exploitation

With knowledge of the fixed record ID cycles—or the ability to derive them from public data—one can trivially unshuffle CVRs or ballot images produced by the affected tabulators. This does not require any malicious action by election insiders, access to election equipment, advanced technical capabilities, or participation by voters. Since the vulnerability can be exploited passively using only publicly available information, exploitation carries little potential risk for attackers.

Given the ability to unshuffle ballot-level data, there are several ways that an attacker can learn how specific people voted. Some require making deliberate observations during the election, while others can be performed retrospectively (perhaps years later) if certain side information is available.

Public Counters In most jurisdictions, precinct-based scanners continuously display a public counter that shows how many ballots they have recorded thus far during the election (see Figure 6a). The counter value is equivalent to the index in the unshuffled ballot sequence of the most recently cast ballot. Anyone near the machine—such as poll workers, other voters, or, depending on the layout of the polling place, public observers—can note the counter value after a victim votes in order to later identify that person's ballot. Of the many threats this facilitates, one noteworthy example is spousal coercion. Suppose a man uses a scanner immediately after his wife. The value of the counter before he scans his ballot tells the man the index of his wife's ballot, which lets him learn how she voted if vulnerable ballot-level data becomes public.

Election Observers Poll workers or election observers could also keep a complete ordered record of who uses the scanner, which would allow them to deanonymize all ballots cast at the polling location or to target public figures or specific people they know. In many states, the right of partisan poll watchers to observe inside polling places and to note the identity of voters is enshrined in law (see, e.g., [39, 48]). Organized

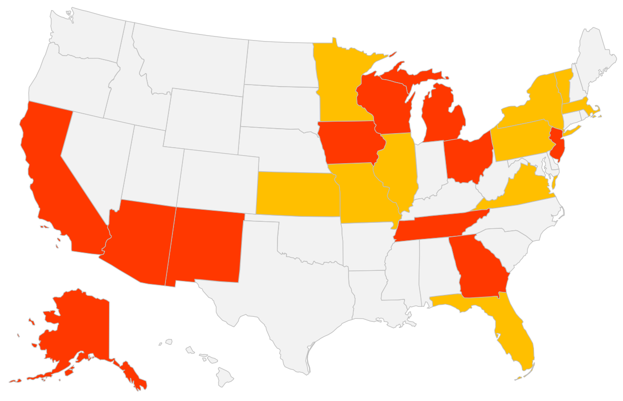
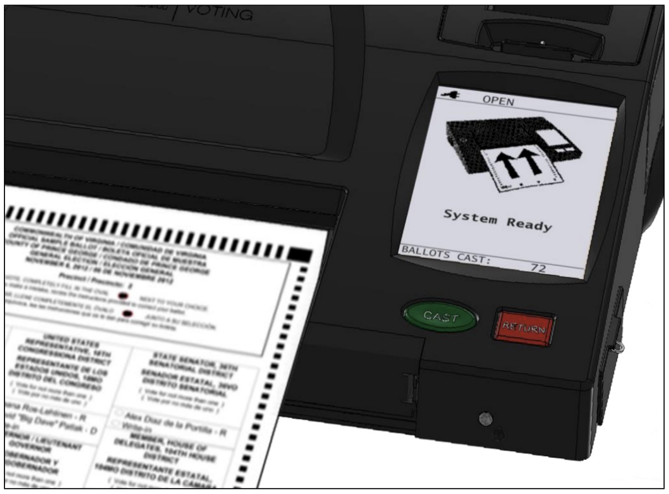
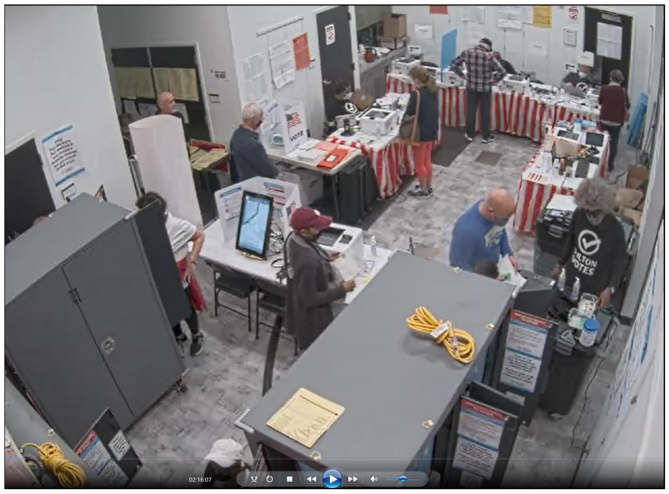


Figure 5: States vulnerable to DVSorder. Parts of 21 U.S. states (*red/amber*) used the vulnerable Dominion scanners as of Nov. 2022, as did Puerto Rico and some Canadian localities. We found vulnerable data published online from recent elections in 11 states (*red*).



(a) Public counter ("ballots cast") displayed by ICP scanner



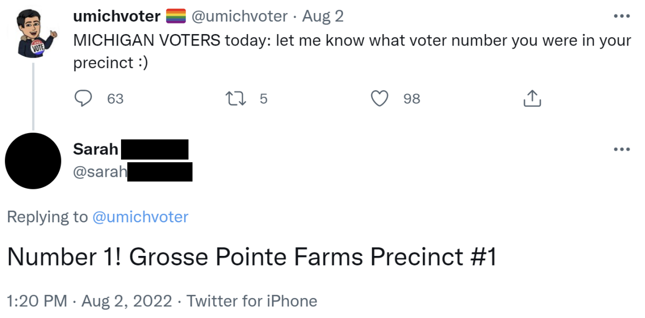
(b) Video from polling place showing voter using ICP scanner (blue shirt)

Figure 6: Exploiting voted order to deanonymize ballots. An attacker can identify who cast which ballot by making observations during the election or, after the fact, using side information. For example: (*a*) Other voters or poll workers can note the scanner's public counter when victims vote. (*b*) Some localities record video of everyone who uses the scanner, such as this footage from Georgia obtained via a public records request.

political machines could exploit these rights to deanonymize ballots as part of efforts to buy votes or intimidate voters.

Poll Books When a ballot is issued, the voter's identity is typically recorded in a digital or paper-based log called a "poll book" to ensure they do not vote again later. Depending on the design of these poll books, they sometimes record the time at which individual voters received their ballots. This provides a strong proxy to the order in which voters cast their ballots and could reveal many voters' choices, especially during slow parts of the day when voters are widely spaced in time.

Scanner Log Files Some localities publish log files from their ballot scanners. In the Dominion system, these logs by themselves pose little risk to privacy, but they can be combined with DVSorder to determine the exact time that each ballot was cast (subject to the accuracy of the scanner's clock). To exploit this, attackers could infer the time a particular individual used the scanner based on the time they leave the polling place. This could be gathered by watching from outside the building during the election or even after the fact from mobile geolocation data. As another example, journalists sometimes film or photograph candidates and other public figures as they vote. Such media is typically timestamped, which, combined with scanner log files, could be used to deanonymize those individuals' ballots even long after the election.



umichvoter @umichvoter · Aug 2
MICHIGAN VOTERS today: let me know what voter number you were in your precinct :)

63 5 98

Sarah [redacted]
@sarah[redacted]

Replying to @umichvoter

Number 1! Grosse Pointe Farms Precinct #1

1:20 PM · Aug 2, 2022 · Twitter for iPhone

Figure 7: Some voters publicly disclose their voting locations and voter numbers, as in this thread from Twitter (which we have redacted). This is sufficient information to identify the unsuspecting voter's ballot if vulnerable CVRs or ballot image files are available.

Disclosures by Voters Some voters publicly reveal their polling places and voter numbers, as in the Twitter thread shown in Figure 7. As long as the voter has accurately stated their position in the ballot sequence, this lets anyone determine their exact set of votes from vulnerable ballot-level data. For most voters who reveal such information, this is an unexpected violation of privacy, but voters could also disclose this information deliberately (in public or private) in response to vote buying or coercion. Sharing one's voter number before the ballot-level data becomes public provides a mechanism for a voter to "claim" a specific ballot, thereby allowing them to prove how they voted to third parties with some confidence.

Surveillance Footage Some localities record surveillance video inside polling places. For instance, Figure 6b is from a day-long video filmed by a county in Georgia that was obtained by others via a public records request prior to our work. The footage shows everyone who used the scanner, in order. If a jurisdiction where such footage is a public record also releases vulnerable CVRs or ballot images, anyone could unshuffle the data and associate each ballot with *video of the voter casting it*. Attackers could then identify people in the video whom they recognized or use commercially available facial recognition systems to identify voters en masse.

5 Mitigation and Prevention

Correcting DVSorder is simple in theory: modify the scanners to assign record IDs securely at random. This would ensure that ballot-level data cannot be unshuffled and prevent all attack scenarios we have described. In practice, the solution is more complicated: only Dominion can issue new scanner firmware, deploying it requires action by many states and localities, and third-parties cannot fully validate any purported fix without access to Dominion's proprietary source code.

5.1 Dominion's Software Update

Following our disclosure, Dominion did issue a new software version, Democracy Suite 5.17, that, according to certification documents, incorporates an "[i]mproved pseudo random number algorithm" [49, §1.1.2]. The software achieved federal certification in March 2023, clearing the way for use in most states [58]. However, 5.17 includes a host of other changes, making it a major upgrade rather than a focused patch; this adds to the cost and complexity of deployment. As discussed in Section 6, many jurisdictions continue to use older, vulnerable versions more than a year after 5.17 became available.

Dominion has not shared any details about the new PRNG. We do not have access to the new software, and we have been unable to locate any ballot-level data produced by it, so we cannot confirm whether 5.17 adequately addresses DVSorder. Considering the incomplete and potentially misleading guidance that Dominion initially gave its customers following our disclosure (Section 6.3), some skepticism may be warranted.

5.2 Sanitizing Ballot-Level Data

Until corrective scanner firmware is deployed, we recommend that election officials sanitize ballot-level data prior to making it public. This can be done with one of two strategies: either completely remove the record IDs and randomly shuffle the records, or securely rerandomize the record IDs and sort the records. If correctly performed, these approaches will prevent DVSorder from being exploited with the published data.

Remove Record IDs and Randomly Shuffle This strategy is comparatively easy for jurisdictions to implement, so long as they only publish CVRs. (It does not work for ballot images, since removing the record IDs from their filenames causes many images to have the same name.) For CSV-format CVRs, officials should remove the "RecordId" and "CvrNumber" columns. For JSON-format CVRs, officials should remove the "RecordId" and "ImageMask" fields (see Figure 1a).

On its own, deleting these fields is insufficient to reliably guard against exploitation; officials must also randomly shuffle the records. This is because Dominion CVR data sometimes contains batches of ballots that are sorted by record ID. For these batches, an attacker can restore the deleted record IDs using the following attack. Typically, a batch of ballots will contain a mix of ballot *styles* corresponding to populations who vote in different sets of contests. An adversary can learn the sequence of ballot styles that were voted, in order, either from the scanner log files (if available) or by stationing an observer in the polling place. For a large batch of ballots, only one starting part in the scanner's cycle will result in record IDs that map the observed sequence of ballot styles to the sequence contained in the CVRs. Using a brute-force search over the 1,000,000 cycle starting positions, the attacker can infer the deleted record IDs and unshuffle the ballots. To prevent this, it is essential to randomly shuffle the CVRs.

A downside of removing the record IDs is that it can disrupt some desirable uses of public election data. For instance, without record IDs, it is impossible to reliably associate ballot images with the corresponding CVRs to confirm that the choices were correctly interpreted by the scanner. Likewise, if a jurisdiction releases multiple versions of a dataset, it is impossible to precisely determine which records have changed.

Rerandomize Record IDs and Sort To preserve the intended utility of the record ID field while preventing the attack, officials can replace the IDs with new random identifiers that are consistent across all public datasets from the election. This maintains the association between sanitized ballot images and their corresponding sanitized CVRs, and it facilitates review of incrementally published data. Sorting the records by their new IDs will also remove any residual information in their order. Special consideration is required for archives of ballot images, which should be recreated to ensure that the original order of the data is destroyed.

To help officials apply this strategy, we created a free and open-source sanitizer tool, available at DVSorder.org. Our tool supports all common forms of Dominion ballot-level data. To rerandomize the record IDs, it chooses a random key and replaces each ID with a truncated form of its AES ciphertext. If there are multiple datasets from an election, they can be processed with the same key to ensure that records about any given ballot will always be assigned the same ID. Although this might seem like an ideal application for format-preserving encryption [5], we instead designed the tool to change the format of the IDs so as to make sanitized and unsanitized data easily distinguishable and reduce the risk of accidental release.

Neither of these mitigation strategies is perfect. Even if CVRs and ballot images are fully sanitized before publication, election workers will continue to have access to the unsanitized data and so retain the ability to deanonymize ballots. Unsanitized data could also potentially be stolen in a data breach and used to compromise privacy.

5.3 Protections for Raw Scanner Data

To fully address the risk of ballot-ordering attacks, Dominion needs not only to change its record ID generation algorithm but also to correct the fact that its scanners store ballot images [29] and CVRs (Section 3.1) in voted order on their memory cards. To safeguard privacy against attackers who have access to such raw scanner data, Dominion and other voting system vendors should shuffle ballots in-memory using an appropriate randomized data structure (e.g., [41]).

6 Disclosure

Our experience disclosing DVSorder is instructive, as it exposed challenges to security information sharing and widespread gaps in vulnerability management practices within the elections sector. The following narrative highlights opportunities to improve coordination and remediation processes so as to better ensure that future vulnerabilities in election equipment are corrected before they can affect voters.

6.1 Ethical Considerations

Disclosure of a serious vulnerability in an election system requires a careful balance of the equities involved. One of the most consequential decisions we faced was when and how to make information about the vulnerability public, particularly given that we discovered the flaw only a few months prior to the November 2022 U.S. general election.

In weighing our options, we saw our foremost obligation as protecting voters. Completely protecting against DVSorder would require officials to patch their election equipment, but patching election systems often takes years, and ballot-level data released in the meantime would remain vulnerable to deanonymization forever. There was also a risk that malicious parties might already have discovered the flaw, since we had been able to do so using only public information. In light of these considerations, we judged that the best way to protect voters would be to disclose the vulnerability to state and local governments as soon as possible, before further data from affected systems would be released. Arming election officials with information about DVSorder, even in the absence of a patch, would allow them to prevent public exploitation by choosing to release less data or by appropriately sanitizing it.

We quickly realized that there was no viable route to notifying the responsible officials without effectively making the vulnerability public. U.S. election administration is decentralized, and the vulnerable equipment is operated by many hundreds of local municipal governments. None of our potential partners—federal agencies and Dominion—appeared to be prepared to privately inform these jurisdictions about the flaw. Even if they were, sanitizing ballot-level data would involve publicly visible changes to datasets that were under intense scrutiny from election skeptics [24], so any such change would require public justification. We concluded that public disclosure prior to the November election would be necessary.

Another central consideration was the effect of our disclosure on public trust. Many prominent election conspiracy theories concerned Dominion, and publishing information about a real vulnerability in the company's products could fan these flames. Yet withholding true vulnerability information would also be harmful to trust in elections. If we had reported the flaw to Dominion and government officials but collaborated to conceal it from the public, that might seem to validate claims of a conspiracy. After considering these risks, we concluded that transparency about the problem had the least potential for harm and was the best course of action.

6.2 Disclosure Process

We discovered the vulnerability in early July 2022. Within days, we contacted the Cybersecurity & Infrastructure Security Agency (CISA), the arm of the Department of Homeland Security responsible for election infrastructure [17]. CISA operates a coordinated vulnerability disclosure (CVD) program that assisted researchers with two earlier election security disclosures by interfacing with the vendors and providing guidance to election officials [18, 53]. We hoped that CISA's CVD program would be an effective way to notify all responsible parties about the problem ahead of the November election. However, on July 22, CISA informed us that it declined to coordinate disclosure for DVSorder and advised us to work directly with Dominion. This caused a lengthy delay, as, in light of Dominion's recent high-profile defamation lawsuits, we sought a careful legal review before approaching them.[3]

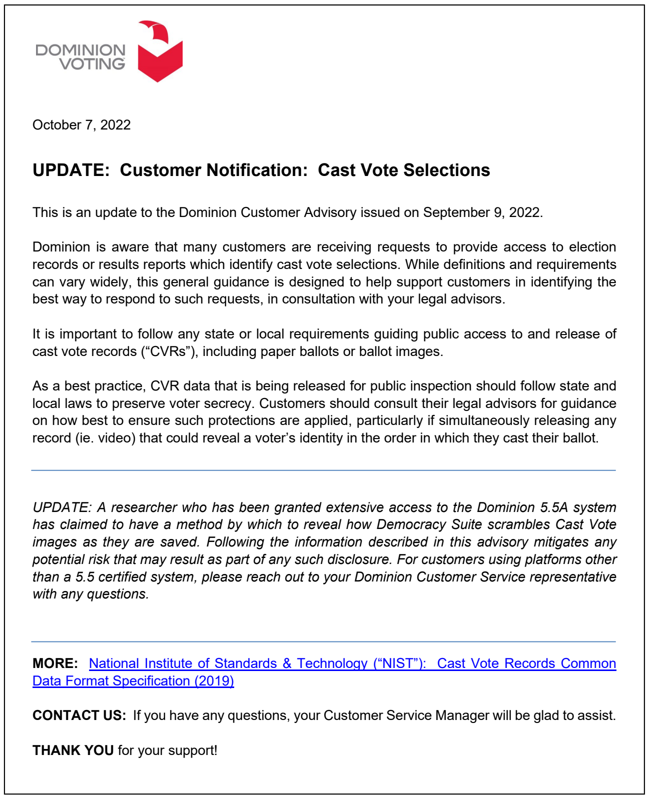
We began disclosure to Dominion on August 22 (day zero) with a call to the company's CEO, followed the next day with a letter (available at DVSorder.org) that detailed the flaw and offered our assistance in understanding and mitigating it. Given the short time before the November election, we told Dominion that we intended to make the information public as soon as 30 days later. The company acknowledged receipt of our letter on August 29 (+7 days) but did not further communicate with us about the issue, and we determined that we would have to notify election officials without their help.

On September 2 (+11 days), we sent formal notifications to CISA and to the U.S. Election Assistance Commission (EAC), the federal agency that oversees testing and certification of election equipment [15]. Dominion had not yet informed either agency about the flaw. We requested their assistance in communicating our disclosure to localities that operated affected equipment. EAC informed us that it did not have a comprehensive list of who used the scanners, as not all states require EAC certification for their equipment. CISA initially offered to put us in contact with state-level officials from affected states, but, on September 19 (+28 days), provided only a link to a public contact page on the National Association of State Election Directors website [42], adding further delay.

The lack of infrastructure for coordinated disclosure made identifying and contacting every affected locality nearly impossible, so we directed our notifications to the officials at the state level who were best positioned to communicate the problem and its mitigations to local officials. On October 10 (+49 days), 29 days before the November election, we sent notifications to the state election directors of the 21 states where at least some localities used affected machines according to data from the nonprofit organization Verified Voting [60].[4]

[3] Although Dominion maintains a vulnerability disclosure policy that offers a legal safe harbor for good-faith research, the company reserves the right "to change any aspect of our coordinated disclosure process at any time without notice, and to make exceptions to it on a case-by-case basis" [19].

[4] Due to an oversight, we neglected to notify authorities in Puerto Rico or Canada. This underscores the difficulty of conducting a disclosure of this complexity without the assistance of the vendor or a coordinating authority.


DOMINION VOTING

October 7, 2022

UPDATE: Customer Notification: Cast Vote Selections

This is an update to the Dominion Customer Advisory issued on September 9, 2022.

Dominion is aware that many customers are receiving requests to provide access to election records or results reports which identify cast vote selections. While definitions and requirements can vary widely, this general guidance is designed to help support customers in identifying the best way to respond to such requests, in consultation with your legal advisors.

It is important to follow any state or local requirements guiding public access to and release of cast vote records ("CVRs"), including paper ballots or ballot images.

As a best practice, CVR data that is being released for public inspection should follow state and local laws to preserve voter secrecy. Customers should consult their legal advisors for guidance on how best to ensure such protections are applied, particularly if simultaneously releasing any record (ie. video) that could reveal a voter's identity in the order in which they cast their ballot.

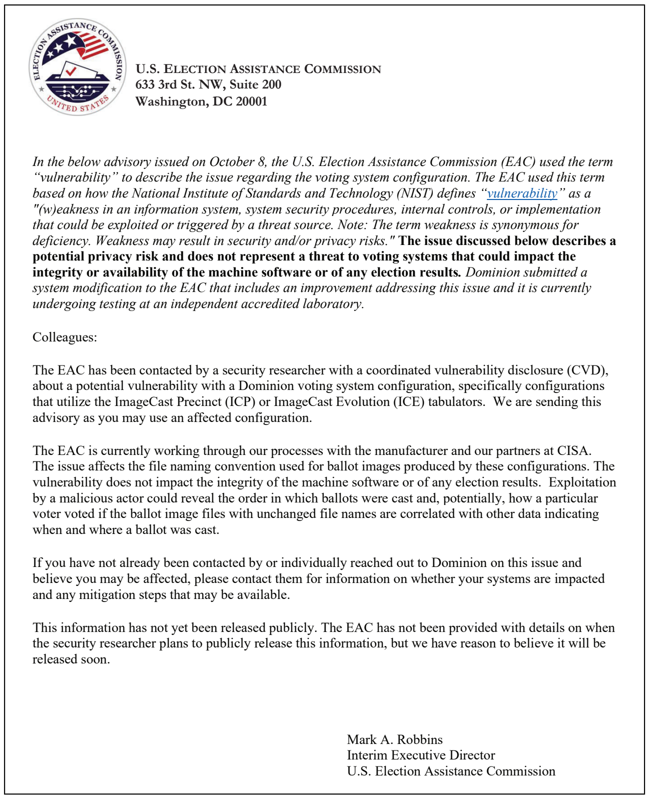
UPDATE: A researcher who has been granted extensive access to the Dominion 5.5A system has claimed to have a method by which to reveal how Democracy Suite scrambles Cast Vote images as they are saved. Following the information described in this advisory mitigates any potential risk that may result as part of any such disclosure. For customers using platforms other than a 5.5 certified system, please reach out to your Dominion Customer Service representative with any questions.

MORE: National Institute of Standards & Technology ("NIST"): Cast Vote Records Common Data Format Specification (2019)

CONTACT US: If you have any questions, your Customer Service Manager will be glad to assist.

THANK YOU for your support!

(a) Dominion Notification (Oct. 7, 2022)


U.S. ELECTION ASSISTANCE COMMISSION
633 3rd St. NW, Suite 200
Washington, DC 20001

In the below advisory issued on October 8, the U.S. Election Assistance Commission (EAC) used the term "vulnerability" to describe the issue regarding the voting system configuration. The EAC used this term based on how the National Institute of Standards and Technology (NIST) defines "vulnerability" as a "(w)eakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. Note: The term weakness is synonymous for deficiency. Weakness may result in security and/or privacy risks." **The issue discussed below describes a potential privacy risk and does not represent a threat to voting systems that could impact the integrity or availability of the machine software or of any election results.** *Dominion submitted a system modification to the EAC that includes an improvement addressing this issue and it is currently undergoing testing at an independent accredited laboratory.*

Colleagues:

The EAC has been contacted by a security researcher with a coordinated vulnerability disclosure (CVD), about a potential vulnerability with a Dominion voting system configuration, specifically configurations that utilize the ImageCast Precinct (ICP) or ImageCast Evolution (ICE) tabulators. We are sending this advisory as you may use an affected configuration.

The EAC is currently working through our processes with the manufacturer and our partners at CISA. The issue affects the file naming convention used for ballot images produced by these configurations. The vulnerability does not impact the integrity of the machine software or of any election results. Exploitation by a malicious actor could reveal the order in which ballots were cast and, potentially, how a particular voter voted if the ballot image files with unchanged file names are correlated with other data indicating when and where a ballot was cast.

If you have not already been contacted by or individually reached out to Dominion on this issue and believe you may be affected, please contact them for information on whether your systems are impacted and any mitigation steps that may be available.

This information has not yet been released publicly. The EAC has not been provided with details on when the security researcher plans to publicly release this information, but we have reason to believe it will be released soon.

Mark A. Robbins
Interim Executive Director
U.S. Election Assistance Commission

(b) EAC Advisory (Oct. 8, 2022)

Figure 8: Dominion and EAC issued these advisories without coordinating with us. Dominion's stopped short of confirming the flaw and inaccurately suggested that only 5.5-series products were affected. EAC's mentioned only ballot images. Neither provided effective mitigations.

Finally, on October 14 (+53 days), we disclosed the vulnerability publicly at DVSorder.org. In designing this website, we sought feedback from a range of subject matter experts to help ensure that it clearly explained the issue to voters and election officials without creating unwarranted fear or providing fodder for conspiracy theories. Our public messaging stressed that DVSorder was a *privacy* flaw that *could not directly change votes*, and that there were actions officials could take to prevent exploitation in the upcoming election. We also provided a pair of software tools to help officials do that: our test tool, which checks whether a particular dataset is vulnerable (Section 3.3) and our data sanitizer tool (Section 5.2).

6.3 Responses by Dominion and the EAC

While we were working to notify jurisdictions, Dominion issued a pair of one-page "customer notifications" related to our findings. The company did not provide copies to us, and we only learned about them weeks after they were distributed.

Dominion's initial notification was dated September 9 (+18 days). While it discussed ballot-level data and privacy, it *did not mention* that the scanners had a vulnerability that revealed the order ballots were cast. This left customers to rely on Dominion's earlier, inaccurate assertions that the scanners securely randomized ballot-level data. Instead, the notification merely directed officials to "follow any state or local requirements guiding public access to and release of cast vote records" and to "consult their legal advisors for guidance on how best to ensure that [voter secrecy] protections are applied, particularly if simultaneously releasing any record (i.e., [sic] video) that could reveal a voter's identity in the order in which they cast their ballot."

Dominion issued a revised notification (shown in Figure 8a) on October 7 (+46 days), three days before we notified states. In a one-paragraph "update" appended to the original text, the company acknowledged our claims but stopped short of confirming the flaw. Oddly, the document says that it offers a way to "mitigate[] any potential risk", even though the only action it directs jurisdictions to take is to consult their lawyers. It also appears to suggest that only 5.5-series versions of Dominion products were affected, when in fact every version for which we were able to locate ballot-data was vulnerable. These omissions and potentially misleading statements likely contributed to instances where jurisdictions failed to mitigate the flaw.

On October 8 (+47 days), the EAC issued its own advisory, also without informing us. This notice (Figure 8b) described the flaw only as affecting ballot images, which likely compounded state and local officials' confusion, as Dominion's notice from the day before had seemed to focus on CVRs.

Not mentioned in either notification was that Dominion had rapidly developed a software change, apparently in response to our findings, that had begun the EAC certification process on October 4 (+43 days) as part of Democracy Suite 5.17 [58]. EAC offers a rapid review process for "minor" changes, which can be used to quickly provide security patches [59], but Dominion did not utilize this process. Instead, it rolled the change into a major update, necessitating a lengthy review that was not complete until March 16, 2023 (+206 days).

Although 5.17 is said to incorporate an "[i]mproved pseudo random number algorithm" (Section 5.1), the user manual for this version contains new guidance advising officials to "remove the RecordId information" "[i]f the Cast Vote Record Export is going to be shared" [20, p. 97]. This suggests that the new PRNG may not fully address the flaw. If removal of the record ID field is indeed warranted under 5.17, this guidance seems dangerously incomplete; it does not mention removal of the "CvrNumber" field from CSV-style CVRs or the "ImageMask" field from JSON-style CVRs, that ballot image filenames must be sanitized, or that records must be randomly shuffled (Section 5.2). Moreover, it is unclear why Dominion has left removal of the record ID field to end-users instead of performing it programmatically.

6.4 Disclosure Outcomes

Our disclosure's strategy of directing contact through state-level election offices proved less than fully effective, as officials did not always propagate information they received to the affected local jurisdictions. Only two states asked us follow-up questions or reported back to us about their mitigation plans. Though some states and localities applied appropriate defensive measures, jurisdictions in at least Alaska, California, and Georgia published vulnerable data in November 2022.

The city of San Francisco provides an illustrative example. It uses Dominion tabulators to count precinct-voted ballots and has published ballot-level data in every election since at least 2019. If the city continued to follow its ordinary procedures, tens of thousands of voters' privacy would be at risk. Our October 10 notification to California's Chief of Elections explicitly named San Francisco as an affected jurisdiction, yet the state neglected to inform the city [52].

On November 8—the day of the general election—San Francisco began publishing new vulnerable data to its elections website. We quickly attempted to contact the city's Department of Elections via its website and Twitter, but we received no response, and San Francisco posted even greater volumes of vulnerable data over the next several days. The city only began to address the problem months later, after we wrote to the San Francisco Elections Commission [28], which conducted its own technical investigation of the issue [32].

Even after San Francisco had completed its investigation, issues persisted into the city's March 2024 primary, 16 months after our initial disclosure. In that election, which again used a vulnerable version of the Dominion software, the city redacted the "RecordId" field from published CVRs, but it neglected to redact the "ImageMask" field, which contains a copy of the record ID. It also failed to remove the record IDs from ballot image filenames. These incomplete redactions happen to exactly match Dominion's guidance from the 5.17 user manual, but unfortunately they leave DVSorder fully exploitable.

Other states likewise continue to be vulnerable. One example is Georgia, which uses vulnerable scanners state-wide and requires ballot-level data to be published by law. The Georgia Secretary of State announced in 2023 that he would defer upgrading to 5.17 until after the 2024 presidential election, more than two years after our public disclosure, citing the cost and logistical complexity of deployment [61]. To our knowledge, the state has yet to issue any guidance on sanitizing ballot-level data, and Georgia counties have published vulnerable datasets as recently as May 2024.

On the other hand, our disclosures prompted effective action in at least some jurisdictions. Minnesota's Secretary of State, for example, said in a statement that his office was "working directly with the counties that use the impacted machines to ensure they are aware of the vulnerability and mitigation measures" [40]. The state subsequently amended its election law to allow jurisdictions to withhold ballot-level data from public release if it was "likely to facilitate associating votes with particular voters, such as showing the order in which the votes were cast" [25]. Several other states, including Michigan, have completed deployment of Dominion's updated software, which may remediate the vulnerability.

6.5 Effects on Public Trust

Making DVSorder public appears to have triggered little mis- or dis-information. Researchers at the University of Washington found in the wake of our public disclosure that nearly every high-engagement post about DVSorder retained our wording that it was a *privacy flaw*, and that conspiracy theories that referenced the issue gained little traction [12]. As noted above, we intentionally emphasized the privacy nature of the vulnerability, and these findings suggest that our strategy was successful at mitigating risks to public trust. This outcome suggests that public communication concerning election vulnerabilities, if appropriately clear and precise, can be constructive and responsible even in an age of disinformation.

In a reflection of the partisan politics that have infected the topic of election security, our findings were reported only by right-wing media outlets. This is unfortunate, as all voters deserve to be accurately informed about the risks elections face and steps officials take—or fail to take—to defend them.

7 Related Work

We are far from the first researchers to draw attention to the threat of ballot-ordering attacks. While DVSorder is unusual in that it can be discovered and exploited using only public information, it follows in a long series of vulnerabilities in voting systems from around the world over the past two decades.

7.1 Past Ballot-Ordering Flaws

Many examples of ballot-ordering flaws exist in the literature. We survey some of the more notable examples below:

Diebold AccuVote TS (2004) As part of the first academic security review of a deployed electronic voting machine, Kohno et al. discovered that the AccuVote TS assigned ballot serial numbers using an LCG that was seeded in a predictable manner, allowing the CVRs to be unshuffled [34]. It is remarkable that the Dominion machines we studied had such a similar design flaw nearly 20 years after scientists first warned of the danger of using an LCG to generate ballot identifiers.

Diebold AccuVote TSX (2007) Calandrino et al. found that Diebold had implemented a new PRNG in response to the findings above, but the generator was still not securely seeded, allowing the order of ballots to be recovered [11, Issue 5.2.21].

Sequoia AVC Edge (2007) In an examination of the AVC Edge, Blaze et al. found that it shuffled ballots into only ten possible orders. An attacker with access to the raw data could determine the seed and fully unshuffle the ballots [8, §4.4.8].

ES&S iVotronic (2007) Blaze et al. discovered that the iVotronic's internal data structures were insufficiently randomized, allowing a partial ordering to be determined from the raw files [7, §7.2.15–16]. Moreover, individual audit trail printouts contained a bar code that included the time the ballot was cast, allowing the order to be fully reconstructed.

Hart eSlate and eScan (2007) Inguva et al. analyzed eSlate and eScan machines, which they found used a poorly randomized data structure for CVR storage, allowing ballots to be recovered in voted order from the raw data files [31, Issue 25].

Sequoia AVC Advantage (2008) Appel et al. found that the AVC Advantage shuffled its ballot storage using a 32-bit seed that could be extracted by poll workers [3, §49].

Scytl Pnyx.core (2008) Clarkson et al. found that remote voting software developed by Scytl relied on an LCG for ballot shuffling [14, §4.3.4.1]. They reported that it used the C standard library's `rand()` function, which is notoriously insecure, and had a bug that resulted in a seed of 8 bits instead of the intended 32, allowing votes to be unshuffled with ease.

India (2010) Wolchok et al. demonstrated that paperless electronic voting machines used nationwide in India stored votes internally in the order they were cast [63]. Given brief physical access to the machine, ballots could be extracted in voted order by connecting a device to the memory chip.

Brazil (2014) Aranha et al. discovered that voting machines used throughout Brazil relied on the C `rand()` function to shuffle ballots [4]. This generator was seeded with the current time in seconds when the software was initialized, which usually occurred between 7 and 8 AM on the day of the election. The seed space thus included only 3600 possible values. Moreover, the machine printed the time it was initialized on a paper poll tape, publicly revealing the exact value of the seed.

Australia (2018) Wilson-Brown discovered that, since 2001, the voting technology used by the Australian Capital Territory *did not shuffle ballots* before they were published online [62]. The election authority dismissed the vulnerability as "purely theoretical" [30], and as of 2020 it had not been fixed [55].

Dominion ICP (2021) In a court-authorized examination, Halderman and Springall found that ICP scanners record ballot images to their removable memory cards in the order they were cast, and, by default, unencrypted [29, §11.2]. Election workers with access to the cards could exploit this to recover the ballots in voted order, even if DVSorder was corrected.

That these errors have occurred time and again for 20 years suggests a persistent failure by vendors and regulators to safeguard voter privacy. Even after this long line of research, ballot-ordering flaws appear not to be recognized within the election sector as important vulnerabilities, despite being a low-risk exploit accessible even to unsophisticated attackers and despite increasing public availability of ballot-level data.

7.2 Other Ballot Secrecy Risks

Protecting privacy in ballot-level data requires more than just concealing the order of casting. Voters who are bribed or coerced can reveal which ballots are theirs by marking them in a distinctive fashion [9], by entering a unique name as a write-in candidate, or by selecting a specific pattern of choices in down-ballot contests [47]. Some voters find themselves in very small anonymity sets: in the U.S., voting districts at different levels sometimes intersect such that only one or a few voters cast ballots with a particular set of contests, and voters with disabilities often use ballot-marking devices that print ballots that appear different from ones marked by hand [36]. Other, more exotic attacks—such as techniques for automatically fingerprinting ballots based on natural imperfections in the paper surface [13] or on voters' individual marking styles [10]—may provide additional avenues for deanonymization.

8 Discussion

DVSorder points to systemic gaps in the design and regulation of election technology with regard to voter privacy. It also provides a case study of challenges to effective vulnerability disclosure and mitigation within the election sector. We discuss these issues below and offer recommendations.

8.1 Design, Certification, and Testing for Privacy

That ballot-ordering vulnerabilities continue to persist indicates an ongoing inattentiveness to voter privacy by voting system developers and regulators. Privacy-conscious system design should have prevented the introduction of these flaws, and privacy-conscious security review should have uncovered them before they placed voters at risk.

Much responsibility falls on the election system vendors. DVSorder resulted from Dominion's use of a class of pseudorandom number generator that was known to be insecure since the 1970s [50]. The company's attempts at obfuscation suggest that it knew the approach was unsound but was unwilling to apply sufficient resources or expertise to implement a correct solution. Responsibility also falls on state and federal testing authorities, who repeatedly passed the vulnerable Dominion systems, even though the vulnerability should have been readily apparent in a source code review.

U.S. Federal Standards One reason that such problems have been overlooked during system certification is that testing standards are insufficient. In the U.S., the most important regulatory standards for election equipment are the Voluntary Voting System Guidelines (VVSG), a set of criteria maintained by the EAC. Though compliance is not federally mandated, many states require it [43]. Equipment is tested against these standards by private entities called Voting System Test Laboratories (VSTLs) that are accredited by the EAC.

While the EAC guidelines include a number of provisions related to voter privacy, they are evidently ineffective. For example, VVSG 1.0 provides that "[t]he voting process shall preclude anyone else from determining the content of a voter's ballot, without the voter's cooperation" [57, §3]. The affected Dominion equipment is certified to this standard, and despite its plain language, EAC representatives have informed us that they have not found the equipment to be in violation.

A recently adopted revised standard, VVSG 2.0, adds more specific privacy requirements among many other changes. It provides that "[t]he 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" [56, §10.2.2-B]. Though such a provision would seem to squarely address DVSorder, it is still possible that affected systems could be certified under the new requirements. This is because VSTLs test systems against a predefined set of "test assertions", which serve as a checklist for determining whether or not the system meets the VVSG's requirements. There is no test assertion that requires a VSTL to assess compliance with this provision [16], so a nonconforming system could still be certified under current testing procedure.

Recommendations Voting system vendors should incorporate a focus on security and privacy throughout the product development cycle, as is best practice in the broader technology industry. Vendors should design systems to be secure against a threat model that includes deanonymization attacks, and they should rigorously test their systems for success against that model, whether through in-house or by engaging qualified penetration-testing firms.

Likewise, federal and state regulators should place greater value on privacy. In particular, the EAC should ensure that VSTLs test for all privacy-related requirements in VVSG 2.0. While such open-ended requirements can be difficult to reduce to checklist or "test assertion" form, many problems like DVSorder could be uncovered if the agency required VSTLs to perform a privacy-focused source code review [16].

8.2 Challenges to Effective Disclosure

When vulnerabilities slip through certification and are discovered in live election systems, mitigation requires getting timely, accurate, and actionable information to all responsible parties, which potentially include hundreds or thousands of local jurisdictions. Challenges in conducting this complex communication can delay remediation and increase risk.

During disclosure of DVSorder (Section 6), we were surprised that CISA and EAC did not provide effective assistance. Neither agency helped to coordinate disclosure or even to broker communication between us and state officials. We were also surprised to learn that neither federal agency has the ability to comprehensively identify (much less contact) the U.S. localities affected by a particular voting system vulnerability.

Some state-level authorities did not convey our disclosure to localities. This leaves local officials reliant on their system vendors for information about vulnerabilities in their products. Unfortunately, Dominion chose not to cooperate with us and issued guidance that was incomplete and misleading. As a result of these challenges, many localities did not know they were vulnerable even after we publicly announced the flaw.

Adding to the confusion, when states and localities did receive notifications about the flaw, those notifications were inconsistent. Dominion and the EAC issued separate advisories without coordinating with us that characterized the problem in different and incomplete ways. Faced with conflicting statements about the problem coming from researchers, the federal government, and the system vendor, election officials were left to puzzle out what to do for themselves.

Recommendations The federal government should establish a coordinated vulnerability disclosure program that reliably serves as a clearinghouse for election system vulnerability disclosures. An existing program, such as CISA's, could serve this role if it had the resources and mandate to assist in cases such as DVSorder. Vendors and federal authorities should coordinate with researchers who identify vulnerabilities in order to ensure that communication about the problems is accurate and consistent. State-level election agencies should establish a uniform contact method for reporting security disclosures (perhaps coordinated by a body such as NASED). Vendors like Dominion should accept responsibility for vulnerabilities in their products. By working cooperatively with third-parties who report such problems, vendors can convey more accurate information to localities and better facilitate mitigation.

8.3 Challenges to Effective Patching

Communication is only half the battle. Once state and local entities know about a vulnerability, they need to take effective steps to mitigate it. Often one of the most important mitigations is deploying software updates. This can be logistically challenging when it involves many thousands of systems operated by localities across a state, since election equipment is typically not connected to the Internet and updates must be installed manually by a technician. Unfortunately, as DVSorder illustrates, the election sector's approach to vulnerability management lags far behind enterprise best practice.

From the time we disclosed the problem to Dominion, it took almost seven months before a software update to address it was developed and federally certified. Even after the update was available, some jurisdictions stated that they would need years to deploy it, and others have no plans to do so. Now, almost two years after our disclosure, states such as California and Georgia are still using vulnerable system versions.

Past election system vulnerabilities have seen even worse response times. For example, in 2007, researchers publicly disclosed buffer overflows and numerous other flaws in a paperless voting machine called the Diebold AccuVote TSX [11], but Georgia used the machines state-wide for 12 more years *without ever patching them* [26]. States will need to significantly mature their vulnerability management practices to match the rapid response times that are expected in industry and other areas of critical infrastructure.

Recommendations Voting system vendors should work to make software fixes available more quickly, and make them easier to deploy, by producing focused patches for security issues rather than incorporating them into major updates. Vendors should also commit to back-porting security fixes to all in-service versions of their products.

States and localities should build capacity to implement election software updates and other mitigations quickly. One potentially effective approach would be to plan and budget for deploying such updates on an annual cycle. This would maintain institutional familiarity with the logistics, which would also help ensure that emergency fixes could be deployed rapidly before an election if necessary.

The costs of keeping up-to-date with security patches vary dramatically with the style of election technology. States like Georgia that require all in-person voters to use computer ballot-marking devices will need to update and maintain many times as many units of equipment as states where most voters mark ballots by hand. States should consider and price-in these costs when selecting a voting system.

The EAC could encourage timely patching under its existing regulatory authority by withdrawing certification from versions of election systems that lack important security fixes. This would encourage vendors to rapidly develop security patches and prompt many jurisdictions to apply the patches so as to maintain compliance with state law.

9 Conclusion

Researchers have uncovered ballot-ordering vulnerabilities at a steady cadence for more than two decades, but DVSorder is a particularly concerning instance of such a flaw. It allows unsophisticated and passive attackers to learn how individuals vote without risk of being detected, and it comes at a time when more ballot-level data is public than ever before.

Jurisdictions that use affected equipment should ensure that their systems are patched to eliminate the flaw, and, until then, that their ballot-level data is properly sanitized before publication. Unfortunately, some states and localities have yet to implement these steps and continue to release vulnerable data today, nearly two years after our disclosure.

DVSorder should be a wake-up call that the election sector needs to take voter privacy more seriously. How is it possible that in 2022, a leading voting system vendor thought shuffling ballots with an LCG was appropriate? Why didn't this raise red flags with regulators during any one of their dozen certification reviews? Why can't jurisdictions as large as Georgia or as sophisticated as San Francisco manage to address the problem even after they know about it? If the right to a secret ballot is to mean anything, actors across the election technology space must do their part to protect it.

Acknowledgements

We thank the many people whose insights or other assistance helped make this work possible, including: Andrew Appel, Jack Bernard, Genya Coulter, Aleks Essex, Rop Gonggrijp, Robert Graham, Harri Hursti, Chris Jerdonek, Sarah Madden, Marilyn Marks, Neal McBurnett, Kevin Skoglund, Mike Specter, Vanessa Teague, the anonymous reviewers, and J.A.E. Halderman. Figures 2 and 3 are adapted from the work of LaKyla Thomas. This material is based upon work supported by the U.S. National Science Foundation under Grant No. CNS-1518888 and by the Andrew Carnegie Fellowship.

References

[1] T. Adams et al. *Scientists say no credible evidence of computer fraud in the 2020 election outcome, but policymakers must work with experts to improve confidence*. Open letter. Nov. 16, 2020. URL: https://www.mattblaze.org/papers/election2020.pdf.

[2] P. Anderholm. *Republicans want to make it easier to access vote records*. Democracy Docket. Nov. 1, 2023. URL: https://www.democracydocket.com/analysis/republicans-want-to-make-it-easier-to-access-vote-records/.

[3] A. W. Appel, M. Ginsburg, H. Hursti, B. W. Kernighan, C. D. Richards, and G. Tan. *Insecurities and inaccuracies of the Sequoia AVC Advantage 9.00H DRE voting machine*. 2008. URL: https://www.cs.princeton.edu/~appel/papers/advantage-insecurities-redacted.pdf.

[4] D. F. Aranha, M. M. Karam, A. Miranda, and F. Scarel. "Software vulnerabilities in the Brazilian voting machine." In: *Design, Development, and Use of Secure Electronic Voting Systems*. Ed. by D. Zissis and D. Likkas. IGI Global, 2014. Chap. 8, pp. 149–175. DOI: 10.4018/978-1-4666-5820-2.ch008.

[5] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. "Format-preserving encryption." In: *16th Workshop on Selected Areas in Cryptography*. Vol. 5867. Lecture Notes in Computer Science. 2009, pp. 295–312. DOI: 10.1007/978-3-642-05445-7_19.

[6] M. Bernhard, K. Kandula, J. Wink, and J. A. Halderman. "UnclearBallot: Automated ballot image manipulation." In: *4th International Joint Conference on Electronic Voting (E-Vote-ID)*. Vol. 11759. Lecture Notes in Computer Science. 2019, pp. 14–31. DOI: 10.1007/978-3-030-30625-0_2.

[7] M. Blaze, A. Aviv, P. Černý, S. Clark, E. Cronin, G. Shah, and M. Sherr. "ES&S specific weaknesses and their implications." In: *EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing*. 2007, pp. 59–82. URL: https://www.eac.gov/sites/default/files/document_library/files/EVEREST.pdf.

[8] M. Blaze, A. Cordero, S. Engle, C. Karlof, N. Sastry, M. Sherr, T. Stegers, and K.-P. Yee. *Source code review of the Sequoia voting system*. Part of the California Secretary of State's "Top-to-Bottom" Voting Systems Review. 2007. URL: https://votingsystems.cdn.sos.ca.gov/oversight/ttbr/sequoia-source-public-jul26.pdf.

[9] M. M. Brown. "Elections: Improper marking of ballots." In: *Michigan Law Review* 23.5 (1925), pp. 522–527. DOI: 10.2307/1279163.

[10] J. A. Calandrino, W. Clarkson, and E. W. Felten. "Bubble trouble: Off-line de-anonymization of bubble forms." In: *20th USENIX Security Symposium*. 2011, pp. 267–280. URL: https://www.usenix.org/conference/usenix-security-11/bubble-trouble-line-de-anonymization-bubble-forms.

[11] J. A. Calandrino, A. J. Feldman, J. A. Halderman, D. Wagner, H. Yu, and W. P. Zeller. *Source code review of the Diebold voting system*. Part of the California Secretary of State's "Top-to-Bottom" Voting Systems Review. 2007. URL: https://votingsystems.cdn.sos.ca.gov/oversight/ttbr/diebold-source-public-jul29.pdf.

[12] M. Caulfield and Z. Kharazian. *Election vulnerability disclosure becomes fodder for dueling conspiratorial narratives on Telegram*. Election Integrity Partnership. Oct. 28, 2022. URL: https://www.eipartnership.net/blog/election-vulnerability-disclosure-conspiratorial-narratives-telegram.

[13] W. Clarkson, T. Weyrich, A. Finkelstein, N. Heninger, J. A. Halderman, and E. W. Felten. "Fingerprinting blank paper using commodity scanners." In: *30th IEEE Symposium on Security and Privacy*. 2009, pp. 301–314. DOI: 10.1109/SP.2009.7.

[14] M. Clarson, B. Hay, M. Inge, A. Shelat, D. Wagner, and A. Yasinsac. *Software review and security analysis of Scytl remote voting software*. 2008. URL: http://www.soc.southalabama.edu/~yasinsac/Evoting/ODBPFinalReportSept2008.pdf.

[15] Congressional Research Service. *The U.S. Election Assistance Commission (EAC): Overview and selected issues for Congress*. Feb. 23, 2023. URL: https://crsreports.congress.gov/product/pdf/R/R45770#page12.

[16] B. Crimmins, D. Narayanan, J. A. Halderman, and A. Springall. *Public comment re VVSG 2.0*. June 6, 2023. URL: https://www.regulations.gov/comment/EAC-2023-0001-0009.

[17] Cybersecurity and Infrastructure Security Agency. *Election Security*. URL: https://www.cisa.gov/topics/election-security.

[18] Cybersecurity and Infrastructure Security Agency. *Vulnerabilities affecting Dominion Voting Systems ImageCast X*. ICS Advisory ICSA-22-154-01. June 3, 2022. URL: https://www.cisa.gov/news-events/ics-advisories/icsa-22-154-01.

[19] Dominion Voting Systems. *Coordinated vulnerabilty disclosure policy*. URL: https://www.dominionvoting.com/coordinated-vulnerability-disclosure-policy/.

[20] Dominion Voting Systems. *Democracy Suite EMS Results Tally & Reporting user guide*. 2023. URL: https://www.sos.state.co.us/pubs/elections/VotingSystems/DVS-DemocracySuite517/documentation/UGRTRUserGuide5-17CO.pdf.

[21] C. Fitzgerald, P. Smith, and S. Goodman. *The secret ballot at risk: Recommendations for protecting democracy*. Electronic Privacy Information Center, Verified Voting, and Common Cause. 2016. URL: https://secretballotatrisk.org/Secret-Ballot-At-Risk.pdf.

[22] S. Fowler. *What does Georgia's new voting law SB 202 do?* Georgia Public Broadcasting. Mar. 27, 2021. URL: https://www.gpb.org/news/2021/03/27/what-does-georgias-new-voting-law-sb-202-do.

[23] K. Friel and J. Singh. *Voter intimidation and election worker intimidation resource guide*. Brennan Center for Justice. 2022. URL: https://www.brennancenter.org/our-work/research-reports/voter-intimidation-and-election-worker-intimidation-resource-guide.

[24] A. Gardner and P. Marley. *Trump backers flood election offices with requests as 2022 vote nears*. The Washington Post. Sept. 11, 2022. URL: https://www.washingtonpost.com/nation/2022/09/11/trump-election-deniers-voting/.

[25] M. Hailperin. *The 2023 legislature did a lot on election laws. Here's a (nearly) comprehensive look.* Minnesota Reformer. May 24, 2023. URL: https://minnesotareformer.com/2023/05/24/the-2023-legislature-did-a-lot-on-election-laws-heres-a-nearly-comprehensive-look/.

[26] J. A. Halderman. *Security analysis of the Dominion ImageCast X*. Freedom to Tinker. June 2023. URL: https://freedom-to-tinker.com/2023/06/14/security-analysis-of-the-dominion-imagecast-x/.

[27] J. A. Halderman. "The Antrim County 2020 election incident: An independent forensic investigation." In: *31st USENIX Security Symposium*. 2022. URL: https://www.usenix.org/system/files/sec22-halderman.pdf.

[28] J. A. Halderman. *Voter privacy flaw affecting San Francisco's ImageCast Evolution tabulators*. Email correspondence. Jan. 9, 2023. URL: https://www.sf.gov/sites/default/files/2023-02/Halderman_Email_to_Commission.pdf.

[29] J. A. Halderman and D. Springall. *Security analysis of Georgia's ImageCast X ballot marking devices*. Expert report submitted on behalf of plaintiffs Donna Curling, et al. in *Curling v. Raffensperger*, Civil Action No. 1:17-CV-2989-AT, U.S. District Court for the Northern District of Georgia, Atlanta Division. 2021. URL: https://storage.courtlistener.com/recap/gov.uscourts.gand.240678/gov.uscourts.gand.240678.1681.0.pdf.

[30] J. Hayne and A. Bogle. *Elections ACT dismisses concerns electronic ballots could be traced to voters*. ABC News. 2018. URL: https://www.abc.net.au/news/2018-08-14/voters-in-act-election-could-have-ballot-choices-identified/10115670.

[31] S. Inguva, E. Rescorla, H. Shacham, and D. S. Wallach. *Source code review of the Hart InterCivic voting system*. Part of the California Secretary of State's "Top-to-Bottom" Voting Systems Review. 2007. URL: https://votingsystems.cdn.sos.ca.gov/oversight/ttbr/Hart-source-public.pdf.

[32] C. Jerdonek. *DVSorder confirmation and observations*. Feb. 2023. URL: https://www.sf.gov/sites/default/files/2023-02/Memo_re_DVSorder_Jerdonek.pdf.

[33] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*. 1st ed. Vol. 2. Addison-Wesley, 1969.

[34] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach. "Analysis of an electronic voting system." In: *IEEE Symposium on Security and Privacy*. 2004. DOI: 10.1109/SECPRI.2004.1301313.

[35] P. L'Ecuyer. "Testing random number generators." In: *Winter Simulation Conference*. 1992. URL: https://www.informs-sim.org/wsc92papers/1992_0033.pdf.

[36] J. Lazar. "Segregated ballots for voters with disabilities? An analysis of policies and use of the ExpressVote ballot marking device." In: *Election Law Journal* 18.4 (2019), pp. 309–322. DOI: 10.1089/elj.2018.0531.

[37] Lycoming County Court of Common Pleas. *Honey vs. Lycoming County Office of Voter Services*. Dec. 16, 2022. URL: https://www.lycolaw.org/uploads/opinions/2022-honey121622li.pdf.

[38] G. Marsaglia. "Random numbers fall mainly in the planes." In: *PNAS* 61.1 (1968), pp. 25–28. DOI: 10.1073/pnas.61.1.25.

[39] Michigan Compiled Laws §168.733. *Challengers; space in polling place; rights; space at counting board; expulsion for cause; protection; threat or intimidation*. URL: https://legislature.mi.gov/Laws/MCL?objectName=MCL-168-733.

[40] Minnesota Secretary of State. *Office statement on additional steps to protect ballot privacy*. Oct. 14, 2022. URL: https://www.sos.state.mn.us/about-the-office/news-room/office-statement-on-additional-steps-to-protect-ballot-privacy/.

[41] D. Molnar, T. Kohno, N. Sastry, and D. Wagner. "Tamper-evident, history-independent, subliminal-free data structures on PROM storage -or- How to store ballots on a voting machine." In: *IEEE Symposium on Security and Privacy*. 2006. DOI: 10.1109/SP.2006.39.

[42] National Association of State Election Directors. *Our members*. URL: https://www.nased.org/members.

[43] National Center for State Legislatures. *Election Assistance Commission updates Voluntary Voting System Guidelines*. June 2022. URL: https://www.ncsl.org/state-legislatures-news/details/election-assistance-commission-updates-voluntary-voting-system-guidelines.

[44] Ordros Analytics. *Public CVR repository*. URL: https://ordros.com/cvr/index.html. Archived at https://web.archive.org/web/20220915000000*/https://ordros.com/cvr/index.html.

[45] H. Peltoa, H. Söderlund, and E. Ukkonen. "SEQAID: A DNA sequence assembling program based on a mathematical model." In: *Nucleic Acids Research* 12.1 (1984), pp. 307–321. DOI: 10.1093/nar/12.1part1.307.

[46] J. B. Plumstead. "Inferring a sequence generated by a linear congruence." In: *23rd Annual Symposium on Foundations of Computer Science*. IEEE, 1982. DOI: 10.1109/SFCS.1982.73.

[47] S. Popoveniuc and J. Stanton. "Undervote and pattern voting: Vulnerability and a mitigation." In: *IAVoSS Workshop on Trustworthy Elections (WOTE)*. 2007.

[48] *Privileges and Conduct of Poll Watchers*. 6 Alaska Administrative Code §25.020(c).

[49] Pro V&V. *Test plan for EAC 2005 VVSG certification testing: Dominion Voting Systems Democracy Suite (D-Suite) version 5.17 voting system*. Dec. 2, 2022. URL: https://www.eac.gov/sites/default/files/voting_system/files/Dominion%20Voting%20Systems%20D-Suite%205.17%20Test%20Plan-Rev.%2001.pdf.

[50] J. Reeds. "'Cracking' a random number generator." In: *Cryptologia* 1 (1 1977), pp. 20–26. DOI: 10.1080/0161-117791832760.

[51] San Francisco Department of Elections. *Past election results*. URL: https://sfelections.sfgov.org/past-election-results.

[52] San Francisco Elections Commission. *Draft letters re DVSorder to California Secretary of State and U.S. EAC*. Mar. 10, 2023. URL: https://www.sf.gov/sites/default/files/2023-03/DVSorder_Letters_DRAFT_v2.pdf.

[53] M. A. Specter, J. Koppel, and D. Weitzner. "The ballot is busted before the blockchain: A security analysis of Voatz." In: *29th USENIX Security Symposium*. 2020, pp. 1535–1553. URL: https://www.usenix.org/system/files/sec20-specter.pdf.

[54] State of Michigan. *Voting system hardware, software, and services contract terms, Contract #071B7700117*. 2017. URL: https://www.michigan.gov/-/media/Project/Websites/dtmb/Procurement/Contracts/MiDEAL-Media/008/7700117.pdf.

[55] V. Teague and P. Keyzer. "Electronic Australian elections: Verifiability of accuracy is a design goal, which must be mandated by law and deliberately designed into electronic electoral processes." In: *Law in Context* 37 (1 2020), pp. 42–65. DOI: 10.26826/law-in-context.v37i1.119.

[56] U.S. Election Assistance Commission. *Requirements for the Voluntary Voting System Guidelines 2.0*. 2021. URL: https://www.eac.gov/sites/default/files/TestingCertification/Voluntary_Voting_System_Guidelines_Version_2_0.pdf.

[57] U.S. Election Assistance Commission. *Voluntary Voting System Guidelines: Volume 1.0: Version 1.0*. 2005. URL: https://www.eac.gov/sites/default/files/document_library/files/VVSG.1.0_Volume_1.PDF.

[58] U.S. Election Assistance Commission. *Voting equipment: Democracy Suite 5.17*. 2023. URL: https://www.eac.gov/voting-equipment/democracy-suite-517.

[59] U.S. Election Assistance Commission. *Voting system testing and certification program manual version 3.0*. Nov. 2022. URL: https://www.eac.gov/sites/default/files/TestingCertification/Testing%20and%20Certification%20Program%20Manual%20Version%203.0%20(2).pdf.

[60] Verified Voting. *The Verifier—Equipment search: Dominion*. 2022. URL: https://verifiedvoting.org/verifier/#mode/search/year/2022/make/Dominion.

[61] D. Wickert. *Georgia officials defend timetable for election software update*. The Atlanta Journal-Constitution. June 21, 2023. URL: https://www.ajc.com/news/georgia-news/georgia-officials-defend-timetable-for-election-software-update/D27QJOC2RFH5JGBC7ZNDHIDGYY/.

[62] T. Wilson-Brown. *Possible vote disclosure in ACT elections*. 2018. URL: https://github.com/teor2345/Elections2018/blob/master/ElectionsACTDisclosure.md.

[63] S. Wolchok, E. Wustrow, J. A. Halderman, H. K. Prasad, A. Kankipati, S. K. Sakhamuri, V. Yagati, and R. Gonggrijp. "Security analysis of India's electronic voting machines." In: *17th ACM Conference on Computer and Communications Security*. 2010. DOI: 10.1145/1866307.1866309.

[64] Zebraduck.org. *Georgia election files*. URL: https://www.zebraduck.org/election-files/. Archived at https://web.archive.org/web/20240112023425/https://www.zebraduck.org/election-files/.

A Mathematics of the PRNG

The generator used by Dominion has interesting properties that do not appear to have been discussed in prior literature.

A.1 Mathematically Equivalent Generators

We describe the Dominion ICP generator as follows, where s defines the substitution rule and p defines the permutation rule. For convenience, let m denote 1000000.

$$x_{n+1} = x_n + 864803 \mod m,$$

$$s_x = [5,0,8,3,2,6,1,9,4,7], \quad p = [2,6,1,5,3,4].$$

This is mathematically equivalent to the following generator:

$$y_{n+1} = y_n + 135197 \mod m,$$

$$s_y = [7,4,9,1,6,2,3,8,0,5], \quad p = [2,6,1,5,3,4].$$

To understand why these generators produce equivalent cycles, observe that $864803 + 135197 \equiv 0 \mod m$. This means that $x_n + y_n \equiv x_{n+1} + y_{n+1} \mod m$. Choose without loss of generality initial states $x_0 = 0$ and $y_0 = m - 1$. By induction $x_n + y_n \equiv m - 1 \mod m$ for all n, and because $x_n + y_n$ must be in range $[0, 2m - 1]$, we know $x_n + y_n = m - 1$.

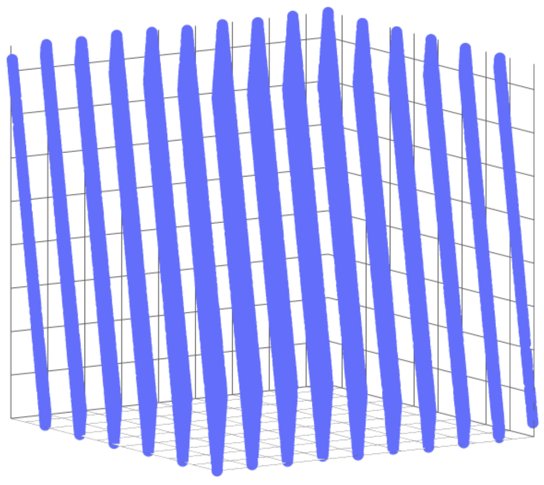
We rewrite this equation as $x_n = 999999 - y_n$. Observe that on a digit-by-digit level this means x_n has a 0 wherever y_n has a 9, x_n has a 1 wherever y_n has an 8, and so on. Observe also that our substitution rules s_x and s_y are reversed relative to each other, so that a 0 in x_n becomes a 5 in the generator's output and a 9 in y_n likewise becomes a 5, etc. This means that for all x_n and y_n, the generators' output would be the same.

The existence of a complementary generator can be generalized to any LCG which uses a multiplicative term of 1, a substitution rule, and a modulus that is the power of that substitution rule's base. We need only that $b_x + b_y = m$ and that s_x and s_y are reversed relative to one another.

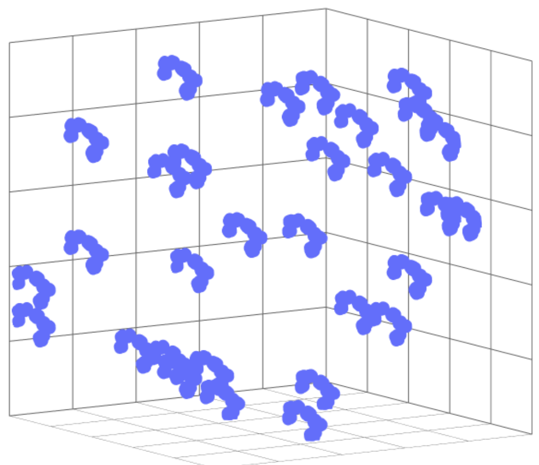
A.2 Graphical Analysis

Linear congruential generators are known to produce output such that when consecutive terms are plotted as three-dimensional points—i.e., points in the form (x_n, x_{n+1}, x_{n+2})—these points are not uniformly distributed but instead fall into planes [38]. The so-called "spectral test" measures the relative strength of a linear congruential generator by the distance between these planes [35], and it is not uncommon to use a graphical representation of these points to provide some degree of intuition of a generator's relative strength.

The Dominion PRNG's output, when graphed, does not fall into planes; the substitution cipher and digital permutation destroy this pattern. The resulting pattern still appears highly non-random, however, and to our knowledge has not been discussed in the literature. Figure 9 depicts a typical LCG's output compared to that of the PRNG from the ICP scanner.



(a) Typical LCG (RANDU)



(b) Dominion's Obfuscated LCG

Figure 9: Unlike typical LCGs, Dominion's does not produce output that falls into planes, but other highly nonrandom patterns still result.