Secure, Accessible, Virtual Voting Infrastructure (SAVVI): Reducing Barriers for Disabled and Overseas Voters

Conference Paper · May 2023

DOI: 10.23919/MIPRO57284.2023.10159972

2 authors, including:

Justin Pelletier
Rochester Institute of Technology
17 PUBLICATIONS **68** CITATIONS

SEE PROFILE

Secure, Accessible, Virtual Voting Infrastructure (SAVVI): Reducing Barriers for Disabled and Overseas Voters

Ted Selker and Justin Pelletier
Rochester Institute of Technology
Rochester, NY 14623
Email: jxpics@rit.edu

***Abstract*—We describe a way to deploy a secured ballot return for overseas, vision, or dexterity-impaired voters. SAVVI is a secure, accessible, virtual voting infrastructure. It uses multiple communication channels with synchronized multi–factor authentication, encryption, and hashing to preserve privacy, confidentiality, and vote integrity. Our usability goal is to supply voters easy access to a hardened system that will present secure instances of their familiar browser and email clients. A key to its viability is synchronizing authentication through two low-tech verifications such as phone calls. This strives to enhance security and usability for remote voting by mimicking best practices for in-person polling place procedures. Other more standard cryptographic measures include layered encryption and dynamically provisioning a secure virtual container–a virtual voting machine (VVM)–to process each ballot. In aggregate, the design of SAVVI seeks to allow remote voting while reducing difficulty for the voter, programming complexity for the election administrator, and material procurement for the voting authority.**

I. Introduction

Mail-in ballots are problematic for some classes of voters. Overseas Americans and Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA) voters have had difficulties getting mail-in ballots to their home polling places in a timely way [49]. Blind voters, as well as people with limited manipulation, are not able to indicate selections and return a mail-in ballot privately and independently. For these classes of voters, some U.S. states such as North Carolina and Massachusetts allowed ballots to be delivered, marked, and returned electronically in 2020 [35]. This paper proposes an approach to assemble existing technologies to allow usable secure electronic return for these voters.

Previous work, such as Bruck, Jefferson, and Rivest's proposal for frog voting, considers architectures that separate vote generation from vote casting[17]. Our proposal also separates vote generation from casting. We also add multifactor authentication, dynamic provision of the vote generation machine, and provide some consideration of usability for disabled voters.

Our approach improves security for election officials who need to receive UOCAVA or disabled voter's ballots remotely. In doing this, we seek to minimize the need for an expensive procurement process, creating new software, or using security approaches that require deep expertise.

II. Background

In this section, we elaborate on the context for our goals to integrate Confidentiality, Integrity, and Accessibility for disabled and overseas voters. Confidentiality demands independent ballot marking and submission and protecting ballot confidentiality in transit. Integrity requires that voters can verify votes are cast as intended and counted as cast. We also strive to reduce the attack surface of voting infrastructures. Finally, we seek to integrate these security features while simultaneously improving Accessibility by requiring fewer and more familiar steps.

A. Voting Security Challenges

Many security problems result from the trade–offs that increase usability of voting systems, and a large number of reported problems arise from the user-interface. Though improved ballot–design practices might yield usability improvements, all practices need continual reassessment to balance security needs.

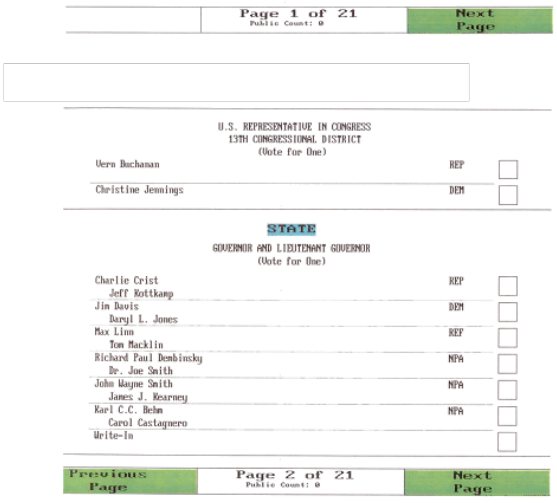

Fig. 1: 2006 Sarasota ballot where 13% of voters made no selections on Congressional CD13

Additional security concerns plague voting. People might rely on incorrect voter information that inhibits them from voting [33][31]. People might find it easier to vote for a selection that is first in a list [55]. Mechanical problems have at times kept votes from being counted, as in the incomplete selections from the hanging chad problems in the 2000 Florida election [56]. Tallying problems sometimes require recounts. Some voters notice that they have voted for the selection adjacent to the one they intended (called flipping). Sometimes data-record manipulation has been found, such as occurred in Volusia County in the 2000 election[24].

The question of how to securely deploy a system with complete privacy continues. Many people have stated that paper ballots are the best approach and they may be for now. Unfortunately, at times paper ballots have been compromised – even in wholesale ways – with reports of deliberate ballot access limitations, incorrectly printed ballots, chain voting to systematically coerce voters, lost ballot boxes, fraud in hand counting, and "new" ballots turning up in recounts [6], [29].

B. Mail-in Voting and Accessibility

During the COVID-19 pandemic, mail-in voting became critical for allowing vulnerable people to vote safely. Still, there are risks to mail-in voting [11]. Some ballots never get to the voter, get lost by the voter, or get spoiled by the voter. Some ballots do not get returned in time to be counted, are sent to the wrong addresses, or have incorrect use of the envelope and are thrown out. Various other simple system and user problems stop votes from being included in the vote. Electronic voting from home might still have some of these problems because voters might not set up the electronic voting system correctly. They might miss the emails. They might not use the software correctly to mark the ballot. They might not return the ballot correctly. Our architecture attempts to mitigate these risks.

There are even more challenges for disabled voters seeking to remotely return ballots. This remains a particularly important consideration because "individuals with disabilities also report voting by mail at much higher rates than do individuals not reporting a disability." [4]

1) Disabled Voters.: Section 508 under title 29 of the United States Code (https://section508.gov) defines functional requirements for technologies that accommodate nine classes of Americans with disabilities: (a) without vision, (b) with limited vision, (c) without perception of color, (d) without hearing, (e) with limited hearing, (f) without speech, (g) with limited manipulation, (h) with limited reach and strength, and (i) the need to minimize photosensitive seizure triggers.

We consider these requirements in light of specific requirements relating to absentee ballot preparation and submission. While deaf and mute voters should be able to use typical methods for voting, other disabilities present more challenges.

Voters with no/limited vision have used magnifiers, braille, and audio to vote. Although only some 10% of blind voters use braille [2] , Rhode Island for example, has made braille ballots available [7] Audio is a typical and accessible option made available to blind voters [36], [8], [5], [25].

Cognitively-disabled voters includes several populations. For example, more than 14% of voters have reading disabilities and more than 6% of voters have some short-term memory problems [47]. Structuring and the presentation of ballots for these voters can greatly reduce their errors. The practice of filling out a sample ballot to help them fill out the actual ballots reduces errors as well [26] [46].

Physically-disabled voters can have mobility or manipulation problems. Mobility problems might make it difficult for them to stand in line with their mobility solution. It has also been a challenge for polling places to make themselves adequately accessible to people with mobility problems. Manipulation problems are more difficult. People with manipulation problems can use prostheses to vote. Vote–marking machines are equipped with interfaces. Examples include special large buttons, mouth-sticks that can press buttons, or sip–and–puff devices to allow them to make selections with their mouth.

Mail-in voting presents privacy problems for blind and manipulation-challenged voters. We know of no workable prosthetic for allowing blind people to privately and independently vote on mail-in ballots. For people with manipulation problems, marking the ballot, putting it in the envelope, and mailing it without assistance can be impossible. Physically-disabled voters might also have difficulties assuring their paper ballots are received and delivered [35].

2) Overseas Voters.: Pernicious ballot delivery delays continue to impact military personnel and people living abroad who cannot be at home to vote. Some 3 million expatriates and military personnel are eligible to vote using UOCAVA. In 2006, only 26.5% of those requesting a ballot succeeded at having it counted [32], [41].

The voting security community is increasingly concerned with voting integrity verification. By letting voters know their vote is included in the count, the new verification approaches give much-needed evidence to increase confidence in outcomes[39]. Other projects such as Remotegrity [59] and VoteXX[19] concern technology that could be used for UOCAVA voters and some of those, such as Helios[9] and Election Guard[14], are even available as open-source software. However, these systems do not address how people with disabilities can vote privately and independently. Though some of these systems may have superior security properties, they retain notable usability limitations.

C. Usable Security in Electronic Voting

Concerns of electronic fraud have impacted the way we think about elections. Sadly, misinformation campaigns are common for reducing voting turnout. These are now being purveyed successfully online. For example, foreign messaging interference in the 2016 presidential election was impactful [12] and bad actors may turn or may have already turned to focusing on the voting systems themselves. To minimize risks, the Federal Election Commission and the Election

Assistance Commission have promulgated voting system certification recommendations. Every jurisdiction can choose how it will certify voting machines. Most states follow the EAC's Voluntary Voting Systems Guidelines (VVSG) for voting machine hardware and software [1] following rigorous testing and scrutiny defined by the National Institute of Standards and Technology (NIST) Handbook, pages 150-22. Still, each state has the right to control its approach to voting, so states mandate their specific certification procedures.

Due to past lost votes for the some 6 million expatriates, states also specify alternative approaches fulfilling UOCAVA [3]. The security standardization and the best practices in the Department of Defense (DoD) Inspector General's 2016 report on the department's policies, procedures, and practices for information security management [28] generally describe the same needs identified by the National Institute of Standards and Technology's (NIST) 2008 threat analysis of UOCAVA voting systems [42]. Though specific technologies have continued to change, both reports suggest three fundamental security control requirements: 1) strong authentication, 2) secure email, and 3) encrypted attachments.

Despite these controls, security concerns persist. For example, a body of research investigates a hypothetical user named Johnny and his use of encryption in email clients [54], [48], [44].

Researchers found that people can't encrypt because of usability concerns. They fail to properly implement pretty good privacy (PGP) because of the steps involved, or find it so difficult to use that they don't even try to encrypt. Most recently, Ruoti, Andersen, Zappala, and Seamons discovered that the usability of PGP-based email encryption has not advanced beyond a 10% success rate[44]. They attributed these failures to the perceived complexity of the encryption mechanism requiring both sender and receiver to materially participate in the PGP key exchange. The SAVVI system seeks to address this usability problem by automating the email encryption process, so Johnny doesn't have to encrypt: the secure container that processes his ballot does it for him.

While estimates vary, probably less than 0.3% of Americans have total blindness [30]. They have historically lower-than-average turnout [4], but weak security for them could still lead to fraudulent results and possibly even an overturned election. Layering best-practice protections can improve the security and access for UOCAVA and voters with other disabilities as well.

The component that remains missing is a secure way of transmitting electronic votes: an architecture that can allow a balance between accessibility, confidentiality, and integrity for remote voters while simultaneously minimizing the burden on the local elections board and the voter.

In the next section, we present our motivation and design rationale to integrate standard and non-centralized tools. Follow-on work will specifically test usability impacts of specialized CAPTCHAS, biometrics, and secure VMs in this voting infrastructure.

[1] https://www.eac.gov/voting-equipment/

III. Security Design Rationale

In the run up to the 2019 presidential election most people were concerned with blunting the pandemic by avoiding respiratory exposure to others. The authors of this paper met when a jurisdiction began asking: "What is possible for this election to allow people with disabilities to vote privately and independently?" We considered if there were inexpensive ways of giving people with disables a way to vote that was less prone to interference than what was currently in use. In exploring this, we looked at possibilities for improving the confidentiality, integrity, and accessibility for UOCAVA voters too. The challenge was posed by election officials who were concerned that there was no time to change anything. Could a safer, easier-to-administer system be organized inexpensively? SAVVI was the result of speaking to the above challenges.

The legitimizing question is which voting methods best reveal potential problems before they materialize and simultaneously lose the least votes. Questions of how to measure this are important too. Subsequently, the design of SAVVI focused on the following requirements:

1) **Prove to voters that their vote was cast as intended and recorded as cast.** Rationale: Trustworthy voting is the foundation of democracy. Giving voters a way to see their vote was tallied–while avoiding vote–buying–could be a way to increase trust in the electoral process.
2) **Resilient to technology change.** Rationale: Technologies deprecate and improve at a high rate. The infrastructure should be flexible enough to incorporate best-in-class security and virtualization tools as they become available. Also, replacing legacy (non-virtual) voting machinery can be tremendously expensive to elections districts.
3) **Allow remote voting from personal devices.** Rationale: Personal devices are more familiar to voters with specialized access technology implementations. Furthermore, using personal devices might reduce the anxiety of voting and may lead to less mistakes.
4) **Leverage process familiarity.** Rationale: The in-person voting process includes several steps that voters and election districts expect, such as checking in with a person at both the beginning and the end of each voting experience. These processes may increase both security and trust for voters. Process familiarity may also increase adoption rates among election districts.

In the next section, we present a design proposal that meets these requirements.

IV. Secure, Accessible, Virtual Voting Infrastructure (SAVVI)

The SAVVI system starts with the design rationale of analysing concerns in using private computers for voting. It then considers what can be done to address these concerns for disabled or UOCAVA voters. We propose defense–in–depth with synchronized multi–factor authentication (MFA) and ballot–integrity verification using Commercial Off–The–Shelf (COTS) security techniques in a 6–step protocol. These

steps are listed briefly here and described in detail in the remainder of this section.

1) Step One: Voter Registration. Voters register for electronic voting using their personal device.
2) Step Two: Check-in. Voters corroborate setting up their voting system with a synchronized telephone or video-conference connection with human volunteers at a check–in desk.
3) Step Three: Voting Machine. Voters log in to a secure, virtual voting machine (VVM) to mark, secure, and submit their ballot.
4) Step Four: Check-out. Voters establish a second synchronized telephone or video–conference connection with human volunteers at a check-out desk.
5) Step Five: Ballot Tally. The system decrypts the message and tallies the ballot.
6) Step Six: Vote Certification. The system publishes a record of the ballot in a way that does not disclose the voter's identity or the ballot's contents.

A. Step One: Voter Registration

Voters must pass a test of "humanness", such as by completing the recently proposed No Nonsense CAPCHA perceptual recognition and response puzzle or suitable alternative[38].

Voters have to share something about themselves, such as their voice audio or a video of their face, stating their name, address, and phone number. A critical part of this might be the simultaneity of interacting with a real person as part of the step.

The electronic voting registration database should also use a hash of a biometric signature such as a "voice–print" or "face–print". Instead of storing the biometric signature itself, a cryptographic hash is a pre–image resistant function that can be used to verify the integrity of a file and its contents. There are many off–the–shelf voice–print and face–print systems. There exist other promising techniques that may allow specific accommodations for specific disabilities, including mobile device-based hand recognition [57]. Also, several ear biometric methods may achieve high accuracy [53]. Biometric data can be practical, using off–the–shelf components that some researchers say can add `58` bits of security, even when "assuming the attacker knows one of the fingerprints of the user" [18]. While valuable as corroborating factors, privacy and public acceptance considerations direct us to not depend entirely on such biometric techniques [27], [22], [34]. We recommend simultaneity with a human verifier during the check–in process, described in Step Two. Furthermore, we consider privacy–preserving synchronization to assure that a voter's identity will not be revealed during this process because they are never directly identified alongside their ballot, which is described in Step Three.

At the conclusion of this step, the voter receives and retains their *voter registration identification number* (`PIN1`) through a separate communication channel, such as the U.S. Postal Service, as voters get registration verification today. The voter will use this in the voter check–in process. A signature of the device could be captured during the registration. The device signature might augment multi–factor authentication, which is described in Step Two.

B. Step Two: Check–in

As with requirements to show up with credentials at governmental offices the world over, proving you are who you claim to be and that you are authorized to vote remotely must include strong authentication. This step begins with another test of humanness and multi–factor authentication (MFA) at a virtual check–in desk. Such tests of humanness are central to the polling place experience and can be an important component in authorizing remote voting. Against practical threat models directed against off-the-shelf MFA, using two or more factors of MFA provides at least `58` bits of security [18]. The following MFA factors can help assure that an actual registered voter who has not previously voted in this election is obtaining a ballot and retains the ability to submit it for counting:

- **Something you have.** The check-in desk will verify the voter's *device signature* against the one captured during the initial voter registration in Step One. This corresponds with Alaca's findings on use of device signatures as an MFA augmentation technique [10]. If this fails, or as added proof, voters will demonstrate proof of residence in accordance with their jurisdiction's requirements.
- **Something you know.** Voters will provide their *voter registration number*, `PIN1`.
- **Something you are.** The check-in desk will verify the voter's *biometric signature*

Following successful authentication, the check-in volunteer follows instructions that automate deployment of a virtual container and whitelist the device and its IP address. The check-in volunteer then directs the voter to establish a secure connection to access the virtual container. The worker at the check-in desk then provides the voter a *ballot identification number* (`PIN2`), which the voter will use to authenticate with the voting machine in Step Three.

C. Step Three: Voting Machine

The system dynamically provisions a unique voting machine. This allows for the separation of personal registration information obtained in Step One and confirmed in Step Two from the private voting data generated in Step Three. The separation of personal identifiers from voting data is especially important in light of criticisms such as those directed against the accumulation of personally identifying information within the Voatz system [50].

1) Virtual Voting Machine.: Connecting to the internet and browsing the web without protection can compromise a device. The creators of secure software rely on computers to be clean of malware or cleaned before loading their software. A fresh browser 'voting system' install and updated security patches is a way to ensure there are no harmful extensions or plugins, but is not in itself adequate to protect a voting activity. There

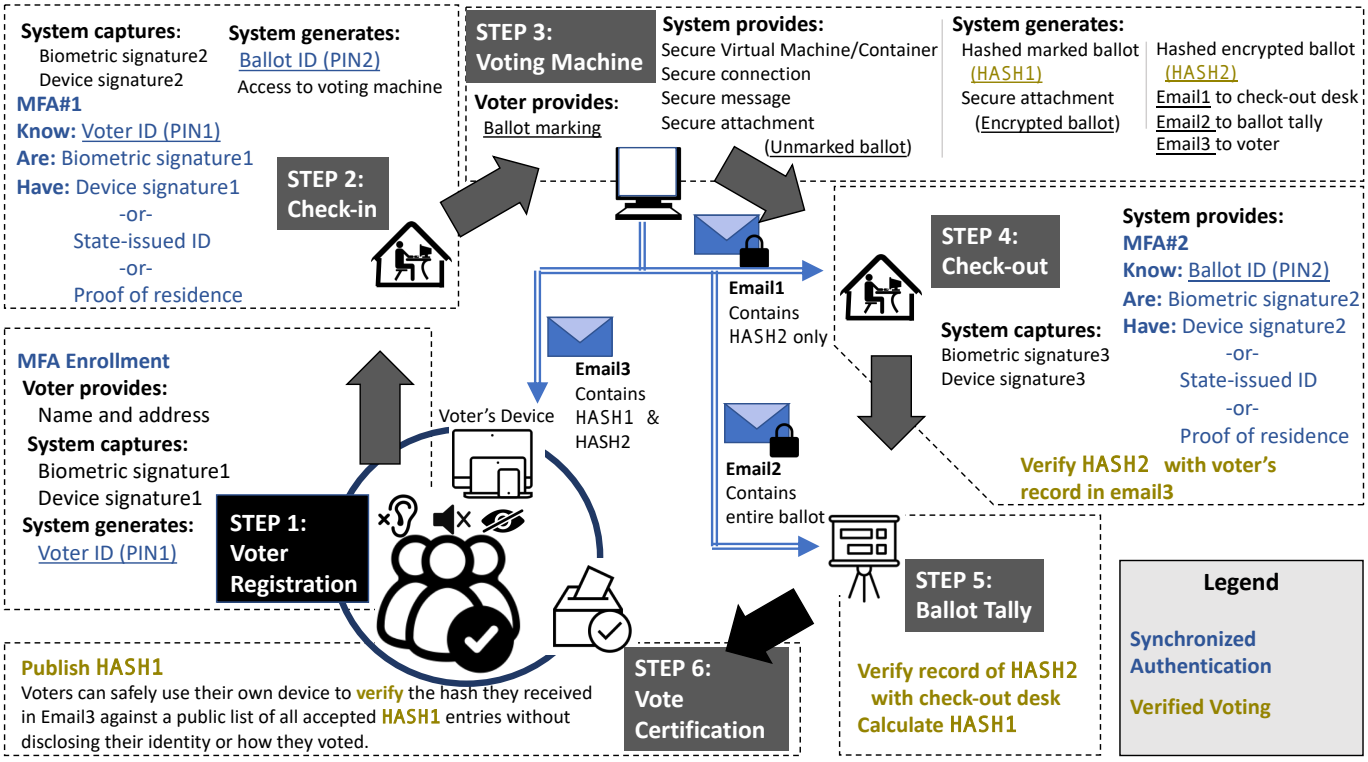

Fig. 2: Secure, Accessible, Virtual Voting Infrastructure (SAVVI)

may remain vulnerabilities in the underlying system that runs the browser.

Clean and secure implementations of browser software can solve some problems presented in today's insecure and inadequate vote-by–email procedures, such as in UOCAVA. However, SAVVI's use of a Virtual Voting Machine is intended to further secure these vote–by–email systems. It's important to note that the approach is made with corrupting possibilities in mind, as they are described in the 2018 National Academies of Science, Engineering, and Medicine (NASEM) report titled "Securing the Vote: Protecting American Democracy" [40]. The NASEM report calls out the possibility that a system can be corrupted at any layer and could flip votes. A protection against this in SAVVI is the dynamic provisioning of a Virtual Voting Machine upon voter check–in, and destruction of that machine upon voter check-out. The reduction of temporal attack surface makes it unlikely that any exploit chain originating on the user's device will be able to change the vote before check–out. Furthermore, different voters will use different systems on different networks. A system that flips ballots has to be aware of the correct ballot for the voter in the local machine used by the voter. As race choices change, some jurisdictions define their final ballot in the last day before providing it. For example, Los Angeles County provides something like 3000 different ballots to voters. It is hard to see wholesale power in creating specific code for changing particular voting choices.

Though it is not required for our system, and may not be immediately available as a COTS component, we envision a virtual voting machine (VVM) running on a secure virtual machine or virtual container that contains a popular, patched desktop. An example is a Security-Enhanced Linux container running a modern Windows Desktop. Clear step-by-step instructions could be displayed on the desktop background or in a simple graphical user interface. Also, to maximize compatibility with modern access technology for the blind and deaf, the secure VVM should automate the creation of a secure connection, a secure email, and presents a way to produce a secure, usable, and verifiable ballot attachment. Furthermore, the dynamic provision of the VVM will automate the correct installation and proper use of security software that requires administrative privilege on the VVM operating system, as in the VeraCrypt software recommended in the *Encrypt* section below.

A key advantage to using a virtual container is that each voting machine can be constructed in real time through common infrastructure automation tools like *Terraform*[2] and *Ansible*.[3] This makes provisioning a unique voting machine scalable, which greatly simplifies the verification of system security and usability for non–experts at the check–in desk. An open source repository of secure voting machine containers, virtual machines, and *Terraform*/*Ansible* scripts/playbooks will

[2] Terraform is an infrastructure compiler that translates program commands into pre–configured infrastructure. https://www.terraform.io/ [3] Ansible is another infrastructure compiler that specifically integrates IBM's *Red Hat* Security–Edition Linux containers. https://www.ansible.com/

simplify configuration. Furthermore, the dynamic creation and destruction of each machine in the span of only minutes presents reduced attack surface for a malicious actor seeking to compromise each voting device.

We recognize that the use of any phone or computer for secure communication does require testing, setting up, and rechecking. To address this concern, we recommend the voter accesses the VVM through a virtual private network (VPN). Several off-the-shelf VPN technologies exist and VPN applications are well-integrated with modern browsers [52]. If voters are unable to configure a VPN, there is still potential for secure connection using their favorite web browsers.

2) Secure connection.: On the secure machine, voters choose between secured ballot return channel options. This should include secured versions of popular email services running on popular browsers. A browser–based email service can enforce Transport Layer Security (TLS), a widely-adopted encryption scheme that adds security to websites and browser-based email. It uses public key encryption to secure the channel using a shared secret for each session of the connection. Several mechanisms can be used to secure the channel, and several are built into popular email–handling systems like *Gmail* and *Outlook*. The TLS 1.3, for example, "allows client/server applications to prevent eavesdropping, tampering, and message forgery over the internet" [43]. In 2017, Cremers, Horvat, Hoyland, et al. published a proof of the symbolic model of the TLS 1.3 [20]. The minimum effective key strength of TLS 1.3 implementations provide `128` bits of security [13]. The symbolic model of TLS 1.3 is proven, secure, and automated [15]. Still, there have been concerns about the implementation of the protocol, since insecure browser configurations can create susceptibility to TLS version–downgrade attacks, which then allow attackers to exploit vulnerabilities of previous TLS versions[16]. As of this writing, the National Vulnerability Database (NVD) indicates all the vulnerabilities specific to TLS 1.3 have been patched [1]. Ongoing attention to the NVD and maintenance of the secure container's contents will ensure that election managers avoid future security issues. Furthermore, we propose that a novel vulnerability discovery during the voting period could be obviated by our defense-in-depth strategy.

3) Secure message.: Pretty good privacy (PGP) and secure/multipurpose internet mail extensions (S/MIME) are widely–adopted for email encryption and digital–signature. These technologies add a layer of security to the message itself, which can preserve integrity and confidentiality even if the channel is compromised. S/MIME works with Gmail, Outlook, and other email systems. The current version of S/MIME is 4.0, which facilitates digital signature for authentication, integrity, and non-repudiation of the email. It also defines a standard for mail encryption exceeding `100` bits using public key infrastructure[45].

4) Secure attachment.: Pre-loaded scripts to hash, encrypt, then hash again will minimize user error and supplement vote integrity.

Hash. Hashing is a one–way function that transforms a message (or file) of any length into a string of fixed length. It can be used for integrity verification to check whether the contents of a message have been altered. SAVVI requires a voting jurisdiction to use hashing to help ensure a ballot is not tampered with and can also provide voters with a way to verify that their votes have been counted as cast. This concept of vote certification is discussed below in Section 3.6 (Step Six). The most common hashing functions are built into common utilities provided by all major device types, as it is with Microsoft's `certutil` tool.

```
H(marked_ballot}) =: HASH1 =:
   SHA512(marked_ballot)
E(marked_ballot) =:
   randomized cipher cascade
H(E(marked_ballot) =: HASH2 =:
   SHA512(E(marked_ballot))
```

Fig. 3: Pseudocode for a hash-encrypt-hash algorithm

Sensitive federal information must verify integrity with a minimum 256 bit Secure Hashing Algorithm (SHA 256) [28]. We recommend use of a 512 bit hashing algorithm (SHA 512), which presents complexities greater than 2^{158} even against methods such as the boomerang attack[58].

The system sends a copy of `HASH2` to the check–out desk, described in Step Four below. The system sends a copy of the first hash of the un–encrypted ballot `HASH1` to the voter, which allows the voter to independently verify that their electronic ballot was properly recorded after the ballot was decrypted. The handling of these hashes during the vote–tally and vote–certification steps are described below in Sections Five and Six, respectively.

Encrypt. There are diverse approaches to encrypt attachments and best practices are almost certain to evolve. For example, Meijer and van Gastel found vulnerabilities in hardware–based encryption and "strongly encourage users to instead use an open source... audited encryption software solution" and specifically recommend *VeraCrypt* as an exemplary software-based encryption tool [37]. *VeraCrypt* extends the popular cryptographic system, TrueCrypt (2004–2011). [4] A security evaluation in 2020 found that VeraCrypt protects confidentiality of data in an encrypted volume [23]. As of early 2021, the suite offers five encryption algorithms, each with a minimum practical security of `100` bits, and ten multi-layer encryption combinations (called cipher cascades). VeraCrypt allows encryption with any of its fifteen algorithms. This creates a randomization opportunity for added security: a script placed on the voting machine would frustrate an attacker's attempt to guess the encryption algorithm as a step toward decrypting the message.[5]

The security of this system can be calculated as $\exp(\kappa + \min\{\kappa(l' - 2)/2), n(l' - 2)/l'\})$, where l = the number of rounds in the multiple-encryption cipher cascade, κ = the

[4] https://www.veracrypt.fr/code/VeraCrypt/ [5] For an example implementation on the *Windows* operating system, see here: https://github.com/JP3L/SAVI

security-bit strength of each encryption (we assume here that $\kappa \geq 100$), and where $exp(t) = 2^t$, and where $l' = 2\lceil l/2 \rceil$ is the smallest even integer greater than or equal to l, for all $l \geq 1$. [21] For our system, where $l = 1$ to 3 depending on which cipher cascade is randomly selected, the security will range between κ and $\kappa + min\{\kappa, n/2\}$, which is between 100 and $100 + 100 = 200$ bits. That said, a cipher cascade is not reliably independent and we should not assume a full 200 bits of security for the system. Instead we consider the cipher cascade as a hedge against the possibility of a cryptographic flaw in one of the algorithms. To calculate this security in practical application such as our proposal, we assume that E() and E()' are two different block ciphers each with a 100-bit key and propose the combined security for two general ciphers is effectively 101 bits, as per the person–in–the–middle attack [51].

D. Step Four: Check-out

Vote logging takes place at the virtual check–out desk. The desk confirms the voter received the email generated from Step Three. The desk also compares the voter's email receipt of `HASH2` with the desk's email receipt of `HASH2`. This biometric signature can be captured as a verified voice or image excerpt from the phone or video feed as part of a simultaneous second-channel support.

E. Step Five: Vote Tally

During this step, the system verifies the record of `HASH2` with the check-out desk, decrypts the *encrypted ballot*, calculates the hash of the unencrypted *marked ballot* as `HASH1`, and relays `HASH1` to the next step for vote certification.

F. Step Six: Vote Certification

Vote certification is an emergent security property of the SAVVI system. Once the vote has been decrypted and tallied, voters receive a certification notice confirming their ballot has been counted, along with public, anonymous access to the published `HASH1`. This notice includes instructions to verify their vote by matching hashes with the email they received at Step Five. One can verify their vote by searching for their ballot hash on a public website. A statistical verification can also assure integrity; non–matches strive to reveal issues before the tally is certified.

V. Security Considerations

Multiplying the probability of uncoupled things gives the probability that both will happen at the same time. It is the diversity of the sources of security and the use of them together–not only each of them individually–that increases security in a multi-factor authentication (MFA). The SAVVI protocol further synchronizes two MFA events, check–in, and check–out to achieve an enhanced security effect. We do not claim full independence of these security systems, but we propose that these individual components will work together in our architecture to defeat multiple types of threat.

The attack surface of our system includes many individual components, including the operating system on the virtual voting machine (VVM), the hosting platform for the VVM, the builders and operators of those machines and platforms, the local machine of the user, and so on. This enhances a standing web application by assuming that no single system will be completely secure for any longer than a few minutes. We instead propose a resilient architecture that presents a narrow temporal attack surface for the actual voting machine.

Technique	Bits of security	Threat Addressed
MFA1	58	Spoofing
TLS 1.3	128	Person-in-the-Middle (session)
S/MIME	100	Person-in-the-Middle (email)
Cipher cascade	101	Person-in-the-Middle (ballot)
MFA2	58	Spoofing

TABLE I: Each of the techniques proposed here provides protection against a specific security threat in the electronic marking and submission of a ballot. The authentication problem known as Spoofing remains the most vulnerable to attack, with 58 bits of security.

To further analyze the security of SAVVI, we consider five realistic and high–impact attack models: denial of service, spoofing, phishing, shoulder surfing, and person–in–the–middle. These attacks might prevent votes from being cast or recorded properly and we describe here the vulnerabilities relative to each type of attack and consider the potential for SAVVI to mitigate those vulnerabilities.

A. Denial of Service

Typical concerns surrounding electronic ballot returns include Denial of Service (DOS) attacks. These attacks usually bring down a service by overloading it. This was especially important when everyone voted on Election Day. Much of the country has early voting, with ballots sent to them as far ahead as 60 days so DOS attacks may be less of a concern. Nevertheless, communications could be disrupted by malicious parties. Denial of service attacks are a significant threat to email-based voting systems. Attackers could flood election email servers with large amounts of illegitimate traffic. This could prevent emails from arriving, or make it difficult for officials to distinguish valid ballots. SAVVI's email server could defeat such an email flood by white-listing only emails sent from the dynamically–provisioned Virtual Voting Machine, and including the white-list access control list entry as part of the container's provisioning process. Once the ballot is cast and the voter checks-out, the email server would remove the white–listed entry.

B. Spoofing

Spoofing attackers present the credentials of another to act like them. Multi–factor authentication in the voter registration phone call and again at the phone call for check-in can defend against such an attack. The voter identification log using biometric signatures facilitates detection and provides a potential remediation of such fraudulent voting attempts. In particular, it is plausible that this three-pronged system could provide adequate assurance that there is an actual person

interacting with the check-in desk and that it is the same person who previously registered to vote with PIN1. SAVVI requires synchronized authentication in voter registration and check–in/check–out, which is likely to detect a spoof attempt. Furthermore, the system might recognize the voice-print as other than the voter and could route the attacker to a honeypot where the attacker demonstrates their attempted fraud and thereby could contribute to future security improvements to the elections infrastructure. Geo–locating the fraudulent caller might also facilitate arrest.

C. Phishing

Phishing attacks use malicious links or attachments in an email to steal a person's credentials or compromise their device. The procedures described in this document create a unique voting email accessible only through a secure virtual container. These systems become live once the voter verifies with an election official for one-time use. Though an attacker might guess the set of all potential email addresses and spray phishing emails across them, the secure virtual container could purge all emails from the inbox upon loading the email client. This is built into the Application Programming Interface of popular email clients.[6] This provides a severely restricted attack surface within the email address and client provisioned within the Virtual Voting Machine that is probably not susceptible to phishing.

D. Shoulder Surfing

A person physically collocated with the voter could view or coerce actions. That said, SAVVI voters may be better able to control the timing, location, and potential for interlopers to reduce this concern in the safety of their own homes. SAVVI's process of check-in includes opportunities for uncovering and also discouraging coercion. The check in process can explicitly require affirmation that the voter is alone and voting privately.

E. Person-in-the-Middle

Person–in–the–Middle (PIM) and eavesdropping are potential threats in internet communications and especially threaten unencrypted communications. Marked ballots show how an individual voted, and may sometimes contain sensitive personal information about the voter. Anyone with access to the infrastructure could read or even modify email messages. In particular, email servers often store messages for a short period of time before passing them on to the next server or to the intended recipient. System operators for these servers could intercept or modify emailed ballots. It is difficult for election officials to identify ballots that have been modified in-transit. Also, emailed ballots are at risk before and after they are sent to election officials. The process of validating a voter, then giving them a new email or other ballot delivery channel is designed to detect bad actors and PIM attacks. Additionally, the provisioning of a secure virtual container would automate the encryption of the webmail channel (TLS 1.3), the marked ballot that is attached to the email channel (cipher cascade), and the email message itself (S/MIME). These three layers of security are complimentary and probably produce a cumulative security benefit. However, it is important to remember that a theoretical security bit-strength can be obviated by a side channel attack such as an endpoint compromise. The dynamic provisioning of a secure virtual container that contains a one-time generated voting machine, though, reduces this attack surface to a brief window of opportunity that makes it unlikely for an external attacker to penetrate. Also, the diversity of voter–owned end point devices makes it impracticable for attackers to create a wholesale disruption. Finally, the hash digests allow election officials to verify the vote was both cast as intended and recorded as cast. The NASEM authors' concerns about a single compromised layer in the application, operating system, BIOS, microprocessor, disk drive firmware, or across the internet reflect points of vulnerability to a PIM attack. The defense–in–depth approach described here provides multiple layers of encryption and dynamic provisioning of Virtual Voting Machines to reduce attack surface and reduce the potential exposure to PIM.

[6] For example, gmail, as in https://developers.google.com/gmail/api/guides.

VI. Conclusion

The difficulty of protecting privacy while establishing accuracy and integrity has made electronic voting a special concern for security experts. Integrity, accuracy, and privacy are all problematic with paper ballots too. Usable security for UOCAVA and disabled voters can be achieved without requiring them to become security experts themselves.

This paper came out of an exercise in contemplating what might be done for voters who could not present a paper ballot to their jurisdiction while preserving privacy and independence. Throughout 2020, several states were grappling with how to support ADA requirements. Disability advocates invited proposing a secure, accessible voting infrastructure. We started by considering blind voters, who generally cannot fill out paper ballots independently and privately. Some physically disabled voters have the same problem. Many UOCAVA voters were sending ballots by regular, unsecured email. These issues inspired devising the Secure, Accessible, Virtual Voting Infrastructure (SAVVI) framework to allow election officials to implement a secure ballot delivery system without significant new procurement or programming activity.

SAVVI uses available technology that can be implemented by a voting jurisdiction with relatively standard technical competence. Advanced Encryption Standard, Transport Layer Security, and Secure Multi–purpose internet Mail Extension represent available and open–source components that can work together to protect information in transit. SAVVI automates the preparation and orchestration of these tools to provide increased usability and security. It also verifies the voter and their ballot transfer using multi–factor authentication. The system represents a practical application of strong authentication, secure email, and encrypted attachments to achieve better usability than traditional mail–in paper ballots.

Potential implementations of this system–and future research–should consider usability impacts of specialized

CAPTCHAS, biometrics, and secure Virtual Voting Machines. These all require careful analysis and consideration that are beyond the scope of the current paper.

Detection is a critical part of judging security problems. A second idea lies in our suggestion of honeypot voting systems to discover and even entrap attempts to defraud the system.

Secure Accessible Virtual Voting Infrastructure is a flexible architecture that can accept evolving security tools that can be used for each part of the process described in Figure 2. Experiments testing these ideas through pilot voting demonstrations will provide deeper support for the approaches.

VII. Acknowledgements

This research was funded in part by ongoing activities in the Eaton Cybersecurity SAFE lab at Rochester Institute of Technology's ESL Global Cybersecurity Institute. We would like to thank the attorneys at Brown, Goldstein, and Levy for introducing us and encouraging us to help make voting safer, more secure, and more accessible to homebound voters with disabilities. J.M. Pelletier would also like to acknowledge the ongoing support he receives from the *Ordo Praedicatorum*.

References

[1] National vulnerability database. https://nvd.nist.gov/vuln/search/results?form_type=Basic, Accessed 15 October, 2020 with the exact-match search term *TLS 1.3*.

[2] The braille literacy crisis in america, Mar 2009.

[3] Schumer releases survey suggesting ballots of one in four troops deployed overseas went uncounted in '08 election, may 2009.

[4] *Recommendations to improve accessibility for absentee voting among recently injured service members*. Task 7 final technical report edition, 2012.

[5] Clemson university: Research alliance for accessible voting, 2014.

[6] Electoral fraud, Feb 2021.

[7] Rhode island general laws title 17. elections 17-19-8.1. ballots for voters who are blind, visually impaired or disabled, Feb 2021.

[8] Voting accessibility, Feb 2021. https://www.eac.gov/election-officials/voting-accessibility.

[9] Ben Adida. Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.

[10] Furkan Alaca and Paul C Van Oorschot. Device fingerprinting for augmenting web authentication: classification and analysis of methods. In *ACM ACSAC '16: Proceedings of the 32nd annual conference on computer security applications*, pages 289–301, 2016.

[11] Stephen Ansolabehere and Charles Stewart III. Residual votes attributable to technology. *The Journal of Politics*, 67(2):365–389, 2005.

[12] Adam Badawy, Emilio Ferrara, and Kristina Lerman. Analyzing the digital traces of political manipulation: The 2016 russian interference twitter campaign. In *2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, pages 258–265. IEEE, 2018.

[13] Elaine Barker. Recommendation for key management: Part 1 – general, May 2020.

[14] J Benaloh. Electionguard preliminary specification v0. 85. *GitHub. https://github.com/microsoft/electionguard/wiki/Informal/ElectionGuardSpecificationV0*, 85:62–70.

[15] Karthikeyan Bhargavan, Bruno Blanchet, and Nadim Kobeissi. Verified models and reference implementations for the tls 1.3 standard candidate. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 483–502. IEEE, 2017.

[16] Karthikeyan Bhargavan and Gaëtan Leurent. Transcript collision attacks: Breaking authentication in tls, ike, and ssh. 2016.

[17] Shuki Bruck, David Jefferson, and Ronald L Rivest. A modular voting architecture (" frogs"). 2001.

[18] Rahul Chatterjee, M Sadegh Riazi, Tanmoy Chowdhury, Emanuela Marasco, Farinaz Koushanfar, and Ari Juels. Multisketches: Practical secure sketches using off-the-shelf biometric matching algorithms. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1171–1186, 2019.

[19] David Chaum, Richard T Carback III, Jeremy Clark, Chao Liu, Mahdi Nejadgholi, Bart Preneel, Alan T Sherman, Mario Yaksetig, and Filip Zagórski. Votexx: A remote voting system that is coercion resistant. *UMBC Student Collection*, 2020.

[20] Cas Cremers, Marko Horvat, Jonathan Hoyland, Sam Scott, and Thyla van der Merwe. A comprehensive symbolic analysis of tls 1.3. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1773–1788, 2017.

[21] Yuanxi Dai, Jooyoung Lee, Bart Mennink, and John Steinberger. The security of multiple encryption in the ideal cipher model. In *Annual Cryptology Conference*, pages 20–38. Springer, 2014.

[22] Fiebig, Tobias, Krissler, and Jan. Security impact of high resolution smartphone cameras. *Workshop on Offensive Technologies (WOOT)*, Aug, 2014.

[23] Fraunhofer Institute for Secure Information Technology (SIT). Security evaluation of veracrypt, 2020.

[24] 24 October 2003 Friday. Diebold memos disclose florida 2000 e-voting fraud, 2003.

[25] Dan Gillette, Shama Hoque, and Edwin Selker. Improving write-in candidate text entry for audio-only voting interfaces. *Journal on Technology & Persons with Disabilities*, 2014.

[26] Jonathan Goler, Edwin Selker, and Lorin Wilde. Augmenting voting interfaces to improve accessibility and performance. *Proceedings CHI 06*, 38(4):1113–1136, Apr 2009.

[27] Marta Gomez-Barrero and Javier Galbally. Reversing the irreversible: A survey on inverse biometrics. *Computers & Security*, 90:101700, 2020.

[28] Carol N Gorman. Dods policies, procedures, and practices for information security management of covered systems (redacted). Technical report, Department of Defense Inspector General Alexandria United States, 2016.

[29] Cole J. Harvey. Changes in the menu of manipulation: Electoral fraud, ballot stuffing, and voter pressure in the 2011 russian election. *Electoral Studies*, 41:105–117, Mar 2016.

[30] Tagg Henderson. Blindness: total loss of vision is rare, Apr 2012.

[31] Astead W. Herndon. Georgia voting begins amid accusations of voter suppression, Oct 2018.

[32] Tom Intorcio. Absentee voting an 'obstacle course' for military and overseas citzens. In *National Conference of State Legislatures*. NCSL, 2009.

[33] Douglas W. Jones. A brief illustrated history of voting, 2003.

[34] Jan Krissler. Jan krissler the biometrics acker. *Workshop on Offensive Technologies (WOOT)*, Nov 12, 2018.

[35] Chris Larabe. Blind, visually impaired mass.voters have a variety of options. *The Daily Free Press*, Oct 29, 2020.

[36] Stephanie Lecci. Talking voting machines, other equipment ensure voters with disabilities can cast ballots, Mar 2016.

[37] Carlo Meijer and Bernard Van Gastel. Self-encrypting deception: weaknesses in the encryption of solid state drives. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 72–87. IEEE, 2019.

[38] Hendrik Meutzner, Santosh Gupta, Viet-Hung Nguyen, Thorsten Holz, and Dorothea Kolossa. Toward improved audio captchas based on auditory perception and language understanding. *ACM Transactions on Privacy and Security (TOPS)*, 19(4):1–31, 2016.

[39] Murat Moran, James Heather, and Steve Schneider. Automated anonymity verification of the threeballot voting system. In *International Conference on Integrated Formal Methods*, pages 94–108. Springer, 2013.

[40] Engineering National Academies of Sciences, Medicine, et al. *Securing the Vote: Protecting American Democracy*. National Academies Press, 2018.

[41] Federal Voting Assistance Program. The uniformed and overseas citizen absentee voting act overview, Feb 2021.

[42] Andrew Regenscheid and Nelson Hastings. *A Threat Analysis on UOCAVA Voting Systems*. US Department of Commerce, National Institute of Standards and Technology, 2008.

[43] Eric Rescorla and Tim Dierks. The transport layer security (tls) protocol version 1.3. 2018.

[44] Scott Ruoti, Jeff Andersen, Daniel Zappala, and Kent Seamons. Why johnny still, still can't encrypt: Evaluating the usability of a modern pgp client. *arXiv preprint arXiv:1510.08555*, 2015.

[45] J Schaad, B Ramsdell, and S Turner. Secure/multipurpose internet mail extensions (s/mime) version 4.0 message specification. Technical report, RFC 8551, April 2019. https://doi. org/10.17487/RFC8551. https://rfc-editor . . . , 2019.

[46] Ted Selker, Dan , Gillette, Shama Hoque, Kate Liu, Minhy Pham, and Mike Vroomen. 'research in accessible voting report, Jul 2014.

[47] Ted Selker. The technology of access: Allowing people of age to vote for themselves. *McGeorge Law Review*, 38, 2007.

[48] Steve Sheng, Levi Broderick, Colleen Alison Koranda, and Jeremy Hyland. Why johnny still can't encrypt: evaluating the usability of email encryption software. In *Symposium On Usable Privacy and Security*, pages 3–4. ACM, 2006.

[49] Captain Smith and Samuel F. Eleven california counties are late in sending absentee ballots for 2012 primary. *Law Review*, 1256, 2012.

[50] Michael A Specter, James Koppel, and Daniel Weitzner. The ballot is busted before the blockchain: A security analysis of voatz, the first internet voting application used in {US}. federal elections. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1535–1553, 2020.

[51] Bing Sun. Provable security evaluation of block ciphers against demirci-selçuk's meet-in-the-middle attack. *IEEE Transactions on Information Theory*, 67(7):4838–4844, 2021.

[52] Haydar Teymourlouei and Vareva Harris. Effective methods to monitor it infrastructure security for small business. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 7–13. IEEE, 2019.

[53] Zhaobin Wang, Jing Yang, and Ying Zhu. Review of ear biometrics. *Archives of Computational Methods in Engineering*, pages 1–32, 2019.

[54] Alma Whitten and J Doug Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *USENIX Security Symposium*, volume 348, pages 169–184, 1999.

[55] Matthew Wills. Ballot position: It matters, Aug 2016.

[56] Carter Yand. Presidency hinges on tiny bits of paper. Nov 12, 2000.

[57] Yilin Yang, Yan Wang, Yingying Chen, and Chen Wang. Echolock: Towards low-effort mobile user identification leveraging structure-borne echos. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 772–783, 2020.

[58] Hongbo Yu, Yonglin Hao, and Dongxia Bai. Evaluate the security margins of sha-512, sha-256 and dha-256 against the boomerang attack. *Science China Information Sciences*, 59(5):052110, 2016.

[59] Filip Zagórski, Richard T Carback, David Chaum, Jeremy Clark, Aleksander Essex, and Poorvi L Vora. Remotegrity: Design and use of an end-to-end verifiable remote voting system. In *International Conference on Applied Cryptography and Network Security*, pages 441–457. Springer, 2013.