A Brief Technical Analysis of Claims Made by Some Researchers from MIT

[Version 1.1 - Last updated: July 13, 2020]

1. Overview

Some students/researchers at MIT in February 2020 published a report about Voatz. The Voatz security team evaluated each of their claims over multiple iterations to determine the status and relevance. Based on this evaluation, numerous errors and misrepresentations were detected in their report and some of these are highlighted in Section 2 of this document.

The approach used by the researchers was fundamentally flawed due to:

- Its sole reliance on the partial reverse engineering of a small, outdated portion of the system
- Usage of a series of incorrect assumptions and creation of purposefully biased data to satisfy their claims
- Lack of real world evidence provided for any of the claims
- Clear lack of experience and maturity in terms of how to build, test and deploy an election system
- Clear lack of understanding regarding blockchain implementations and their usage in election systems

It is also evident that the authors of the report were ideologically motivated to oppose any progress in the field of internet voting, to create fear using the media and lacked any semblance of objectivity that is required to analyze a complex system designed and built by award winning mobile security and telecommunication security experts. Their report should more appropriately be categorized as a '*point of view*' written to accomplish ideological objectives rather than a proper scientific analysis.

Voatz has conducted 70+ successful elections since its inception in 2015 and has never had a successful breach or compromise of its election systems in the field. All attempts to break into or tamper with the system have been intercepted and blocked successfully. Each one of our government elections has been audited and every marked oval has matched successfully.

In July 2020, Voatz became the 1st remote voting platform to successfully complete comprehensive system testing with a Federally-certified VSTL (Voting Systems Test Laboratory). This test covered several key aspects such as Security, Accessibility, Usability, Functionality and Accuracy. Additionally, Voatz continues to conduct frequent security assessments and is fully committed to a process of continuous improvement.

2. Errors & Misrepresentations

2.1 Blockchain 51% Attack

On page 10 of their report, the researchers make a series of incorrect assumptions about the Voatz blockchain implementation.

> Note that this is an optimistic analysis of the use of the blockchain in this system. It is unlikely that every interaction is stored via the blockchain, and their own documentation of the West Virginia election indicates that the verifying servers are split equally between Amazon AWS and Microsoft's Azure — indicating that their scheme is vulnerable to Microsoft or Amazon surreptitiously adding resources and executing a 51% attack, or performing a selfish mining attack that requires only 1/3 of the compute [26].

The Voatz implementation doesn't use a Proof-of-Work mining mechanism and is not susceptible to a traditional 51% attack. Rather, the Voatz implementation utilizes a modified PBFT (Practical Byzantine Fault Tolerant) algorithm that requires a near 100% consensus from diverse entities (including the election management bodies and independent auditors) thereby making our approach far more resilient as compared to other approaches.

Furthermore, the claim that Microsoft or Amazon could surreptitiously add resources to the network is incorrect. Neither entity has any such capability. Addition of any new nodes to the network involves a complex approval process and only designated trusted administrators are permitted to make any changes to the network configuration at this time.

In a nutshell, their claims here are baseless and totally ignorant of how the Voatz blockchain infrastructure continues to evolve based on the feedback from the various pilot election programs.

2.2 Speculative Commentary about Voter Verified Receipts

On page 13 of their report, the researchers admit their lack of understanding about the receipt process like many other parts of the system.

> understand these tradeoffs, and without further information, a full analysis of these receipts is not possible.

Upon a successful submission of a mobile ballot, each pilot voter receives an out-of-band confirmation receipt. The current receipt process includes the following safeguards:

a) The receipt is password protected using a credential available only to the voter.
b) The receipt is digitally signed by Voatz.
c) The receipt includes hidden watermarks to detect receipt tampering.
d) The jurisdiction receives an anonymized copy of the receipt to facilitate a pre/post election audit process.
e) The receipt is sent from an email address familiar to the all the pilot voters and uses a trusted, dedicated system that utilizes industry best practices such as DKIM, SPF and DMARC.
f) Training is provided to voters on how to verify their receipts.
g) All the pilot jurisdictions have a cure process in place to handle any complaints, issues reported by the voters.

Based on the feedback collected from pilot voters thus far, more than 50% of pilot voters are checking their receipts diligently and feedback has been highly satisfactory.

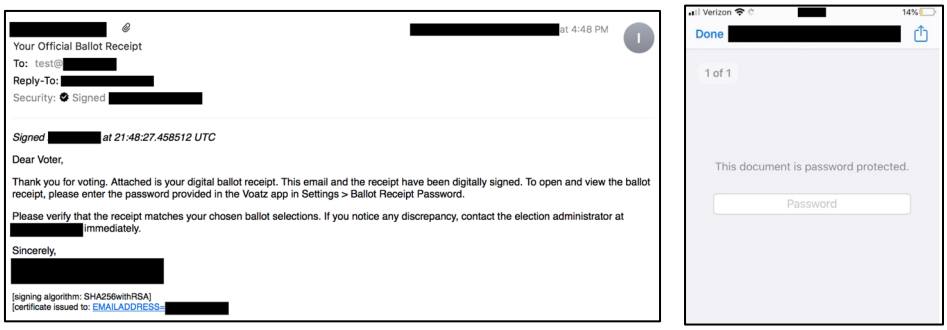


Fig 2.2.1: Sample ballot receipt email

The entire process is described in a detailed blog post title "*How Do You Know That Your Vote Counts?*" published in October 2019.

2.3 Incorrect Analysis of Handshake Mechanism

On page 5 of their report, the researchers admit their lack of understanding about the handshake process and yet proceed to make an incorrect assertion:

> 5. Out of the 100 public keys sent by the Server, the App selects the 57th pubkey (PK_S), and finishes the ECDH handshake to create the ECDH shared key SK_{ecdh}. Finally, it decrypts and parses the AES-GCM parameters(SK_{aes}, N, T).

During the handshake process wherein public keys are exchanged between the device and server, both sides pick a key at **random** - the hard coded index was removed long ago. Also, the purpose of this process is to exchange keys securely and not simply obfuscation.

2.4 Dubious Concerns Around Jumio, Crashlytics, Location & Privacy

On pages 5, 8 and 12, the researchers make various speculative comments around our integration with Jumio [G], a well-known identity proofing services provider.

> language. Identifying data is provided to Jumio and Voatz, and, to the best of our knowledge, Voatz makes no representations to its users about how long such information is retained, stored, or if it is shared beyond a general privacy policy that does not explicitly discuss Jumio. Worse, if Jumio were to prove truly malicious, it is possible it could refuse to validate particular users at all. Furthermore, we note that the app requests permissions to read the user's GPS upon first login, though we have not identified what exactly the app does with this information.

Voatz was the first remote ballot marking system to incorporate strict identity proofing as a security mechanism. Any online voting system that doesn't incorporate remote identity proofing cannot be considered viable in today's world.

i. Voatz uses the services provided by Jumio in selected jurisdictions only for **one** stage of our identity verification process.
    a. The determination made by any external service is not treated as final and every pilot voter's identity is manually verified before activation of any mobile ballots.
    b. The depth of other verification processes built in to the system would detect any malicious activity on the part of Jumio. While this could present scaling

challenges in the future, it is a very robust process that has guaranteed the accurate verification of each and every one of our 1000+ pilot voters so far.

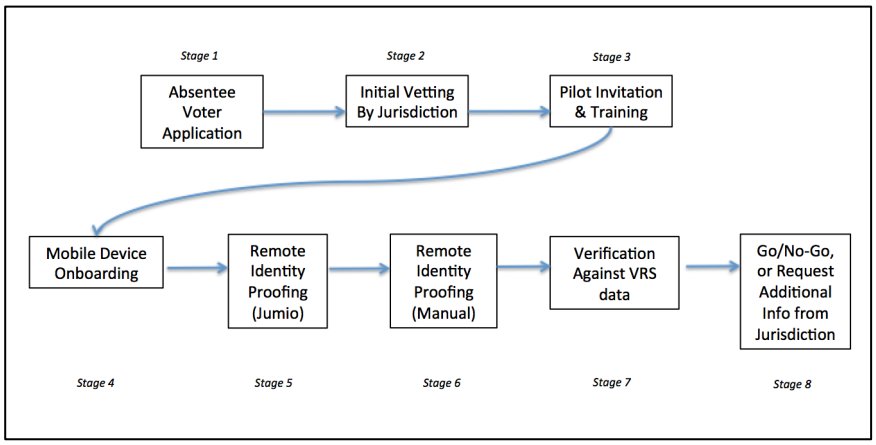


Fig 2.4.1: Multi-stage Identity Proofing Process

ii. All identity documents and photographs provided by the voters are deleted once the verification is completed (~usually within 24 hours). Any stored data even if temporary is always encrypted at rest.

iii. All voters are informed about this process as part of the pilot invitation and using the training material.

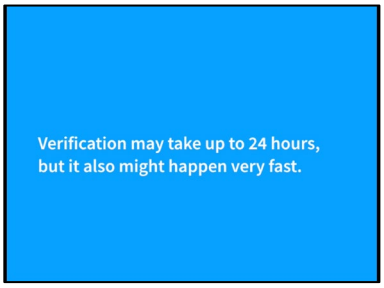


Fig 2.4.2: Screen grab from Voatz training video

iv. Voatz's use of Jumio and Crashlytics is documented as per industry standard best practices in the 'Licenses' section of the mobile applications.

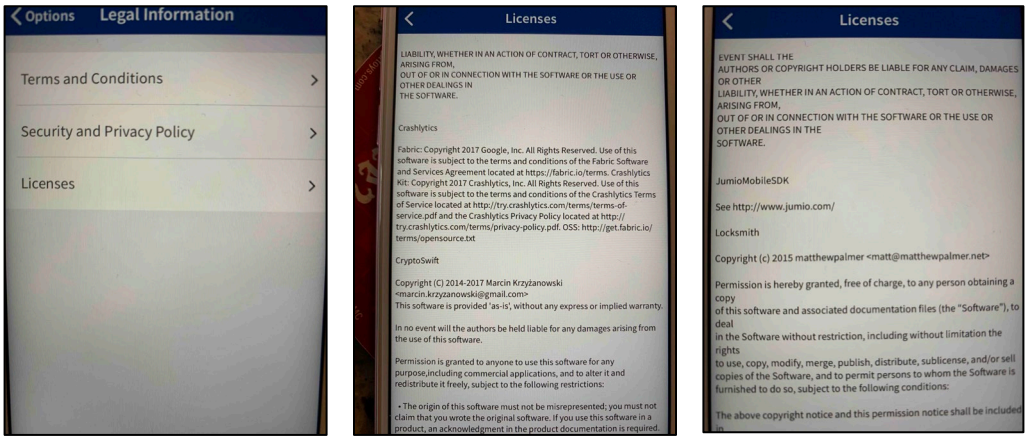


Fig 2.4.3: Screenshots from Voatz -> Options -> Licenses

v. Voatz's T&Cs [H] and privacy policy clearly specify the need for identity data along with a mechanism for users to confirm that their private data has been deleted.

2. Registration. In order to use certain parts of the Service, you may be required to provide us with your first name, last name, email address, telephone number, provide a photograph of your drivers license or other form of identification, take a selfie, create a password and register with us. We may also request additional information from you, including but not limited to demographic information. You represent and warrant to us that you will provide us with

Fig 2.4.4: Section from Voatz T&Cs

1. You can request that any Personal Information stored by us to be deleted at any time by contacting us at privacy@voatz.com.

Fig 2.4.5: Section from Voatz Privacy Policy

vi. Location preferences are under the user's full control and if enabled, are only used to provide official Election Day related information to voters who request it. Location details are never publicly shared with any 3rd party or used for any marketing or geo-targeting activities.

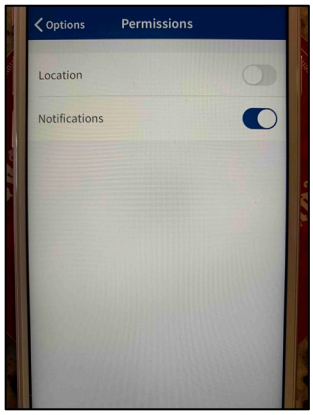


Fig 2.4.6: Screenshot from Voatz -> Options - > Permissions

vii. Mobile voting is an optional voting method for all pilot participants and no voter is ever forced to use this method.

2.5 Concerns Around Coercion

On page 13 of the report, the researchers make some incorrect and irresponsible claims about the Voatz application during their commentary about the risk of coercion.

Susceptibility to Coercion: As mentioned in 4.2, the app never requires the voter to re-enter their PIN at log-in after registration, and does not appear to show the user if a ballot has been re-voted or spoiled.

This indicates that the app leaves users vulnerable to coercion attacks. Consider a voter asleep or otherwise incapacitated. Assuming the attacker has physical access to the device and user, and that the device is unlockable via the user's fingerprint, an attacker would easily have the ability to cast a vote on behalf of the user. This threat model is very relevant in the case of intimate partner abuse [23,45].

The Voatz application **requires** the user to re-authenticate using their fingerprint, face picture or PIN at multiple stages including at each login, whenever the application goes into the background, prior to ballot submission, etc.

It is well known that there is no absolute technology-based guarantee against coercion, vote buying/selling, etc. regardless of the method of voting (in-person at your precinct, using mail-in ballots, email, facsimile or mobile voting). Legislation and enforcement are the best safeguards against coercion and vote buying/selling.

2.6 Risk of Side-loading & Unsupported Devices

On page 13 of the report, the researchers make some incorrect claims regarding side-loading and unsupported devices.

Risks of Sideloaded Malware & Unsupported Devices
Voatz's security requires that the app only be available on certain devices, in particular modern phones with up-to-date operating systems. They implement this via app store preferences; the Google Play Store will only allow certain device models to install the app, and will not make the app visible if the device does not meet Voatz's criteria.

This enables attackers to trick users of unsupported devices into installing an app containing malware by establishing a legitimate-looking website with information about how to vote and directing the reader to install a malicious version of Voatz's app. This is not a hypothetical concern — after the popular game Fortnite was released outside the Play Store to avoid Google's fees, malware authors tricked many naive users using very similar tactics [18].

Firstly, the claim that device restrictions are implemented merely via Play Store preferences is incorrect and possibly demonstrates their ignorance or lack of experience in publishing real world election software products.

Voatz requires users to have certain minimum device specifications from a security and compatibility perspective. These include the minimum Android OS version levels and these restrictions are built into the application as part of the build process and are not dependent on any Play Store preferences [K].

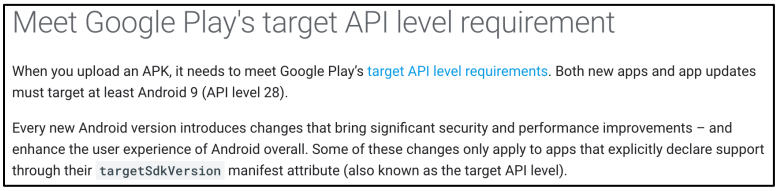


Fig 2.6.1: Screenshot of Android Build Targets from [K]

Secondly, the claim about tricking voters into downloading a malicious version of the application is needlessly alarmist and doesn't taking into the series of well-defined procedures which have been put in place for the conduct of these pilot programs. These include:

a) Pilot participation is by invitation only.
b) Each invite includes the link to a unique landing page created by each participating pilot jurisdiction. This includes links to the app stores, training videos, FAQ, etc.

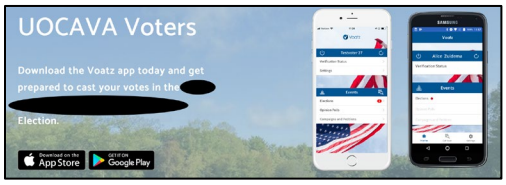


Fig 2.6.2: Landing Page Sample for Voatz Pilots

c) Voters are instructed to follow the links on the landing page to download the apps.
d) Merely downloading an application from a store without an official invitation from a participating jurisdiction will not give you any kind of access to an election.
e) Every pilot user has to go through a multi-stage verification process as discussed in Section 6.4
f) Voatz deploys licensing checks, side-loading prevention mechanisms and out of band verification methods that would detect and disrupt a Fortnite-style attempt. Moreover, the voter would never receive an authentic receipt and would raise an alarm instantly.

This blog post details how our threat detection works in the real world.

2.7 Incorrect Information About The 2016 Utah Convention

On page 1 of their report, the researchers make another incorrect claim that Voatz was used at the 2016 Utah GOP Convention. That year, the party used a solution provided by a different company. This is evidenced by public information available here.

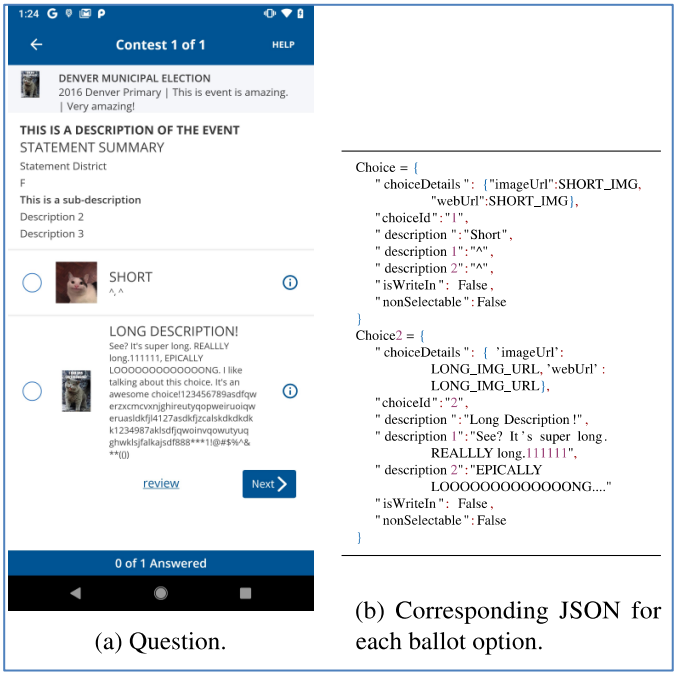
2.8 Speculation About End-to-End Vote Encryption

On page 3 of the report, the researchers speculate about end-to-end vote encryption. On the Voatz platform, the voter's ballot remains encrypted all the way through its active lifecycle – i.e. upon submission on the mobile device, transmission over the Internet, arrival at the server side infrastructure and persistent storage. An anonymized canonical representation is stored on the blockchain and is used to print the paper ballots for tabulation. Only authorized election officials have permissions to decrypt the digital lockbox for printing the fully marked paper ballots for tabulation. Please see this blog post for the overall flow.

2.9 Using Purposefully Flawed Data To Depict A Side-Channel Attack

In the section 5.3 titled '*Network Adversary*' of their report, the researchers make claims about exploiting a side-channel attack. However, they conveniently skip the part about the basic flaws in their purported claim – if you make a hypothesis, then create biased data to support that hypothesis, you cannot simply claim that your initial hypothesis was accurate.

Flaw-1: The deliberately created a completely unrealistic ballot design – one that wouldn't pass even the most basic of ballot design rules. Such a ballot design would easily fail the server side JSON validations each time and would not be accepted by the server. The researchers provide zero evidence on how they would bypass this server side validation.



(a) Question. (b) Corresponding JSON for each ballot option.

Fig 2.9.1: Flawed and invalid ballot design used by researchers.

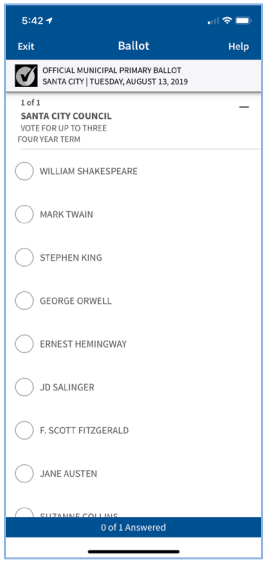


Fig 2.9.2: Well Formed Ballot Design

Fig 2.9.3: Well Formed JSON

Flaw-2: If the researchers had used a more recent and/or authorized version of the Android application, it would have been apparent that the system already supported cryptographic metadata padding as an additional mechanism to defeat such an attack. See these logs from our test runs that refute the claims made by the researchers.

GET	/us/i/US@2x.png		403	453	HTML	png	403 Forbidden
POST	/v1/868654bc51d042f69d9734a107ed0e48	✓	400	875	text		
POST	/v1/c4169049d00947df8ad61250c0896d7a	✓	200	1076	JSON		
POST	/v1/5ed78d8625854f6183a884c73ed07581	✓	200	831	text		
POST	/v1/868654bc51d042f69d9734a107ed0e48	✓	400	875	text		
POST	/v1/c4169049d00947df8ad61250c0896d7a	✓	200	1076	JSON		
POST	/v1/5ed78d8625854f6183a884c73ed07581	✓	200	831	text		
GET	/us/i/US@2x.png		403	453	HTML	png	403 Forbidden

Fig 2.9.4: Log snapshot showing similar network data packet sizes that contradict the claims made by the researchers

Finally, the researchers fail to explain how they would actually collect this data in the real world. It would require them to determine ahead of time which voter is participating in our pilot program, detect their overseas or military locations, compromise or sniff their wireless transmission network (whether WiFi or cellular), collect packet data, reverse engineer packet data, compromise all the encryption protocols, build up the intelligence by repeating this process for multiple voters until they can finally intercept the next voter during the act of voting. While theoretically anything is possible, practically speaking it is extremely difficult to pull off the above steps during the election window without remaining undetected and without triggering a whole series of trip wires available in the system.

2.10 Speculative Claims About On-Device Security Circumvention

In section 5.1.1 of the report, the authors make claims about being able to subvert device side defensive measures and remained undetected while providing no real evidence of how they would defeat the out-of-band communication. Simply disabling a small section of the trigger code using hooking or similar techniques will not accomplish anything as the layered trust cycle will be broken and the system will block the device from successfully submitting a valid ballot.

It is also evident from the report that the authors were unable to even detect the other device security measures and were likely fooled easily by the presence of canaries. A more advanced actor would likely have at least noted the existence of some of those measures.

The diagram below provides a high level representation of how mobile threat detection is implemented using the multichannel component communication in the Voatz system.

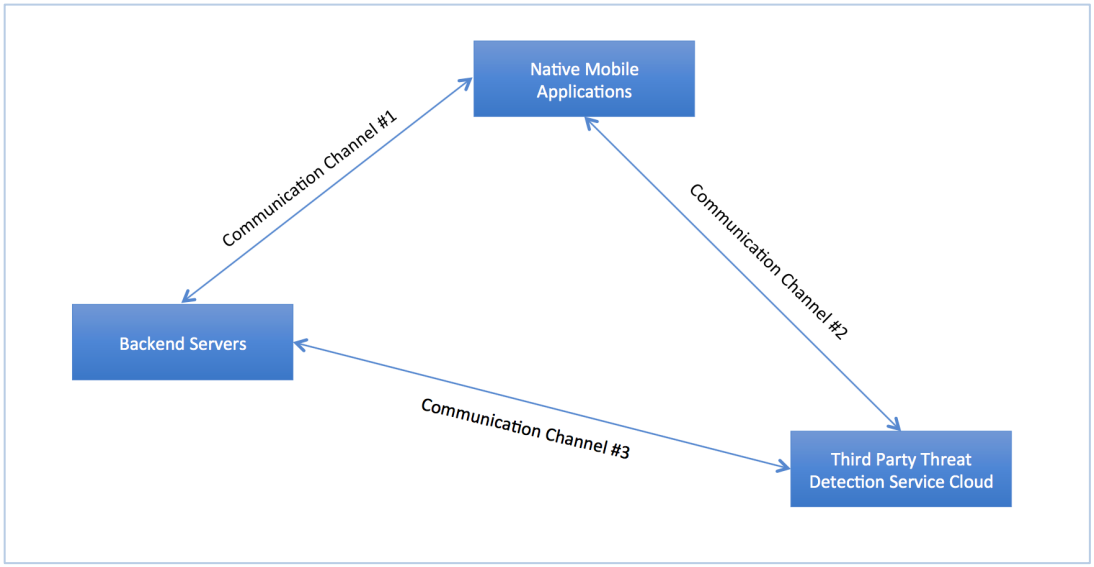


Fig 2.10.1: Multichannel component communication for mobile threat defense

All the three communication channels must be fully active and authenticated for a device to successfully transact with the system. Disruption or impersonation of any of the three channels will lead to the device losing the ability to submit a vote.

2.11 Speculative Claims About Server Compromise

In Section 5.2 of the report title '*Server Attacks*', the authors make speculative claims about server being capable of altering the user's vote or controlling the outcome of the election but provide no real world evidence to support their claims. Instead, they continue to rely on their incorrect assumptions regarding the protocols used by the Voatz system.

We highlight some of the main errors in their analysis:

a) The purported analysis of the device-to-server protocol is not only incomplete but also incorrect.
b) The hypothetical claim that the API server could execute an active MITM attack on its own operations and remain undetected is so outside the realm of possibility that it is mind-boggling. Even if an adversary somehow figured out a way to execute it in the real world, such an attempt would easily be discovered not only by the voter immediately (by verifying the authenticity and accuracy of the receipt) but also by the election officials (using their copy of the anonymized receipts) and independent auditors making it largely futile.
c) The claim that there is no public key authentication performed as part of the initial voter on-boarding is blatantly incorrect and once again demonstrates the severe lack of understanding about the nature of the Voatz system.

It is worth pointing out here that the authors attempted to break into the Voatz servers in December 2019 and miserably failed in their exercise. That abject failure was perhaps their motivation for making such fictitious and unethical claims. Refer to section 2.13 for more details on their failed attempts.

2.12 Flawed Claims About Stealth GUI Modification & Data Exfiltration

In Section 5.1.2 of the report, the authors claim that it is straightforward to modify the app and somehow fool the voter about this. No evidence is provided regarding the following speculative claims:

a) How they would accomplish this in the first place on a remote device
b) How they would bypass the series of application integrity and signature checks
c) How they would alter a vote and remain undetected
d) How they would enable the server to interact with an altered application

A claim is also made about potentially stealing user authentication data and somehow using that to impersonate a voter in order to interact with the Voatz servers. The Voatz system uses several layered protection mechanisms that require the voter interactions to originate on an un-tampered smartphone. Attempts to initiate such interactions outside of the device subsystem are summarily blocked as was evident in an unsuccessful attempt made by certain individuals to break into the system during the West Virginia elections in 2018.

2.13 Not Revealing Their Failed Attempts

In three separate documents (screenshots below), the researchers claimed that:

1. They did not attempt to connect to the Voatz servers
2. They used a version of the application from January 2020
3. The Voatz server was down at the time of their analysis

Snippet from the Initial document sent to Voatz via DHS/CISA on 1/29/20

Methodology

We performed a clean-room reverse-engineering of the latest Android app at time of analysis (v1.1.60). We never intentionally accessed any Voatz servers (they were down at the time of analysis). We performed our analysis in a cleanroom environment, and prevented the application from loading any content from Voatz.com, crashlytics, Jumio, or any other parts of Voatz's system. We disabled

Snippet from the paper sent to Voatz via NYT on 2/11/20

3 Experimental Methodology

As performing a security analysis against a running election server would raise a number of unacceptable legal and ethical concerns [46], we instead chose to perform all of our analyses in a "cleanroom" environment, connecting only to our own servers.[7] Special care was taken to ensure that our static and dynamic analysis techniques could never directly interfere with Voatz or any related services, and we went through great effort so that nothing was intentionally transmitted to Voatz's servers.[8]

To gain a better understanding of Voatz's infrastructure, we began by decompiling the most recent version of their Android[9] application as found on the Google Play Store as of January 1, 2020[10] and iteratively re-implemented a minimal server that performs all election processes visible from the app itself. This included every interaction involved in device registration, voter identification, and vote casting. We used two devices for our dynamic analysis and development: a Voatz-supported device running an up-to-date and fully patched version of Android, and a Voatz-unsupported device running the Lineage OS, both jailbroken with the Magisk framework [2].

[7]Unless otherwise specified, throughout this paper, any reference to communication we performed with "a server" or "the server" refers to our own server infrastructure.
[8]Indeed, at the time of analysis, Voatz's servers appeared to be down including for an unmodified app running on an supported and up-to-date device.
[9]We did no analysis on and make no claims about Voatz's iOS app.
[10]Version 1.1.60, SHA256 191927a013f6aae094c86392db4ecca825866ae62c6178589c02932563d142c1

Snippet from the public paper released by researchers on 2/13/20

concerns [53], we instead chose to perform all of our analyses in a "cleanroom" environment, connecting only to our own servers.[7] Special care was taken to ensure that our static and dynamic analysis techniques could never directly interfere with Voatz or any related services, and we went through great effort so that nothing was intentionally transmitted to Voatz's servers.[8]

To gain a better understanding of Voatz's infrastructure, we began by decompiling the most recent version of their Android[9] application as found on the Google Play Store as of January 1, 2020[10] and iteratively re-implemented a minimal server that performs election processes as visible from the app itself. This included interactions involved in device registration, voter identification, and vote casting. We used two devices for our dynamic analysis and development: a Voatz-supported Pixel 2 XL running Android 9, and a Voatz-unsupported Xiaomi Mi 4i running the Lineage OS with Android 8, both jailbroken with the Magisk framework [2].

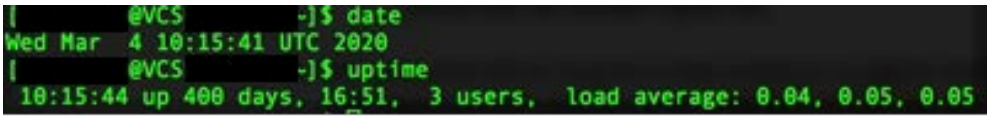
[7]Unless otherwise specified, throughout this paper, any reference to communication we performed with "a server" or "the server" refers to our own server infrastructure.
[8]Indeed, at the time of analysis, Voatz's servers appeared to be down including for an unmodified app running on an supported and up-to-date device.
[9]We did no analysis on and make no claims about Voatz's iOS app.
[10]Version 1.1.60, SHA256 191927a013f6aae094c86392db4ecca825866ae62c6178589c02932563d142c1

Each of the above claims is incorrect. Furthermore, throughout the public report the researchers write in at least **ten** different instances that they did not attempt to connect to the server.

Note the partial snapshot from our server logs indicating the **intentional** attempts made by the researchers to access our server between December 4, 2019 and December 10, 2019.

Fig 2.13.1: The above log snapshot is a partial listing indicating the IP addresses that were used. Additional evidence is available that ties the researchers' devices to the above attempts.

Below is a snapshot taken March 4, 2020 indicating that our server has been operational for the past 400 days, which far exceeds the time window of the researchers' analysis:

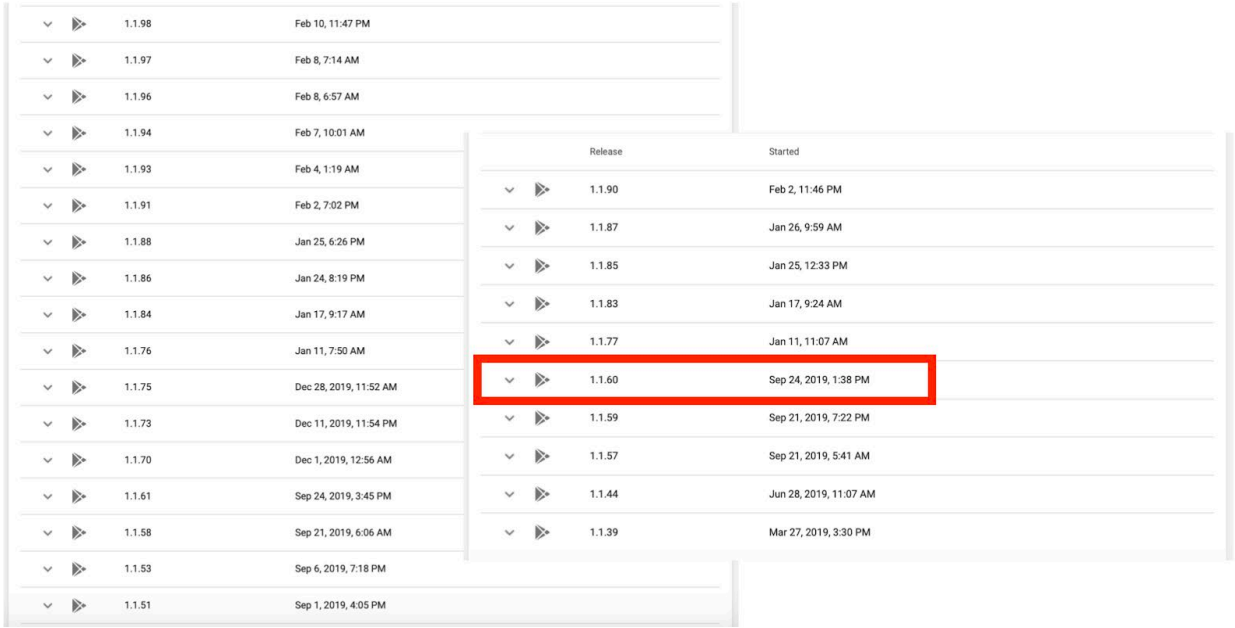


The above evidence contradicts the researchers' claim of not attempting to connect to the server, and also their claim that the "*server was down*". It also brings into question their claim around the timing of when the application was downloaded.

It should additionally be noted that the researchers did not reveal the models of the Android devices they used until the publishing of the public report. We suspect that this was an intentional tactic to prevent or delay detection and hide the fact that they used these devices in their failed attempts to break into the servers.

2.14 Incorrect and Invalid Version Numbers

The researchers claim to have used the Android application version from January 2020, yet the version number they cite (1.1.60) was a test version from September 2019 (see Google Play, Play Beta screenshots below). At the time the researchers initially reported their findings, the latest release was 1.1.88, a full 28 versions beyond 1.1.60. Additionally, the initial paper sent to the NYT reporter curiously cited a download date of the app as January 1, **2019** further bringing into question the legitimacy of the claims made in the report.



3. Security by Design

Security has been at the forefront of the Voatz solution architecture since the very beginning, including the company's earliest roots in winning first prize at the 2014 SXSW hackathon. The founders have always believed that security must sit at the heart of the company's design principles, and the technology's development closely follows this thesis.

The architecture of the Voatz solution sits on hardware and software designed to provide platform security. This security architecture spans all devices, servers, and networks used by the Voatz solution and incorporates device verification, real-time mobile threat detection and mitigation, remote identity proofing, distributed ledger-based data security, and a user-centric approach to end-to-end vote verification. Inherent in the Voatz culture is the philosophy of continuous improvement. Voatz management and shareholders require regular third-party evaluations, daily security testing, and constant enhancements in the presence of real-world threats, all aimed to supplement and continuously strengthen this architecture.

Core security tenets at the heart of the Voatz technology

All layers of the system enable an end-to-end process to ensure that all ballots are counted as intended and verified by the voter: (1) The platform produces a paper ballot for the jurisdiction to tabulate; (2) The system automatically sends the voter a password-protected, anonymized ballot receipt; and (3) The system uses a blockchain-based, tamper-resistant ledger to secure the aggregate vote and enable rigorous post-election audits.

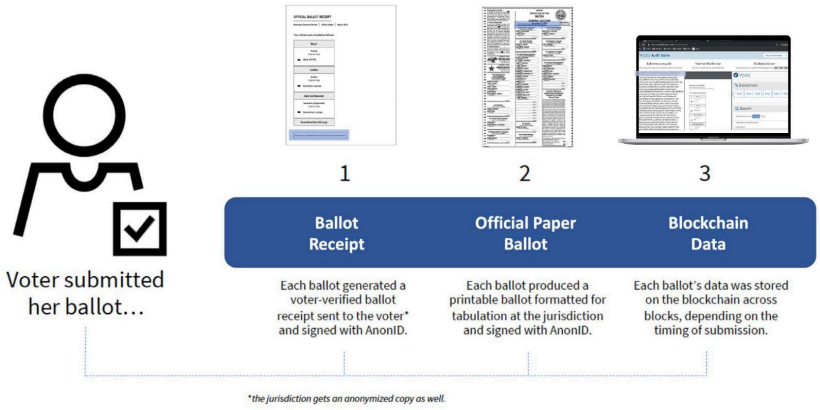


Diagram showing the multiple ballot trails generated by every mobile ballot submission, which facilitate a robust post-election audit

These checks and balances ensure that every single voter can verify their vote, that every election official can tabulate a paper ballot, and that the stakeholders involved in the election process can audit the integrity of the overall count without revealing voter identity.

3.1 How Mobile Voting Works

Voatz offers a new channel of voting within the traditional voting landscape. If a voter cannot vote in-person at the polls, early nor by mail, they now have an additional option: mobile voting.

Across its voting platform, Voatz leverages the latest smartphone security features and pairs them with multifactor authentication, including biometrics and facial recognition technology, to verify and validate the identity of the voter. The voter experience is seamless: 1) sign up to "vote mobile" on your absentee request form; 2) download the app and verify your identity; and 3) vote.

Privacy and security are inherent in the design of the Voatz solution. As soon as the voter's identity is verified, all identifying documents are deleted, and the voter's identity is anonymized.

The voter immediately receives a ballot receipt to verify their selections. A record of the marked ballot is written to the blockchain to assure data security and support the post-election audit process. Finally, a paper ballot is produced for the jurisdiction to tabulate alongside ballots received via the traditional voting methods.

At the close of every election, the jurisdiction has the option to host an open, public audit of all electronic ballot submissions. Any citizen can sign up to be an auditor. These auditors gain access to an audit portal with each mobile ballot submission's paper ballot, their anonymized ballot receipt, and the data on the blockchain. These audits are amongst the first in history to be fully open and transparent. This expansion of the audit process is part of an ongoing effort to widen a community of stakeholders, to build trust, and foster integrity in our critical infrastructure.

How Voatz works for a voter and how the system integrates with a jurisdiction

3.2 Election Industry Innovations Pioneered by Voatz

A comprehensive list of innovations in the Voatz platform includes:

- Native smartphone applications for highly accessible (ADA regulation compliant) remote ballot delivery, marking and return
- Remote identity proofing of voters using government-issued photo IDs paired with cutting-edge liveness and facial recognition technology
- Auditable, automated, fully-marked and formatted paper ballot generation for each mobile vote for tabulation
- Remote ranked-choice voting using an accessible interface

- Use of distributed ledger technology to secure the aggregate vote and enable post-election audits
- Real-time mobile threat detection and mitigation
- Visual and voter-centric approach to citizen-led post-election audits
- Coercion detection capabilities
- Public bug bounty programs and continuous third-party security assessments as input to Voatz's continuous improvement philosophy
- First smartphone based remote voting system to successfully complete comprehensive testing with a Federally Certified VSTL (Voting Systems Test Laboratory).
- First elections company to publicly release security/threat data (at the 2020 DefCon Voting Village).

3.3 Defense in Depth

The Voatz platform incorporates the security principle of *Defense in Depth.* There are multiple layers of security controls deployed across the platform, each approaching risk in different ways to build layers of defense around each asset.

Some key examples include our approach to remote identity proofing to determine voter eligibility, mobile device threat detection, and mitigation, botnet attack mitigation, etc.

3.4 A Model Based on Continuous Improvement

Voatz has been committed to the process of continuous improvement since its inception. The company conducted its very first white box, third party security assessment in 2016, and continues to pursue examinations of this kind since. In 2019, Voatz voluntarily submitted its platform to CISA (under the U.S. Department of Homeland Security) for an infrastructure security assessment (HUNT). In 2020, Voatz pursued a critical product evaluation (CPE) and continues to work with relevant private cybersecurity assessment firms for additional testing and evaluation.

Assessments of this kind are essential to the pursuit of continuous improvement as Voatz works to stay ahead of ever-evolving cyber threats. Any relevant issues detected during these audits are triaged and resolved promptly, or mitigated as needed. Recently, Voatz became the first mobile voting solution to successfully undergo a comprehensive assessment by a federally certified VSTL (Voting Systems Test Laboratory). Phase 1 was completed in May 2020, and Phase 2 was completed in July 2020.

Voatz has been the subject of intense media scrutiny and criticism by some security academics who have attempted to break into the system unsuccessfully on multiple occasions. Voatz remains the most battle-tested remote voting platform, has never had a successful security breach, nor experienced any voter fraud, and has thwarted every break-in attempt. In a recent election involving thousands of voters, the Voatz platform detected and prevented an unprecedented number of advanced mobile device threats in real-time, including insecure wireless networks, to fully ensure the integrity of the electoral process. Recently, Voatz become the first elections company to publicly release its security/threat data for independent analysis and feedback – an unprecedented feat in US election history.

4. About Voatz

4.1 Team & Advisors

The Voatz team includes experts spanning mobile security, high-performance SaaS, product design, election systems management and certifications, financial technology, and beyond. It is due to this unique blend of expertise that Voatz has managed to maintain and press forward with progress in the space.

The Voatz Advisory Board includes eminent professionals with sector expertise spanning elections, cybersecurity, nation-state threat mitigation, financial technology, politics, government, civic innovation, and business.

4.2 Investors & Awards

A committed group of investors backs Voatz's focus on next-generation technologies, blockchain, and civic innovation. The company is a graduate of both the Techstars Boston 2017 and MassChallenge Boston 2017 startup accelerator programs and has raised an aggregate of $9.2 million across two rounds of venture funding. Voatz is also the winner of several technical, civic innovation awards, including the MassChallenge 2017 Gold Award Winner, Microsoft Civic Innovation Award 2017, Election Center's Democracy Award (Denver County) 2019, Innovative Entrepreneurship in Blockchain Award (Public Sector Services) 2019, and was a finalist at the GSMA Mobile World Congress 2020 Awards for Best Mobile Innovation for Accessibility and Inclusion.