Phrase-Verified Voting: Verifiable Low-Tech Remote Boardroom Voting

(How We Voted on Tenure & Promotion Cases during the Pandemic)

Anonymous authors
Redacted Institutions

Abstract

We present a voter-verifiable remote voting system for small private elections assembled from common available technology. Easy to understand, and without using any complicated mathematical cryptography, the system enables each voter to verify that their ballot selection was included in the tally. We describe our system and experiences using it in fall 2020, in Tenure & Promotion Committees to vote remotely during the COVID-19 pandemic. Voters found the system easier to use, and providing greater privacy and outcome integrity, than the commonly used alternatives of paper ballots or voting by email.

In this *Phrase-Verified Voting* system, each voter fills out a form in the cloud with their vote V (YES, NO, ABSTAIN) and a passphrase R—two words entered by the voter that are memorable but do not identify the voter. The system generates a verification spreadsheet of the (R,V) pairs and tally of the votes, organized to help visualize how the votes add up. After the polls close, each voter verifies that this table lists their (R,V) pair and that the tally is computed correctly from these data.

consider suggestion to use random word generator

spreadsheet, table?

Because the system would allow a coercerto demand that their victim use a specified passphrase — it should not be used in applications where such malfeasence would be likely or would go undetected The system was well accepted, performed effectively for its intended purpose, and introduced users to the concept of voter-verified elections.

USENIX Symposium on Usable Privacy and Security (SOUPS) 2021.
August 8–10, 2021, Vancouver, B.C., Canada.

1 Introduction

The COVID-19 pandemic prompted most universities to carry out their operations remotely. This paper describes a voting system made for a university that had been voting in person with paper.

In this context, the chair of the T&P Committee in the Computer Science and Electrical Engineering (CSEE) Department approached us for a recommendation on how voting should take place remotely in fall 2020. The system needed to require few steps, be easy to use and compatible with the institutional constraints of the university. We devised, implemented, and fielded a new remote voter verifiable "boardroom" voting system, called *Phrase-Verified Voting* using google forms as its communication infrastructure. This paper describes the system and experiences using it in fall 2020 in the T&P committees of two separate departments.

We propose what turned out to be a simple and practical solution to the real problem of how to conduct elections remotely, for small elections that require voter privacy and outcome integrity—assurance that the tally correctly reflects the voter intentions. Policy decisions are often made by voters. In some public referenda, such as in the U.S. Congress, individual votes are public; other referenda, such as T&P voting described , require voter privacy.

We designed Phrase-Verified Voting for remote boardroom elections. A traditional boardroom election [4] is an election with a small number of voters who can all fit into one room where they can see and hear each other. Many important decisions take place in boardrooms, in-person and remotely, including about corporate strategies and board of directors actions. In this case the boardroom was a online video meeting.

A variety of voting options exist. Prior to the pandemic, the described P&T committees voted by paper ballots. By contrast, P&T committees in some other schools typically vote by voice without voter privacy[1].

In fall 2020, most P&T committees at UMBC voted by email, with members sending their ballot selections to a

[1] Private correspondence with Ronald Rivest.

trusted third party, such as the department secretary.

Desirable properties of voting systems include outcome integrity, voter privacy, ease of use, reliable operations, and simplicity. MIT's voice voting offers high outcome integrity but no voter privacy. UMBC's voting by email offers no voter verifiability and requires complete trust in a third party for voter privacy and outcome integrity. Helios [2] is a free open-source electronic voting system that offers voter verifiability with quite strong security properties, and was considered as an alternative for UMBC. Some UMBC administrators expressed concern that using Helios running on servers outside UMBC might violate UMBC policy, while installing Helios on UMBC servers would add complexity.

Low-tech solutions offer many advantages: they are easy to understand, implement, and use. Unlike Helios, Phrase-Verified Voting does not use any complicated mathematical cryptography, and does not require a PhD to understand. Nevertheless, Phrase-Verified Voting empowers each voter to verify that their vote was included in the tally.

In Phrase-Verified Voting, each voter fills out a Google form in which they enter their vote V (YES, NO, ABSTAIN) and a passphrase R—two words entered by the voter that are memorable but do not identify the voter. The system generates a verification spreadsheet of the (R,V) pairs and a tally of the votes, organized to help visualize how the votes add up. After the polls close, each voter verifies that this spreadsheet lists their (R,V) pair and that the tally is computed correctly from these data.

Phrase-Verified Voting worked effectively for P&T voting, and voters reacted favorably to it. Because Phrase-Verified Voting facilitates coercion—a coercer could demand that their victim use a specified passphrase—the system should not be used in applications requiring a high level of security. Because the stakeholders for P&T voting are known and accountable, Phrase-Verified Voting is appropriate for that application. Phrase-Verified Voting also provides a gentle way to introduce the concept of voter-verifiable elections.

[clarify trust assumptions in Rebecca]

Our contributions include: (1) A new remote voting system with voter privacy that is low-tech and voter verifiable, and (2) deployment of our system with two voter groups using it a total of 14 times, and assessment of its use through a voter survey.

2 Background and Previous Work

To the best of our knowledge, we are the first to propose a low-tech verifiable procedure for remote boardroom voting. Blanchard et al. [4] propose a low-tech verifiable procedure for traditional boardroom voting, and there are several systems for verifiable cryptographic remote boardroom voting—e.g., Javani and Sherman [11].

Hao and Ryan [9] survey modern electronic voting systems, including end-to-end (E2E) verifiable systems including Scantegrity II [5] and Helios [2, 3]. Chaum et al. of the VoteXX Project [8] designed a remote voting system that is coercion resistant.

3 Problem Specification and Adversarial Model

We seek a method by which members of the UMBC CSEE Department's P&T Committee can vote remotely. The method should not add significant effort, time, or expense to the process. The goals are to: (1) Assure that the outcome is accurate; (2) enable voters to verify the correctness of the outcome and that their votes are correctly recorded and tabulated; (3) provide ballot secrecy; (4) produce a list of people who voted and a list of eligible voters who did not vote; (5) require few steps or learning to use or administer; and (6) be acceptable to the UMBC administration [13].

The approximately 35 eligible voters are mainly the tenured faculty in the CSEE Department; each holds a PhD in computer science or a related field. Because the stakeholders are known and accountable, the method does not have to provide an extremely high level of security and privacy. Simplicity and usability are very important.

One to a few candidates might be voted on in each meeting. For each candidate, there are up to four referenda: teaching, service, research, and overall. Each meeting lasts approximately one hour.

Each ballot choice is one of YES, NO, ABSTAIN. Voting ABSTAIN is different from not voting. Voters may vote during a committee meeting, or, with permission, they may vote by absentee ballot before the meeting. All eligible voters not on sabbatical are required to vote.

The adversary could be anyone, including voters, staff, candidates, and administrators. The goals of the adversary are to modify the outcome or to learn how certain individuals voted. We assume that the adversary is covert in the sense that they do not wish to be caught. Undue influence—including voter coercion and vote buying—is out of scope. We are not very concerned about denial-of-service attacks, or discredidation attacks that aim to cast doubt on the validity of the outcome.

The system may use a trusted party (e.g., the department secretary, preferably from a different department), for trust involving privacy.

4 Overview of Solution

We now explain the main steps of phrase-verified voting, discuss alternatives, define our notation, and state the main principles underlying our design. Figure 2 illustrates the main steps of phrase-verified voting.

[explain figure]

For each referendum, the election authority (EA) configures a first Google Form with details pertaining to the specific

explain the figure. We already gave a one-par explanation in abstract and

```
Y:
1. disagree imperial
2. assume jockey
3. friendly,root
N:
1. presidential, shock
2. frank 99
.
A:
1. k b

.
Total
Yes: 3
No: 2
Abstain:1
```

Figure 1: Example of formatted audit data.

vote. Part of this configuration involves restricting access to users within the organization, and enabling one-time submission. The Google Forms service handles authentication of users and enforces a single submission, but does not not reveal the email identification (ID) of voters to the EA. The EA optionally provides a list of voter emails to facilitate automated distribution of the ballot access link, though the access link could be distributed through other means.

Each voter anonymously accesses fills out a Google form in which they enter their vote V (YES, NO, ABSTAIN) and a passphrase R (two words chosen by the voter that are memorable but do not identify the voter). The Forms service automatically generates a spreadsheet of the (R, V) pairs and a tally of the votes that the EA can access at any time. EA can choose to periodically announce the number of votes cast. The EA closes the vote by disabling further submissions.

After the polls closes, the EA downloads the vote spreadsheet. The EA must then perform the following formatting of the audit data: remove timestamps for privacy; sort the data by V and separate into three enumerated lists; and append a tally of V. A representative example of the formatted audit data is shown in Figure 1.

The formatted audit data is inserted into a second Google form. Unlike the ballot form, this second Google form is configured to automatically collect voters IDs. As a best practice, the form access link is distributed by an email embedded with a copy of the audit data contents for voters to keep in their records. Google Forms automates this when provided proper form settings and a ID (email) list.

Each voter uses the verification form access link to perform an identified verification that the formatted audit data lists their (R, V) pair and that the tally is computed correctly. Each voter is required to fill in the second Google form that automatically records their ID, how they voted (absentee ballot, at meeting, did not vote), whether they attended the meeting, and the results of their vote verification.

4.1 Alternatives

In making our recommendation, we also considered three competing alternatives:

1. (Baseline) With essentially no security, each voter emails their vote to a trusted party who tabulates them. Variation: email the votes to two separate trusted parties, each of whom tabulates them independently.

2. (Helios) Use the Helios [2, 3] verifiable online voting system, which is available through a web interface [1]. Helios is a free open-source electronic voting system that offers voter verifiability with quite strong security properties [6, 7, 10, 12].

3. (Public Vote) With high integrity and no privacy, each voter fills in a Google form that generates a public spreadsheet that lists, for each voter, their name and vote.

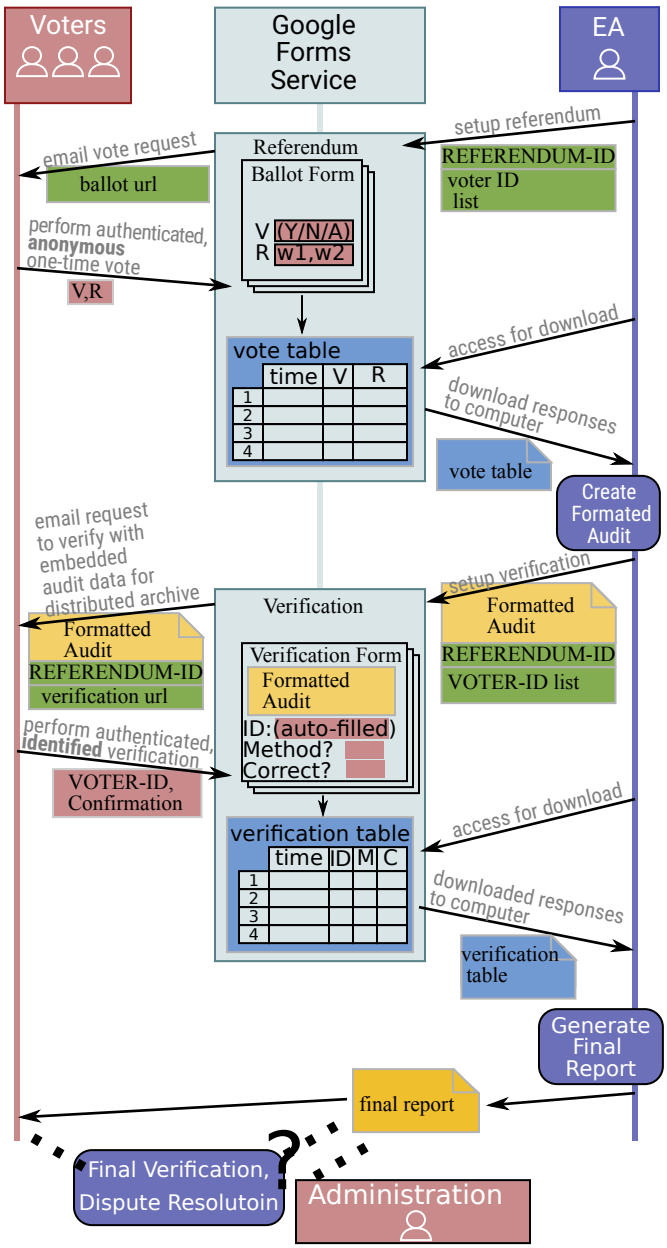
We feel that the university department ought to achieve higher outcome integrity than offered by Baseline, and greater voter privacy than offered by Public Vote. Public Vote may also violate university guidelines.

Although Helios provides considerable integrity and is fairly easy to use, we have three concerns with it. First, it is not clear if the UMBC administration would find it acceptable, given that the servers do not run on the UMBC network, though UMBC has a special relationship with Google that permits the use of Google servers for certain activities. Second, considerable time and effort might be required to run it at UMBC. Third, CSEE faculty have experience using and administering Google forms but not using and administering Helios. Nevertheless, Helios is our recommendation for those who seek greater outcome integrity and are willing to administer a more complicated system.

4.2 Notation and Terminology

We use the following notation and terminology.

- EA denotes the *Election Authority*.
- T_1 and T_2 denote two trusted parties.
- V denotes a vote (one of YES, NO, ABSTAIN).
- R denotes a passphrase, which is a pair of words selected by the voter.
- The *Adjudication Panel* includes T_1 and T_2 and others.
- A *referendum* is a specific question about one matter (e.g., research for the candidate).



[A protocol (ladder) diagram showing messages sent (from whom , to whom) voting form, verification sheet, verification form, audit data.] [Overview not intended to be comprehensive]

Figure 2: Main steps of phrase-verified voting.

- Each referendum has a REFERENDUM-ID, which uniquely identifies the referendum.
- A *committee meeting* is a meeting during which one or more referenda are voted on.
- A *ballot* is a Google form that includes a REFERENDUM-ID, referendum date, and place to enter an (R, V) pair.

4.3 High-Level Principles

We state six high-level principles and operational policies that enhance the voting process: (1) All voters (including absentee voters to the extent possible) must use the same voting procedure (i.e., fill out an identical but individual Google form privately). (2) All absentee votes are due at least one hour before the scheduled committee meeting at which a referendum is to take place. Each absentee voter must verify their vote immediately after voting. (3) Late votes will not be permitted. The Google voting form will not accept any votes after the close of voting. (4) Election outcomes are announced at the committee meeting, immediately after voting ends, by displaying Google spreadsheets. No preliminary results are announced before voting ends. (5) All voters (including absentee voters) are required to vote and to engage in the vote verification step. Each voter present at the meeting must verify their vote during the meeting. Each referendum is separately verified. (6) Later in the semester, using Docusign, each committee member signs the committee report, which includes the two spreadsheets produced by the voting and verification steps.

Principle (3) helps ensure that the voting process will terminate during the meeting and in a way that can be verified by the voters during the meeting.

Principle (5) enhances outcome integrity and facilitates the creation of a list of eligible voters who voted, and a list of eligible voters who did not vote, as required by UMBC policy.

5 The Protocols

[ATS has not written]

5.1 Voting

1. The EA announces all referenda and the dates and times of the committee meetings at which they will take place.
2. For each referendum, the EA announces the number of eligible voters and who they are by displaying a Google document, and the deadline by which voters may request permission to vote absentee.
3. The EA identifies trusted parties T1 and T2, who do not vote in any way in the tenure and promotion process, and who preferably are not associated with the CSEE Dept.

4. For each referendum, voters vote by filling out a simple Google voting form, in which they enter R, V. This form does not record the voter's identity.

5. Each voter selects a fresh randomly-chosen R on each voting day. The voter may reuse their R for all referenda held on the same day.

6. The Google voting form creates a spreadsheet of (R,V) pairs.

7. For each referendum, immediately after voting ends, the EA releases the (R,V) pairs as described below, and the vote tally. These voting data are made available to all voters and the Adjudication Panel. To facilitate verification of the tally, these data are listed as follows. The (R,V) pairs are arranged in three groups by the three possible votes (YES, NO, ABSTAIN). Within each group, the (R,V) pairs are sorted by R values. Within each group, each (R,V) pair is listed on a separate numbered line, with the first line of each group numbered 1.

8. After Step 7, each committee member must fill out a simple Google verification form. People attending the meeting immediately fill out the form during the meeting.

9. Immediately after the in-meeting verification step ends, the EA makes the verification results spreadsheet available to all voters and the Adjudication Panel.

10. Throughout the process, T2 receives the spreadsheets created by the Google forms at the same time as does T1. T2 also receives all communications from T1 to the voters. T2 verifies that the audit data (the two spreadsheets made available to the voters) are consistent with the data T2 received from the Google forms.

Absentee voting. (a) If an eligible voter cannot attend the meeting, they must send an email to Rebecca requesting to vote absentee and stating a reason. Permission to vote absentee will be granted only for the most compelling reasons. (b) To vote absentee, the voter will fill out the Google voting form at least one hour before the meeting. Immediately after voting absentee, the voter must fill out the absentee acknowledgment form. 4 (c) After the final audit data become available, the voter will be able to check that the audit data include their (R,V) pair and vote. Absentee voters will not fill out a verification form.

notes: To collect the names automatically on the verification form, each voter must authenticate themselves via Google. Although this step can create a small usability cost, it deters proxy voting. 2. Experts estimate that, with human-chosen word pairs, the chance is very small (less than 0.1%) that voters will select the same passphrase. Voters are welcome to use random word generators available on the web (e.g. https://randomwordgenerator.com/ or https://www.wordgenerator.net/random-wordgenerator.php). Doing so may unnecessarily complicate the voting experience and be superfluous, and duplicates do not necessarily compromise election integrity. 3. The proposed method is not intended to protect against undue influence (e.g., coercion).

FORMS:

Google Voting Form Ballot for referendum REFERENDUM-ID held on DATE. Enter two neutral words not associated with you; you might use this generator): Enter your vote: (checkbox selection:) ☐ YES ☐ NO ☐ ABSTAIN Google Verification Form for Voting at Meeting Verification of public voting data for referendum REFERENDUM-ID held on DATE. Name: (automatically recorded) Select one: (checkbox selection:) ☐ I attended the meeting and voted there. ☐ I attended the meeting but did not vote. ☐ I attended the meeting but voted absentee. Are your word pair and vote accurately shown? (checkbox selection:) ☐ YES ☐ NO Are the vote totals correctly shown? That is, is it true that: 5 the number of votes does not exceed the number of eligible voters, and the vote totals are correctly listed for YES, NO, and ABSTAIN ? (checkbox selection:) ☐ YES ☐ NO Google Acknowledgment Form for Absentee Voting Verification of public voting data for referendum REFERENDUM-ID held on DATE. Name: (automatically recorded) Select one: (checkbox selection:) ☐ I voted absentee. ☐ I did not vote.

5.2 Verification

5.3 Dispute Resolution

1. To make a valid claim that their vote is incorrectly listed, the voter must reveal their name and (R, V) values to the Adjudication Panel. The Adjudication panel will broadcast to the voters that someone (without identifying this person) has filed a dispute involving the (R,V) values. For each valid claim, the EA will correct the V in the audit record and correct the tallies accordingly. 2. Any other dispute will be adjudicated by the Adjudication Panel, which will deliberate remotely in a conference call. The Adjudication Panel should include members who are not associated with the department. 3. The Adjudication Panel will report the number of claims it received and, for each, a summary of the general nature of the claim and whether and how it was resolved.

6 Security Notes

We informally discuss the system's security properties with regard to outcome integrity, dispute resolution, discreditation, and voter privacy, .

First, we note that there are several points in the process where the system creates an indelible record and commitment as a result of sending emails and posting files on Google Drive.

In particular, voting and verification forms are sent by email, and the final audit data are posted on Google Drive.

Second, because the verification form (but not the voting form) requires the voter to be signed into their UMBC account, the system provides some degree of assurance and identification of who filled out the verification form.

Third, the system is "software independent" **??** in that any software fault that affects the tally can be noticed by voters.

6.1 Outcome Integrity

The system provides strong outcome integrity: Voters have opportunities to detect any manipulation of the tally. For example, because voters know the number of eligible voters and the number of people who voted, voters can detect if additional votes were inserted into the tally by inspecting the vote counts and by noticing too many (R,V) entries in the verification form.

Each voter can observe that the verification form lists their (R,V) pair. If the R values are distinct, each voter can detect any attack that changed their (R,V) pair. If exactly two voters pick the same R value, the only case where an adversary could change one vote without detection is when both voters also cast the same corresponding vote.

It is possible but highly unlikely that two voters pick the same R value. Experts estimate that, with human-chosen word pairs, the chance is very small—less than 0.1%) Experts estimate that, with human-chosen word pairs, the chance is very small (less than 0.1%—that voters will select the same passphrase []. Voters are welcome to use random word generators available on the web.[2] Doing so may unnecessarily complicate the voting experience and be superfluous, and duplicates do not necessarily compromise outcome integrity.

Although the situation for absentee voters is slightly different than for regular voters, absentee voters can verify their votes from the final audit data.

It is important that all voters participate in the verification process. If the adversary knew that a particular voter would not verify, and if the adversary could identify that (R,V) pair, the adversary could attempt to modify that pair without detection.

If the adversary compromises the voter's machine, the adversary could attempt to modify the voter's vote and display a compromised verification form to the voter. This attack risks detection if the voter checks the verification form or final audit data from a separate uncompromised machine.

6.2 Dispute Resolution and Discreditidation Attacks

We now consider what happens when a voter claims that their (R,V) pair is not present in the verification form. When only the V value is in dispute, the dispute resolution procedure provides an adequate resolution, albeit at the cost of the voter revealing their identity to the Adjudication Panel. Namely, the Adjudication Panel can adjust the tally by replacing the disputed (R,V) pair with the corrected one offered by the complainant and posting this action into the audit record.

If the adversary changes one or more R values, voters can detect such change, regardless of whether the adversary additionally modifies any V values. The system does not have capability to correct such anomalies. Relatedly, the system does not prevent a dishonest voter from claiming falsely that the verification form does not list their R value. Therefore, the system is vulnerable to a discreditation attack by dishonest voters who falsely claim their R is not present, and by adversaries who modify one or more R values.

6.3 Voter Privacy

The system offers reasonable privacy to honest voters, but none to dishonest voters. A dishonest voter can intentionally identify themself through their choice of R value, similar to the way a dishonest voter could write an identifying mark on a paper ballot. Unfortunately, this limitation facilitates undue influence via vote buying and coercion. For example, a coercer could demand that the victim vote with a particular (R,V) pair. Mitigating undue influence is the most daunting challenge of Internet voting [8].

Just because a R value might be someone's name or initials does not necessarily mean that the voter is the identified person. It is possible that a dishonest voter intentionally entered the R value in question for the purpose of making it appear that the identified person voted in a certain way.

An adversary who compromises a voter's machine could learn how the voter voted. Furthermore, although the voting form does not overtly identify the voter, it is possible that careful technical analysis of network traffic could identify the voter from an intercepted voting form.

There are fundamental limits on voter privacy resulting from the number and identity of regular and absentee voters. This limitation particularly affects absentee voters because there are typically few of them. Although the (R,V) pairs of the absentee voters are not identified to the entire electorate as such, these votes are automatically entered into a Google spreadsheet before the regular votes are entered. Therefore, anyone with access to this spreadsheet, including the EA, would likely be able to identify the absentee (R,V) pairs. If, for example, there were two absentee voters, and they both voted NO, then anyone who knows the absentee votes would know how each absentee voter voted.

7 Evaluation and user study

To evaluate the protocol proposed, and because we were answering a call for solutions, Phrase Verified Voting was used

[2] E.g., https://randomwordgenerator.com/ or https://www.wordgenerator.net/random-wordgenerator.php .

to handle the votes of Tenure and Promotion committees. A total of 26 referenda were held in two departments — Computer Science and Electrical Engineering (CSEE) and Information Systems (IS) — from October 5th to November 23rd 2020. Between 8 and 26 voters participated in each referendum, depending on the department and the set of eligible voters — some people having voting power on certain committees only. A follow-up survey was administered after those initial 26 referenda, and 17 users answered — from a set of at least 43 voters who used the system. The administrator for the CSEE votes also answered this survey.

7.1 Survey and results

7.2 Feedback received

from users and from admins

8 Discussion

8.1 Major Design Decisions

8.2 Usability

8.3 Decision Process

[talk about important issues in the process, beyond voting system]

example: negative votes on teaching (but not overall) vs. range voting

8.4 Limitations

8.5 Recommendations

8.6 Open Problems

some discussion points/ideas:

In discussion: talks about option of using harder system if someone is doing discreditation/denial of service

Enforcement of voting once is problematic (with Google forms, need to log in to limit votes to one)

Curtis's role starting stopping election

separate design from implementation

details: sabbatical voters (allowed to vote, not required to vote)

9 Conclusion

We have proposed and deployed a simple and easy-to-use system for private remote boardroom voting that is low tech and voter verifiable. It is useful to have a range of voting options for various applications, including ones with different tradeoffs among simplicity, ease of use, privacy, and outcome integrity. *Phrase-Verified Voting* offers a new option that emphasizes simplicity and ease of use, while still providing voters the ability to verify that their vote was correctly recorded and tabulated. Results from a user survey from two voter groups show that the system was well accepted.

Our system offers greater voter privacy and outcome integrity that does the commonly-used alternative of sending votes to a trusted party by email. Our system, however, does not provide as much privacy and integrity than does the more complex Helios system. Phrase-Verified Voting gently introducing voters to the idea of voter-verified systems; as such, it can serve as a stepping stone toward eventual adoption of more complex and secure choices.

Preliminary draft (January 31, 2021). To be submitted to SOUPS 2021. There is a 12-page limit (excluding refs and appendix), but shorter papers are encouraged. Registration is due Feb 18 and submissions due Feb 25. Notifications on may 21st (except for early reject).

References

[1] Helios: Trust the vote. https://heliosvoting.org/.

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

[3] Ben Adida, Olivier De Marneffe, Olivier Pereira, Jean-Jacques Quisquater, et al. Electing a university president using open-audit voting: Analysis of real-world use of helios. *EVT/WOTE*, 9(10), 2009.

[4] Enka Blanchard, Ted Selker, and Alan T. Sherman. Boardroom voting: Verifiable voting with ballot privacy using low-tech cryptography in a single room. https://arxiv.org/abs/2007.14916, 2020.

[5] Richard Carback, David Chaum, Jeremy Clark, John Conway, Aleksander Essex, Paul S Herrnson, Travis Mayberry, Stefan Popoveniuc, Ronald L Rivest, Emily Shen, Alan T Sherman, and Poorvi L Vora. Scantegrity II municipal election at Takoma Park: The first e2e binding governmental election with ballot privacy. In *Proceedings of USENIX Security*. USENIX Association, 2010.

[6] Anthony Cardillo and Aleksander Essex. The threat of SSL/TLS stripping to online voting. In *International Joint Conference on Electronic Voting*, pages 35–50. Springer, 2018. https://whisperlab.org/blog/2016/security-analysis-of-helios.

[7] Nicholas Chang-Fong and Aleksander Essex. The cloudier side of cryptographic end-to-end verifiable voting: a security analysis of Helios. In *Proceedings*

of the 32nd Annual Conference on Computer Security Applications, pages 324–335, 2016. https://www.researchgate.net/profile/David_Duenas-Cid/publication/327980266_Third_International_Joint_Conference_on_Electronic_Voting_E-Vote-ID_2018_TUT_Press_Proceedings/links/5bd99588299bf1124fafaba2/Third-International-Joint-Conference-on-Electronic-Voting-E-Vote-ID-2018-TUT-Press-Proceedings.pdf#page=337.

[8] David Chaum, Richard Carback, Mario Yaksetig, Jeremy Clark, Mahdi Nejadgholi, Alan T. Sherman, Chao Liu, Filip Zagórski, and Bart Preneel. Votexx Project. https://votexx.org/, 2020.

[9] Feng Hao and Peter YA Ryan. *Real-World Electronic Voting: Design, Analysis and Deployment*. CRC Press, 2016.

[10] Helios. Attacks and defense. https://documentation.heliosvoting.org/attacks-and-defenses, 2011.

[11] Farid Javani and Alan T Sherman. BVOT: Self-tallying boardroom voting with oblivious transfer. *arXiv preprint arXiv:2010.02421*, 2020. https://arxiv.org/abs/2010.02421.

[12] Ben Smyth. Ballot secrecy: Security definition, sufficient conditions, and analysis of Helios. Technical report, Cryptology ePrint Archive, Report 2015/942, 2018. https://eprint.iacr.org/2015/942.pdf.

[13] UMBC. Faculty Handbook. https://provost.umbc.edu/policies/faculty-handbook/.

A Sample Verification Form

B Survey Forms