VoteXX: A Remote Voting System that is Coercion Resistant

David Chaum,[1] Richard T. Carback III,[1] Jeremy Clark,[2]
Chao Liu,[3] Mahdi Nejadgholi,[2] Bart Preneel,[4]
Alan T. Sherman,[3] Mario Yaksetig,[1] Filip Zagórski[5]

October 29, 2020

1 Introduction

The *VoteXX Project* imagines a future in which citizens can vote on their smart phones with security and privacy. Such remote Internet voting offers numerous advantages: convenience, low cost, more accurate ballot marking, fast reporting of results, and improved usability and accessibility, including support for multiple languages and long ballots. This white paper introduces recent new approaches by the VoteXX Team that offer promising strategies to mitigate the daunting challenges of Internet voting, including *voter coercion*.

Although some countries have already forged ahead despite security and privacy vulnerabilities, some leading computer scientists—including MIT's Ronald Rivest [5]—admonished in 2010 that, given current limitations of technology, Internet voting is akin to "drunk driving."

Recent examples showing the limitations of Internet voting bear out this opinion. For example, Lewis et al. [8, 7, 6] show that in Switzerland's SwissPost-Scytl sVote voting system, a corrupt system could change valid

[1]xxnetwork, Los Angeles, CA, USA, david@chaum.com, rick@xx.network, mario@xx.network

[2]Concordia University, Canada, pulpspy@gmail.com, m_nejadg@encs.concordia.ca

[3]Cyber Defense Lab, University of Maryland, Baltimore County (UMBC), MD, USA, sherman@umbc.edu, chaoliu717@umbc.edu

[4]Katholieke Universiteit, Leuven, Belgium, bart.preneel@esat.kuleuven.be

[5]Wroclaw University of Science and Technology, Poland, filip.zagorski@gmail.com

votes into invalid votes. Teague [13] explains that the iVote system has the same weakness. Specter et al. [12] point out serious vulnerabilities in the Voatz blockchain system used in West Virginia, enabling adversaries to modify and expose votes cast from mobile phones.

Widely used mail-in ballots are not susceptible to malware attacks at the voter's end, but they too are vulnerable to undue influence. For example, a dishonest or coerced voter might take a video of them filling out, sealing in an envelope, and mailing the ballot. Similarly, a voter could also covertly take a video of them voting in a precinct—curiously, in some precincts such videos are even permitted. It should not be possible for a voter to prove how they voted.

A form of remote voting, vote-by-mail is here to stay. Our techniques will help improve the security and privacy of vote-by-mail.

Three daunting challenges make Internet voting difficult:

- The lack of a secure physical voting precinct facilitates undo influence, including vote selling and coercion.
- Malware on the voter's phone might undetectably modify votes and spy on voters.
- Determined adversaries might try to launch an online attack, including causing outages.

Voting systems must also be usable, accessible, low cost, easy to administer, and compatible with open-source data formats (to facilitate compatibility with other components of an election system). These additional properties, however, can be solved with existing techniques.

We believe these daunting challenges can be adequately solved and this white paper proposes techniques for solving them. Of particular note is our new approach to mitigating voter coercion by introducing the idea of "vote flipping." The appendix includes five slides that give a technical glimpse of how the VoteXX system works.

2 Background: Scantegrity and Remotegrity

In 2009 and 2011, *Scantegrity II* [2, 3] demonstrated an effective approach to dealing with possible malware. The City of Takoma Park, Maryland, used

the Scantegrity II voting system to elect the mayor and city council members. The 2009 election was the first time an end-to-end [10] secure voting system was used in a binding municipal election. Scantegrity II does not depend on the correct operation of any hardware or software: any change in the correct outcome could be detected by voters or anyone by inspecting the audit data posted on the web. Voters can perform such verifications, even though they cannot prove how they voted.

The 2011 Takoma Park election also demonstrated a related system, *Remotegrity* [14], for Internet voting. Voters reveal vote codes from scratch-off cards mailed to them, which they use in sending in their ballot over the Internet. If malware changed any votes, voters could detect such malfeasance, without being able to prove how they voted. Furthermore, the scratch-off card provides physical proof of malfeasance.

3 Mitigating Coercion

To date, researchers have attempted to address vote selling and coercion with revoting, fake credentials [4], or decoy ballots. We believe a new technique, "*vote flipping*," provides a much more robust and practical solution for Internet voting.

Mitigating voter coercion is one of the hardest challenges in Internet voting. The VoteXX team came up with a powerful new approach to mitigating coercion. They introduce a new step in voting—*vote flipping:* after the polls close, and before the results are announced, a voter has the option to "flip" (change or cancel) their vote. They do so using a "flip code" which they establish during registration.

The coercer can demand that the voter reveal all of the secret keys the voter possesses, including the flip code. Revealing the flip code to the coercer enables the coercer to flip the voter's vote, but revealing the flip code does not prevent the voter from flipping their vote.

This approach also introduces the new concept of *hedgehogs*, trusted associates of the voter. Optionally, each voter can enlist one or more hedgehogs and reveal the flip code to them. To flip their vote, it would be sufficient for the voter to signal at least one hedgehog. The signal could be subtle and covert—such as moving a specified potted plant on a balcony. Hedgehogs are useful when a coercer closely monitors the voter.

Vote flipping achieves the theoretical limit of what is possible in mitigat-

ing coercion: a coercer could simply threaten a voter not to vote or even register to vote. With flipping, undue influence is not possible because the coercer or vote buyer cannot be certain that the voter followed the coercer's demands.

Previous approaches to coercion depended on very strong assumptions. For example, some systems assume that the voter cannot be coerced during registration [4]. Others allow the voter to vote multiple times but assume that the voter can vote freely the last time. The vote-flipping approach does not have these limitations.

4 Mitigating Online Attacks

Mitigating DOS attacks is a separate type of challenge that is common to many other computer systems. A key defense would be to expand the voting period, say, to three weeks (results, however, would not be tabulated until the end of the voting period). Experience shows that it is extremely difficult for criminals to cause systems to fail continuously for such a long period of time. Furthermore, to cast a ballot, each voter would need to be able to connect to the election system only once, and they could attempt to do so in multiple ways. A long voting period would also provide the authorities with ample time to respond.

With code voting, the only publicly accessible part of the voting system simply collects vote codes. To cause an outage of such a system, an adversary would have to deny access to the many collection sites. As technologies for website defense are becoming more effective, and voting periods are lengthened—especially with remote voting—such attacks are unlikely to affect an election outcome.

5 Mitigating Malware

To thwart malware the system uses end-to-end verification [10]: using public audit data, voters verify that their ballots were cast, collected, and counted properly, without revealing how they voted. In this way, the system is "software-independent" [11]: without assuming trust in the correct operation of any hardware or software, voters can detect any error in the tally.

In particular, VoteXX uses a form of "remote code voting," pioneered at

Takoma Park in 2011, using *Remotegrity*. Voters received scratch-off cards with secret codes by mail, protecting and hiding voter choices from the computer they used to cast their ballots via the Internet.

6 Conclusion

Designs for remote voting fall into three types: (1) Ballots mailed to voters and returned by mail. (2) Ballots mailed to voters and returned by Internet. (3) Ballots provided to voters via the Internet and returned via the Internet.

The way we carry out vote-by-mail today highlights positive and negative aspects of remote voting. Vote-by-mail is efficient, reduces the risk of exposure to pathogens, and increases voter participation [1, 9]. Vote-by-mail is also vulnerable to vote selling and coercion. While presently there is little evidence of attacks, in the future, such threats could potentially become a serious problem as remote voting becomes more widespread.

However, using the techniques outlined in this white paper (and variations thereof), remote voting—including Internet voting—is possible to do safely. In particular, the idea of vote flipping can be used to mitigate voter coercion for each of the three types of remote voting.

Our ongoing plans include applying the techniques of VoteXX to make vote-by-mail as safe as possible and refining and implementing VoteXX as a coercion-resistant Internet voting system.

Acknowledgments

Sherman was supported in part by the National Science Foundation under SFS grant DGE-1753681, and by the U.S. Department of Defense under CySP grant H98230-19-1-0308.

References

[1] Camhi, T.: How oregon became the first state to vote by mail in a presidential election. OPB, https://www.opb.org/news/article/history-vote-by-mail-oregon-elections/ (October 2020)

[2] Carback, R.T., Chaum, D., Clark, J., Conway, J., Essex, A., Hernson, P.S., Mayberry, T., Popoveniuc, S., Rivest, R.L., Shen, E., Sherman, A.T., Vora, P.L.: Scantegrity II municipal election at Takoma Park: the first E2E binding governmental election with ballot privacy. In: USENIX Security Symposium (2010)

[3] Carback, R.T., Chaum, D., Clark, J., Essex, A., Mayberry, T., Popoveniuc, S., Rivest, R.L.: The scantegrity voting system and its use in the takoma park elections. Real-World Electronic Voting: Design, Analysis and Deployment p. 237 (2016)

[4] Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Towards Trustworthy Elections, pp. 37–63. Springer (2010)

[5] Kane, C.: Voting and verifiability. Vantage Magazine **7**(1) (2010), https://people.csail.mit.edu/rivest/pubs/Kan10.pdf

[6] Lewis, S.J., Pereira, O., Teague, V.: Trapdoor commitments in the swisspost e-voting shuffle proof. https://people.eng.unimelb.edu.au/vjteague/SwissVote.html

[7] Lewis, S.J., Pereira, O., Teague, V.: Addendum to how not to prove your election outcome. https://people.eng.unimelb.edu.au/vjteague/HowNotToProveElectionOutcomeAddendum.pdf (March 2019)

[8] Lewis, S.J., Pereira, O., Teague, V.: How not to prove your election outcome. https://people.eng.unimelb.edu.au/vjteague/HowNotToProveElectionOutcome.pdf (March 2019)

[9] Nichols, C.: Does voting by mail lead to higher turnout in red, blue and purple states? it's not that simple. Politifact, https://www.politifact.com/article/2020/may/18/does-voting-mail-lead-higher-turnout-red-blue-and- (May 2020)

[10] Popoveniuc, S., Kelsey, J., Regenscheid, A., Vora, P.: Performance requirements for end-to-end verifiable elections. In: Proceedings of the 2010 international conference on Electronic voting technology/workshop on trustworthy elections. pp. 1–16 (2010)

[11] Rivest, R.L.: On the notion of 'software independence'in voting systems. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **366**(1881), 3759–3767 (2008)

[12] Specter, M.A., Koppel, J., Weitzner, D.: The ballot is busted before the blockchain: A security analysis of voatz, the first internet voting application used in u.s. federal elections. https://internetpolicy.mit.edu/wp-content/uploads/2020/02/SecurityAnalysisOfVoatz_Public.pdf (2019)

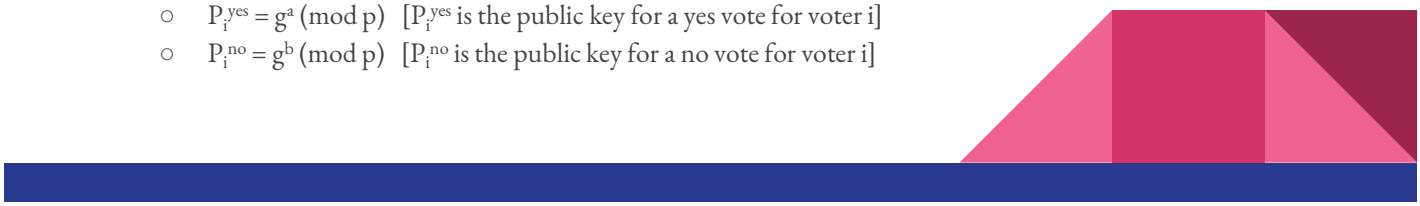
[13] Teague, V.: Faking an ivote decryption proof. https://people.eng.unimelb.edu.au/vjteague/iVoteDecryptionProofCheat.pdf (November 2019)

[14] Zagórski, F., Carback, R.T., Chaum, D., Clark, J., Essex, A., Vora, P.L.: Remotegrity: Design and use of an end-to-end verifiable remote voting system. In: International Conference on Applied Cryptography and Network Security. pp. 441–457. Springer (2013)

A Summary of How VoteXX Works

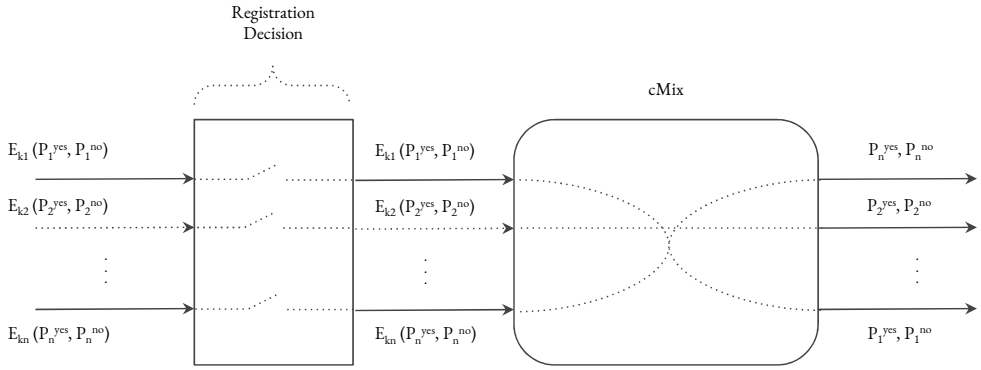
Assumptions

- Each user can share its keys with a hedgehog privately.
  - A hedgehog is an entity who interacts with the Election Authority MPC on behalf of a voter; hedgehogs help provide coercion resistance.
- The system has a roster of public keys of the voters (two per voter).
  - This roster does not link to any voter identity, which ensures ballot privacy.
- There exists a mechanism for each voter to register their two voting public keys. Each user has two Diffie-Hellman key pairs:
  - $P_i^{yes} = g^a \pmod p$ [P_i^{yes} is the public key for a yes vote for voter i]
  - $P_i^{no} = g^b \pmod p$ [P_i^{no} is the public key for a no vote for voter i]



Registration

Outputs permuted (P_i^{yes}, P_i^{no}) pairs

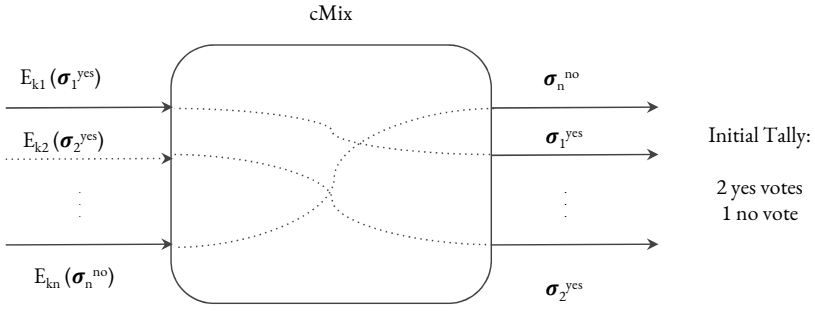


k_i is the symmetric key established between the mix network and user i.

Voting

Step 1: Construct Votes.
$\boldsymbol{\sigma}_1^{yes}$ is the signature from voter 1 using the secret key corresponding to P_1^{yes}; similarly, $\boldsymbol{\sigma}_1^{no}$ is the signature using the secret key corresponding to P_1^{no}.

Step 2: Output permuted votes.



Initial Tally:

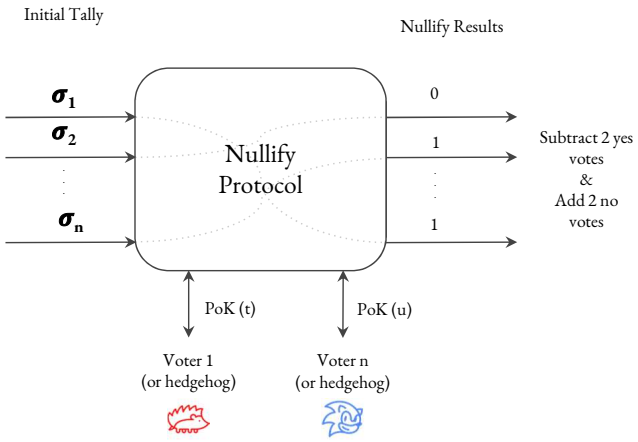
2 yes votes
1 no vote

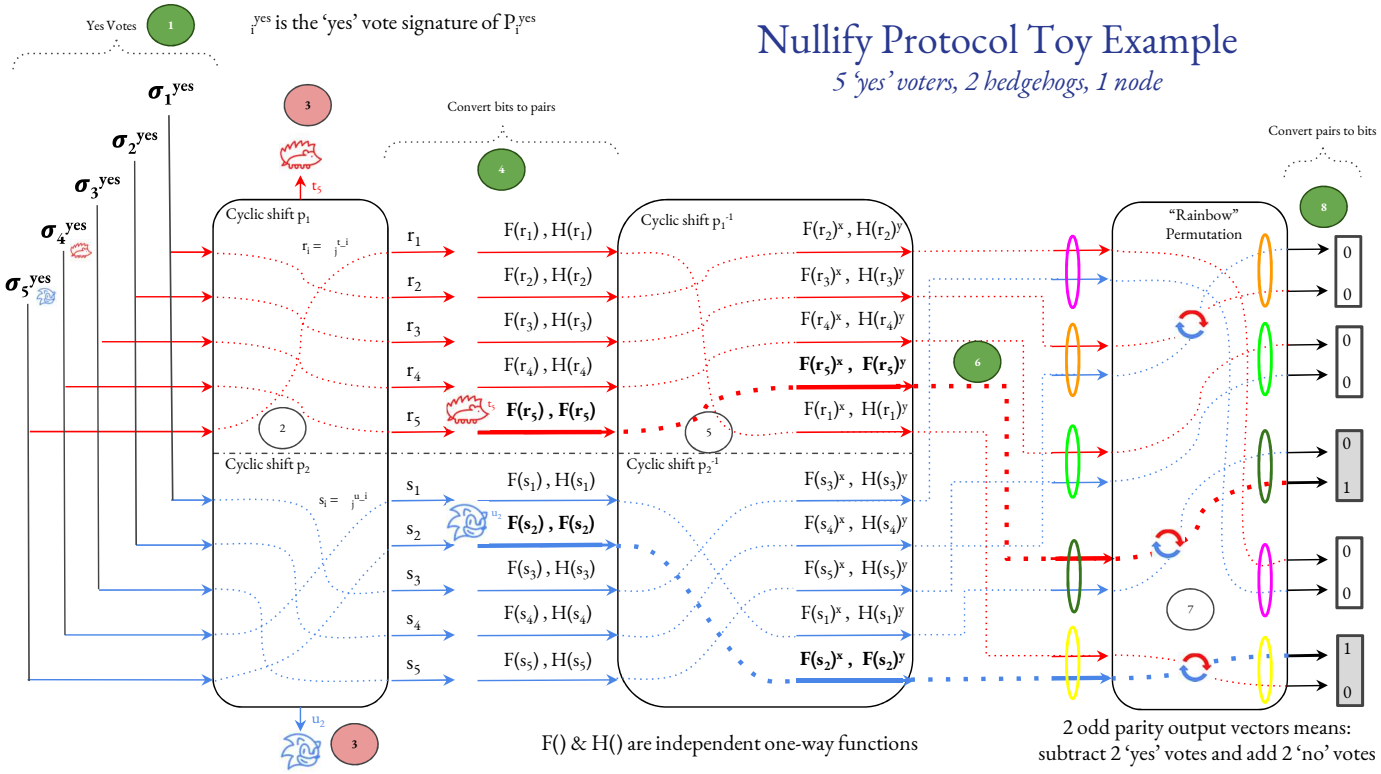
Nullify Stage Overview

Voters (or their hedgehogs) interact with the mix nodes to change or nullify their vote.

- **Input:**
  - List of cast votes (one batch for 'yes' & one batch for 'no')
  - Proof-of-Knowledge (PoK) of secret key by voters (or hedgehogs)

- **Output:**
  - Number of votes to be added/removed, which is calculated from the sum of the output values.




Nullify Protocol Toy Example
5 'yes' voters, 2 hedgehogs, 1 node
Yes Votes
$_i^{yes}$ is the 'yes' vote signature of P_i^{yes}
Convert bits to pairs
Cyclic shift p_1
Cyclic shift p_2
Cyclic shift p_1^{-1}
Cyclic shift p_2^{-1}
"Rainbow" Permutation
Convert pairs to bits
F() & H() are independent one-way functions
2 odd parity output vectors means:
subtract 2 'yes' votes and add 2 'no' votes