# The State of Cryptographic Voting Systems

Joshua Friede
A term project for CS:4980:0004 Spring 2020,
Electronic Voting at the University of Iowa

May 15, 2020

This version is intended for public distribution

# 1 Introduction

Is cryptography the solution to securing elections? Electronic voting systems have become mainstream due to their convenience and cost efficacy over paper, but concern exists about their ability to be hacked or rigged among other issues. Proponents of cryptographic systems suggest they can provide benefits to electronic voting systems such as ballot secrecy, verifiable audit trails, and resistance to hacking and fraud. Do these claims hold up, and are cryptographic voting systems practical to implement in real-world elections?

# 2 Scantegrity II

Scantegrity II is an election framework built by David Chaum that implements end-to-end verification on top of conventional scannable paper ballots. It creatively uses invisible ink to print confirmation codes inside the bubbles of a typical optical paper ballot that are revealed when a voter fills in their choices with a decoder pen. This, in combination with randomly ordering candidate names on ballots, allows encoded information to be made public for integrity purposes, while preserving the secret ballot. To generate these ballots, election officials must first create a seed to a pseudorandom number generator (PRNG) using a secret-sharing cryptographic scheme such that no single official can decode ballot information. Under this scheme, the officials combine their shares on a trusted machine to generate confirmation codes and the following tables, as described in Chaum's paper [1]:

> P : A table containing the confirmation codes in the order in which they were generated by the PRNG. Table P specifies the correspondence between confirmation codes and candidates on each ballot. Row i corresponds to ballot i and column j corresponds to candidate j, so that the confirmation code in position (i, j) is printed on ballot i within the bubble for candidate j. Table P is never published and is used to generate table Q.
> Q : A table in which the confirmation codes in each row of P have been pseudorandomly permuted. Thus, row i corresponds to ballot i, but each column does not correspond to a fixed candidate. The election officials commit to each confirmation code in table Q and publish these commitments on the election website.

R : A table in which each row i corresponds to an underlying confirmation code from Q. Each row contains a flag, which will be raised in the post-election posting phase if a vote is made for the underlying confirmation code, and two pseudorandom pointers — a "Q-pointer" specifying the position of a confirmation code in table Q and an "S-pointer" specifying the position of the same confirmation code in table S (described below). The election officials generate these pointers using the PRNG, commit to each Q-pointer and S-pointer, and publish these commitments on the election website. Essentially, table R provides two random shuffles of the confirmation codes and will be used in the audit process via randomized partial checking (Jakobsson).

S : A table in which each element corresponds to an underlying confirmation code. Each element is a flag, which will be raised if a vote is made for the underlying confirmation code. Each column j contains the confirmation codes for candidate j. Table S (initially empty) is published on the election website.

Paper ballots are printed with randomly ordered candidate names, and corresponding bubbles that when filled in by a voter's decoding pen, reveal a confirmation code. The voter copies this confirmation code onto their receipt. The voter then inserts their ballot into a conventional optical scanner, and a poll worker stamps their receipt. Voters may request two ballots, so they can vote using one, and reveal all the confirmation codes on the other to take home and audit at the close of the election. To tally the results, "The electronic ballot images from the scanner and table P are used to translate the votes into the confirmation codes which were revealed on the cast ballots. The election officials open the commitments in table Q to the confirmation codes that have been revealed to voters and flag the entries in tables R and S corresponding to those codes. Anyone can now compute the number of votes for each candidate as the sum of the number of flagged entries in the candidate's column in table S."[1] Voter's can end-to-end verify their ballot was counted correctly by matching the ballot number and confirmation codes on their receipt to the publicly available table Q. The election table made public online can be audited using random partial checking, where an election official releases either the Q-pointer or S-pointer for each element in R, decided randomly using a publicly verifiable pseudorandom coin flip. Using this information, anybody can verify that flags are mapped unchanged from table Q through table R to table S. Furthermore, voters that took home an auditing ballot can verify the posted confirmation codes match those on the paper audit ballot, and check that the confirmation codes are unique for each race. In short, Scantegrity II provides end-to-end voter verification that ballots are tallied as intended, on top of a familiar optical voting system.

This system was used once, in the November 3, 2009 Takoma Park, Maryland city election. This implementation of Scantegrity II was considered an overall success. While the system is highly complex and some voters reported confusion about the decoder pen and the hidden numbers, both voters and poll workers were satisfied in this real-world scenario.[2] While Scantegrity II protects against many threats in comparison to a traditional optical voting system, there are some potential areas of concern. First, by nature of ballots being uniquely identifiable, there is an increased risk to voter privacy. For example, an attacker could attempt to link ballot numbers to names by logging the IP address of voters as they check their receipts on the public website.[2] Moving on, an add-on to support write-in voting with the Scantegrity II system was implemented by creating a generic write-in bubble choice on ballots. However it relies on an election authority counting and transcribing written names, and therefor write-in votes won't reap all the verification benefits of Scantegrity II.[2]

# 3 SCRATCH & VOTE

A hybrid electronic/paper election framework called Scratch & Vote leverages homomorphic encryption to provide many guarantees. Scratch & Vote utilizes paper ballots divided such that one half contains checkboxes and the other contains the corresponding candidate labels in random order. The ballot also contains a tracking number and a barcode containing encrypted candidate order information. The decryption key is hidden under a scratch off area on the ballot. The voter fills in the checkbox, discards the half of the ballot containing the candidate names, then presents it to a poll worker. The poll worker tears off the scratch off area, scans ballot into an online repository, and returns to ballot to the voter as a receipt. The voter can verify their ballot was cast correctly by searching for their tracking number in the online repository. To verify the ballots themselves are not rigged, a voter can request an additional ballot to audit by scratching off the encryption key (which voids the ballot), decrypting the barcode, and matching it with the printed order of candidate names.[3] “To tally the votes, a computer reads each barcode to reveal the encrypted value that corresponds to the checkmark position on the voter’s ballot. Each of these encrypted values was created using homomorphic encryption. This property allows the computer to aggregate all of the encrypted values to arrive at one encrypted tally for each race. Since all of the ballots are available online, anyone can perform these same steps to verify that the election officials have correctly tallied the ballots. Finally, a quorum of election officials decrypts the single encrypted counter, and then posts a proof of correctness that any voter can verify.”[3] This framework is cost effective (using paper, barcodes, and an online repository) and provides many benefits. Scratch & Vote produces a voter-verified audit trail via paper receipts and their respective tracking numbers. Additionally, voters can verify if the ballots are rigged and if their ballots were cast correctly. Voters’ candidate choices remain secret, as neither the public online ballot repository nor receipts can be used to show which candidates correspond to the voter’s check marks. Finally, election tallies can be mathematically proven correct. On the downside, Scratch & Vote fails to support write-in voting. Also, the Scratch & Vote framework is a fairly complicated process involving tearing multiple sections off of the paper ballot at the polling place. Additionally, it is unreasonable to expect voters to be able to audit ballots, a task requires interpret barcodes and perform decryption. Moreover, the physical security of discarded secret keys and sensitive discarded sections of the ballot must be taken into account. To date, this system has not been used.

# 4 Polys (Blockchain)

Many startup companies are proposing using blockchain technology to carry out elections online. Voting over the internet provides many benefits in terms of convenience, cost efficiency and speed of tallying results, but it also brings many security concerns. It is common knowledge that any system connected to the internet is vulnerable to hacking, but blockchain voting system vendors make sweeping security claims. For example, blockchain voting system Polys claims that its system is impossible to hack or manipulate, and is impossible to expose voters’ candidate choices.[4]

Blockchain systems work by trusting the collective action of several observers to maintain an immutable public ledger. In the case of a blockchain voting system, the election observers, including the election commission and trusted independent organizations, would control these nodes. The main benefits of using a blockchain for voting are the tamper resistance of votes, transparency via the public ledger of votes, and potential to vote from personal devices. However, additional (and quite complicated) layers would need to be added to ensure ballot secrecy. To further evaluate such a system, consider Polys, a multipurpose blockchain voting system prototype unveiled by the Kaspersky Innovation Hub in February 2020. At the start of a Polys election, election observers (the election commission and trusted independent organizations) form a blockchain based on Ethereum. According to the whitepaper, the blockchain's contents are encrypted with a unified balloting key generated via a homomorphic secret sharing scheme such that each observer holds a share of the key. This method prevents independent observers from decrypting ballots. Moving onto voter authentication, Kaspersky Innovation Hub presents in a press release that regional election teams could use a special Polys Printer to print and distribute unique QR tokens to eligible voters.[5] These tokens can be used to authenticate into a mobile app or any Polys voting machine. The app or machine encrypts the contents of the vote via the unified balloting key and signs it with the voter's token, allowing the voter to later log into a web app using their token (QR code) to check if their vote was cast intact. The Access Control List Ethereum smart contract verifies the identity of the voter based on their token and prevents a user from casting multiple votes. To prevent voter coercion (for example, if a party coerces voter into sending screenshots of their ballot), this system could be modified to allow users to vote multiple times where only the most recent vote from each user is counted. Additionally, a zero-knowledge proof can be used by the system to check if the vote is spoiled. Because the ballots are encrypted using a homomorphic secret sharing scheme, the votes can be added while encrypted, and the final tally can be decrypted by combining a set number of observer's shares of the unified balloting key.[6] Blockchain voting systems such as Kaspersky's Polys provide many benefits, but at a closer look, it's grand claims of being unhackable and impossible to manipulate are questionable. For one, blockchains are inherently vulnerable to collusion, as explained by David Jefferson at the Verified Voting Foundation [7]:

> A [collusion] threat is present when a blockchain is used in elections. The co-owning organizations must reach consensus on each ballot to be stored in the blockchain, and the final set of ballots in the blockchain will be the basis for the final vote counts. But

> a majority of co-owners might agree on a fraudulent set of ballots leading to declaring the wrong winners. Alternatively, outsider attackers such as other nation states or foreign criminal organizations might penetrate the servers, injecting malicious software to create the same effect as collusion to rig the election remotely. The local Election Agency may be unable even to detect such a penetration attack, let alone correct it.

In other words, while difficult, it is possible to hack a blockchain. Furthermore, voting over the internet from mobile devices could open doors to many vulnerabilities. What is to stop identification QR codes from being stolen or sold? For example, hackers could carry out phishing attacks and create fake voting apps and websites to prevent votes from being recorded, or even steal QR codes. On top of it all, Polys does not claim to support write-in voting, a necessity for holding elections in the United States.

## 5 Conclusion

We can see that cryptographic voting systems can benefit election integrity by providing methods of verifying votes are recorded and tallied correctly. However, these systems collectively struggle to securely implement write-in voting. When it comes to blockchain voting, these systems are very new, and we must be cautious when companies claim that blockchain makes their system unhackable. In conclusion, Scantegrity II has been the most promising, as it proved in a one-time example that cryptographic end-to-end verifiable voting systems can be carried out successfully in real world elections. Perhaps governments will pursue such systems for use in the future.

# References

[1] David Chaum. Scantegrity II: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes. https://www.usenix.org/legacy/event/evt08/tech/full_papers/chaum/chaum.pdf, 2008.

[2] Richard Carback. Scantegrity II municipal election at takoma park: The first e2e binding governmental election with ballot privacy. https://www.usenix.org/events/sec10/tech/full_papers/Carback.pdf.

[3] Daniel Castro. Stop the presses: How paper trails fail to secure e-voting. https://itif.org/files/evoting.pdf, 2007.

[4] Polys. Secure internal elections for political parties. https://polys.me/political-parties. Retrieved 2020-05-15.

[5] Kaspersky Press Release. Polys from kaspersky innovation hub presents first blockchain-based voting machine. https://www.kaspersky.com/about/press-releases/2020_polys-from-kaspersky-innovation-hub-presents-first-blockchain-based-voting-machine, 2020.

[6] Roman Alyoshkin. Polys technology (whitepaper). https://docs.polys.me/en/collections/699457-technology-whitepaper. Retrieved 2020-05-15.

[7] David Jefferson. The myth of ”secure” blockchain voting. https://www.verifiedvoting.org/jefferson_themythof_secure_blockchainvoting/, 2018.