Executive Summary

VoteTrackerPlus (VTP) is a 100% open source project that offers a verifiable guarantee directly to the voter that their ballot is correctly cast and tallied as intended. VTP also provides a verifiable guarantee to election officials that proprietary ballot scanning and tabulating software has functioned correctly. VTP accomplished this by providing three novel core capabilities that increase the security, accuracy, and trustworthiness of elections:

1. Provides a cryptographically sealed, anonymized, and yet unencrypted ballot receipt back to the voter, allowing the voter to validate that their ballot has correctly cast and tallied - a.k.a. zero trust End-to-End Verification (E2EV)
2. Cryptographically records and seals the complete history of the election, pre and post election day
3. Allows the public to inspect and validate the contest tallies by executing them, effectively watching the actual counting of their specific vote

VTP is designed fundamentally as both a plugin to existing election equipment as well as a stand-alone election solution. As a plugin VTP can be incorporated today into existing certified paper based election equipment. As a stand-alone solution VTP can fully support internet (TCP/IP) based voting solutions. VoteTrackerPlus creates 100% closed loop, zero trust, end-voter based E2EV solutions without the need for encrypted data-at-rest, supporting a plethora of workflows such as in-person, mail-in, absentee, early voting, and pure internet based voting, all with or without paper based ballots.

The product road map of the core design principles behind VoteTrackerPlus is separated into two products: VoteTrackerPlus which is available now as a demo, and RegistrationTrackerPlus (RTP) which is still in the conceptual stage.

- VoteTrackerPlus secures the vote itself, digitally signing Cast Vote Records and recording the signatures in a public ledger similar to today's cryptocurrencies. However, as the voter's ballot must remain anonymous, VTP ensures this without the private keys associated with cryptocurrencies and without encrypting the data at rest. This solution is zero trust in that no human or organization is needed to permanently secure the cryptographic signatures or the containing public ledger.
- RegistrationTrackerPlus proposes to integrate the public voter registration data into a separate public ledger with similar but separate capabilities. The registration information is aggregated across the electorate as well as across sequential elections, keeping a record of all changes. When integrated with VTP, RTP can print blank ballots on demand.

There are End User License Agreements (EULA’s) with both VTP and RTP to address data privacy and usage concerns.

The rest of this summary only addresses VTP, securing the vote itself.

1. How does VTP provide a verifiably but yet anonymous guarantee back to the voter?

The 10,000 foot summary of how VTP accomplishes this is by returning to the voter a spreadsheet-like table of cryptographic digests, one digest for each contest. The table rows are effectively ballots and each column is a contest. 99 rows are randomly selected from previously submitted ballots and one row is the voter's specific ballot. A key design element is that the voter's specific row number is privately displayed to the voter and is not recorded in the public ledger - it is ephemeral only and disappears. The 99 other rows are random contests from random ballots, which is in contrast to the voter's row which is in fact a single ballot (the voter's).

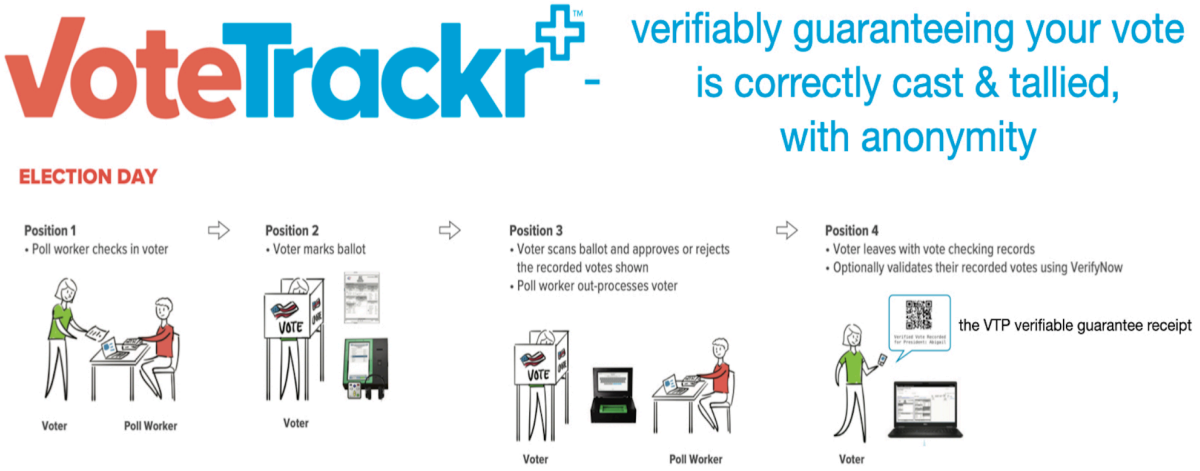
A second key design element is that since there is no identifying information in the public ledger nor the ballot receipts, the public ledger and the ballot receipts need not be encrypted at rest - they are already both anonymous as is.

A third key design element is that the VTP database is a Merkle Tree similar to cryptocurrencies but without the need for either a blockchain proof-of-work component or for private data-at-rest encryption keys. Nonetheless, like a cryptocurrency the history of all transactions, in this case both all votes, all the VTP software, and all the election configuration data (blank ballots and the like), is protected and secured.

2. How does VTP integrate as a plug-in to existing paper based election equipment?

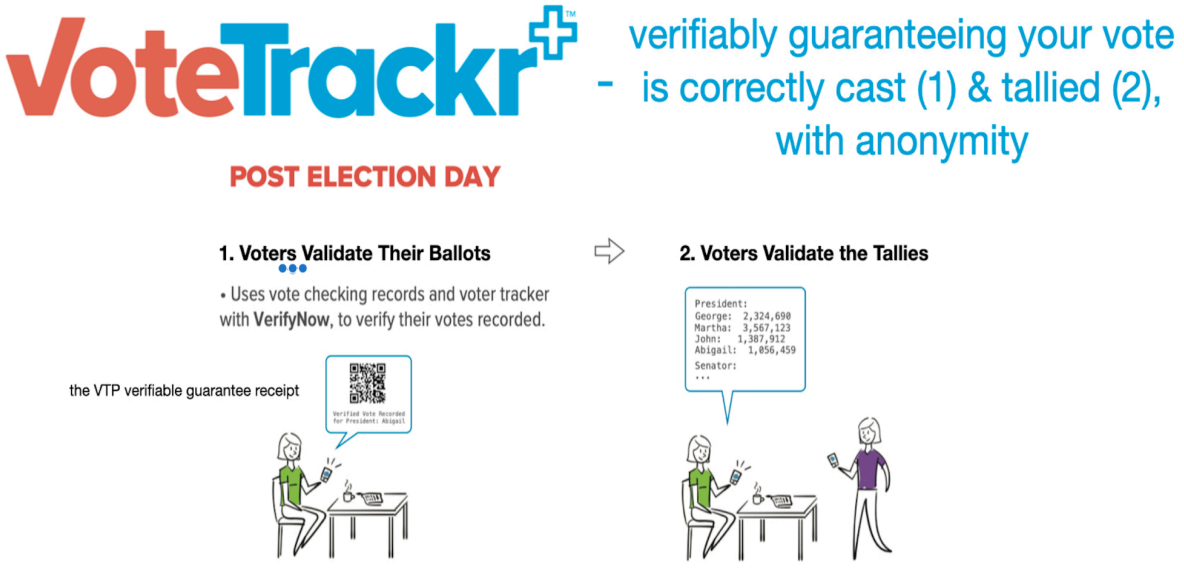
VoteTrackerPlus is a side-car type plugin to existing election equipment. It does not alter the official election data that the existing proprietary and certified systems produce. As a plugin, the existing proprietary software is only modified to pass Cast Vote Records out to the VTP software - no software or data goes the other way. At a technical level existing systems do not need to be recertified.

It is up to the election equipment manufacturer how to integrate VTP into the voter and election official workflows that the equipment supports. If the manufacturer supports ballot marking devices, VTP can be integrated there. If the equipment consumes voter marked ballots and directly scans those, VTP can be integrated there. The following slide is an example of a hypothetical in-person paper based workflow augmented with VTP:



In-Person Voting Workflow

This next slide depicts the end voter workflow after all the polls close and the VTP data is made publicly available by election officials. Regardless of how the equipment manufacturer has delivered the receipt to the voter, this workflow depicts the end voter verifying the guarantee that their vote was cast and tallied as directed:

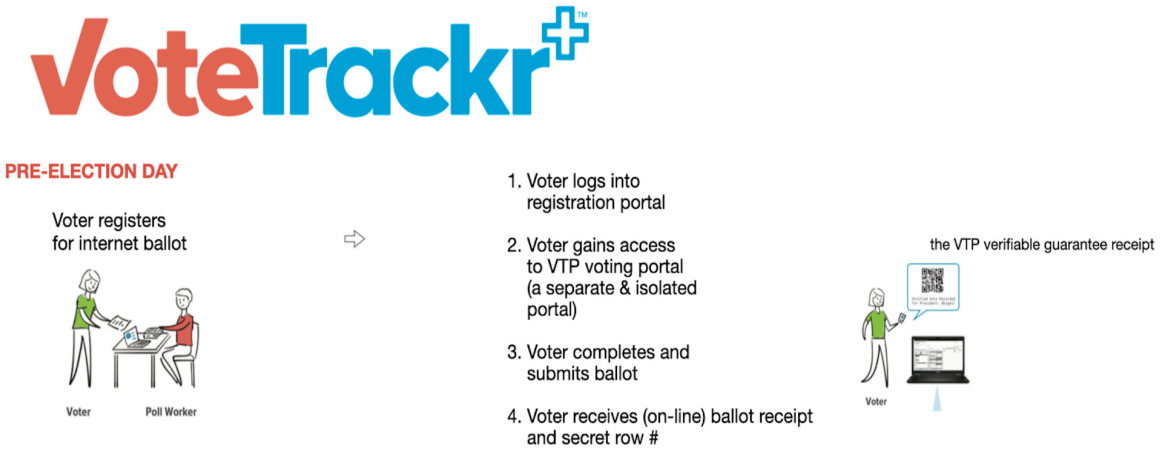


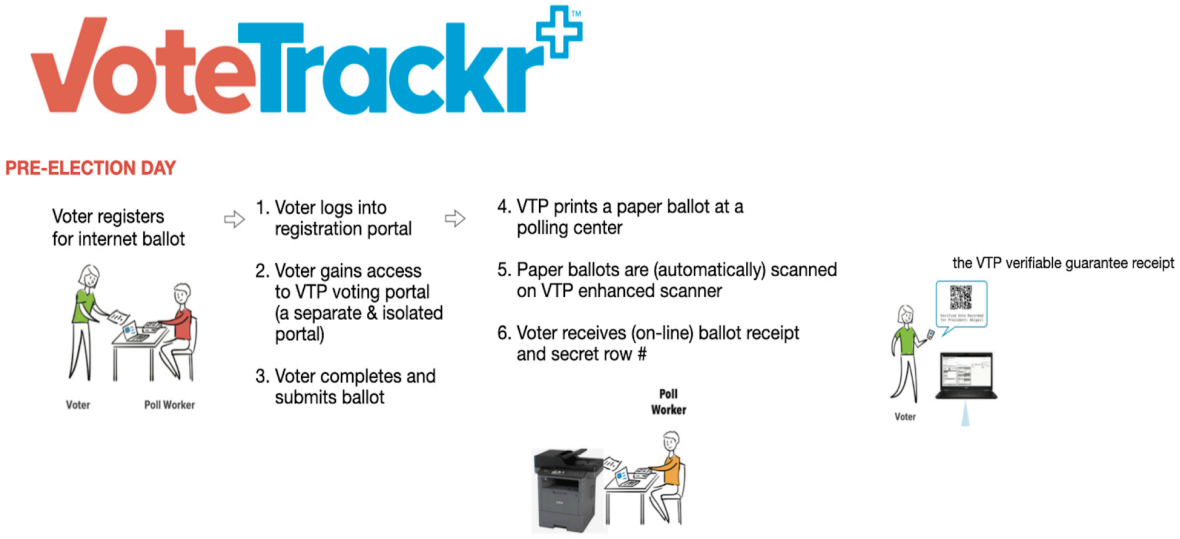
Same for all Voting Methods

3. How does VTP support internet (TCP/IP) based voting?

VoteTrackerPlus works as a complete software based vote scanning and tallying solution. Internet voting (a.k.a. TCP/IP based voting or iVoting) does not require dedicated end-voter facing hardware. VTP can be integrated with a voter registration portal and used as an end voter ballot marking device and tabulator. Once marked on a mobile device, the voter's choices are added to the VTP public ledger and the anonymous ballot receipt is returned to the voter as web page data. The voter is free to store or delete the receipt.

The first image below depicts a hypothetical paperless VTP internet voting workflow. The second image depicts an internet voting workflow where a paper ballot trail is produced, secured end-to-end with VTP.





4. How does VTP support and promote RCV?

VoteTrackerPlus natively supports both plurality and RCV tally algorithms. Note that a fundamental feature and differentiator of VTP is that it allows end voters to execute the contest tallies and watch their specific vote get properly counted.

Note that RCV contests have greater vulnerability to misinformation campaigns as well as software defects due to their increased algorithmic complexity, which may be beyond the understanding of portions of the electorate. With the proliferation of fake news and media outlets augmented with advanced AI, RCV contests have a greater need for transparency and zero trust. VTP with its inherent zero trust E2EV design allows each voter to inspect their own specific vote on either their own personal smart devices or via election official VTP websites. Consider the following RCV contest tally validation. In this command line example, an end voter has entered a specific CVR contest digest to track during the execution of a RCV contest tally.

```
$ tally-contests -c 0001 -t da193d86cca73996735f3c66d3048c01c19c226e
Scanned 122 contests for contest (U.S. Senate) uid=0001, tally=rcv, max_selections=6,
win_by>0.5

--------------------------------
RCV: round 0
Counted da193d86cca73996735f3c66d3048c01c19c226e as vote 10: selection=Gloria Gamma
Total vote count: 122
```

```
 ('Francis Foxtrot', 26)
 ('Emily Echo', 24)
 ('Anthony Alpha', 23)
 ('Betty Beta', 21)
 ('David Delta', 19)
 ('Gloria Gamma', 9)

---------------------------------
RCV: round 1
INSPECTING: da193d86cca73996735f3c66d3048c01c19c226e (contest=U.S. Senate) as vote
10
RCV: da193d86cca73996735f3c66d3048c01c19c226e (contest=U.S. Senate) last place pop
and count (Gloria Gamma -> Betty Beta)
Total vote count: 122
 ('Francis Foxtrot', 28)
 ('Anthony Alpha', 26)
 ('Emily Echo', 25)
 ('Betty Beta', 24)
 ('David Delta', 19)
 ('Gloria Gamma', 0)

---------------------------------
RCV: round 2
INSPECTING: da193d86cca73996735f3c66d3048c01c19c226e (contest=U.S. Senate) as vote
10
Total vote count: 122
 ('Francis Foxtrot', 35)
 ('Anthony Alpha', 30)
 ('Emily Echo', 30)
 ('Betty Beta', 27)
 ('David Delta', 0)
 ('Gloria Gamma', 0)

---------------------------------
RCV: round 3
INSPECTING: da193d86cca73996735f3c66d3048c01c19c226e (contest=U.S. Senate) as vote
10
RCV: da193d86cca73996735f3c66d3048c01c19c226e (contest=U.S. Senate) last place pop
and count (Betty Beta -> Emily Echo)
Total vote count: 122
 ('Francis Foxtrot', 47)
 ('Emily Echo', 41)
 ('Anthony Alpha', 34)
 ('Betty Beta', 0)
```

```
 ('David Delta', 0)
 ('Gloria Gamma', 0)

--------------------------------
RCV: round 4
INSPECTING: da193d86cca73996735f3c66d3048c01c19c226e (contest=U.S. Senate) as vote
10
Total vote count: 121
Final results for contest U.S. Senate (uid=0001):
 ('Francis Foxtrot', 64)
 ('Emily Echo', 57)
 ('Anthony Alpha', 0)
 ('Betty Beta', 0)
 ('David Delta', 0)
 ('Gloria Gamma', 0)
```

In the above output, the end voter can observe that in the first round their vote for Gloria Gamma was the losing choice, so their specific vote was recast to Betty Beta. Then in the third round their vote for Betty Beta was recast to Emily Echo as Betty Beta was the losing candidate. And then in the fourth and last round Francis Foxtrot won with greater than 50% of the vote. Note that VTP will display which specific vote count in the round was the voter's. All this information is repeatable without derivation on any smart device, owned by either an election official or a voter.

5. How does VTP promote agile election operations?

VoteTrackerPlus, being open source and already integrated with GitHub, allows election officials to design and roll out their blank ballots within the GitHub DevSecOps continuous integration and deployment infrastructure - if they are so interested. This includes testing their precinct maps for which blank ballots apply to which addresses as well as ballot scanning and tabulation.

GitHub: https://github.com/OpenVotingTechnologyGroup/VoteTrackerPlus