votingapp

Scan to read
the long version

End-to-End Verifiable Voting

by David Wallick, Ryan Cook & Greg Long

Introduction

Public elections in the United States consist primarily of poll station in-person voting on paper ballots or on highly controlled electronic machines. Vote-by-mail paper ballot voting has not been widely adopted except in a few states prior to the 2020 election.[1, 2] While these methods have been refined for security, accessibility, and privacy concerns, they still have their drawbacks.

In the late 90's many ideas have circulated on using the internet for voting, with goals of providing increased convenience and access to voters, as well as potentially reducing administrative costs to jurisdictions.[3, 4, 5, 6] However, large scale voting over public networks in government sponsored elections has not been a practical possibility in the United States due to security concerns, privacy concerns, and a general lack of verifiability. Voters, election officials and outside observers have not been able to verify the integrity of online elections.[7, 8]

The challenge of voting on public networks is meeting both necessary security and vote verifiability and the need to keep individual voter selections private.[9]

Solution

The voting system uses a variety of well-established hardware and software cryptographic technologies. Included is a commercially available cryptographic hardware chip designed specifically for securing keys and sensitive data, signing and encrypting data with industry standard ciphers, standard certificate chain of trust processes, and the use of blockchain as the ballot box public ledger. Multiple redundant and complementary security measures are utilized to mitigate potential unknown security vulnerabilities.

An end-to-end verifiable (E2EV) system involves three key 'proofs' regarding vote verification.

- Cast as Intended Proof - Voters can verify their vote is cast as they marked their ballot.
- Recorded as Cast Proof - The voter and observers can verify the authenticity and integrity of each vote.
- Tallied as Recorded - All observers of the system can verify all valid votes are included in the final count.

Establishing Trust

A core element of the voting system is the Security Device. It is provisioned at manufacturing time with a Signed Root Certificate from the securely stored private key. Trust is delegated to the Security Device which in turn delegates trust to various election components by acting as a certificate authority. In most cases, trust is established by verifying that the source of the data has a certificate signed by the same Security Device. The exception is the mobile Voting Client. The mobile application will confirm that it can trust the system or data by examining the chain of trust and confirming that the root of trust is a trusted source. See long paper for more information for establishing trust between components.

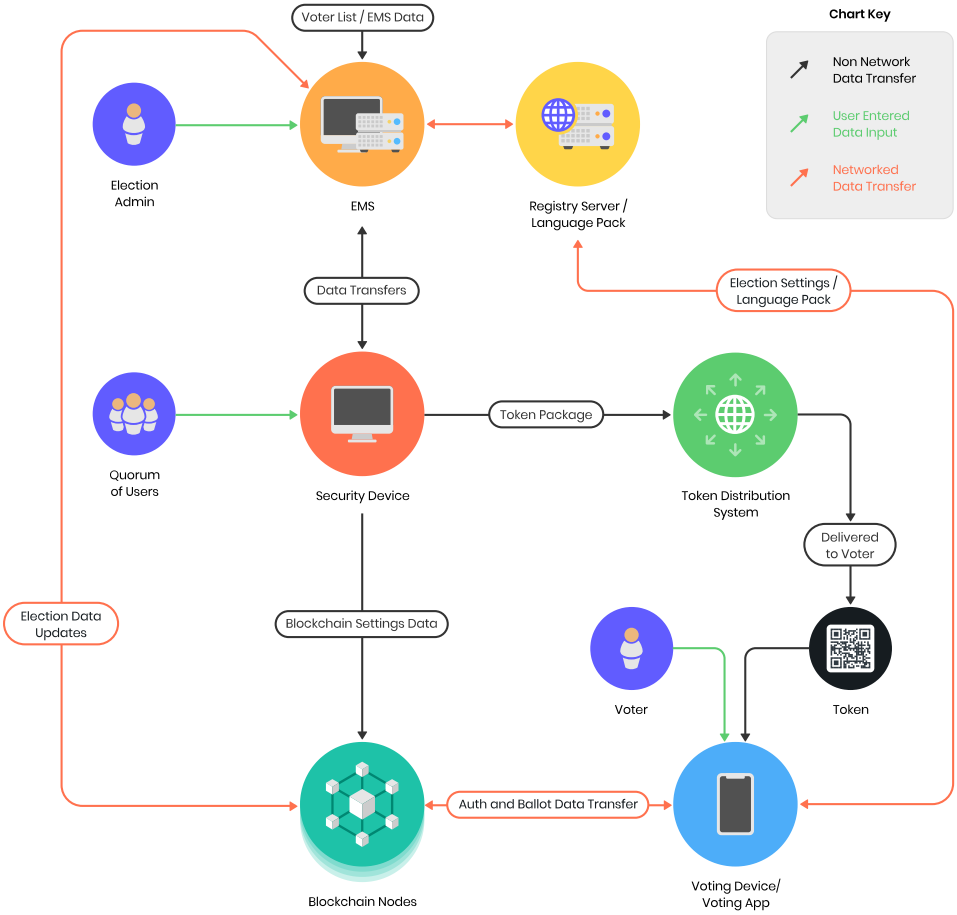


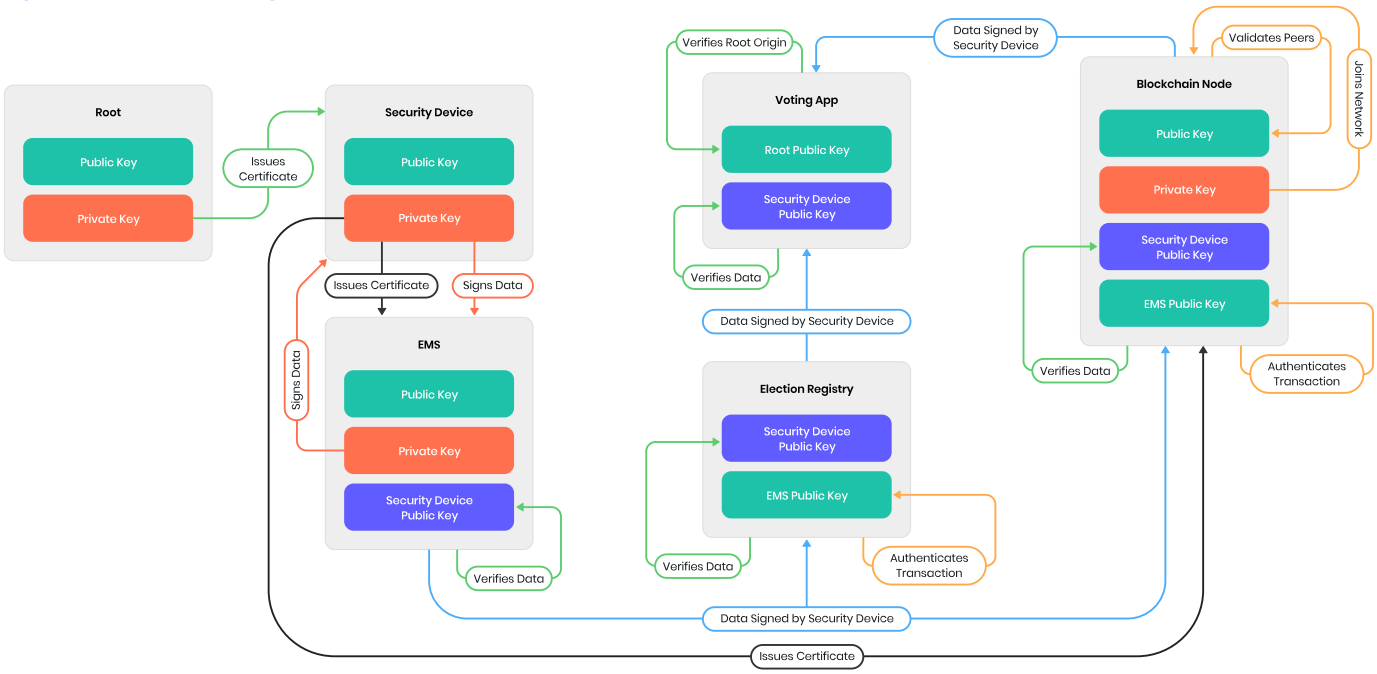
Figure 1: High Level Data Flow

Fully provisioned and operating election network. The communication paths shown represent data transfers during pre-election, live election, and post-election phases. It depicts the connections between components and distinguishes between networked traffic and manual data transfers.

Detailed data workflow diagrams are available in the long version of this paper.

Data within the system is stored and transported in such a way that the origin of the data can be traced, and the contents cannot be tampered with. Data is verified by confirming that it is signed by a trusted source. Using the various keys and certificates provisioned, all election data can be verified. Trust can be verified by following back the entire certificate chain to the root certificate in the Security Device. Multiple methods of verification are used for all operations so that a single participant or component, if compromised, cannot damage the integrity of the system. See Figure 2. Further details are available in the long version of this paper.

Figure 2: Establishing Data Verification



End-to-End Verification

Up to this point the focus has been on establishing the election network and verifying the authenticity and integrity of data transfers between components. While an essential piece to establishing the integrity of the system, on its own these steps do not constitute end-to-end verifiability. To complete this process three proofs must exist. Due to the architecture, described below, for handling cast vote records (CVRs) the three proofs (Cast as Intended, Recorded as Cast, and Tallied as Recorded) are overlapping. As such, a review of the basic structure is provided with proof explanations following.

Vote Submission & Storage

The vote collection and subsequent validations require two aspects:

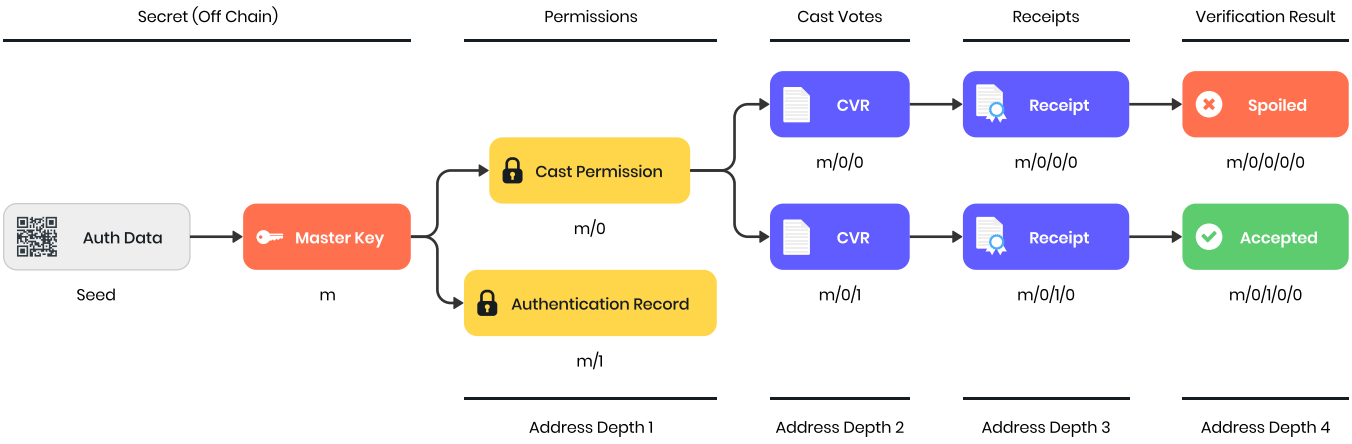
- Voter Receipt - Contains data that fully encapsulates the ballot choices as well as user defined data. It is verified against the cast vote record (CVR) and is recognizable to the voter. The receipt is generated on the voting device during the marking process, delivered to the voter and stored on the ledger as a child address (see below) of the cast vote record within the ledger.
- Hierarchical Deterministic Addressing Protocol (HDAP) - The system uses hierarchical deterministic keys to create a protocol for storing election data in the public ledger. The protocol provides predetermined locations for election records such as CVRs and associated receipts.

Hierarchical Deterministic Addressing Protocol (HDAP)

Our addressing protocol uses hierarchical deterministic key generation based on BIP32[10] to lay out a structure for storing election data in a way that supports end-to-end verification while maintaining the privacy of the voters.

Hierarchical deterministic key generation provides the ability to derive child public key addresses from a parent public key and child private keys from parent private keys. Deriving private keys from public keys is not possible and deriving parents from children is not possible.

Figure 3: Address Structure



The voting system relies on a multi-factor authentication where the voter recreates the correct private keys in order to verify identity by signing transactions. To achieve this the Security Device uses some information known to the voter along with a random token to generate the master key in Figure 3. The private key is discarded, the token is delivered to the voter and the public key is used to pre-populate the ledger with the cast permission and authentication records. The voter uses the token and known information to recreate the key(s) within the Voting Client needed to sign the transactions. No private key is ever stored on any device or server.

Details of the addressing protocol, authentication, and vote casting are available in the long version paper.

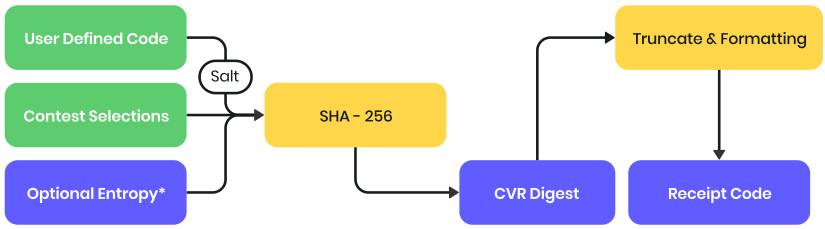
Voter Receipt

The receipt will have the following data elements:

- **Ledger Address** - Location where the receipt data is stored and can be used to look it up from any device.
- **User Defined Code** - A code or password that can be defined by the user.
- **CVR Digest** - Full digest of the contest selections, user defined code and additional optional entropy.
- **Receipt Code** - Version of the CVR digest that is formatted for easier recognition by a human.

The CVR digest and receipt code is generated by creating a digest and truncated digest from the relevant data. This receipt is offered to the voter for storage/printing by the voting device at the time of creation & prior to submission to the public ledger. To increase coercion resistance, the saved/printed receipt has details encrypted by prompting the user for a password or pin.

Figure 4: Receipt Generation



Voter Validation

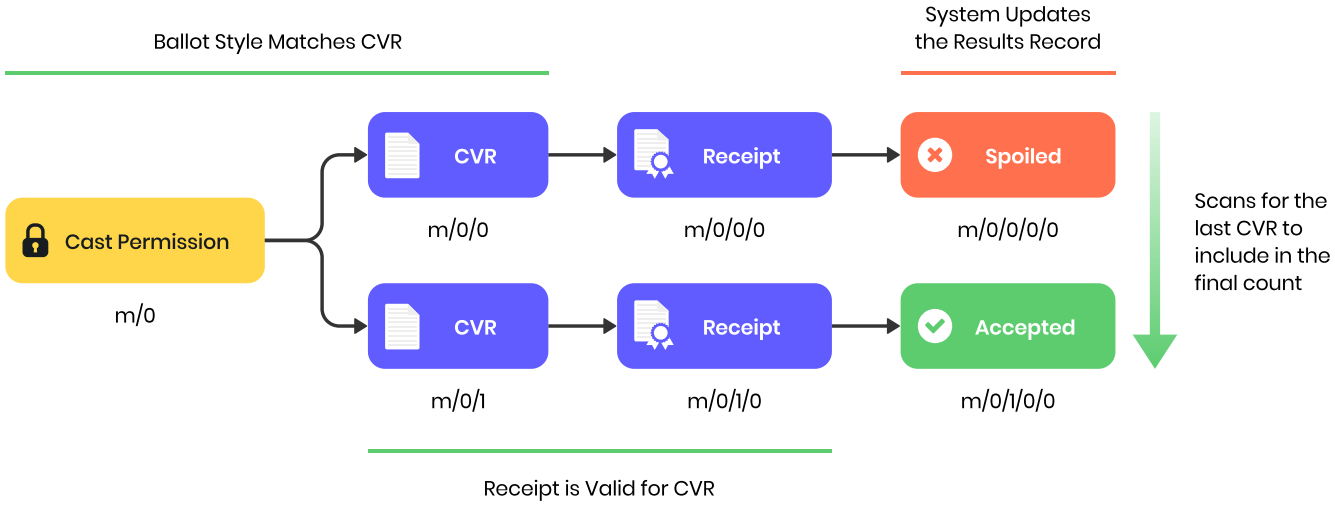
It is important that a technically sophisticated voter can recreate the CVR Digest on their own to validate the accuracy of the receipt. In order to facilitate this the Voting Client allows a user to see a detailed view that lists all of the vote selections used to generate the receipt code as well as an explanation on the digest generation. With this data the user recreates the digest and confirms the receipt is constructed correctly.

Less technically inclined voters still get peace of mind by observing that their receipt code changes in a deterministic manner as they vote because the receipt code displays on all ballot marking and review screens of the Voting Client and changes in real time as they alter selections or change the User Defined Code.
After submission all voters can confirm that the correct receipt is stored in the public ledger and matches the receipt generated on their device.

System Verification

After the election is complete and the CVRs are decrypted, the receipts are verified by the election administrator by recreating the CVR digest and comparing it to the stored receipt. Once verification is complete a record is stored marking the CVR as accepted. Any CVRs found generating receipts that do not match saved receipts are flagged and steps are taken to deal with improperly collected data including invalidating the election if deemed necessary.

Figure 5: Receipt Verification



Cast as Intended Verification

The Cast as Intended Proof uses multiple steps from the process listed above.

- Voter creates and stores the receipt and receipt code generated by the Voting Client on their own device.
- Voter can confirm that the correct receipt is stored at any time.
- When the CVR is decrypted post-election the voter is assured the Cast as Intended Verification completed because:
  - The receipt stored in the ledger matches the receipt in hand; and
  - The receipt stored in the ledger matches the CVR digest stored in the ledger.

Recorded as Cast Verification

- CVR is decrypted and digest generated.
- Receipt is recreated from above digest and compared against stored receipt, if
  - Receipts match and no spoil child is present CVR is deemed 'Accepted'
  - Receipts that do not match CVR are deemed invalid. Investigation can determine 'why' discrepancy exists and steps can be taken to:
    - Invalidate individual CVR (in case where single instance root cause can be found) and child address of 'Rejected' added; or
    - Invalidate entire election because no root cause can be found, or systematic tampering/failure cannot be ruled out

Tallied as Recorded Verification

Unlike other E2EV systems that keep CVR data encrypted, which then requires mathematical proofs for the correctness of vote counting, the VotingApp system ends with unencrypted CVRs that can be treated similarly to traditional paper ballot for tallying, reporting, auditing, and recounts. Assuming the previous two proofs are successfully validated, a jurisdiction can release the CVR data the same way ballot images and CVR data are released in traditional paper systems.

For an in-depth look as well as conclusions and discussion on future white paper topics see the long version of this paper at votingapp.com

References

1. Hastings, N., Peralta, R., Popoveniuc, S., Regenscheid, A. (2011). "Security Considerations for Remote Electronic UOCAVA Voting". Accessed July 15, 2021. https://csrc.nist.gov/publications/detail/nistir/7770/final
2. Regenscheid, A., Beier, G. (2011). "Security Best Practices for the Electronic Transmission of Election Materials for UOCAVA Voters" Accessed July 15, 2021. https://csrc.nist.gov/publications/detail/nistir/7711/final
3. Beroggi, G. (2008). "Secure and Easy Internet Voting" (PDF). Retrieved July 15, 2021. https://sargasso.nl/wp-content/uploads/2008/07/internetvoting.pdf
4. Rubin, A.D. (2001). "Security Considerations for Remote Electronic Voting over the Internet" (PDF). Retrieved July 15, 2021. www.cs.jhu.edu/~rubin/courses/sp03/papers/e-voting.security.pdf
5. U.S. Election Assistance Commission (2011) "A Survey of Internet Voting" (PDF). Retrieved July 21, 2021. https://www.eac.gov/sites/default/files/eac_assets/1/28/SIV-FINAL.pdf
6. Kelleher, W. (2013) "Internet Voting in the USA: History and Prospects" (PDF). Retrieved July 21, 2021. https://www.eac.gov/sites/default/files/eac_assets/1/28/William-Kelleher-Internet-Voting-WPSA-Paper-July-9th.pdf
7. Alvarez, R.M., Hall, T.E. (2004). "Point, click & vote: the future of Internet voting." Brookings Institution Press, Washington D.C.
8. Evans, D., Paul, N. (2004). "Election Security: Perception and Reality" (PDF). Retrieved July 14, 2021. https://www.academia.edu/1126471/Election_security_Perception_and_reality
9. Benaloh, J., Rivest, R., Ryan, P., Stark, P., Teague, V., & Vora, P. (2015). "End-to-end verifiability" (PDF). Retrieved July 9, 2021. https://escholarship.org/content/qt7c9994dg/qt7c9994dg_noSplash_97d64dc5a809c552701079250f47b4cb.pdf
10. Pieter Wuille "BIP-032 - Hierarchical Deterministic Wallets". https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki