TPM Meets DRE: Reducing the Trust Base for Electronic Voting Using Trusted Platform Modules

Russell A. Fink, Alan T. Sherman, and Richard Carback

Direct Recording Electronic (DRE)

Higher usability compared to paper ballot

- Multi-language support
- People with disabilities can easily use it
- Different ordering of candidates
  - reduce primacy phenomenon

DRE Limitation

Users have to trust both hardware, software, and user interface of the DRE since the inner workings of the platform is transparent to the user.

Motivation

Reduce the trusted computing base of DRE.

How?

Use TPM

TPM (Trusted Platform Module)

SRK(Storage Root Key)

- Generated when someone takes ownership of the TPM
- Private SRK never leaves the TPM
- Can manage other keys
    - Encrypt the other keys using SRK private key
    - Can export the encrypted key

Check the integrity of the software

- Checked during boot
- The configuration data are stored in PCRs (TPM Platform Config Registers)

PCRs (Platform Config Registers)?

Volatile memory in TPM

- It can be set to zero or store hash of existing PCR value.
- Cannot be set to specific non-zero value

Values in PCR is used to check the software state

Human Roles

Trusted Election Authority (TEA)

Trusted Tallying Authority (TTA)

Trusted Precinct Judge (TPJ)

Independent Testing Authority (ITA)

Architecture

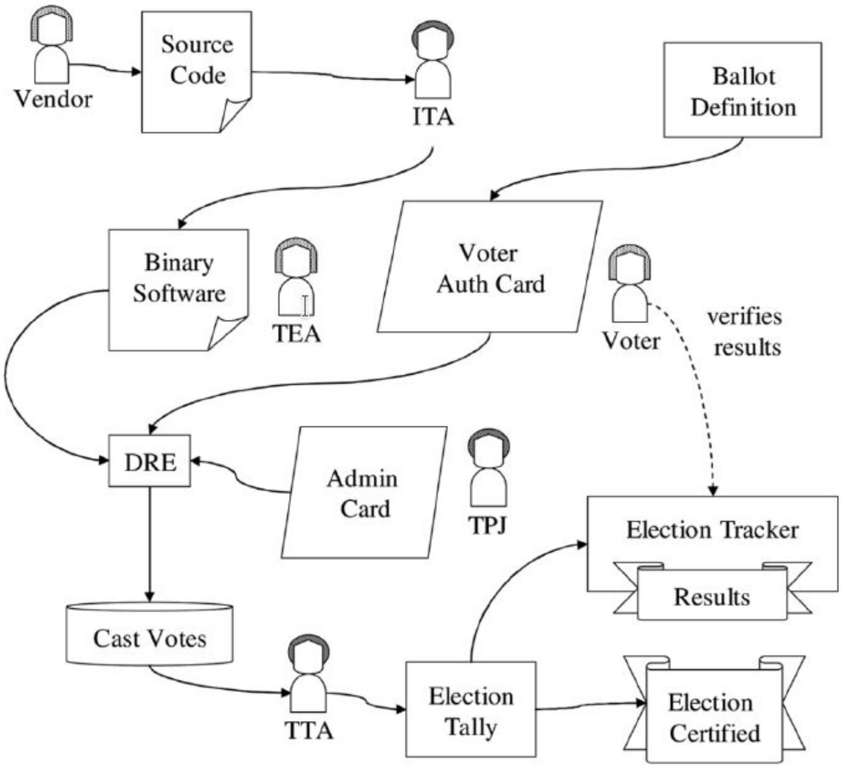
Platform initialization

Voting Start

Poll

Voting Termination

Tallying



Platform Initialization

TEA takes ownership

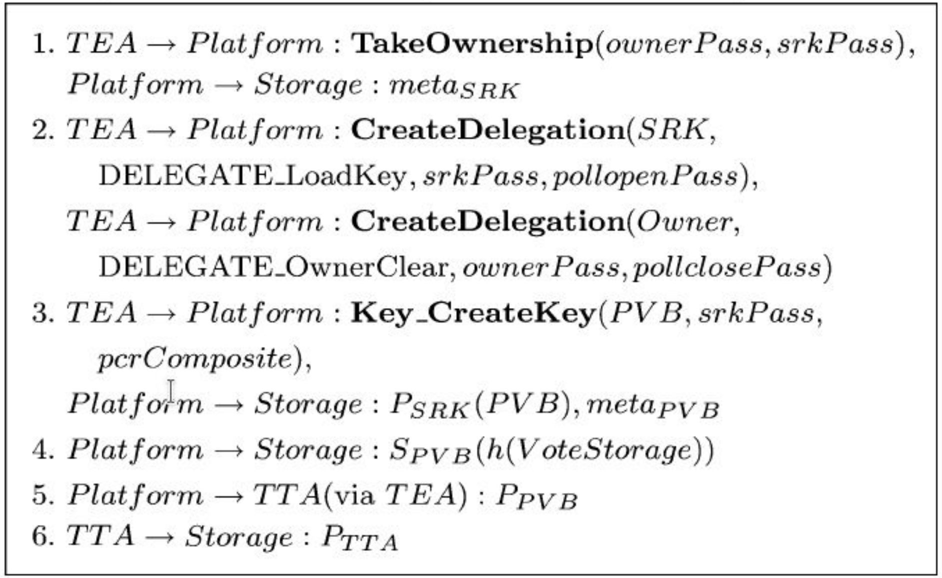
TEA delegates key loading (pollopenPass) and ownership (pollclosePass) to TPJ.

TEA supplies the PCR measurements and creates PVB key

Setup Storage

TPM gives public PVB key to TEA. TEA sends the key to TTA.

The key gets stored in the platform for final encryption.



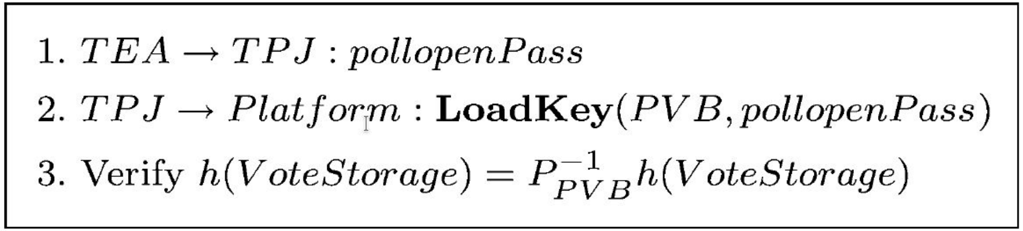
1. $TEA \rightarrow Platform :$ **TakeOwnership**$(ownerPass, srkPass)$,
   $Platform \rightarrow Storage : meta_{SRK}$
2. $TEA \rightarrow Platform :$ **CreateDelegation**$(SRK,$
   DELEGATE_LoadKey$, srkPass, pollopenPass)$,
   $TEA \rightarrow Platform :$ **CreateDelegation**$(Owner,$
   DELEGATE_OwnerClear$, ownerPass, pollclosePass)$
3. $TEA \rightarrow Platform :$ **Key_CreateKey**$(PVB, srkPass,$
   $pcrComposite)$,
   $Platform \rightarrow Storage : P_{SRK}(PVB), meta_{PVB}$
4. $Platform \rightarrow Storage : S_{PVB}(h(VoteStorage))$
5. $Platform \rightarrow TTA($via $TEA) : P_{PVB}$
6. $TTA \rightarrow Storage : P_{TTA}$

Voting Start

TPJ can only use pollopenPass to load the software.

The voting software get loaded and the software state is compared against the supplied PCR.

Ensures that storage is loaded correctly.

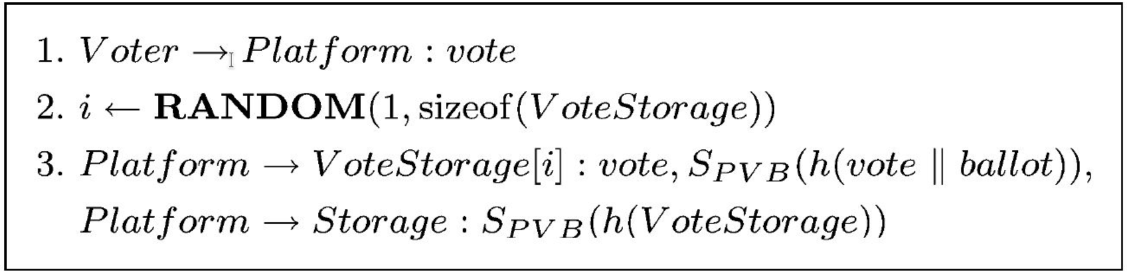


1. $TEA \rightarrow TPJ : pollopenPass$
2. $TPJ \rightarrow Platform : \mathbf{LoadKey}(PVB, pollopenPass)$
3. Verify $h(VoteStorage) = P_{PVB}^{-1} h(VoteStorage)$

Poll

Voter uses electronic ballot supplied by TPJ to vote.

The the vote and hash of the vote and the ballot gets encrypted using private PVB key and get stored in pseudo random location of the storage.



1. $Voter \rightarrow Platform : vote$
2. $i \leftarrow \mathbf{RANDOM}(1, \text{sizeof}(VoteStorage))$
3. $Platform \rightarrow VoteStorage[i] : vote, S_{PVB}(h(vote \parallel ballot)),$
   $Platform \rightarrow Storage : S_{PVB}(h(VoteStorage))$

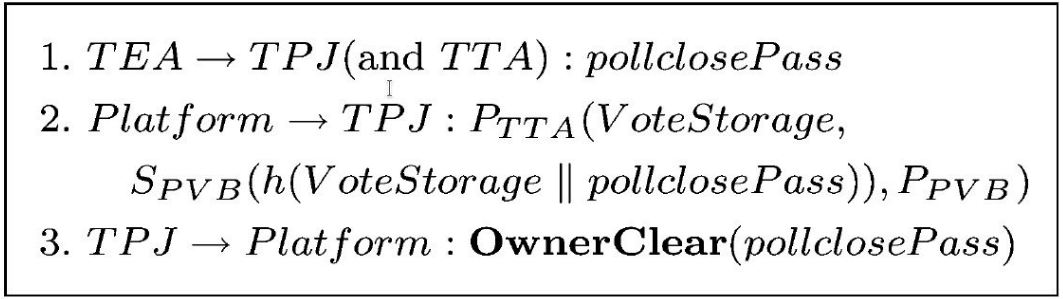
Voting Termination

TPJ uses pollclosePass to initiate voting termination.

The storage is encrypted using TTA public key and is removed. The storage contains vote data, encrypted hash, and public PVB key.

TPJ clears ownership. This cleans the TPM state. Private PVC key is permanently gone.

This process must be witnessed.



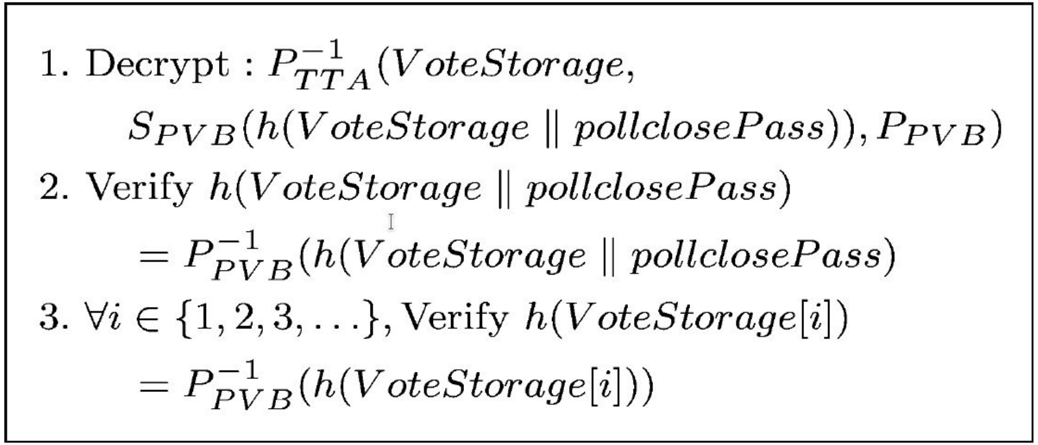
1. $TEA \rightarrow TPJ(\text{and } TTA) : pollclosePass$
2. $Platform \rightarrow TPJ : P_{TTA}(VoteStorage,$ $S_{PVB}(h(VoteStorage \parallel pollclosePass)), P_{PVB})$
3. $TPJ \rightarrow Platform : \textbf{OwnerClear}(pollclosePass)$

Tallying

TTA decrypts the storage and checks the public PVB key by comparing it to the one supplied by TEA.

TTA checks the integrity of the storage by checking the hash in the encrypted storage.

Count and verify each vote.



1. Decrypt : $P_{TTA}^{-1}(VoteStorage,$
   $S_{PVB}(h(VoteStorage \parallel pollclosePass)), P_{PVB})$
2. Verify $h(VoteStorage \parallel pollclosePass)$
   $= P_{PVB}^{-1}(h(VoteStorage \parallel pollclosePass)$
3. $\forall i \in \{1, 2, 3, \ldots\}$, Verify $h(VoteStorage[i])$
   $= P_{PVB}^{-1}(h(VoteStorage[i]))$

What attacks does it mitigate?

- Ballot Modification
- Storage modification
- Software modification
- Observing Vote Order to de-anonymize voters
- Day-before Attack

Discussion

Should the trust be more distributed or centralized in voting systems? Where is the right balance? What is the tradeoff?

Does the assumption that this system make have huge difference compared to paper ballot?

What do you think is the minimal amount of assumptions should a DRE system operate under to ensure voting integrity and voter anonymity?