a tamper-evident, verifiable voting system

DANIEL R. SANDLER
DAN S. WALLACH
RICE UNIVERSITY
DRE voting systems

Current systems shown to have deep flaws
High-profile malfunctions
Vulnerable to attacks
Yet, there are benefits
Accessibility
Feedback
Flexibility
User preference

Greene et al. Is newer always better? The usability of electronic voting machines versus traditional methods. CHI ’08.
building a better electronic voting machine

security properties

minimized software stack
less code to examine → practical audits & certification

fault tolerance
prevent or minimize data loss in case of failure

tamper evidence
proof of failure/attack during & after an election

verifiability
confirm that votes will be cast as intended
techniques
used in VoteBox

1. PRUI: pre-rendered user interfaces
2. Auditorium: replicated secure logs
3. ballot challenge system
PRUI: pre-rendered user interfaces

Move complexity out of the voting machine TCB and into a definition file representing the ballot.

- Ballot artwork & text, pre-rendered into bitmaps
- Ballot layout
- Navigation & selection state machine

Result:

<table>
<thead>
<tr>
<th></th>
<th>Diebold</th>
<th>Sequoia</th>
<th>VoteBox</th>
</tr>
</thead>
<tbody>
<tr>
<td>KLOC</td>
<td>64 (C++)</td>
<td>124 (C)</td>
<td>14 (Java)</td>
</tr>
</tbody>
</table>

Inspired by Pvote

A voting machine is a terrible place to keep ballots

A malicious voting machine might silently alter its own totals

And even honest voting machines can fail, losing votes & audit logs

We can’t trust voting machines to store critical election data

...not without redundancy
the AUDITORIUM polling place network

connects all voting machines (+ supervisor console)
all election events are digitally signed, broadcast to other machines, and recorded in tamper-evident logs
result: tamper-evidence and recoverable data
“cast as intended”

the biggest challenge for DREs

how can the voter trust that a VoteBox captured the voter’s choices faithfully, encrypted the ballot correctly, and stored and broadcast it in the Auditorium?

if the voter’s intent is lost, no amount of procedure or post facto auditing can recover it
ballot challenge

at the end of the voting session:

1. force the machine to **commit** to the contents of the ballot it is about to cast
   - irreverable
   - contents not revealed

2. the voter chooses either:
   - **cast** the ballot, or
   - **challenge** the machine to reveal the contents of the commitment

(challengers should enlist pollworker assistance)
we owe this technique to Benaloh

J. Benaloh. *Ballot casting assurance via voter-initiated poll station auditing.* In *Proceedings of the 2nd USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '07).*

the commitment is an **encrypted ballot**

in Benaloh 07, it was printed under glass
the machine cannot un-print it in case of audit
to challenge: break glass & decrypt

in VoteBox, **Auditorium** is the “printer”

commitments broadcast & logged everywhere
we can send these commitments offsite via one-way link
allows third-party **challenge centers** to supervise and help confirm challenges
conclusion: why VoteBox?

lots of research on **individual pieces** of the e-voting problem

VoteBox uniquely integrates these techniques into a **single system**

it also introduces **Auditorium** and a new **ballot challenge** scheme

offering **security properties** not found in today’s commercial systems

**NB:** some or all of our techniques could be added to those systems
thanks

undergraduates who have worked on VoteBox

Kyle Derr, Emily Fortuna, George Mastrogiannis, Kevin Montrose, Corey Shaw, Ted Torous

designers of the VoteBox ballot

Mike Byrne, Sarah Everett, Kristen Greene

others who have offered ideas and criticism

Ben Adida, Josh Benaloh, Peter Neumann, Chris Piekert, Brent Waters

NSF/ACCURATE
votebox.cs.rice.edu

SOURCE CODE
— booths, supervisor console, ballot creator
— core tech: Auditorium, etc.

RESEARCH PAPERS

OPERATING INSTRUCTIONS

WHAT IS VOTEBOX?
VoteBox is a prototype electronic voting machine created by researchers in the Computer Security Lab at Rice University. It is designed to be a platform for broad e-voting research, particularly in the areas of security and usability. The code is written in Java, and runs on computers with Windows, Macintosh, and Linux operating systems.
ballot challenge voter flowchart

1. make selections
2. review selections
3. FINISHED
   - cast?
     - ballot committed
       - CAST
         - thank you
           - ballot confirmed
           - thanks for challenging
             - ballot challenged
       - CHALLENGE
         - ballot challenged