E-Voting: A Case Study in Software Engineering

Dan Wallach
Rice University
What elections are all about

Task: Measure voter intent

Goal: Convince the loser that he/she actually lost!

How?
How can technology improve elections?

- Anonymity / privacy of voter
- Integrity of vote records / final tally
  - Software correctness / robustness
  - Tamper-resistance
- Human factors / accessibility for voters
- Procedural compliance / robustness
Voting technology glossary

Precinct-based optical scanner

Direct Recording Electronic (DRE)
Voting technology glossary

Voter-verifyable paper audit trail (VVPAT)
Voting Machine Adoption

Counties That Have Changed Voting Equipment

<table>
<thead>
<tr>
<th>KEY TO MAP</th>
<th>NO. OF COUNTIES</th>
<th>PCT.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004 to 2006</td>
<td>1,078</td>
<td>34%</td>
</tr>
<tr>
<td>2002 to 2004</td>
<td>324</td>
<td>10%</td>
</tr>
<tr>
<td>2000 to 2002</td>
<td>388</td>
<td>12%</td>
</tr>
<tr>
<td>Not since 2000</td>
<td>1,351</td>
<td>43%</td>
</tr>
</tbody>
</table>

Percentage of registered voters in counties using each equipment type

Source: Kimball W. Brack, Election Data Services

The New York Times

ACCURATE
Risk 1: Anonymity

Resistance to voter bribery / coercion

- First addressed with “Australian ballot,” 1850’s
- Inherent weakness of mail-in ballots (or Internet voting)

Still of concern today

- Chain voting, pattern voting, camera-phones
- Votes recorded in order (paper-roll VVPAT)
- Timing issues (write-in votes in TX-22)
Risk 2: Bugs

Carteret County, NC
Nov. 2004 election
4438 votes lost (machine memory full)

Not uncommon issues:
- Hardware / smartcard / battery failures
- Inconsistent tallies (operator error?)
Risk 3: Software insecurity

Most studied: Diebold AccuVote-TS / TSx

- Poor software engineering
- Incorrect cryptography / protocols
- Possible for voters to cast multiple votes
- Vulnerable to malicious software upgrades

How *not* to encrypt data

```
#define DESKEY
    ((des_key*) "F2654hD4")
```

One key for every voting machine, everywhere

Doug Jones (Iowa official) found this in 1997

- Still present in current systems!
  (DES replaced with AES, but same key)

Comparable naïveté with other vendors
Diebold’s smart card protocol

Terminal

My password is (8 bytes)

“Okay”

Card

Are you valid?

“Yup”

Cancel yourself, please.

“Okay”
Princeton study of Diebold

All physical locks use the same key
  ● Common for hotel mini-bars, office furniture

Implemented a voting machine virus
  ● Software update from memory card
  ● No authentication of any kind
  ● Infection can spread via memory cards
    (no networking necessary)

http://itpolicy.princeton.edu/voting/
Dutch study of Nedap ES3B

- Poor physical key security
- Easily modified ROM chips
- Easily observed RF emissions
- Demonstration: chess software

Risk 4: Procedural failures

Poll workers have many responsibilities

- Machine setup
  - Validate date, machines zeroed, etc.
- End-of-day tallying / reporting
- Unusual events
  - “Fleeing voters” (forgot to press “cast ballot”)
  - Machine / memory card failures
Webb County (Laredo) experience

March 7, 2006: primary election
First local use of ES&S DRE machines
Margin of victory in Flores v. Lopez
was \(~100\) of \(~50K\) votes (0.2%)

Significant procedural problems

*Joint work with Dan Sandler*
# Normal event logs

<table>
<thead>
<tr>
<th>Votronic</th>
<th>PEB#</th>
<th>Type</th>
<th>Date</th>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>5117865</td>
<td>161061</td>
<td>SUP</td>
<td>03/06/2006</td>
<td>16:31:12</td>
<td>01 Terminal clear and test</td>
</tr>
<tr>
<td>161126</td>
<td></td>
<td>SUP</td>
<td>03/07/2006</td>
<td>07:09:37</td>
<td>09 Terminal open</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>07:13:50</td>
<td>13 Print zero tape</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>07:15:39</td>
<td>13 Print zero tape</td>
</tr>
<tr>
<td>160973</td>
<td></td>
<td>SUP</td>
<td>03/07/2006</td>
<td>12:32:24</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>16:59:19</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:06:23</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:25:56</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:32:18</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:48:54</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:56:03</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td>161126</td>
<td></td>
<td>SUP</td>
<td>03/07/2006</td>
<td>19:01:52</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td>161126</td>
<td></td>
<td>SUP</td>
<td>03/07/2006</td>
<td>19:39:41</td>
<td>10 Terminal close</td>
</tr>
</tbody>
</table>
# Issue #1: Test votes

<table>
<thead>
<tr>
<th>Votronic</th>
<th>PEB#</th>
<th>Type</th>
<th>Date</th>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>5145172</td>
<td>161061</td>
<td>SUP</td>
<td>03/06/2006</td>
<td>15:04:09</td>
<td>01 Terminal clear and test</td>
</tr>
<tr>
<td>161126</td>
<td>SUP</td>
<td>03/06/2006</td>
<td>15:19:34</td>
<td></td>
<td>09 Terminal open</td>
</tr>
<tr>
<td>160973</td>
<td>SUP</td>
<td>03/06/2006</td>
<td>15:26:59</td>
<td></td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td>161126</td>
<td>SUP</td>
<td>03/06/2006</td>
<td>15:30:39</td>
<td></td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td>161126</td>
<td>SUP</td>
<td>03/06/2006</td>
<td>15:38:37</td>
<td></td>
<td>27 Override</td>
</tr>
<tr>
<td>03/06/2006</td>
<td>15:38:37</td>
<td></td>
<td>10 Terminal close</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Election was on 3/7
- 93 votes with the wrong dates
  - Four machines: clock probably set wrong
  - 26 machines: test votes included in final tally
    (one Republican ballot, one Democrat ballot, repeated on each test machine)
## Issue #2: Lost votes?

<table>
<thead>
<tr>
<th>Votronic</th>
<th>PEB#</th>
<th>Type</th>
<th>Date</th>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>5140052</td>
<td>161061</td>
<td>SUP</td>
<td>03/07/2006</td>
<td>15:29:03</td>
<td>01 Terminal clear and test</td>
</tr>
<tr>
<td>160980</td>
<td>SUP</td>
<td></td>
<td>03/07/2006</td>
<td>15:31:15</td>
<td>09 Terminal open</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>15:34:47</td>
<td>13 Print zero tape</td>
</tr>
<tr>
<td>160999</td>
<td>SUP</td>
<td></td>
<td>03/07/2006</td>
<td>15:56:50</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>16:47:12</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:07:29</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:17:03</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:37:24</td>
<td>22 Super ballot cancel</td>
</tr>
<tr>
<td>160980</td>
<td>SUP</td>
<td></td>
<td>03/07/2006</td>
<td>18:41:18</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>18:46:23</td>
<td>20 Normal ballot cast</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>03/07/2006</td>
<td>19:07:14</td>
<td>10 Terminal close</td>
</tr>
</tbody>
</table>

- Most machines cleared on 3/6
- Ten machines cleared and used on 3/7

Poll workers not supposed to do this!
Issue #3: Insufficient audit data

- Many machines cleared after the election
  - Only CompactFlash memory cards remained
- Many “zero tapes” were lost
- No records for “cancelled ballots”
  (Poll workers *supposed* to keep a log)
Issue #4: Unwieldy equipment

ABOUT HALF OF THE IMPOUNDLED MACHINES
Meaningless certification

- All of these systems are “certified”
  - Clear evidence of insufficient processes/laws
- Most certification documents are secret
- Testing authorities skip “hard” tests
  - Or, no evidence of doing them properly
- No consideration of development process
- No consideration of procedural difficulties
- No oversight of testing authorities
Research: Build a better machine

Step 1: Use Moore’s Law
- Computation is free
- Disk storage is infinite
- $N$ is small enough that $O(N^2)$ is still cheap

Implications?
- Never delete anything, ever
- Digitally sign everything
- Store redundant copies everywhere
Networked voting systems

Local network not Internet

I just cast a ballot!

Massive redundancy, but is it secure?
Example: “Protective counters”

Defense against ballot stuffing

- Lever machines: visible mechanical counter
- Diebold has a text file on the flash card
- Our system: records every vote ever cast
Network data handling

Two classes of data: events and votes

- Events are public: sign and log everything
- Timeline entanglement to preserve history *(Maniatis and Baker ’02)*

Need to preserve anonymity of votes

- Option 1: Assume a trusted network
- Option 2: Encrypt the votes
Network vote storage?

Issue: who gets to decrypt?

- Requirements vary from state to state

If local precinct needs vote totals

- Homomorphic encryption (allows computation of vote totals)

If local precinct needs individual ballots

- Verifiable mix networks

Work in progress…
Voting protocol

1. Authorize vote
2. Broadcast result (encrypted)
3. Console decrypts and tallies

Others just record

CONTROLLER:
Machine 1 is authorized to cast this ballot:
<< BALLOT >>
<< NONCE >>
Plaintext votes (local store)

Votes should not be in the order cast
- Option #1: randomize the order
- Option #2: sort the ballots (Chaum)

Simple solution: One sorted list per election
- Election IDs need to be globally unique
Pragmatic benefits

Admin console shows status of all machines
- Votes cast, battery running low, etc.

Admin console tells poll workers what to do
- Less opportunity for poll-worker error

Voting machines are interchangeable
- Add/remove machines on the fly
Software tampering?

Secure bootstrapping / attestation

- Machines can “challenge” each other
- Just log the result, resolve conflicts later

Burn software to ROM (not Flash)

- Ballot definition downloaded for each vote
  - State-specific rules part of the ballot definition
- Less need for software upgrades

And, of course, VVPAT

- Printed ballots should take legal precedence
Software engineering

Strong type systems are security mechanisms
- No concerns about buffer overflows
- Narrow public interfaces between modules
- Easy to verify using `grep`

Less is more
- Diebold: ~35K lines of C++ (plus Windows CE)
- Yee ‘06: 400 lines of Python (pygame, SDL, …)
- Our current prototype: ~4K lines of Java
Recount / auditing process

1. Tally the votes from the admin consoles
2. Sample the VVPAT: ensure consistency
3. Sample the machines: ensure consistency

If inconsistencies occur, study entire precinct
- Computer-aided auditing
Related: hardware separation

Sastry et al., *Designing Voting Machines for Verification* (Usenix Security ’06).

- Property 1: No voter session can interfere with a previous session.
- Property 2: A ballot cannot be cast without voter consent.
- Core idea: separate HW modules, reset after each vote.
Human factors matters
Human factors matters

If it’s not usable, it’s not secure.

- Necessity: human subject experiments

Example question: do people read VVPAT?

- Need a voting machine that lies!

Joint work with Mike Byrne (Rice Psychology)

- Measure usability of voting UI features

- Poll worker usability as well
HF work in progress

- Paper ballots are most consistent across different demographics
- Education / prior experience don’t help
- Error rates are stunning (1% or worse)
ACCURATE Voting Center

NSF research center, $7.5M

- PIs at U.C. Berkeley, U. Iowa, Johns Hopkins, Rice, SRI, Stanford

Research into better voting systems

- Cryptographic protocols
- Verifiable software
- Tamper resistance
- Human factors
- Policy implications

accurate-voting.org
Conclusions

Current DRE voting systems have real problems

- Independent certification is (currently) meaningless
- Significant failures observed in the field
- Non-trivial margins of error

Good science can improve the situation

- Better software engineering
- Better auditability / fault tolerance
- Better human factors