VoteBox: a verifiable, tamper-evident electronic voting system

Daniel R. Sandler
Rice University

February 17, 2009 | The Johns Hopkins University

Talk outline

Background

Trustworthiness of electronic voting machines

Why it's worth improving them

The design of VoteBox

Durability and audit

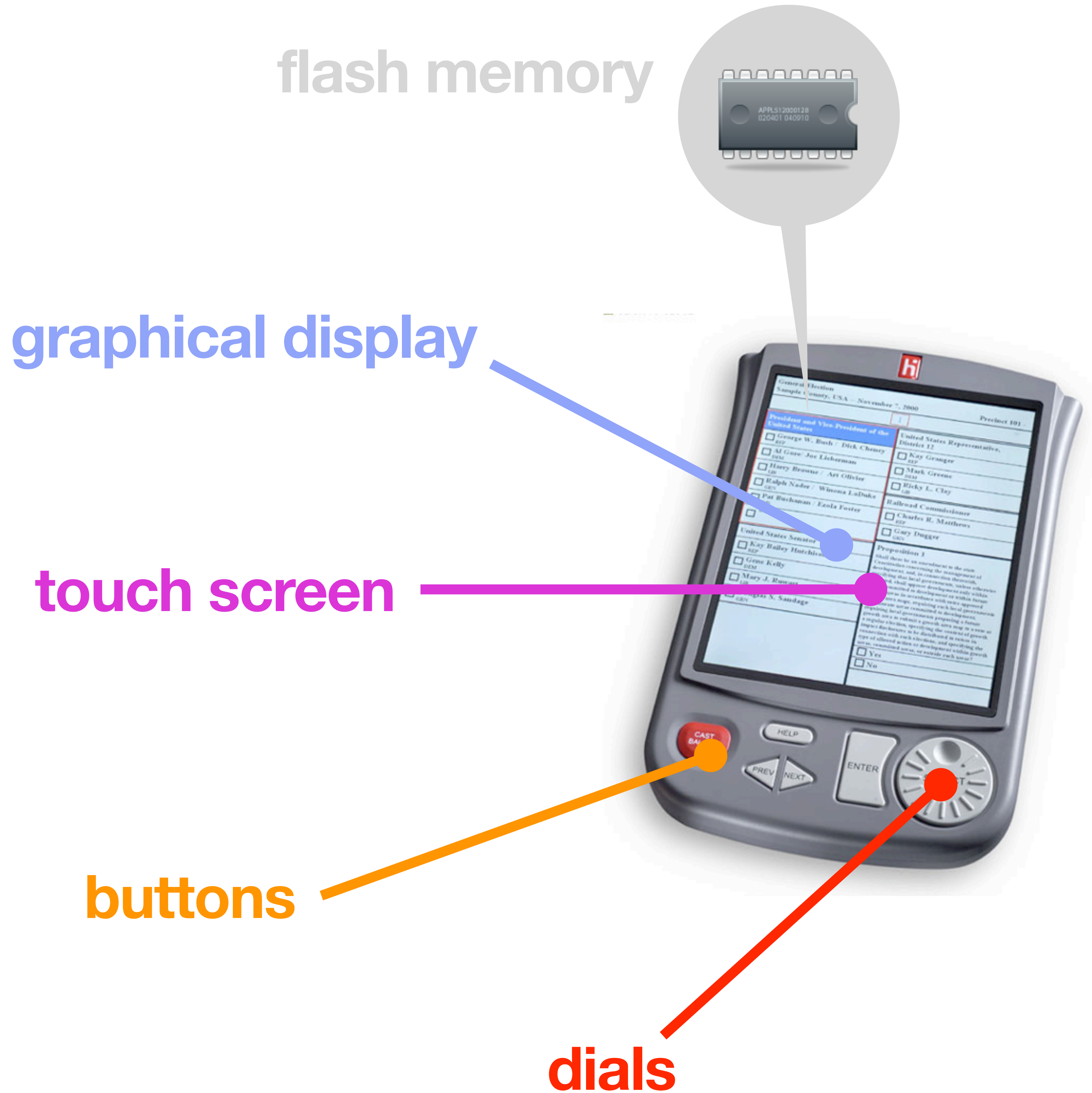
Ballot casting assurance

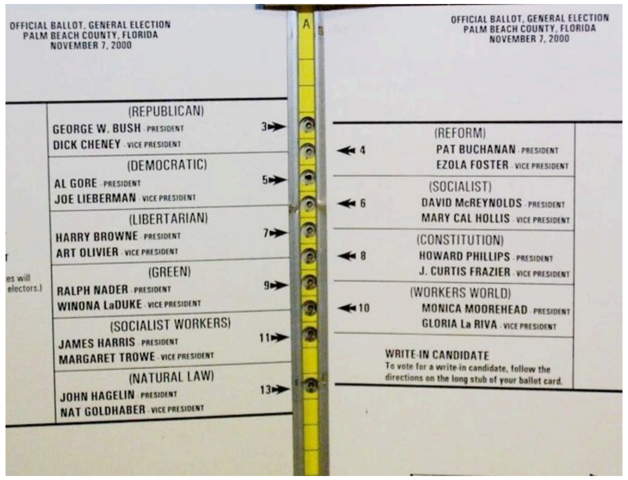
Beyond

1. Background

DRE voting machines

(Direct Recording Electronic)


flash memory
graphical display
touch screen
buttons
dials
HELP
PREV
NEXT
ENTER



US Presidential election (2000)

HAVA (2002)

DREs discredited

High-profile failures in real elections

A few examples:

2006: Sarasota, FL undervoting

~18,000 ballots blank in the congressional race (~15%)
margin of victory: 369 votes

2008: video documentation of “vote flipping”

touch-screen calibration? buggy input filters?

Ongoing: long lines due to complex set-up, equipment problems, etc.

DREs discredited

Software bugs & design flaws identified by e-voting researchers

2003 Analysis of Diebold AccuVote TS

- **Leaked source code analyzed [Kohno et al. 2004]**
- **Poor software engineering, incorrect cryptography, vulnerable to malicious upgrades, multiple voting**

2006 "Voting-machine virus" developed

- **Self-propagating malicious upgrades that spread from machine to machine, altering votes and leaving no trace [Feldman et al. 2006]**

DREs discredited

Software bugs & design flaws identified by e-voting researchers

2007 Involvement by computer scientists in statewide voting systems audits

groundbreaking access to source code of commercial voting systems

Top-To-Bottom Review (California)

- All machines certified for use in CA found to have serious bugs & be vulnerable to attack
- Viral-style attacks found in all systems

EVEREST study (Ohio)

- All machines certified in OH found vulnerable (validating CA-TTBR)
- Showed that hundreds of votes were lost in 2004

malfunctions

could result in changed or lost votes

design flaws

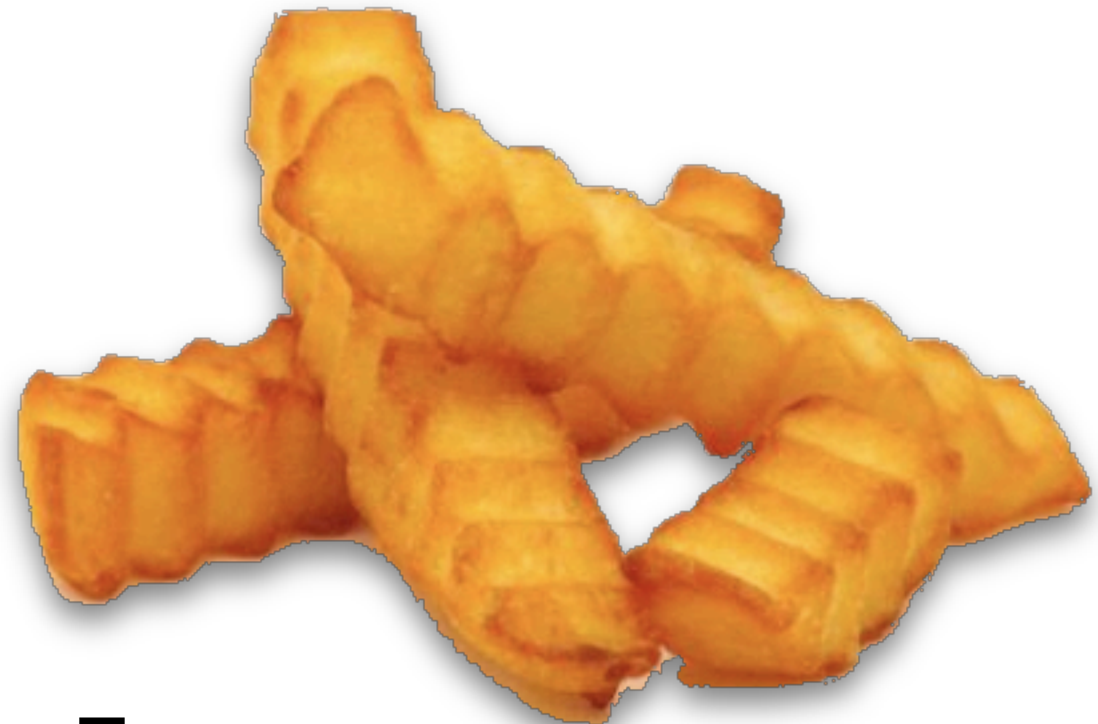
could let attackers alter the election outcome without leaving evidence

Result:
undermined trust in elections

Trash

?

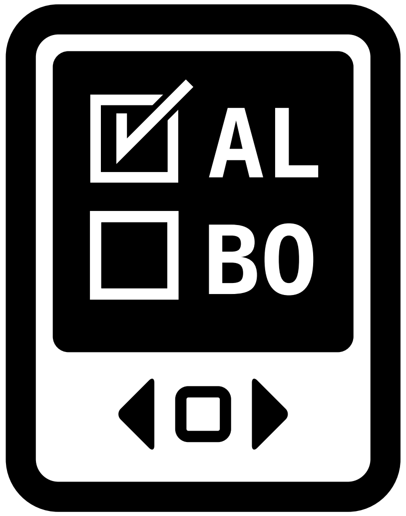
voters **prefer** electronic voting



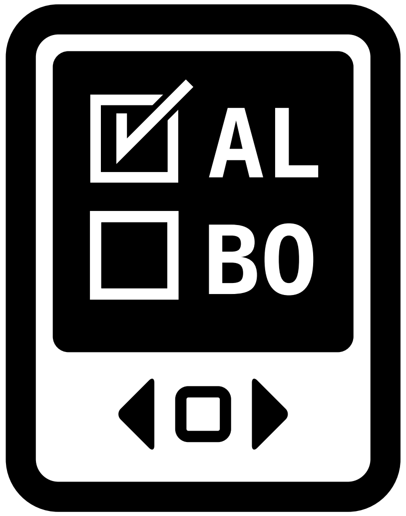
S. P. Everett, K. K. Greene, M. D. Byrne, D. S. Wallach, K. Derr, **D. R. Sandler**, and T. Torous. *Electronic voting machines versus traditional methods: Improved preference, similar performance.* In CHI 2008.

legitimate benefits

accessibility
feedback
flexibility
satisfaction



can we design a better DRE?



"better" = ?

goals

1. **resistance to failure & tampering**
   essential vote data should survive hardware failure, poll worker mistakes, attempts to attack the system

goals

2. **tamper-evidence**

   if we are unable to prevent data loss, we must always be able to detect the failure

goals

3. verifiability

two useful properties:

cast-as-intended

"Was my vote recorded faithfully?"
very, very hard for DREs to satisfy

counted-as-cast

"Has my vote been tallied correctly?"
can be somewhat addressed with recounts

goals

resistance to failure & tampering

prevent or minimize data loss

tamper-evidence

if resistance is futile

verifiability

cast-as-intended; counted-as-cast

(DRE user experience)

a computer science problem

resistance to failure & tampering

replication; gossip

tamper-evidence

secure logs

Auditorium

verifiability

cryptography

Ballot challenge

2. VoteBox

A field study.

Webb County, TX

March 7, 2006:

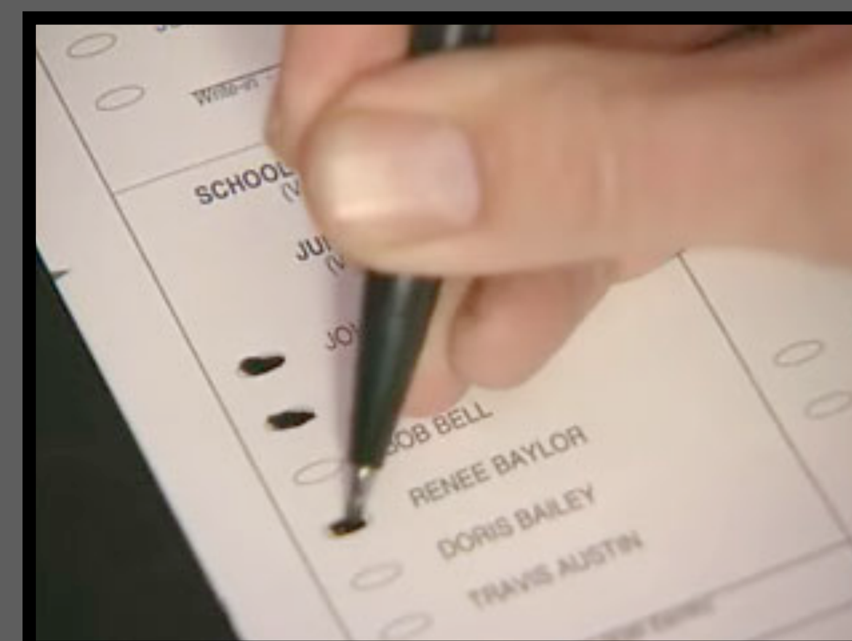
Democratic primary election

(County's first use of DREs)

An unusual situation

Voters given a choice:

OR



DRE

(ES&S iVotronic)

Paper

(central ES&S op-scan)

Flores v. Lopez

~50,000 votes cast

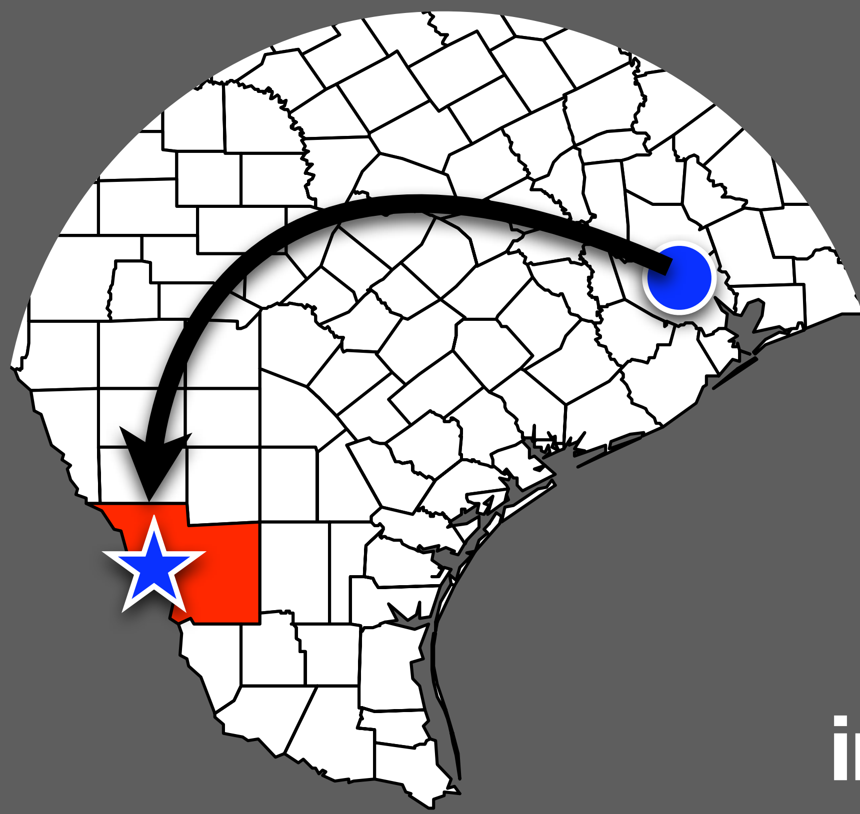
Margin of victory: **~100 votes**

The loser suspected the DREs

…because he looked better on paper

Lawsuit

Bring in the experts.



initial investigation: gathering data

(April 2006)

Webb Co. data

Raw binary data from Compact Flash cards

Opaque, undocumented format

Text output from tabulation process

`IMAGELOG.TXT` (cast ballots)

`EVENTLOG.TXT` (more on that later)

What we found

A smoking gun?

Evil voting machines?

HACKS??

inherently difficult to find evidence with DREs!

What we (really) found

Anomalies in the event logs

Per-machine records

Captured during machine run time

Transferred to tabulator (`IMAGELOG.TXT`)

A timeline of important election events

e.g. “terminal open,” “ballot cast,” …

Example event log

Votronic	PEB#	Type	Date	Time	Event
5140052	161061	SUP	03/07/2006	15:29:03	01 Terminal clear and test
	160980	SUP	03/07/2006	15:31:15	09 Terminal open
			03/07/2006	15:34:47	13 Print zero tape
			03/07/2006	15:36:36	13 Print zero tape
	160999	SUP	03/07/2006	15:56:50	20 Normal ballot cast
			03/07/2006	16:47:12	20 Normal ballot cast
			03/07/2006	18:07:29	20 Normal ballot cast
			03/07/2006	18:17:03	20 Normal ballot cast
			03/07/2006	18:37:24	22 Super ballot cancel
			03/07/2006	18:41:18	20 Normal ballot cast
			03/07/2006	18:46:23	20 Normal ballot cast
	160980	SUP	03/07/2006	19:07:14	10 Terminal close

Problem #1
Logs starting mid-day

```
03/07/2006 15:29:03  Terminal clear and test
03/07/2006 15:31:15  Terminal open
```

Polls opened around 7 AM across Webb Co.

What happened between 7 and 3:30?
Lost votes?

Problem #2
Election events on wrong day

A normal voting pattern:

```
Votronic  PEB#    Type    Date         Time      Event

5142523  161061  SUP   02/26/2006 19:07:05   01 Terminal clear and test
         161115  SUP   03/06/2006 06:57:23   09 Terminal open
                       03/06/2006 07:01:47   13 Print zero tape
                       03/06/2006 07:03:41   13 Print zero tape
         161109  SUP   03/06/2006 10:08:26   20 Normal ballot cast

                            [... 9 more ballots cast ...]

         161115  SUP   03/06/2006 19:29:00   27 Override
                       03/06/2006 19:29:00   10 Terminal close
```

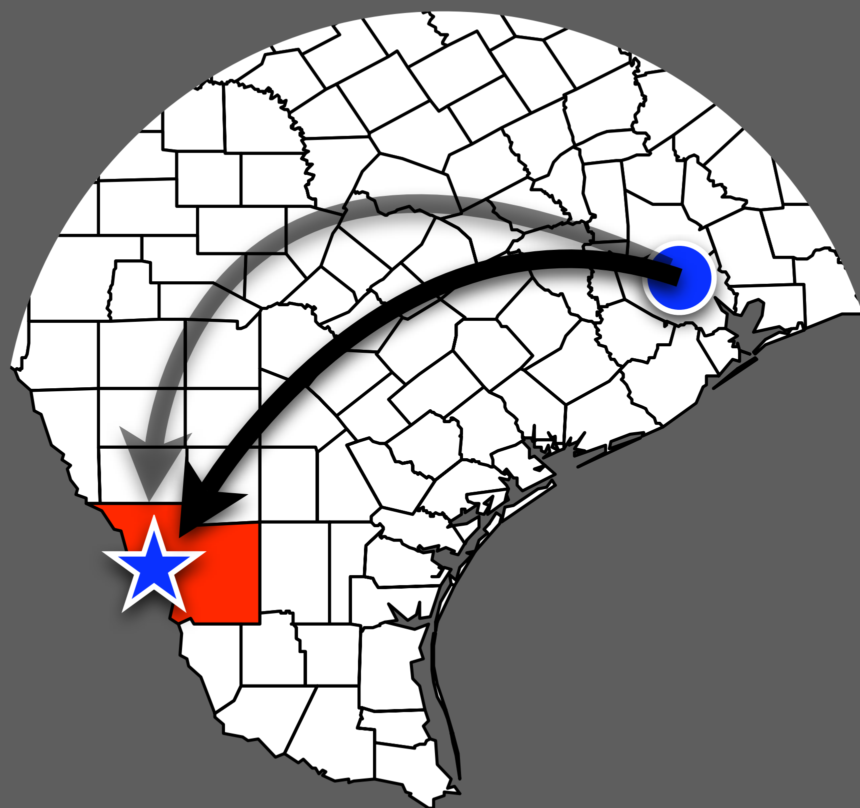
The election was held on 03/07!
Ballot box stuffing the day before?

A different pattern:

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

26 machines with exactly two ballots cast the day before (always for the same guy)

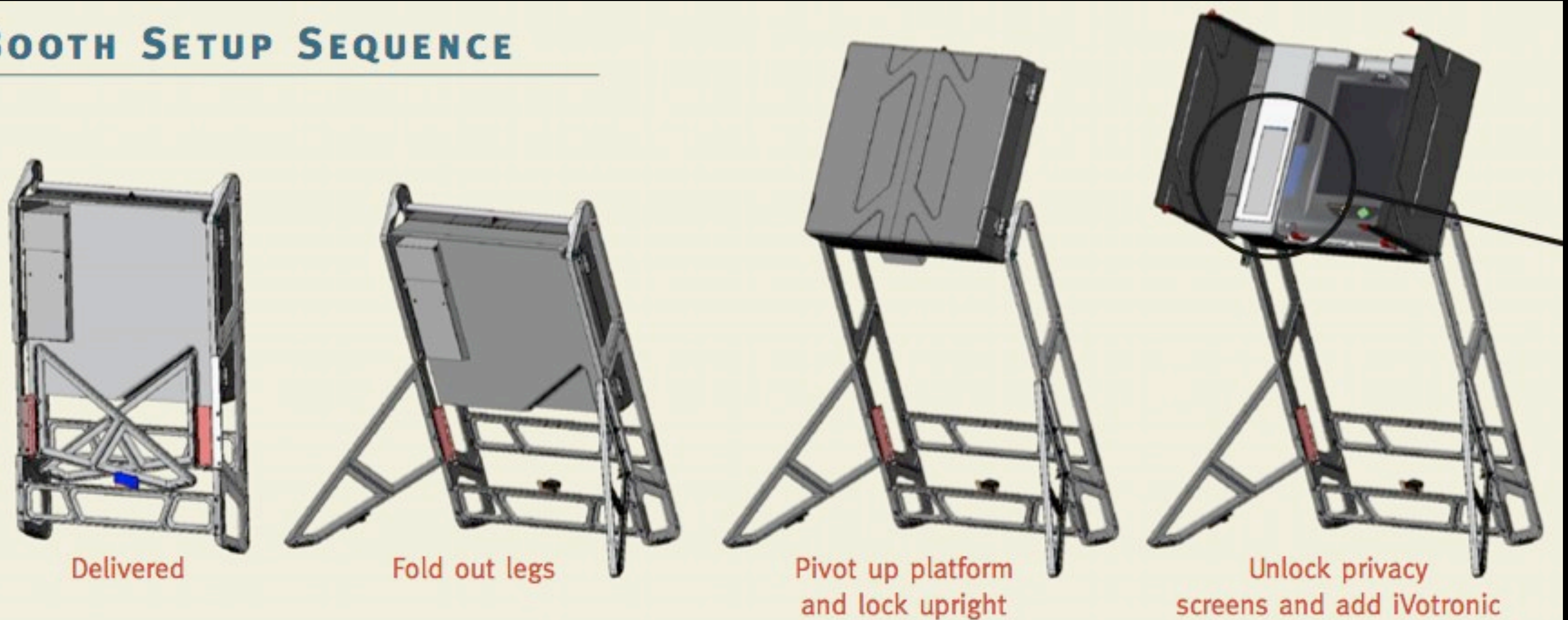
We learned that these might be "logic and accuracy test" votes, erroneously included in the tally



initial investigation

follow-up trip: direct inspection

Booth Setup Sequence



History for Laredo, TX

Tuesday, April 25, 2006 — View Current Conditions

Daily Summary

« Previous Day | April | 25 | 2006 | Go | Next Day »

Daily | Weekly | Monthly | Custom

	Actual:	Average :	Record :
Temperature:			
Mean Temperature	87 °F / 30 °C	-	
Max Temperature	101 °F / 38 °C	85 °F / 29 °C	101 °F / 38 °C (2006)
Min Temperature	73 °F / 22 °C	64 °F / 17 °C	55 °F / 12 °C (2001)

source: wunderground.com

Findings

Machines containing only two votes

Hardware clock appeared normal

Most likely L&A test votes

Others

Hardware clock set incorrectly

…just enough to account for anomaly

This is not proof of correct behavior!

Problem #3
Insufficient audit data

We couldn't collect data from every machine

Many were **cleared** after the election

(Poll workers not supposed to do this!)

Paper records missing

Zero tapes

Cancelled ballot logs

Procedural errors by administrators, pollworkers

(but the machines didn't help)

“Mistakes were made.”

Mistakes were made

Violations of election procedures

Counting test votes in final results

Loss of zero tapes and other paper logs

Erasure of some machines

Local (mis)configuration

Hardware clocks set wrong

These things cast doubt on the results

Honest mistakes
or illegitimate votes?

No way to be sure.
Believable audits
impossible.

Research goals

Make it easier to audit results after the election

Make it harder to make mistakes on election day

In particular:

Prove

every vote tallied is valid

every valid vote is present

Tolerate

accidental loss/deletion of records

election-day machine failure

How?

Connect the machines together.

“The Auditorium”

Auditorium’s approach

Store everything everywhere

Massive **redundancy**

Stop trusting DREs to keep their own audit data

Link all votes, events together

Create a **secure timeline** of election events

Tamper-evident proof of each vote’s legitimacy

D. Sandler and D. S. Wallach. **Casting Votes in the Auditorium.** In Proceedings of the 2nd USENIX/ACCURATE Electronic Voting Technology Workshop (EVT’07).

Ingredient: hash chains

```
             “Machine turned on” (HASH = 0x1234)
“Cast a vote after event 0x1234” (HASH = 0xABCD)
“Cast a vote after event 0xABCD” (HASH = 0xBEEF)
 “Turned off after event 0xBEEF” (HASH = 0x4242)
```

A hash-chained secure log

Every event includes the cryptographic hash (e.g. SHA1) of a previous event

[Schneier & Kelsey ’99]

Result: provable order

If **Y** includes H(**X**), then **Y** must have happened after **X**

Any individual change to the log

invalidates all later hashes (breaks the chain)

To alter, insert, or delete a single record

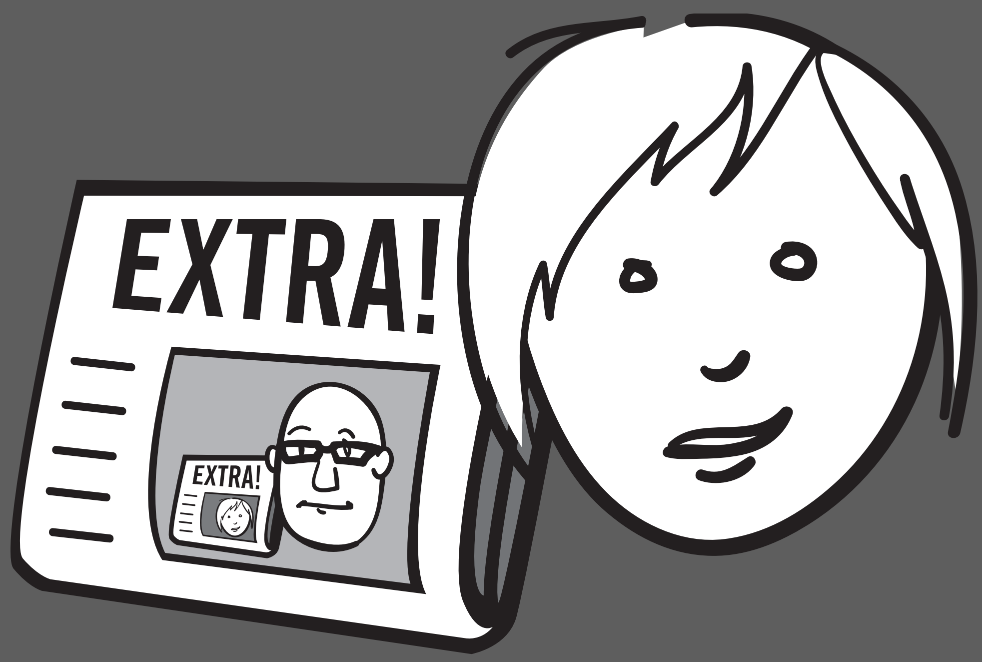
you must alter every subsequent event as well

Ingredient: timeline entanglement

Entanglement = “chain with hashes from others”

Result: event ordering **between** participants

[Maniatis & Baker ’02]



Malicious machines can’t retroactively alter their own logs

it would violate commitments they have already exchanged with others

Ingredient: broadcast

All-to-all communication

All messages signed & sent to every VoteBox

Each machine records each message independently

→ massive **replication**

O($\boldsymbol{N}^2$), but ***N*** is small in a polling place

Mechanism for entanglement

When publishing a new message,

include hashes of recent messages in the log

(regardless of their origin!)

Broadcast entanglement =

Auditorium

A pragmatic benefit

The supervisor console

Assistance for poll workers

Shows status of all machines

Votes cast, battery running low, etc.

Helps conduct the election

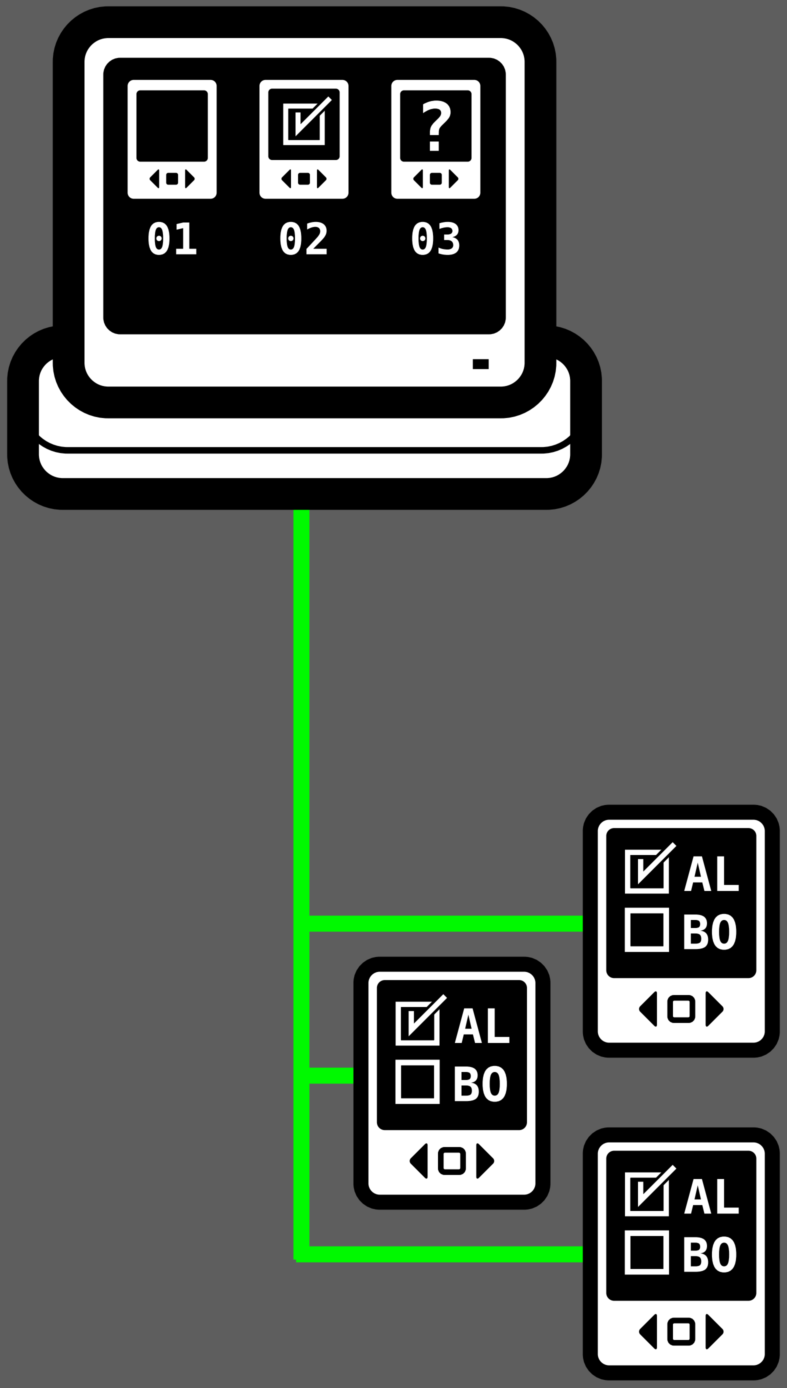
Open/close polls, authorize machines to cast ballots

Less opportunity for poll-worker error

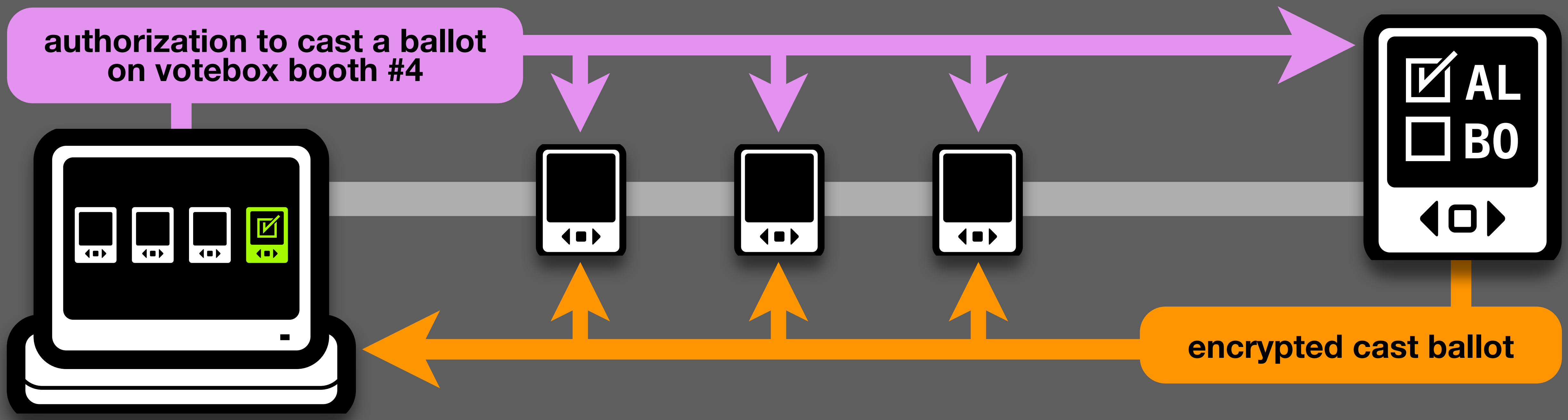
Ballots distributed over the network

Booths are **stateless**, interchangeable

(Supervisor can have a spare as well)



Voting in the Auditorium



voting machines connected in a private polling-place network

all election events are signed and broadcast

each broadcast is logged by every machine

isolated failures won't lose data

secure logs provide a global timeline for meaningful audits

"Everyone hears everything in the Auditorium."

Unusual prior art

The Papal Conclave

Proceedings **closed to outsiders**

All ballots cast **in plain view**

All ballots **secret**

How do you audit a secure log?

> *"Audit logs are useless unless someone reads them. Hence, we first assume that there is a software program whose job it is to scan all audit logs and look for suspicious entries."*
>
> **—Schneier & Kelsey '99**

Where is that program?

"suspicious" is domain-specific

QUERIFIER: an audit log analysis tool

Predicate logic for expressing rules over secure logs

Key predicate: "precedes" — requires graph search

Querifier runs on a complete log ("OK" / "Violation")

or iteratively on a growing log ("OK so far" / "Violation")

D. Sandler, K. Derr, S. Crosby, and D. S. Wallach. **Finding the evidence in tamper-evident logs.** In Proceedings of the 3rd International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'08).

Ballots.

privacy

Privacy

Secure log of votes could be a problem

When decrypted for tallying, votes are exposed in **order**

An observer could match them with voters

Loss of privacy → bribery & coercion*

Anonymity through clever ballot ordering

re-encryption mixnets

lexicographic sorting

These would still require the ballots to be removed from the ordered audit logs

Ballots in VoteBox

logically, a cast ballot is a vector of counters

one per *candidate*

***e.g.,* for one race with three candidates:**

$$\text{ballot} = (a, b, c) \qquad\qquad a, b, c \in \{ 0, 1 \}$$

ballots may therefore be summed

$$\text{tally} = \sum \text{ballot}_i = (\sum a_i , \sum b_i , \sum c_i)$$

Encryption

Ballots should be sealed

protected from prying eyes

once cast, they should be readable only by the parties trusted to count them

But how do we count them?

Remember, we don't want to decrypt them in order

Homomorphic encryption

An encryption scheme with a special property

mathematical operations can be performed on ciphertexts, the result of which is a valid ciphertext

We can use this to tally without decrypting

e.g.,

$$\mathrm{E}(x) \odot \mathrm{E}(y) = \mathrm{E}(x + y)$$

for some homomorphic operation "$\odot$"

Homomorphic ElGamal does this nicely

Other research voting systems use this cryptosystem

Adder [Kiayias et al. '06]; **Civitas** [Clarkson et al. '08]; **Helios** [Adida '08]

ElGamal encrypted ballots

Encryption & decryption:

$$E(c, r, g^a) = \langle g^r, (g^a)^r f^c \rangle$$

$$D(\langle g^r, g^{ar} f^c \rangle, a) = \frac{g^{ar} f^c}{(g^r)^a} = f^c$$

Homomorphic property using multiplication:

$$\langle g^{r}, g^{ar} f^{c} \rangle \cdot \langle g^{r'}, g^{a'r'} f^{c'} \rangle = \langle g^{r+r'}, g^{ar+a'r'} f^{c+c'} \rangle$$

f, g	group generators
c	plaintext (counter)
r	random (chosen at encryption time)
a	(private) decryption key
g^a	(public) encryption key

cast-as-intended

How can I be sure my vote is faithfully captured by the voting machine?

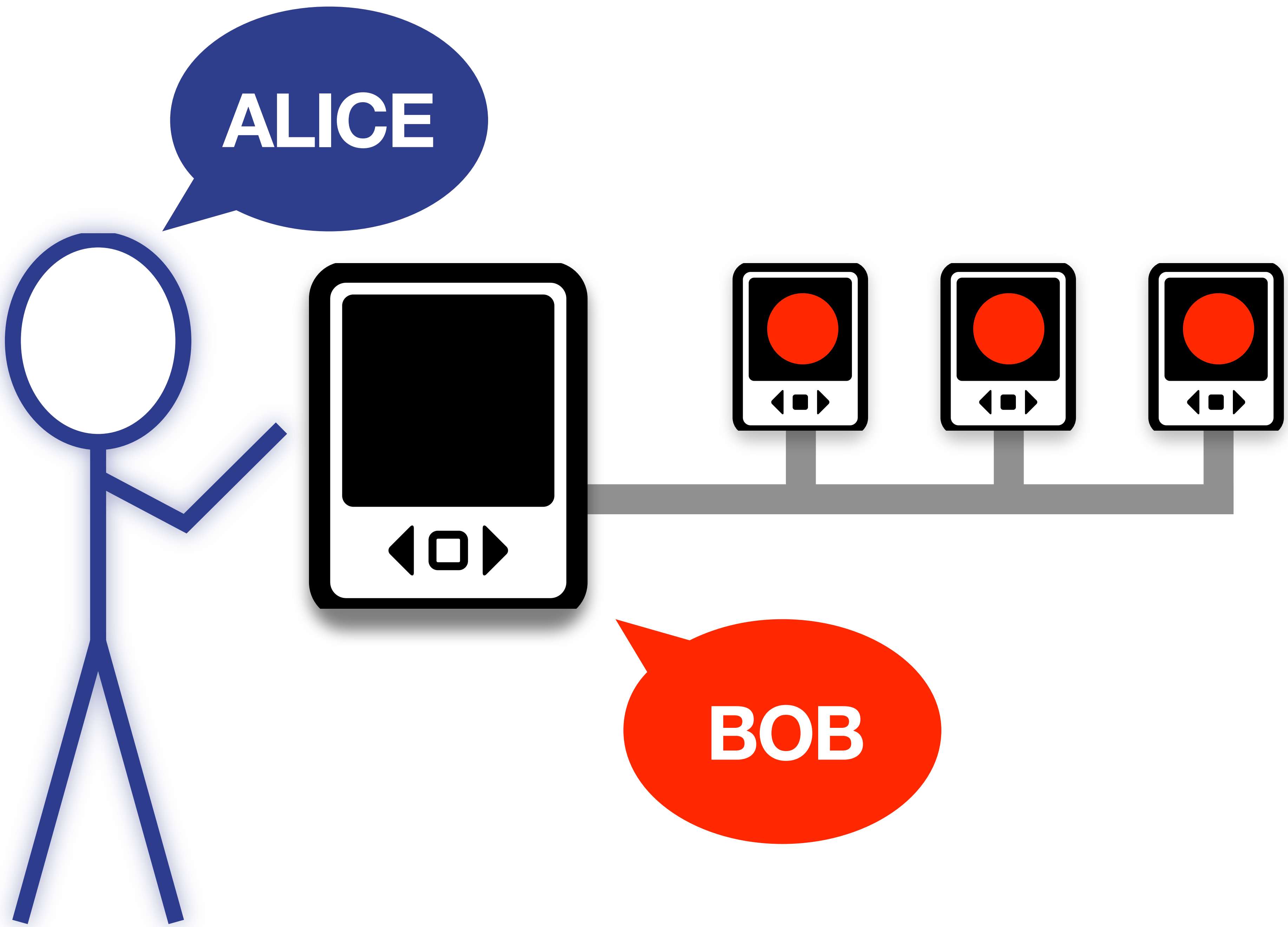
"software independence"

an *undetected* system problem cannot create an *undetectable* change in the results

or,
equipment failures can't affect the result

paper—directly inspect the ballot before casting
electronic—?
current DREs fail this test miserably

polling place



this doesn't work:

"logic & accuracy testing"

this is helpful:

trusted hardware

VoteBox's approach:

ballot challenge

ballot challenge

a technique due to [Benaloh ’07]

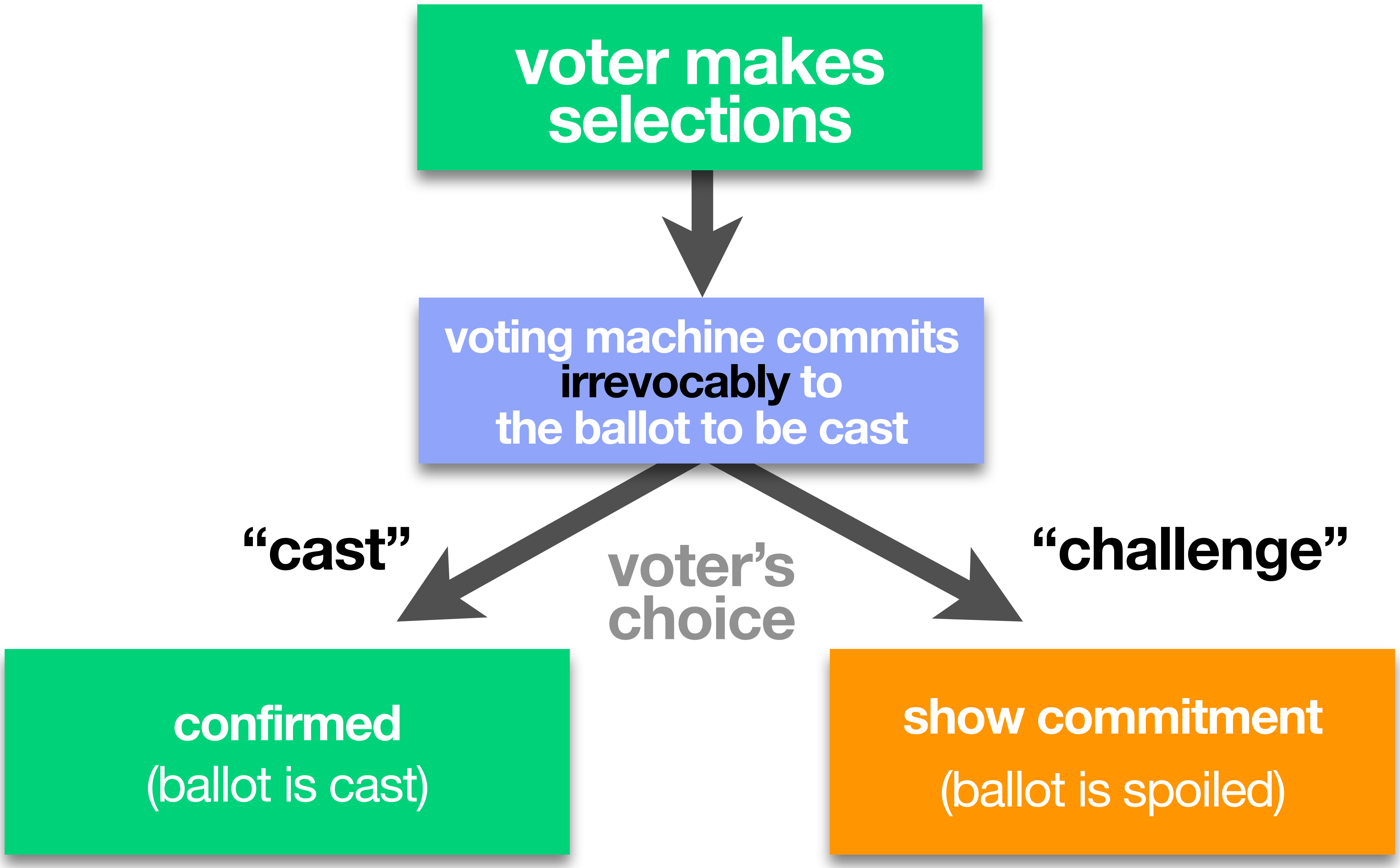
at the end of the voting session:

1. force the machine to **commit** to the ballot it is about to cast
2. the voter chooses to **cast** the ballot or **challenge** the machine to reveal its commitment

the voting machine cannot distinguish this from a real vote

no artificial L&A testing conditions

ballot challenge



ballot commitment

What is the commitment?

How do we force the machine to produce proof of what it's about to cast on the voter's behalf?

Benaloh's proposal

print the encrypted ballot behind an opaque shield

You can't see the contents, but you can see the page

the computer cannot "un-print" the ballot

How do you test the commitment?

Decrypt it.

But decryption requires the private key for tabulating the whole election!

ElGamal encrypted ballots

More than one way to decrypt a counter:

$$E(c, r, g^a) = \langle g^r, (g^a)^r f^c \rangle$$

$$\left.\begin{aligned} D(\langle g^r, g^{ar} f^c \rangle, a) &= \frac{g^{ar} f^c}{(g^r)^a} = f^c \\ D(\langle g^r, g^{ar} f^c \rangle, r) &= \frac{g^{ar} f^c}{(g^a)^r} \end{aligned}\right\} = f^c$$

f, g	group generators
c	plaintext (counter)
r	random (chosen at encryption time)
a	(private) decryption key
g^a	(public) encryption key

challenging the machine

When challenged, the machine must reveal *r*

We can then decrypt this ballot (only) and see if it's what we expected to see

In Benaloh, the encrypted ballot is on paper

An **irrevocable** output medium

decrypting requires additional equipment

VoteBox happens to have its own irrevocable publishing system

One that doesn't run out of ink or paper

Auditorium.

Challenges in Auditorium

When challenged,

a VoteBox must **announce *r* on the network**

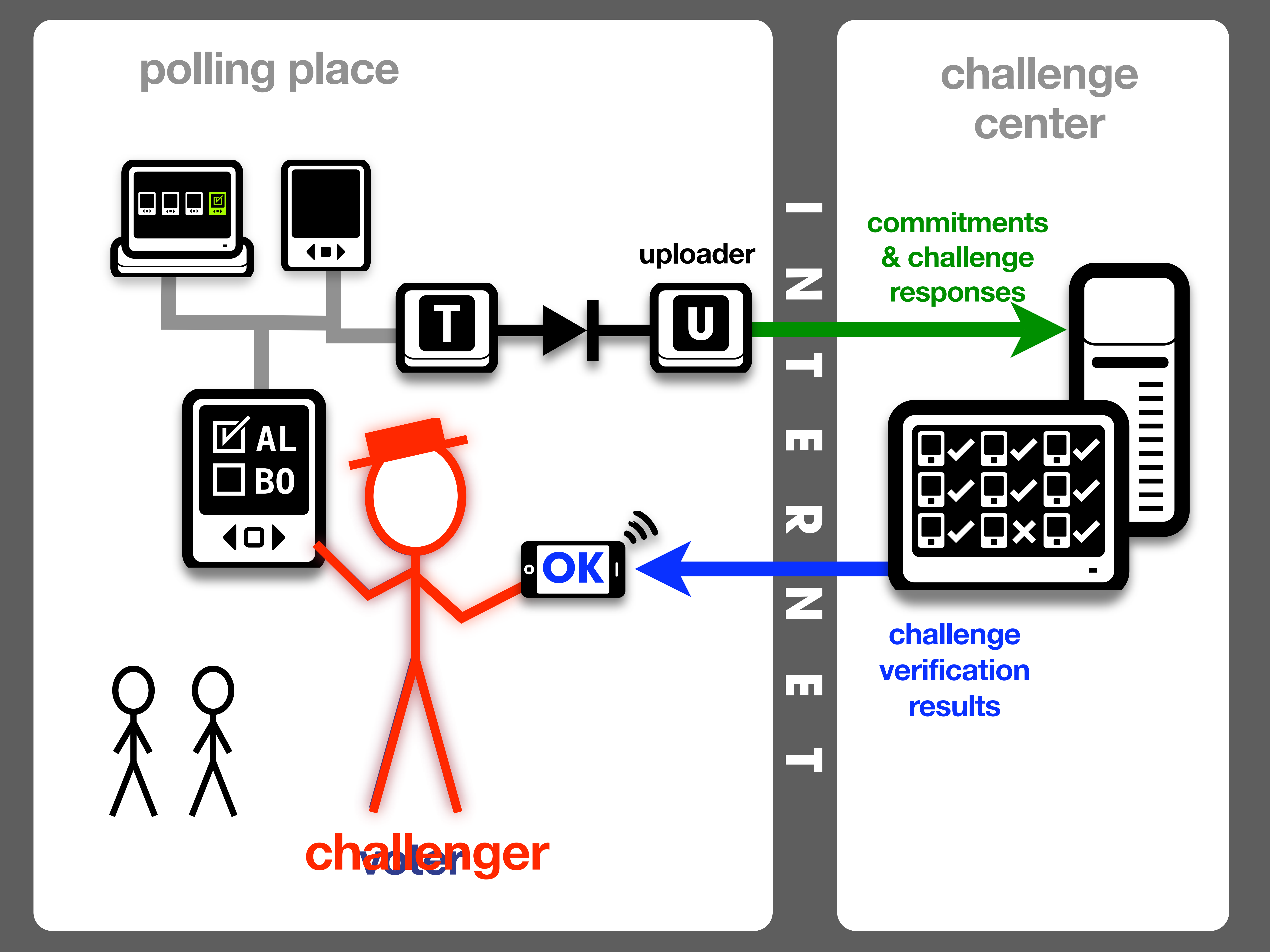
Irrevocable thanks to the properties of Auditorium

We still need help decrypting the commitment, even given ***r***

If we are careful, we can send challenges offsite

Allow a third party to assist in verifying the challenge

Trusted by the challenger!



polling place
challenge center
uploader
T
U
INTERNET
commitments
& challenge
responses
AL
BO
OK
challenge
verification
results
challenger
voter

Ballot challenges:
cast-as-intended verification
preserving privacy
without artificial test conditions.

3. Conclusion

why?

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

VoteBox integrates these techniques in a **single system.**

Auditorium (Sandler et al.)
robustness, tamper-evidence

Ballot challenge (new adaptation of Benaloh)
verifiability

Other techniques
Smaller TCB through pre-rendered UI [Yee ’06]

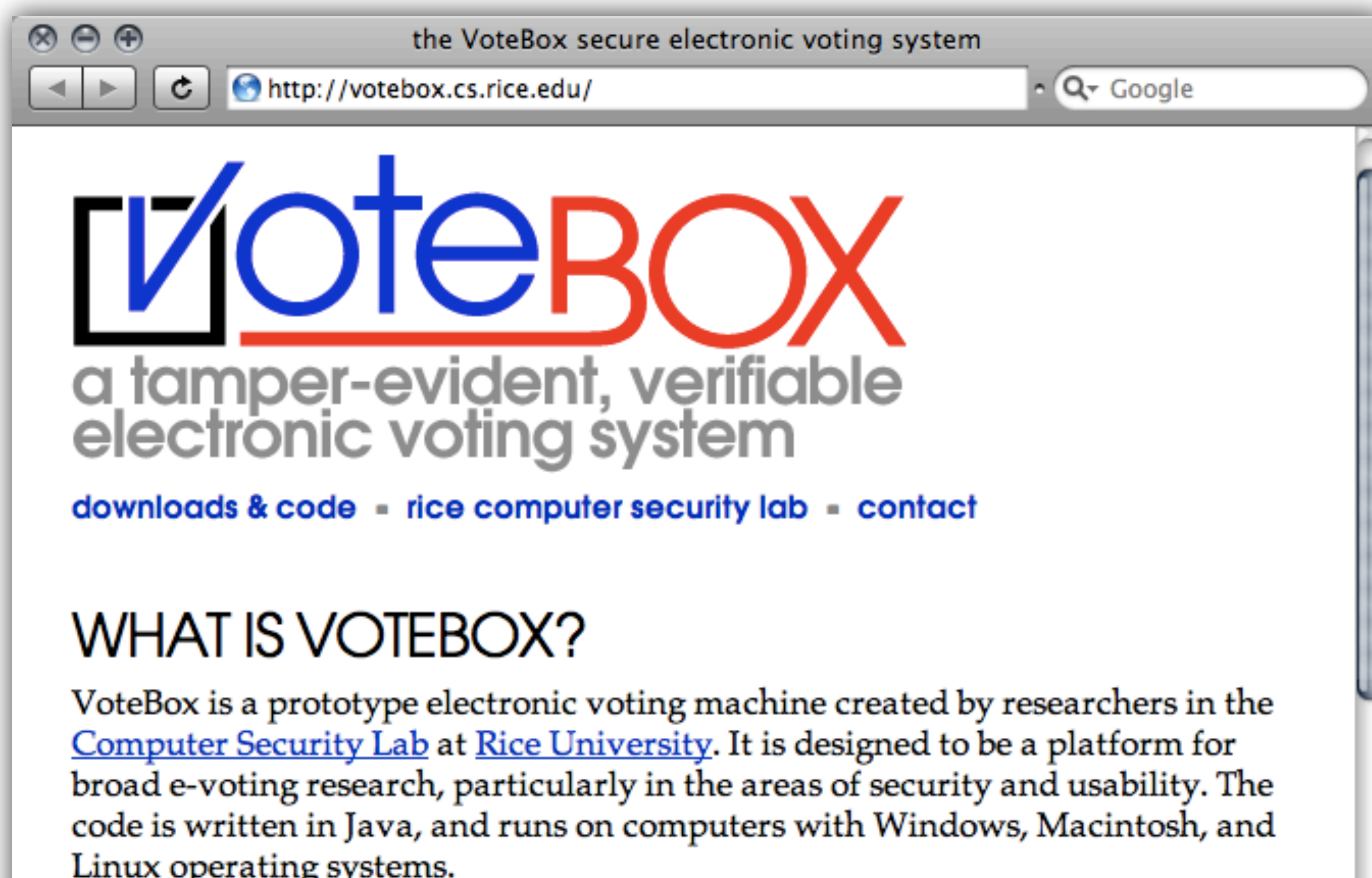
D. R. Sandler, K. Derr, D. S. Wallach. **VoteBox: A tamper-evident, verifiable electronic voting system.** In USENIX Security 2008.

platform

VoteBox is open-source

votebox.cs.rice.edu & code.google.com/p/votebox

suitable for further research, HCI experiments, class projects, security analysis



HCI research

Platform for human factors research & experimentation

VoteBox's ballot designed jointly with CHIL

VoteBox-HF includes extensive instrumentation for HCI work

Questions answered include:

"Do DREs improve performance?"

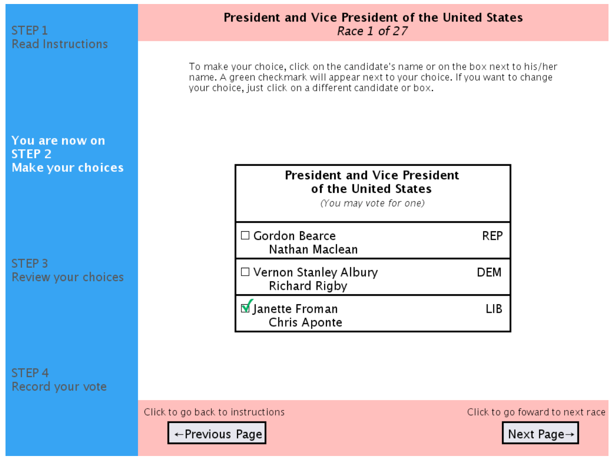
"Do voters notice if DREs malfunction?"

Research output

workshop papers, journal articles, conferences (CHI), two theses

Collaboration ongoing

chil
THE COMPUTER-HUMAN INTERACTION LABORATORY AT RICE UNIVERSITY



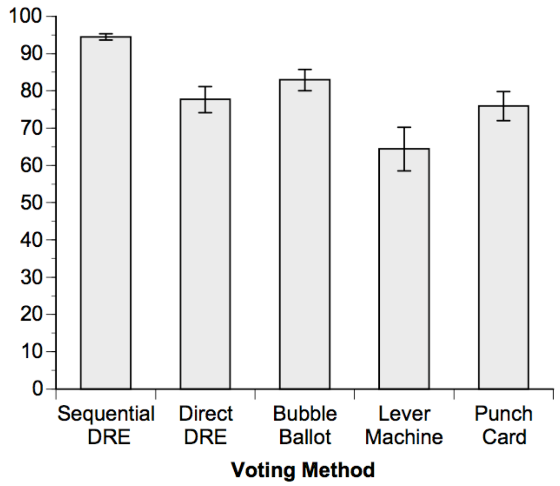
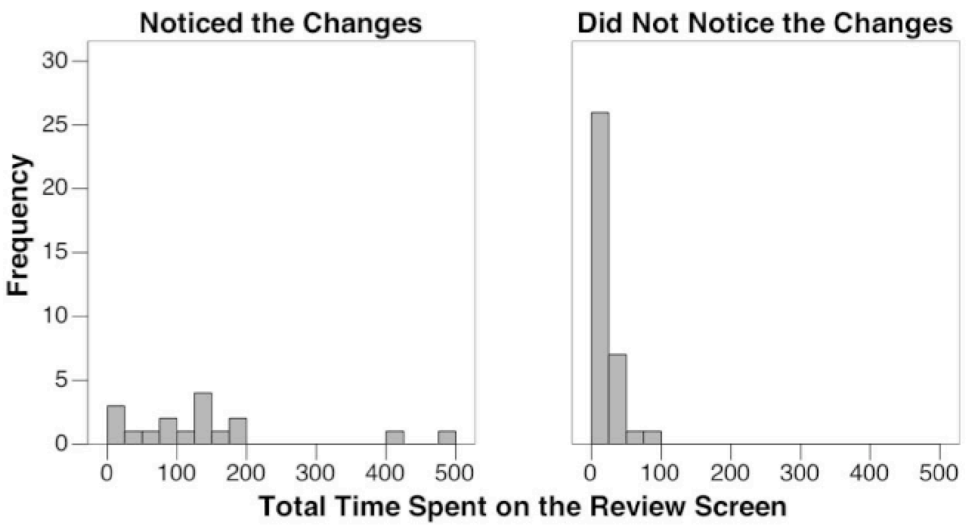


Figure 5. Mean SUS ratings by voting method.



thanks

co-authors

Dan S. Wallach (adviser); Kyle Derr

students who have worked on VoteBox

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