# Civitas

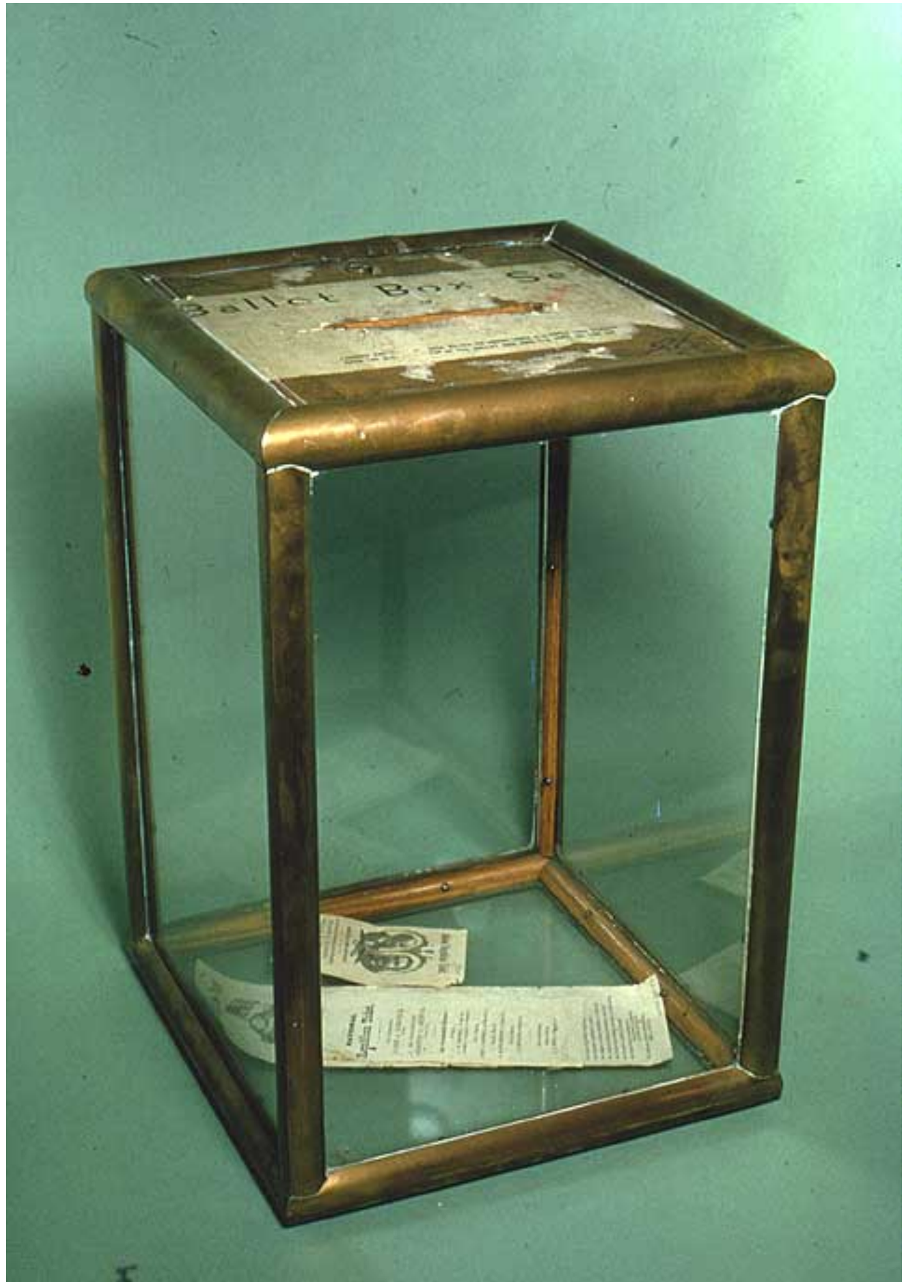
## Verifiability and Coercion Resistance for Remote Voting

Michael Clarkson
Cornell University

15th International School on Foundations of Security Analysis and Design
University Residential Center of Bertinoro, Italy
September 4, 2015

# Secret Ballot


LONGIN IIIV


Ballot Box


OFFICES
HANDICAPPED
ACCESS MAC
ASK AN INSPECTOR F
ASSISTANCE IN LOWER
THE VOTING PANEL
FOR EASIER ACCESS
AUTOMAT


VOTOMATIC
GENERAL BALLOT—ALAMEDA COUNTY
INSTRUCTIONS
TURN OVER FOR PAGE
INSTRUCTIONS

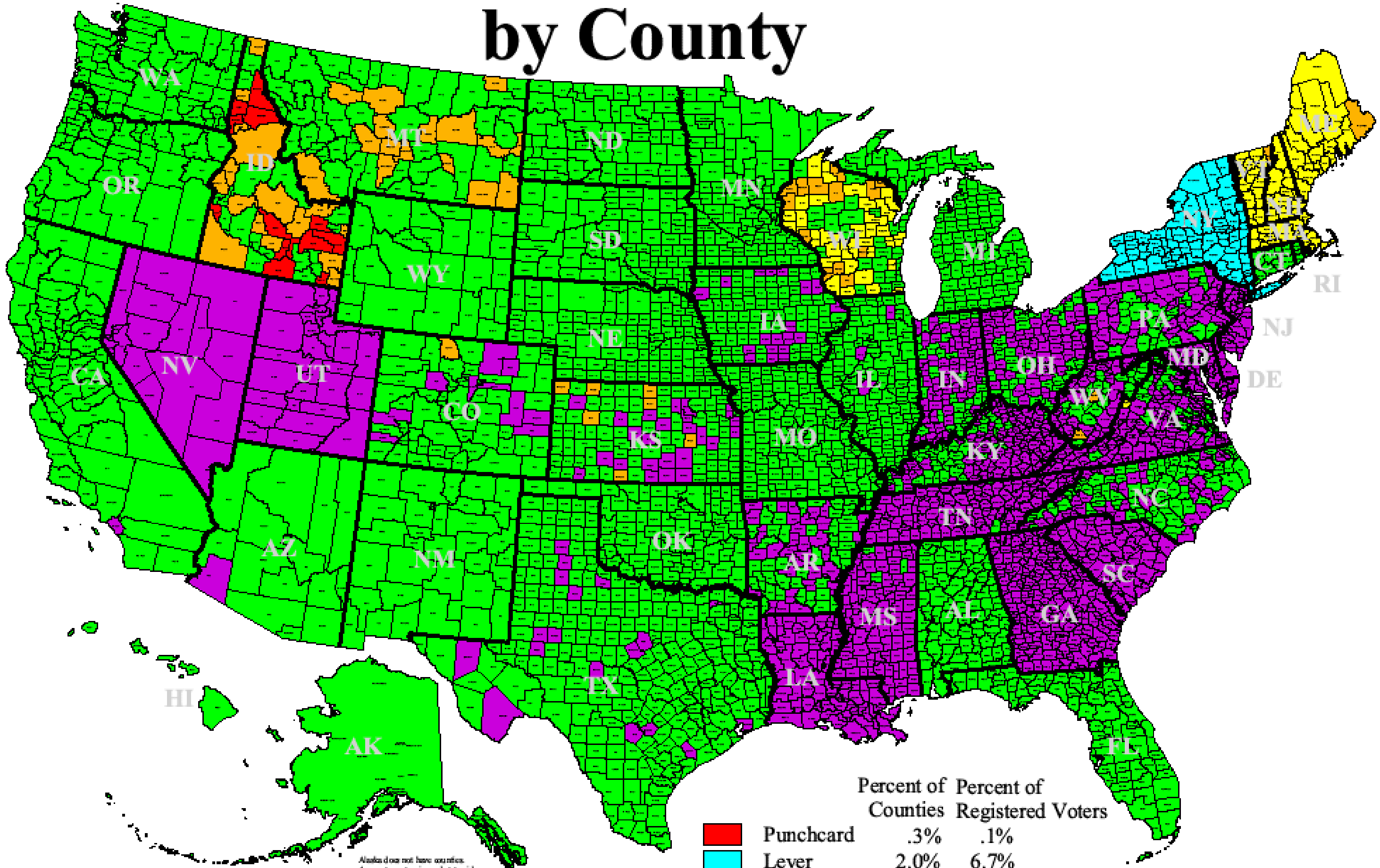

W B
W B


CLERK OF THE CIRCUIT COURT
Vote for One
Terry Bork
Republican
Write in
SHERIFF
Vote for One
BOARD OF EDUCATION

# Florida 2000:
# Bush v. Gore

# “Flawless”

# November 2008 Voting Equipment Usage by County




| | Percent of Counties | Percent of Registered Voters |
|---|---|---|
| Punchcard | .3% | .1% |
| Lever | 2.0% | 6.7% |
| Paper | 1.8% | .2% |
| Optical | 58.9% | 56.2% |
| Electronic | 34.3% | 32.6% |
| Mixed Systems | 2.7% | 4.2% |

Equipment expected to be used in the November 2008 election as reported by state election officials and news media. The map shows equipment used at polling places, not necessarily absentee or disabled balloting.

Election Data Services
(202) 789-2004
6171 Emerywood Court
Manassas, VA 20112-3078
www.ElectionDataServices.com


Printed with information available as of 10/14/2008

# Security FAIL

# Analysis of an electronic voting system [Kohno et al. 2003, 2004]

- DRE trusts smartcards
- Hardcoded keys and initialization vectors
- Weak message integrity
- Cryptographically insecure random number generator
- …

# California top-to-bottom reviews [Bishop, Wagner, et al. 2007]

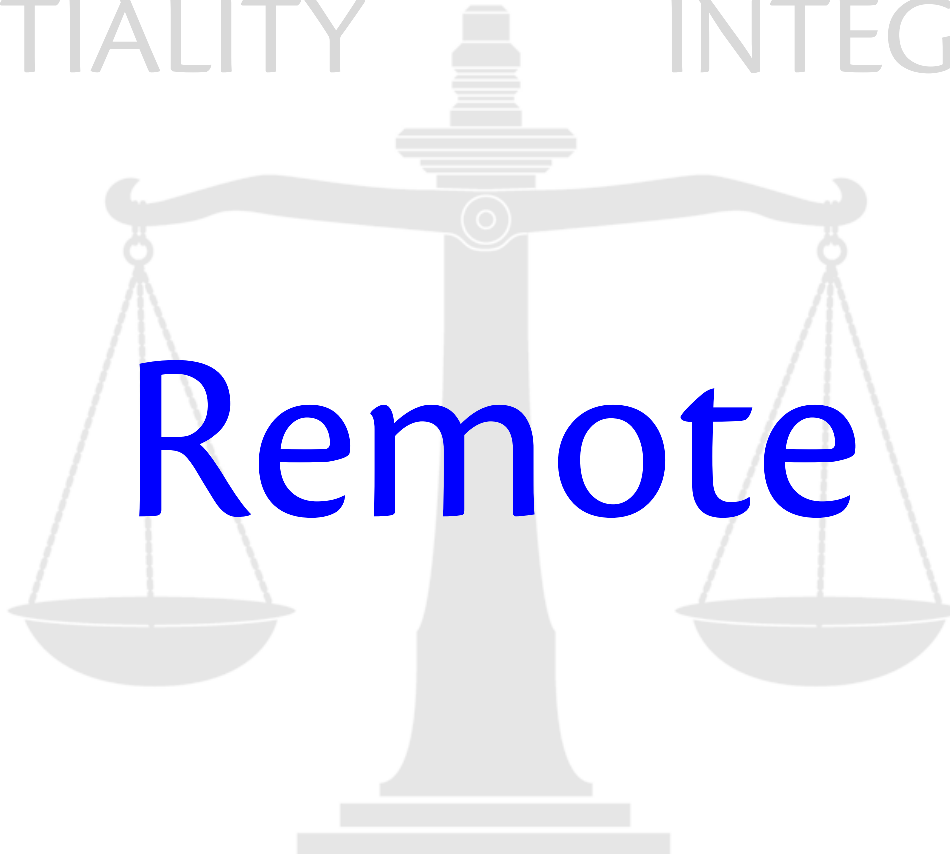
- *"Virtually every important software security mechanism is vulnerable to circumvention."*
- *"An attacker could subvert a single polling place device…then reprogram every polling place device in the county."*
- *"We could not find a single instance of correctly used cryptography that successfully accomplished the security purposes for which it was apparently intended."*

# Why is this so hard?

CONFIDENTIALITY

INTEGRITY

CONFIDENTIALITY INTEGRITY



# Remote

(including Internet)

# Why not Paper?

- What paper does:
  - Convince voter that her vote was captured correctly
- What paper does next:
  - Gets dropped in a ballot box
  - Immediately becomes insecure
    - Chain-of-custody, stuffing, loss, recount attacks…
    - Hacking paper elections has a long and (in)glorious tradition [*Steal this Vote*, Andrew Gumbel, 2005]
    - 20% of paper trails are missing or illegible [Michael Shamos, 2008]
- What paper doesn't:
  - Guarantee that a vote will be counted
  - Guarantee that a vote will be counted correctly

KEY PRINCIPLE:

# Mutual Distrust

# INTEGRITY

**Universal verifiability**

**Voter verifiability**

**Eligibility verifiability**

UV: [Sako and Killian 1994, 1995]

EV & VV: [Kremer, Ryan & Smyth 2010]

New definitions: [Smyth, Frink, Clarkson, work-in-progress]

# Why Verifiability?

- People:
  - Corrupted programmers
  - Hackers (individuals, …, nation-states)
- Software:
  - Buggy code
  - Malware
- Trustworthiness: *fair elections are a basis of representative democracy*

# CONFIDENTIALITY

**Coercion resistance**

better than **receipt freeness**
or simple **anonymity**

RF: [Benaloh 1994]
CR: [Juels, Catalano & Jakobsson 2005]

# Why Coercion Resistance?

- Protect election from *improper influence*
- Protect people from fear of reprisal
- Realize ideals of voting booth, remotely
- Trustworthiness: *fair elections are a basis of representative democracy*

# AVAILABILITY

**Tally availability**

# Recap

- History of voting technology
- Integrity: individual, universal, eligibility verifiability
- Confidentiality: coercion resistance, receipt freeness, anonymity
- Availability: tally avail.

# Security Properties

Original Civitas system:

- Universal verifiability
- Eligibility verifiability
- Coercion resistance

Follow-up projects:

- Voter verifiability
- Tally availability

…under various assumptions

# Adversary

Always:

- May perform any polynomial time computation
- May corrupt all but one of each type of *election authority*
  - ➔ Distributed trust

Almost always:

- May control network (Dolev-Yao)
- May coerce voters, demanding secrets or behavior, remotely or physically

# JCJ Voting Scheme

[Juels, Catalano & Jakobsson 2005]

Proved universal verifiability
and coercion resistance

Civitas extends JCJ

# Terminology

- *Voting system*: (software) implementation
- *Voting scheme*: cryptographic construction
- *Voting method*: algorithm for choosing between candidates

# <Voting Schemes>

*Classification based on cryptographic technique used to achieve confidentiality.*

# Tallying with Cryptography

- Blind signatures
- Mix networks
- Homomorphic encryption

# When is Vote Anonymized?

Before submission

After submission

# Blind Signatures

[Chaum 1983]

# Blind Signature Voting Protocols

Chaum 1983, Fujioka et al. 1992, Sako 1994, Okamoto 1996, 1997, Cranor & Cytron 1997, Herschberg 1997, DuRette 1999, Ohkubo et al. 1999, Joaquim et al. 2003, Lebre et al. 2004, Shubina & Smith 2004, ...

Fallen out of favor?

# When is Vote Anonymized?

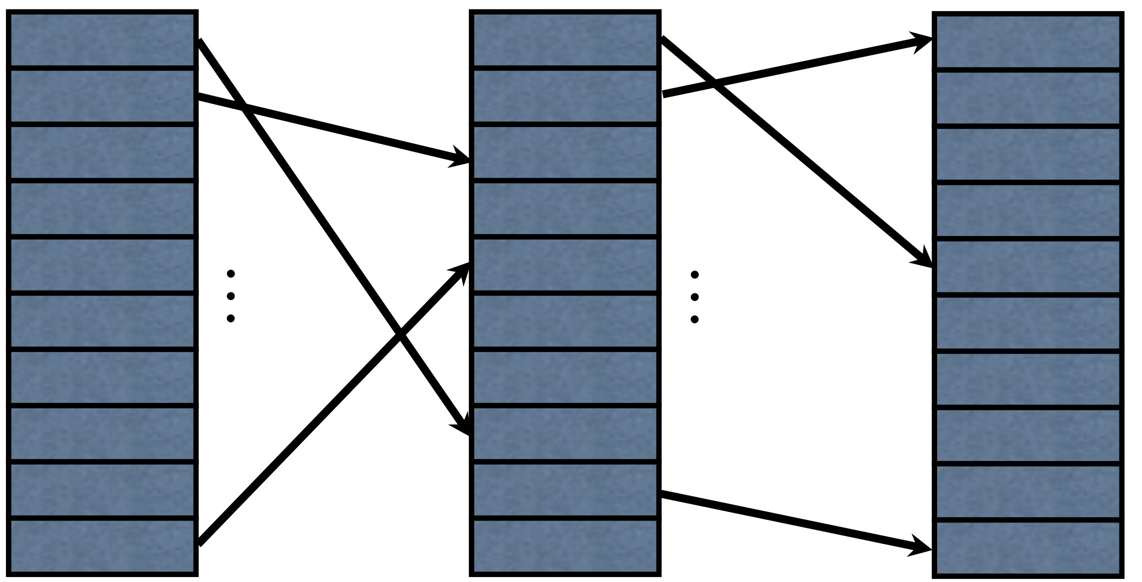
Before submission

After submission

Before tallying

# Mix Networks

[Chaum 1981]



# Simple Mix Network Election Protocol

1. $V \longrightarrow BB$: sign(enc(vote); $k_V$)
2. Talliers: check signatures
3. Mixers: remove signatures, mix votes
4. Talliers: decrypt votes, tally

# Verifiable Mix Networks

- Zero-knowledge proofs
  Park et al. 1993, Sako and Killian 1995, Neff 2001, Furukawa and Sako 2001, Groth 2003, Wikström 2005, Adida and Wikström 2007, ...
- Randomized partial checking
  Jakobsson et al. 2002, Khazaei and Wikström 2012

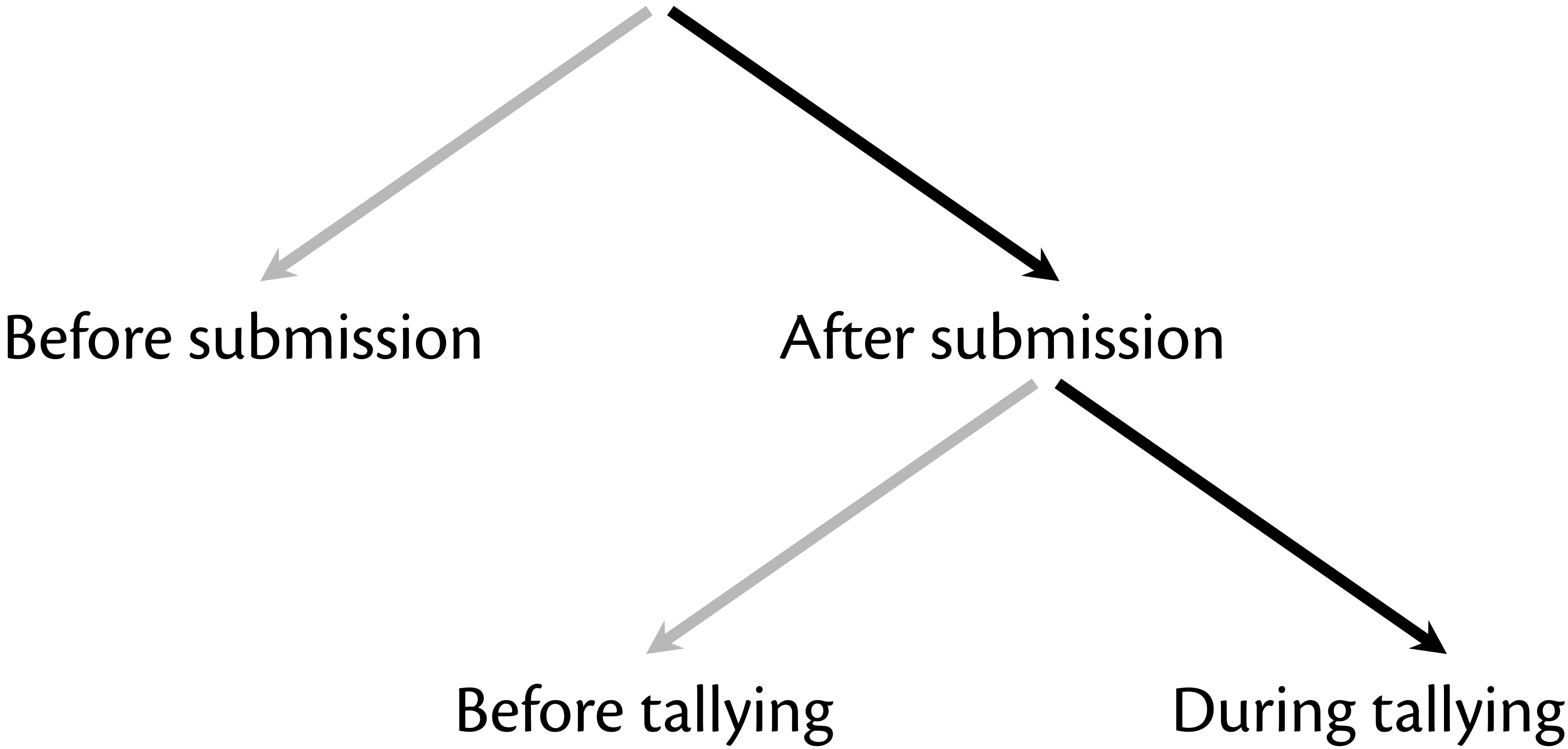
# Mix Network Election Protocols

Papers: Chaum 1981, Furukawa & Sako 1991, Park et al. 1993, Sako & Killian 1995, Ogata et al. 1997, Jakobsson 1998, Abe 1999, Neff 2001, Golle 2002, Jakobsson et al. 2002, Lee et al. 2003, Aditya et al. 2004, Juels et al. 2005, Chaum et al. 2005, Benaloh 2006, Popoveniuc & Hosp 2006, Ryan & Schneider 2006, Chaum et al. 2008, ...

Systems: Civitas (Clarkson et al.), Scantegrity II (Chaum et al.), VoteHere (Neff), Pret à Voter (Ryan et al.), Helios 1.0 (Adida)

Efficient schemes that prevent voter coercion?

# When is Vote Anonymized?



# Homomorphic Encryption

$$
\begin{tikzcd}
G \times G \arrow[r, "{(f,f)}"] \arrow[d, "\circ_G"'] & H \times H \arrow[d, "\circ_H"] \\
G \arrow[r, "f"'] & H
\end{tikzcd}
$$

[Rivest, Adleman, Dertouzos 1978]

$$\mathrm{enc}(v) \times \mathrm{enc}(v') = \mathrm{enc}(v+v')$$

# Simple Homomorphic Encryption Election Protocol

1. $V \rightarrow BB$: sign(enc(vote); $k_V$)
2. Talliers:
   1. check signatures
   2. compute $T = \prod_i \text{enc}(\text{vote}_i)$, which is $\text{enc}(\sum_i \text{vote}_i)$
   3. compute dec(T)

# Homomorphic Encryption Election Protocols

Papers: Cohen (Benaloh) & Fisher 1985, Cohen (Benaloh) & Yung 1986, Benaloh 1987, Benaloh & Tuinstra 1994, Sako & Killian 1994, Cramer et al. 1996, Cramer et al. 1997, Hirt & Sako 2000, Baudron et al. 2001, Kiayias 2006, Sandler 2007, Adida 2008, ...

Systems: Helios 2.0

Efficient schemes that prevent voter coercion?

# Is Cryptography Acceptable?

"The public won't trust cryptography."

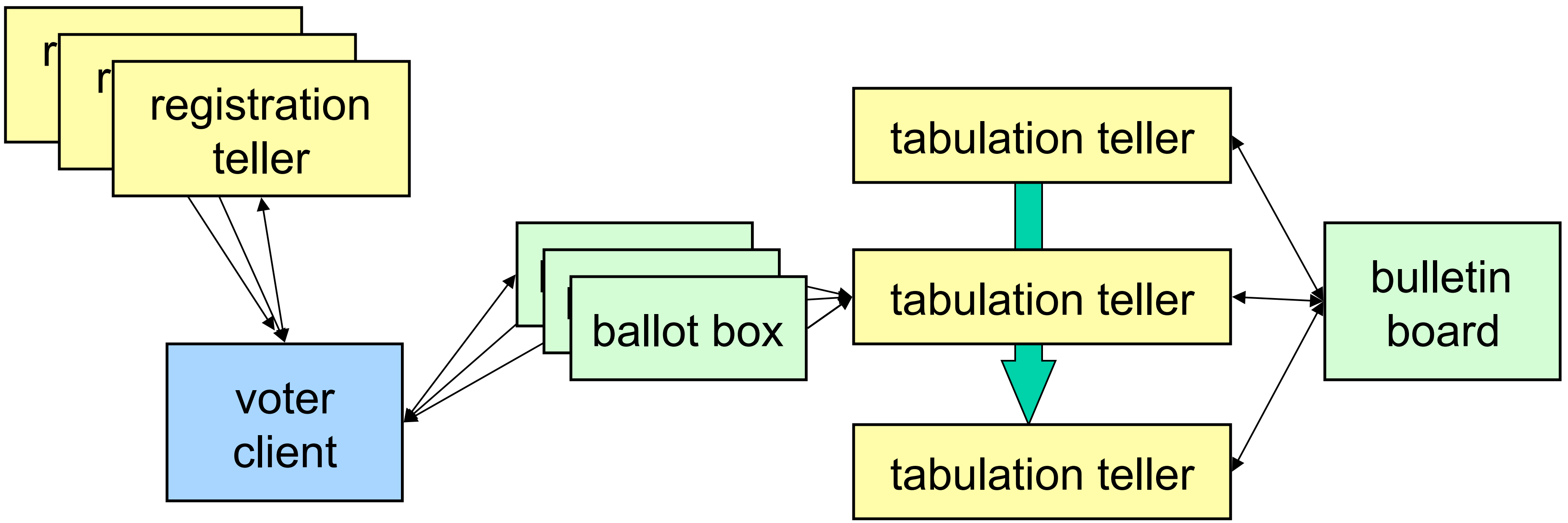
- It already does…
- Because experts already do

"I don't trust cryptography."

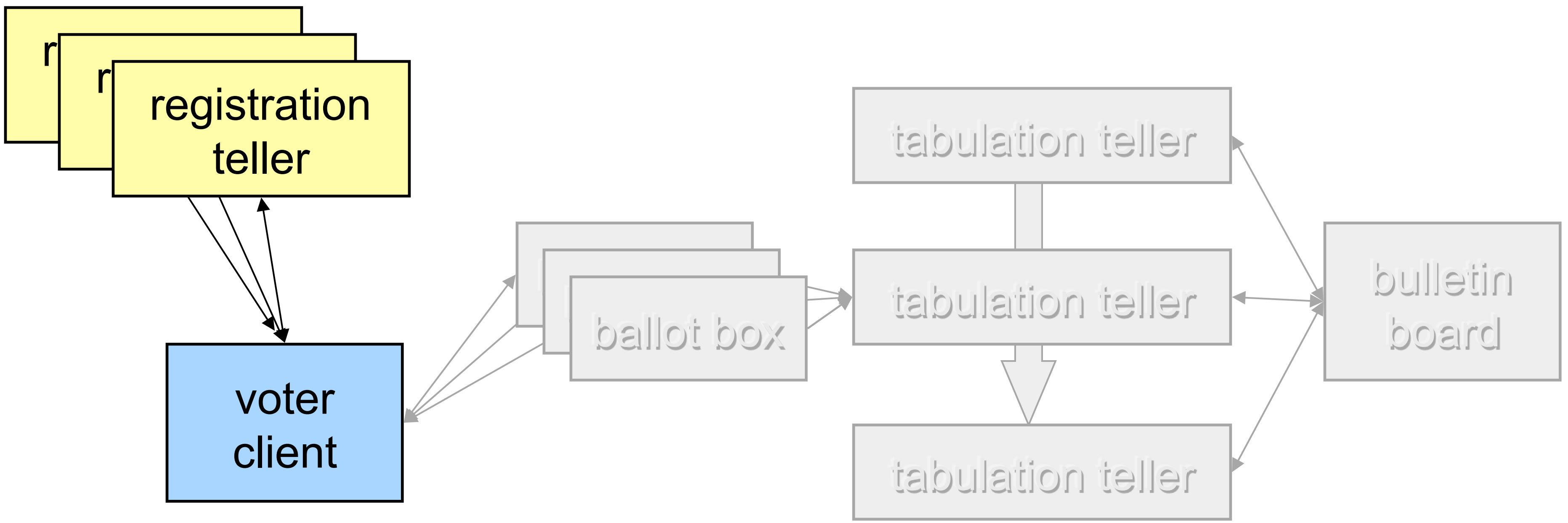
- You don't trust the proofs, or
- You reject the hardness assumptions

# </Voting Schemes>

# Civitas Architecture



# Registration

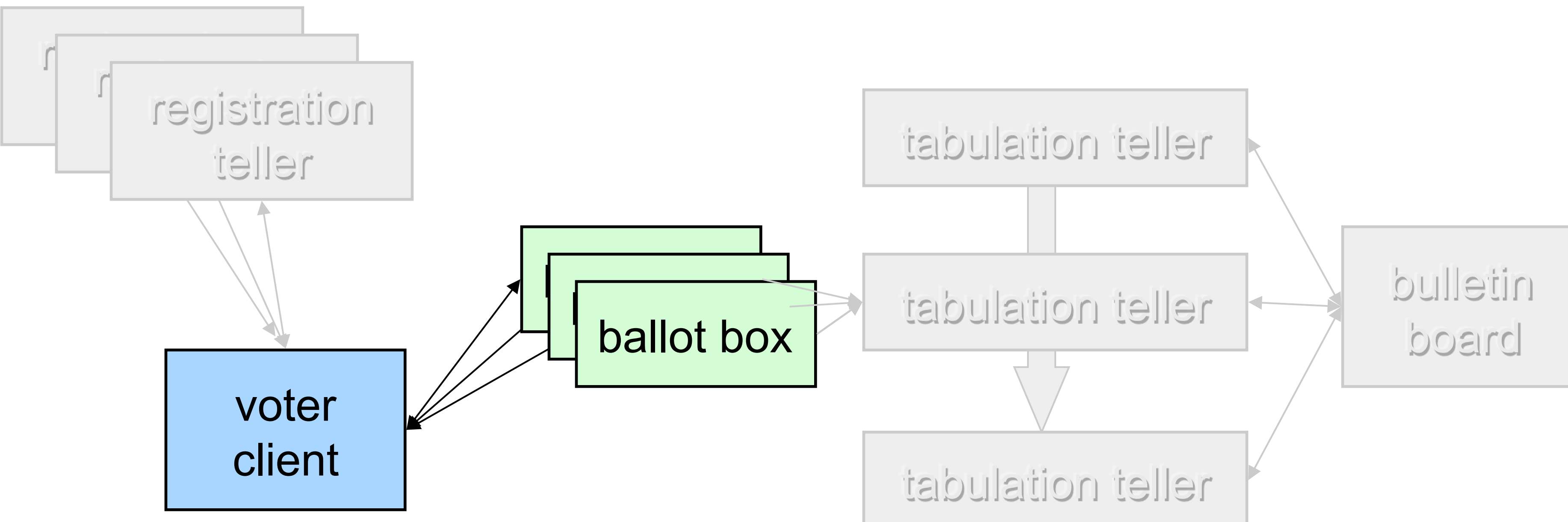



Voter retrieves *credential share* from each registration teller; combines to form *credential*

# Credentials

- Verifiable
- Unsalable
- Unforgeable
- Anonymous

# Voting



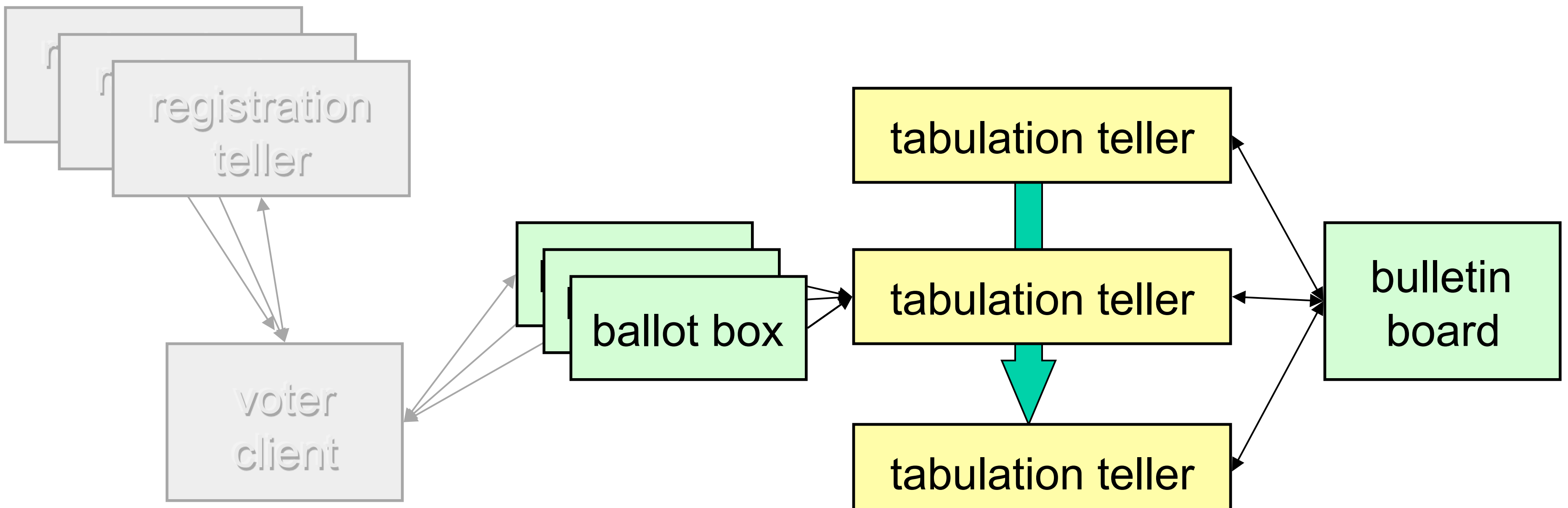

Voter submits copy of encrypted *choice* and credential to each ballot box

# Resisting Coercion:
# Fake Credentials

# Resisting Coercion

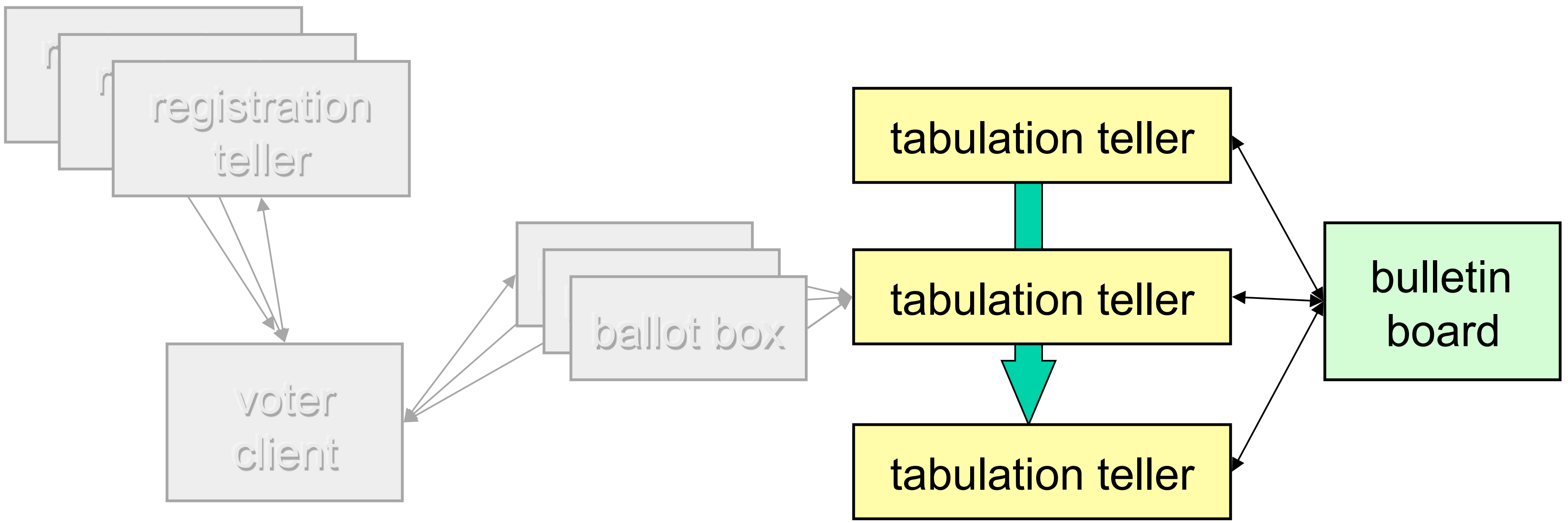
| **If the coercer demands that the voter…** | **Then the voter…** |
|---|---|
| Submits a particular vote | Does so with a fake credential. |
| Sells or surrenders a credential | Supplies a fake credential. |
| Abstains | Supplies a fake credential to the adversary and votes with a real one. |

# Tabulation



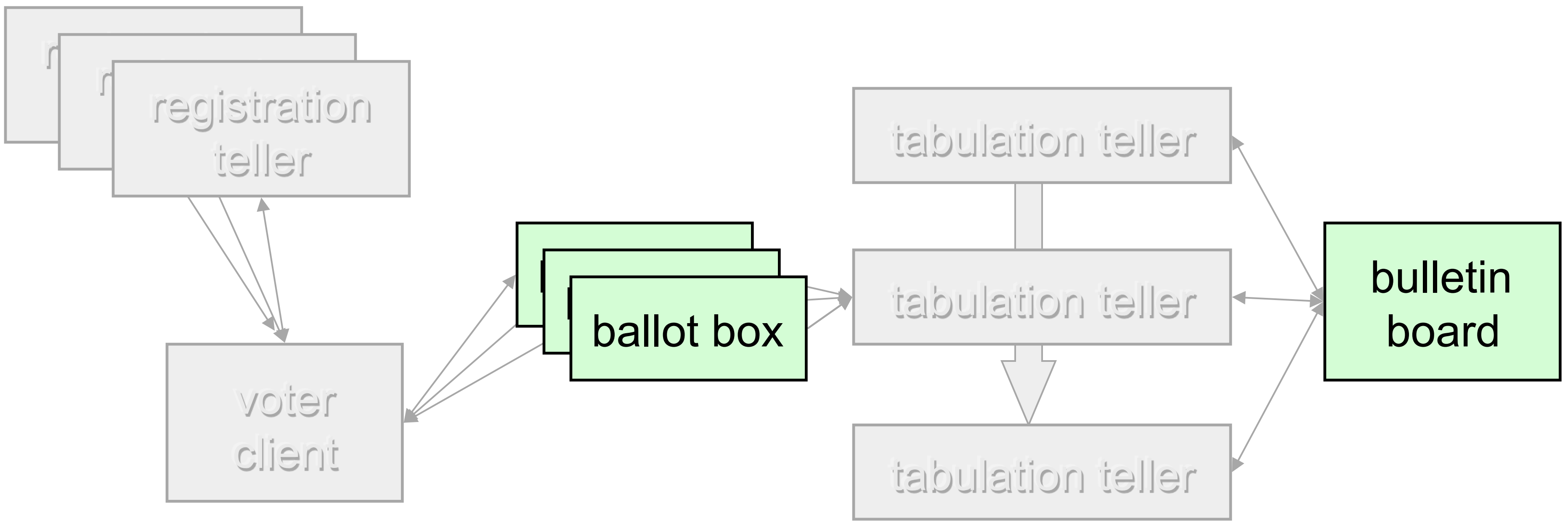

Tellers retrieve votes from ballot boxes

# Tabulation



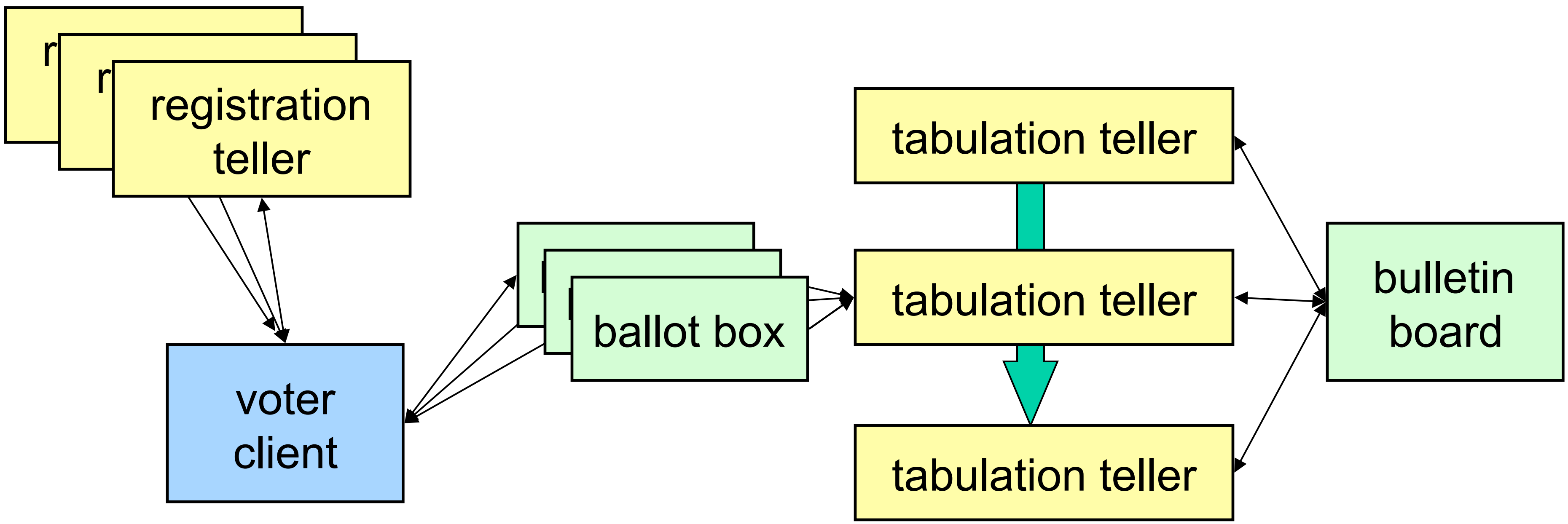

Tabulation tellers anonymize votes;
eliminate unauthorized (and fake) credentials;
decrypt remaining choices.

# Auditing




**Anyone** can verify proofs that tabulation is correct

# Civitas Architecture




**Universal verifiability:**
Tellers post proofs during tabulation

**Coercion resistance:**
Voters can undetectably fake credentials

SECURITY PROOFS

# Protocols

- El Gamal; distributed [Brandt]; non-malleable [Schnorr and Jakobsson]
- Proof of knowledge of discrete log [Schnorr]
- Proof of equality of discrete logarithms [Chaum & Pederson]
- Authentication and key establishment [Needham-Schroeder-Lowe]
- Designated-verifier reencryption proof [Hirt & Sako]
- 1-out-of-L reencryption proof [Hirt & Sako]
- Signature of knowledge of discrete logarithms [Camenisch & Stadler]
- Reencryption mix network with randomized partial checking [Jakobsson, Juels & Rivest]
- Plaintext equivalence test [Jakobsson & Juels]

**Implementation: 21k LoC**

# Cryptographic Techniques

- Zero-knowledge (ZK) proofs
  - Vote proofs, tabulation proofs
- Plaintext equivalence test
  - Elimination of duplicate and unauthorized credentials
- Mix network (already discussed)
  - Anonymization

# Plaintext Equivalence Test

- Special kind of ZK proof
- Tabulation tellers prove (as a group) that Dec(c) = Dec(c' ) without anyone, including the tellers, learning what Dec(c) or Dec(c' ) actually are

# Recap

- Voting schemes: blind signatures, mixnets, homomorphic encryption
- Civitas/JCJ architecture: credentials, PETs

# Trust Assumptions

# Trust Assumptions

1. "Cryptography works."
2. The adversary cannot masquerade as a voter during registration.
3. Voters trust their voting client.
4. At least one of each type of authority is honest.
5. The channels from the voter to the ballot boxes are anonymous.
6. Each voter has an untappable channel to a trusted registration teller.

# Trust Assumptions

Universal verifiability
Coercion resistance

1. "Cryptography works."

---

2. The adversary cannot masquerade as a vote

Coercion resistance

3. Voters trust their voting client.

4. At least one of each type of authority is honest.

5. The channels from the voter to the ballot boxes are anonymous.

6. Each voter has an untappable channel to a trusted registration teller.

# Trust Assumptions

1. “Cryptography works.”

2. The adversary cannot masquerade as a voter during registration.

3. Voters trust their voting client.

4. At least one of each type of authority is honest.

5. The channels from the voter to the ballot boxes are anonymous.

6. Each voter has an untappable channel to a trusted registration teller.

# Trust Assumptions

1. “Cryptography works.”
2. The adversary cannot masquerade as a voter during registration.
3. Voters trust their voting client.
4. At least one of each type of authority is honest.
5. The channels from the voter to the ballot boxes are anonymous.
6. Each voter has an untappable channel to a trusted registration teller.

# Registration

In person.

In advance.

**Con:** System not fully remote

**Pro:** Credential can be used in many elections

# Trust Assumptions

1. "Cryptography works."
2. The adversary cannot masquerade as a voter during registration.
3. Voters trust their voting client.
4. At least one of each type of authority is honest.
5. The channels from the voter to the ballot boxes are anonymous.
6. Each voter has an untappable channel to a trusted registration teller.

# Eliminating Trust

## in Voter Client

**VV:** Use *challenges* (like Helios, VoteBox)

**CR:** Open problem

# Trust Assumptions

1. “Cryptography works.”
2. The adversary cannot masquerade as a voter during registration.
3. Voters trust their voting client.
4. At least one of each type of authority is honest.
5. The channels from the voter to the ballot boxes are anonymous.
6. Each voter has an untappable channel to a trusted registration teller.

# Trust Assumptions

1. “Cryptography works.”
2. The adversary cannot masquerade as a voter during registration.
3. Voters trust their voting client.
4. At least one of each type of authority is honest.
5. The channels from the voter to the ballot boxes are anonymous.
6. Each voter has an untappable channel to a trusted registration teller.

# Trust Assumptions

1. “Cryptography works.”
2. The adversary cannot masquerade as a voter during registration.
3. Voters trust their voting client.
4. At least one of each type of authority is honest.
5. The channels from the voter to the ballot boxes are anonymous.
6. Each voter has an untappable channel to a trusted registration teller.

# Untappable Channel

Minimal known assumption
for receipt freeness and coercion resistance

Eliminate? Open problem.

(Eliminate trusted registration teller? Also open.)

# Trust Assumptions

1. "Cryptography works."
2. The adversary cannot masquerade as a voter during registration.
3. Voters trust their voting client.
4. At least one of each type of authority is honest.
5. The channels from the voter to the ballot boxes are anonymous.
6. Each voter has an untappable channel to a trusted registration teller.

# Trusted procedures?

# Time to Tally

# Blocks

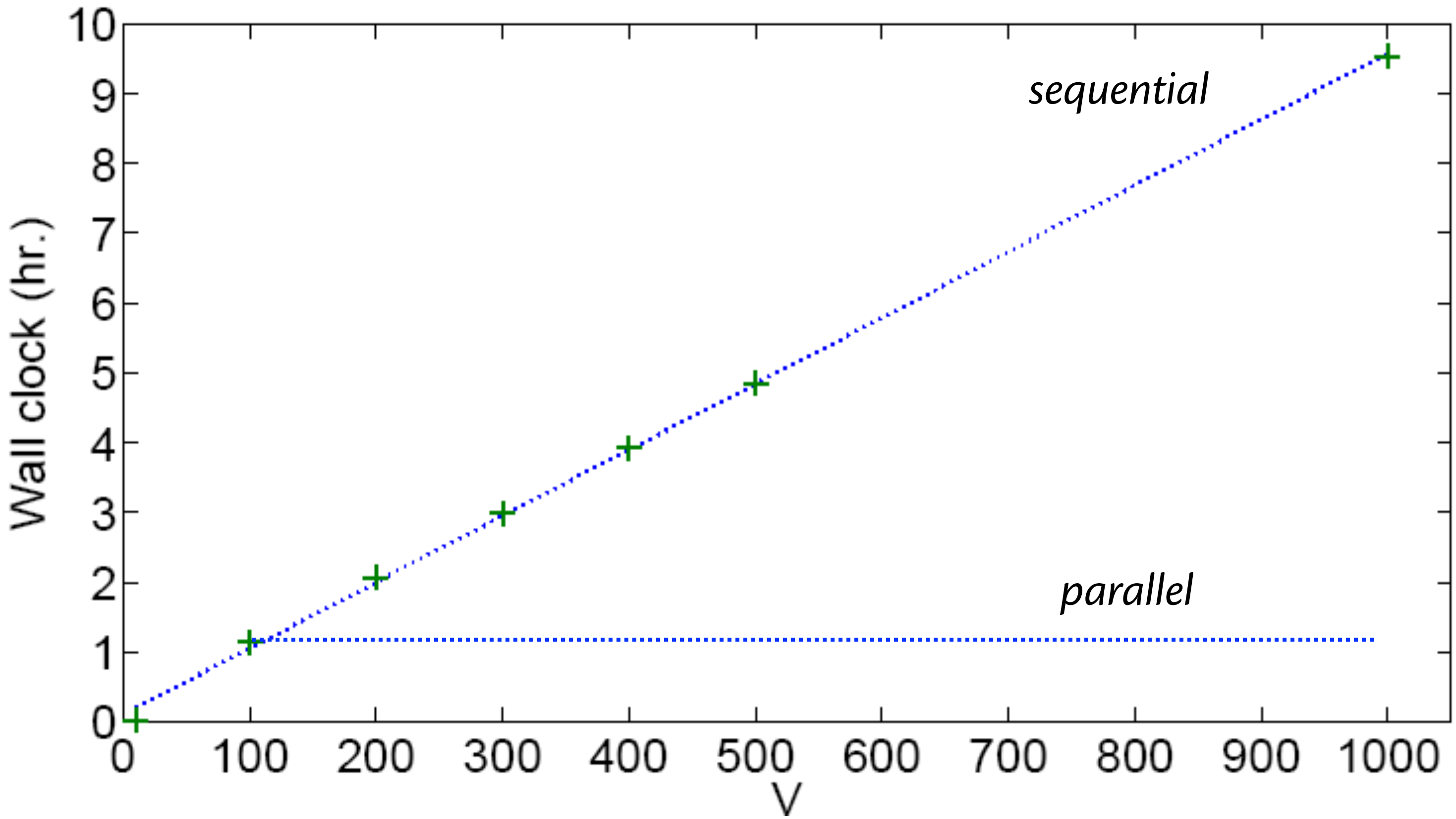
Block is a “virtual precinct”

- Each voter assigned to one block
- Each block tallied independently of other blocks, even in parallel

Tabulation time is:

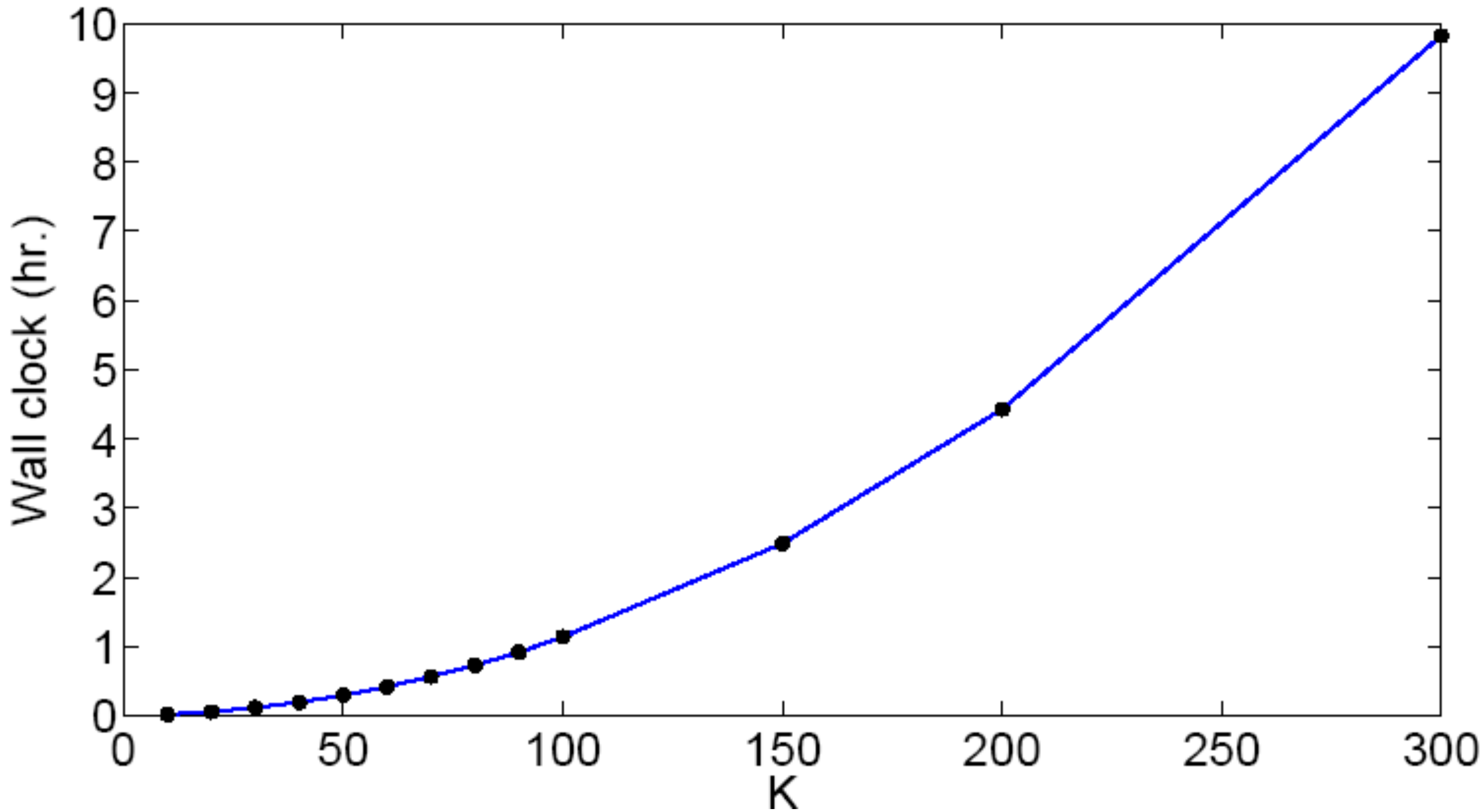
- Quadratic in block size
- Linear in number of voters
  - If using one set of machines for many blocks
- Or, constant in number of voters
  - If using one set of machines per block

# Tabulation Time




K = 100

# Tabulation Time




# voters in precinct = K, # tab. tellers = 4,
security strength ≥ 112 bits [NIST 2011–2030]

# CPU Cost

For 112-bit security level,

CPU time is 39 sec / voter / authority.

If CPUs are bought, used (for 5 hours), then thrown away:

$1500 / machine = $12 / voter

If CPUs are rented:

$1 / CPU / hr = 4¢ / voter

Increased cost…Increased security

# Summary

Can achieve strong security and transparency:

- Remote voting
- Universal (voter, eligibility) verifiability
- Coercion resistance

Security is not free:

- Stronger registration (untappable channel)
- Cryptography (computationally expensive)

# Assurance

## Security proofs (JCJ, us)

**Lemma 2.** $\{RegExp(n, 0)\}_{n \in \mathbb{N}} \approx \{RegExp(n, 1)\}_{n \in \mathbb{N}}$

*Proof.* Define three hybrids:

$$
\begin{array}{lclcl}
H_0 & = & \{K_{\mathrm{TT}}, K_V, S, s, r', P\} & = & RegExp( \\
H_1 & = & \{K_{\mathrm{TT}}, K_V, S, s, r', \tilde{P}'\} & & \\
H_2 & = & \{K_{\mathrm{TT}}, K_V, S, \tilde{s}, \tilde{r}, \tilde{P}\} & = & RegExp(
\end{array}
$$

where $\tilde{P}' = \widetilde{\mathsf{DVRP}}(K_V, S, S'; k_V)$. By the definition of a designated-v

To show that $H_1 \approx H_2$, assume for contradiction that there exist some non-negligible advantage in distinguishing $H_1$ and $H_2$. Using $D$, $A$ that breaks the indistinguishability of the encryption scheme, as fc $K_{\mathrm{TT}}$, challenges $m_0$ and $m_1$, and a ciphertext $c$ that encrypts one of the c

## Secure implementation (Jif)

```
/**
 * Retrieve from the bulletin board the final array of votes f
 * i.e., the votes that contain capabilities that match one in
 */
private Vote{TT<-SUP;TT<-TELLS}[]{TT<-SUP;TT<-TELLS} retrieveF
        PETCache{TT<-SUP;TT<-TELLS} votesToRollCache,
        ElectionCache{TT<-SUP;TT<-TELLS} electionCache,
        ElGamalPublicKey{TT<-SUP;TT<-TELLS} tabTellerSharedKey
        TellerDetails{TT<-SUP;TT<-TELLS} tellerDetails,
        int{TT<-SUP;TT<-TELLS} tellerIndex, int{TT<-SUP;TT<-TE
throws IOException, CryptoException
where caller(TT) {
    if (electionDetails == null || tttUtil == null || votesToF
    BallotDesign ballotDesign = electionDetails.ballotDesign;
    if (ballotDesign == null) return null;
    ElectionID electionID = electionDetails.electionID;
    if (electionID == null) return null;
```

# Secure Implementation

In Jif [Myers 1999, Chong and Myers 2005, 2008]

- Security-typed language
- Types contain information-flow policies
  - Confidentiality, integrity, declassification, erasure

If policies in code express correct requirements…

- (And Jif compiler is correct…)
- Then code is secure w.r.t. requirements

# Civitas Policy Examples

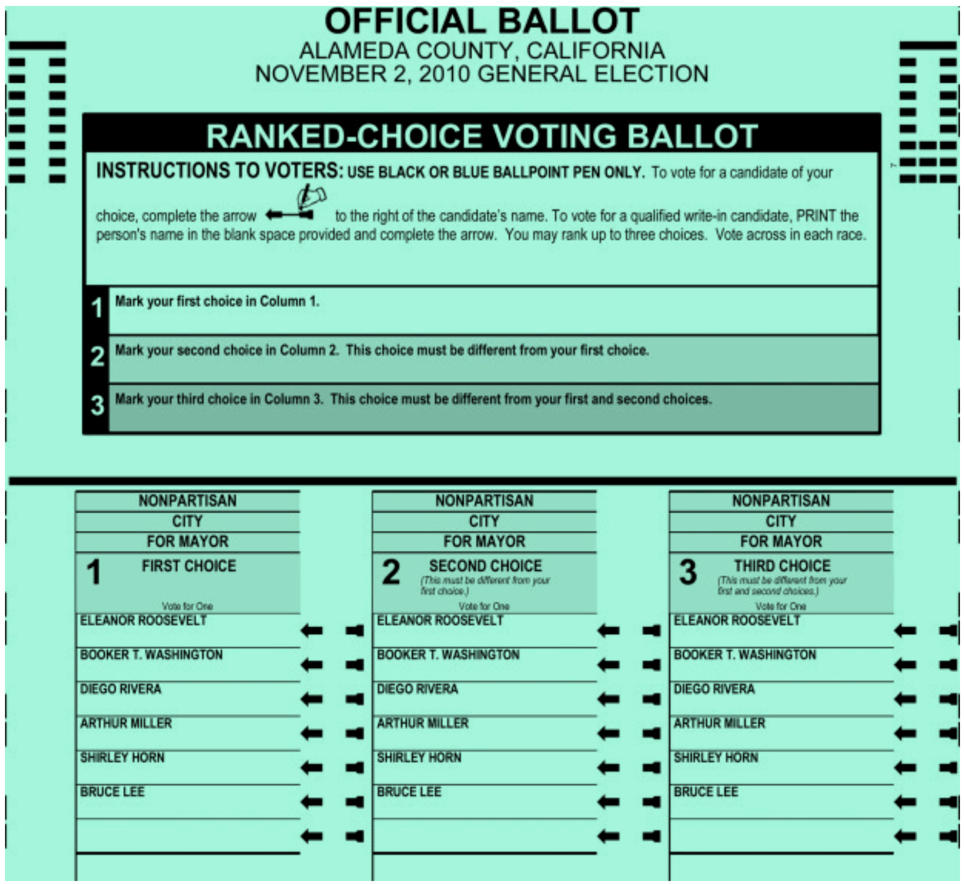
- Confidentiality:
  - Information: Voter's credential share
  - Policy: "RT permits only this voter to learn this information"
  - Jif syntax: RT → Voter
- Confidentiality:
  - Information: Teller's private key
  - Policy: "TT permits no one else to learn this information"
  - Jif syntax: TT → TT
- Integrity:
  - Information: Random nonces used by tellers
  - Policy: "TT permits only itself to influence this information"
  - Jif syntax: TT ← TT

# Civitas Policy Examples

- Declassification:
  - Information: Bits that are committed to then revealed
  - Policy: “TT permits no one to read this information until all commitments become available, then TT declassifies it to allow everyone to read.”
  - Jif syntax: $TT \rightarrow [TT \searrow^{commAvail} \bot]$

- Erasure:
  - Information: Voter’s credential shares
  - Policy: “Voter requires, after all shares are received and full credential is constructed, that shares must be erased.”
  - Jif syntax: $Voter \rightarrow [Voter \,{}^{credConst}\!\nearrow \top]$

# Ranked Voting



# Ranked Voting

Voters submit ranking of candidates

- e.g., Condorcet, Borda, STV
- Help avoid spoiler effects
- Defend against strategic voting

Civitas implements coercion-resistant Condorcet, *approval* and *plurality* voting methods

# Open Problems

- Coercion-resistant voter client?
- Voter-verifiable voter client?
- Eliminate untappable channel in registration?
- Credential management?
- Usability?
- Application-level denial of service? (Efficient coercion-resistant tallying?)
- Scalable secure bulletin board?

http://www.cs.cornell.edu/projects/civitas

(google "civitas voting")

# Civitas

## Verifiability and Coercion Resistance for Remote Voting

Michael Clarkson
Cornell University

15th International School on Foundations of Security Analysis and Design
University Residential Center of Bertinoro, Italy
September 4, 2015