CIVITAS
Toward a Secure Voting System

Michael Clarkson
Cornell University

University of Maryland
March 15, 2011
Secret Ballot
<table>
<thead>
<tr>
<th>Position</th>
<th>Candidates</th>
<th>Party</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clerk of the Circuit Court</td>
<td>Terry Bork</td>
<td>Republican</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Democratic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write-in</td>
</tr>
<tr>
<td>Board of Education District 1</td>
<td>Judy Docca</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Michael Bathez</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write-in</td>
</tr>
<tr>
<td>Sheriff</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Florida 2000: 
Bush v. Gore
“Flawless”
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?
Cryptography
Cryptography

Can cryptography be defended?
Low-tech crypto?
Simple Voting Protocol

1. $V \rightarrow BB$: sign($\text{enc}(\text{vote}); k_v$)

2. Talliers: check signatures

3. Talliers: decrypt votes, tally
Simple Voting Protocol

1. \(V \rightarrow BB: \text{sign}(\text{enc}(\text{vote}); k_V)\)

2. Talliers: check signatures

3. Talliers: decrypt votes, tally

How to build secure, scalable BB?
PRIVACY via cryptography

- Blind signatures
- Mix networks
- Homomorphic encryption
PRIVACY via cryptography

- Blind signatures
- Mix networks
- Homomorphic encryption

Why these three? What others?
When is Vote Anonymized?

Before submission

After submission
Blind Signatures

[Chaum 1983]
unblind( sign(blind(m); k) )
= 
sign(m; k)
\[ V \rightarrow BB: \ sign(\text{enc}(\text{vote}); k_v) \]
V → BB: sign(enc(vote); k_A)
Simple Blind Signature
Election Protocol

1. $V \rightarrow \text{Auth}: V, \text{sign}(\text{blind}(\text{enc}(\text{vote})); k_v)$
Simple Blind Signature Election Protocol

1. $V \rightarrow \text{Auth}: V, \text{sign}(\text{blind}(\text{enc}(\text{vote})); k_V)$

2. $\text{Auth} \rightarrow V: \text{sign}(\text{blind}(\text{enc}(\text{vote})); k_A)$
Simple Blind Signature Election Protocol

1. $V \rightarrow \text{Auth}: \ V, \ sign(\text{blind}(\text{enc}(\text{vote})); \ k_V)$

2. $\text{Auth} \rightarrow V: \ sign(\text{blind}(\text{enc}(\text{vote})); \ k_A)$

3. $V \rightarrow \text{BB [anon.]}: \ sign(\text{enc}(\text{vote}); \ k_A)$
Simple Blind Signature Election Protocol

1. V → Auth: V, sign(blind(enc(vote)); k_V)
2. Auth → V: sign(blind(enc(vote)); k_A)
3. V → BB [anon.]: sign(enc(vote); k_A)
4. Talliers: check signatures, decrypt votes, tally
Blind Signature
Voting Protocols


How to achieve high integrity?
When is Vote Anonymized?

- Before submission
- After submission
- Before tallying
Mix Networks

[Chaum 1981]
Decryption Mix

\[ \text{enc} \left( \text{enc} \left( \text{enc} \left( m ; K_3 \right) ; K_2 \right) ; K_1 \right) \]
Reencryption Mix

\[ \text{enc}(m; K) \quad \text{reenc}(m; K) \quad \text{reenc}(m; K) \]

[Park et al. 1994]
1. \( V \rightarrow BB: \text{sign}(\text{enc}(\text{vote}); k_V) \)
2. Talliers: check signatures
3. Mixers: remove signatures, mix votes
4. Talliers: decrypt votes, tally
Mix Network
Election Protocols

When is Vote Anonymized?

Before submission

After submission

Before tallying

During tallying
Homomorphic Encryption

\[(\mathbb{G} \times \mathbb{G}) \xrightarrow{(f,f)} (\mathbb{H} \times \mathbb{H})\]

\[\mathbb{G} \xrightarrow{\circ_{\mathbb{G}}} \mathbb{H} \xrightarrow{\circ_{\mathbb{H}}}
\]

\[\mathbb{G} \xrightarrow{f} \mathbb{H}\]

[Rivest, Adleman, Dertouzos 1978]

Fully homomorphic?
$\text{enc}(v) \times \text{enc}(v') = \text{enc}(v+v')$
Simple Homomorphic Encryption Election Protocol

1. \( V \rightarrow BB: \text{sign} (\text{enc}(\text{vote}); k_V) \)

2. Talliers:
   a. check signatures
   b. compute \( T = \prod_i \text{enc}(\text{vote}_i) \), which is \( \text{enc}(\sum_i \text{vote}_i) \)
   c. compute \( \text{dec}(T) \)
Homomorphic Encryption
Election Protocols

Definitions of PRIVACY Integrity?
Coercion resistance
Receipt freeness
Vote privacy

[Delaune, Kremer & Ryan 2006]
CIVITAS

[Clarkson, Chong & Myers 2008]
based on [Juels, Catalano & Jakobsson 2005]
1. $V \to BB$: $\text{sign(}\text{enc(vote)}; k_V\text{)}$

2. Talliers: check signatures

3. Mixers: remove signatures, mix votes

4. Talliers: decrypt votes, tally
Voter Credentials

- Registrar → V: cred
- Registrar → BB: enc(cred)  [electoral roll]
- V → BB: enc(cred), enc(vote)
Voter Credentials

- Registrar $\rightarrow$ V: $\text{cred}$
- Registrar $\rightarrow$ BB: $\text{enc(cred)}$ [electoral roll]
- V $\rightarrow$ BB: $\text{enc(cred)}$, $\text{enc(vote)}$
Voter Credentials

- Registrar → V [untap.]: cred, zkpf₁
- Registrar → BB: enc(cred) [electoral roll]
- V → BB [anon.]: enc(cred), enc(vote), zkpf₂
Tallying Protocol

Talliers:
Tallying Protocol

Talliers:

1. Retrieve votes from BB, check proofs
Tallying Protocol

Talliers:

1. Retrieve votes from BB, check proofs

2. Eliminate unauthorized credentials (requires mixes, zkpsfs)
Tallying Protocol

Talliers:

1. Retrieve votes from BB, check proofs
2. Eliminate unauthorized credentials (requires mixes, zkpfks)
3. Decrypt votes, tally
Removing Unauthorized Credentials

Electoral roll, mixed

Submitted votes, mixed

PETs

enc(cred)

enc(cred), enc(vote)
JCJ Credentials

- Verifiable
- Unsalable
- Anonymous
- Unforgeable
Coercion resistant: voters use fake (unauthorized) credentials to comply with coercer
Civitas Architecture

- voter
- client

- registration teller

- ballot box

- tabulation teller

- bulletin board
Civitas
Architecture

JCJ: single trusted registrar
Civitas: distributed trust
...improved privacy and verifiability
Civitas
Registration for Credentials

credential shares

Electoral roll
Civitas
Registration Security

Voter must be able to lie about a credential share

Requires *untappable channel*
Civitas Architecture

JCJ: single trusted registrar
Civitas: distributed trust
...improved privacy and verifiability
JCJ: single trusted registrar
Civitas: distributed trust
...improved privacy and verifiability

JCJ: no ballot boxes
Civitas: distributed storage
...improved availability
JCJ: single trusted registrar
Civitas: distributed trust
...improved privacy and verifiability

JCJ: no ballot boxes
Civitas: distributed storage
...improved availability

JCJ: \(O(V^2)\)
Civitas: \(O(B^2)\), \(B \ll V\)
...improved scalability
Civitas
Architecture

JCJ: single trusted registrar
Civitas: distributed trust
...improved privacy and verifiability

JCJ: no ballot boxes
Civitas: distributed storage
...improved availability

JCJ: $O(V^2)$
Civitas: $O(B^2)$, $B < V$
...improved scalability

JCJ: no ballot boxes
Civitas: distributed storage
...improved availability

Civitas: concrete implementation, 21K LoC
Security Assurance

Design:

- JCJ proof of coercion resistance and verifiability (we extend it)
- Backes et al. (CSF 2008) verification with ProVerif

Implementation:

- In security typed-language
Civitas
Trust Assumptions

1. DDH, RSA, random oracle model.

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.
Secure Implementation

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

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

Compiler verifies code obeys policies
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: \[TT \rightarrow [TT \downarrow^{\text{commAvail}} \bot] \]

**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: \[\text{Voter} \rightarrow [\text{Voter} \uparrow^{\text{credConst}} \uparrow T] \]
Tabulation Time vs. Anonymity

K = # voters in block, # tab. tellers = 4, security strength ≥ 112 bits [NIST 2011–2030], 3GHz Xeons
Tabulation Time vs. \#Voters

\[ K = 100 \]
Tabulation Time vs. #Voters

![Graph showing tabulation time vs. number of voters for sequential and parallel processing. The graph indicates a linear relationship with a slope for sequential processing and a flat line for parallel processing. The x-axis represents the number of voters (V), and the y-axis represents wall clock time (hr.). The constant K is set to 100.](image)
Summary

Security:

- Coercion resistance & universal verifiability
- Distributed trust

Assurance:

- Security proofs & security-typed implementation

Also: Ranked voting
Ongoing Work

- Verifiable voter client
- BFT bulletin board
- Threshold cryptography
Open Problems

- Eliminate untappable channel in registration?
- Credential management?
- Application-level DoS?
www.cs.cornell.edu/projects/civitas

or google “civitas voting”
CIVITAS
Toward a Secure Voting System

Michael Clarkson
Cornell University

University of Maryland
March 15, 2011