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
Florida 2000: Bush v. Gore
“Flawless”
November 2008 Voting Equipment Usage by County

Equipment expected to be used in the November 2008 election as reported by state election officials and news media.

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

Copyright © 2008 Election Data Services, Inc. This product is for informational purposes only and has not been verified by Election Administration. No guarantees of accuracy are made. For the latest information, please visit www.electiondataservices.com.
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?
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*
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


Fallen out of favor?
When is Vote Anonymized?

Before submission

After submission

Before tallying
Mix Networks

[Chaum 1981]
Simple Mix Network
Election Protocol

1. \( V \rightarrow BB: \ sign(\text{enc}(vote); k_v) \)
2. Talliers: check signatures
3. Mixers: remove signatures, mix votes
4. Talliers: decrypt votes, tally
Verifiable Mix Networks

- Zero-knowledge proofs

- Randomized partial checking
  Jakobsson et al. 2002, Khazaei and Wikström 2012
Mix Network
Election Protocols


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?

Before submission

Before tallying

During tallying

After submission
Homomorphic Encryption

$G \times G \xrightarrow{(f,f)} H \times H$

$G \xrightarrow{f} H$

$G \xleftarrow{\circ G} H \xleftarrow{\circ H}$

[Rivest, Adleman, Dertouzos 1978]
\( \text{enc}(v) \times \text{enc}(v') = \text{enc}(v+v') \)
Simple Homomorphic Encryption
Election Protocol

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

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


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 teller
- voter client
- ballot box
- tabulation teller
- bulletin board
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

<table>
<thead>
<tr>
<th>If the coercer demands that the voter...</th>
<th>Then the voter...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submits a particular vote</td>
<td>Does so with a <strong>fake credential</strong>.</td>
</tr>
<tr>
<td>Sells or surrenders a credential</td>
<td>Supplies a <strong>fake credential</strong>.</td>
</tr>
<tr>
<td>Abstains</td>
<td>Supplies a <strong>fake credential</strong> to the adversary and votes with a real one.</td>
</tr>
</tbody>
</table>
Tabulation

Tellors retrieve votes from ballot boxes
Tabulation tellers anonymize votes; eliminate unauthorized (and fake) credentials; decrypt remaining choices.
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 tells prove (as a group) that $\text{Dec}(c) = \text{Dec}(c')$ without anyone, including the tellers, learning what $\text{Dec}(c)$ or $\text{Dec}(c')$ 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

1. “Cryptography works.”

2. The adversary cannot masquerade as a voter.

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 (untappble 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:

\[
\begin{align*}
H_0 &= \{K_{TT}, K_V, S, s, r', P\} = RegExp(
H_1 &= \{K_{TT}, K_V, S, s, r', \tilde{P}'\}
H_2 &= \{K_{TT}, K_V, S, \tilde{s}, \tilde{r}, \tilde{P}\} = RegExp(
\end{align*}
\]

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

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 for \( K_{TT} \), challenges \( m_0 \) and \( m_1 \), and a ciphertext \( c \) that encrytcs one of the \( 2 \) messages, we can distinguish between \( H_1 \) and \( H_2 \).

```
1117 **/
1118 * Retrieve from the bulletin board the final array of votes f
1119 * i.e., the votes that contain capabilities that match one ir
1120 */
1121 private Vote{TT<-SUP;TT<-TELLS}[]{TT<-SUP;TT<-TELLS} retrievef
1122 PETCache{TT<-SUP;TT<-TELLS} votesToRollCache,
1123 ElectionCache{TT<-SUP;TT<-TELLS} electionCache,
1124 E1GamalPublicKey{TT<-SUP;TT<-TELLS} tabTellerSharedKey,
1125 TellerDetails{TT<-SUP;TT<-TELLS} tellerDetails,
1126 int{TT<-SUP;TT<-TELLS} tellerIndex, int{TT<-SUP;TT<-TELLS}
1127 throws IOException, CryptoException
1128 where caller(TT) {
1129   if (electionDetails == null || tttUtil == null || votesToF
1130     BallotDesign ballotDesign = electionDetails.ballotDesign;
1131     if (ballotDesign == null) return null;
1132     ElectionID electionID = electionDetails.electionID;
1133     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: \( TT \rightarrow [TT \downarrow^{\text{commAvail}} \bot] \)

• 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: \( \text{Voter} \rightarrow [\text{Voter}^{\text{credConst}} \Rightarrow T] \)
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