Caveat Coercitor

Mark Ryan and Peter Y. A. Ryan
University of Birmingham University of Luxembourg

TVS/SerTVS meeting
University of Birmingham
6th April 2011
1 Desired properties
2 Approaches
3 Caveat Coercitor
4 Conclusions
Desired properties

Verifiability
- Outcome of election is verifiable by voters and observers
- You don’t need to trust election software

Incoercibility
- Your vote is private
  - even if you try to cooperate with a coercer
  - even if the coercer is the election authorities

Usability
- Vote & go
- Verify any time
Examples

- Verifiable
- Usable
- Incoercible

- Raised hands
- Website voting
- Using Tor

Diagram:

- Verifiable
- Usable
- Incoercible

Overlap:

- Raised hands and using Tor
- Website voting

Question mark for uncertainty.
Approaches

The computer that you interact with encrypts your vote.

Examples:
- FOO and derivatives
- Helios 2.0
- JCJ/Civitas

**Problem:** you need to trust the computer
- to do it correctly
- to keep it secret

The computer that you interact with, if any, does not see your vote.

Examples:
- PaV
- Scantegrity
- code voting

**Problem:** you need to trust the back-end
- to do it correctly
- to keep it secret
Caveat Coercitor

- intended for Internet voting
- intended to balance security & usability
- intended to be deployable
- borrows ideas liberally, but especially from [JCJ/Civitas]
* Make user responsible for privacy
* Give up incoercibility . . .
  but make coercion evident
  ("caveat coercitor")
CC: What the voter does

1. Voter obtains her credentials.
2. Voter chooses platform on which to construct her ballot.
   - smartphone applet
   - standalone bootable program (memtest86-like)
   - app for favourite OS, downloaded from source of choice
   - browser applet from source of choice
   - HTTPS connection to server of choice
3. Voter submits her ballot to the collector.
4. Voter repeats 1-3 as often as she likes. (At most one of them will be counted.)

Observation:
- Voter is required to make a personal judgment about the trustworthiness of the ballot-forming tool she chooses.
  - Something has to be trusted . . .
  - So we give the voter the freedom / responsibility to choose what.
Ballot formation applet for novices

UK Parliamentary Election 2014

Birmingham, Selly Oak constituency

☐ Stephen McCabe (Labour)
☐ Nigel Dawkins (Conservative)
☐ Dave Radcliffe (Liberal Democrat)
☐ Lynette Orton (BNP)
☐ Jeffrey Burgess (UKIP)
☐ James Burn (Green)
☐ Samuel Leeds (Christian)

Voter’s credential:

Calculate Ballot

Your encrypted ballot:

Qa3+MXgqTE2FkHWK14n5QFGbjucvTeeF1NapnbGdGnNqsfVAvgi/Etu+B78hCuB
94MAVQRi+LDo5ckcAUX2pMDCAJJ/k0vPeBNaDTdmtFPjFoXwq5n2U7JCdCqS/1s
q1IRFxsu3SwB+IRuejSyALEqt1nIxzCxqtXEvqX0s6zt8sez1/uApn/eFEG9/8
GgkFwe7X01WKYxTwdMa5HMTS41L0Jq1mzuA77DRIa4FpBsU+Eh06npYqcKvtbv
5uaIY+2foPPkq7Flk3iE2CtNhPJ6QI61Ku2KjSJ6mnyhTbyEB70jp0acSEfzGLV
0H9StCN20nsHAC0uCd/OyDrNHuA==

You should paste this value to the website at election2014.gov.uk.
# Ballot formation applet for experts

## Caveat Coercitor ballot-forming applet

<table>
<thead>
<tr>
<th>pk_R</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>pk_T</td>
<td></td>
</tr>
</tbody>
</table>

**Voter’s credential**

<table>
<thead>
<tr>
<th>rand_R</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>rand_T</td>
<td></td>
</tr>
</tbody>
</table>

**Calculate Ballot**

**Your encrypted ballot:**

```
Qa3+MXgqTE2FkHWK14n5QFGbjucvTeeF1NApnbGdGnNqsfVAvgi/Etu+B78hCuB94MAVQRi+LDo5ckcAUX2pMDCAJJ/k0vPeBNaDTdmtFPjFoXwq5n2U7JCdCqS/1sq1IRFxsu3SwB+IRuejSyALEqt1nIIxzCxtXEvqX0s6zt8sez1/uApn/eFEG9/8GgkiFwe7Xo1WKYxTwdMa5HMTS41L0Jq1mzua77DRIA4FpBsU+Eh06npYqcKvtbv5uaIY+2foPPkq7F1k3iE2CtNhPJ6QI61Ku2KjSJ6mnyhTbyEB70jp0acSEfzG1VOH9StCN20nsHAC0uCd/0yDrNHuA==
```

**rand_R**

```
pSGkxaQRxypkzL08kFo9og==
```

**rand_T**

```
lwf+YABhpvHgcS4KpJYhxg==
```
CC: What the system does

A ballot has the form \( \left( \left\{ v \right\}_{pk_T}^m, \left\{ d \right\}^M_{pk_R}, zkp \right) \), where \( \left\{ \cdot \right\} \) is randomised encryption that supports re-encryption, plaintext equivalence testing, and verifiable threshold decryption (e.g. ElGamal).

On receipt of the ballots, the system:

- removes malformed ballots;
- verifiable re-encryption mixes the remaining ballots
- uses PETs to group ballots into sets corresponding to same credential
- uses PETs to determine if any set contains two different votes
- discards all the ballots of such sets, and all but one of the remaining sets
- uses PETs to discard any remaining ballot not corresponding to a credential on the published electoral roll
- publishes the results of all these calculations
- decrypts and publishes the votes in the remaining ballots

All of these computations can be verified by any observer or voter.
Caveat Coercitor (based on JCJ-Civitas)

Voter with credential \(d\)

Ballot

\[ (\{v\}^{m}_{pk_T}, \{d\}^{m'}_{pk_R}, zk) \]

remove malformed ballots

If credential \(d\) has >1 ballot with different votes, remove all \(d\)'s ballots

verifiably remove ineligible ballots (using PETs)

verifiable threshold decryption

results

Electoral register

\[ (\{d\}^{m''}_{pk_R}, Anne Jones) \]
Coercion evidence

<table>
<thead>
<tr>
<th>n</th>
<th>m</th>
<th>Number of credentials having n ballots corresp. to m different votes</th>
<th>Percentage of ballots</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>40,485,324</td>
<td>83%</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2,128,347</td>
<td>4.3%</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2,654,913</td>
<td>5.4%</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1,748,362</td>
<td>3.6%</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>549,472</td>
<td>1.1%</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3,842</td>
<td>0.0079%</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,755</td>
<td>2</td>
<td>3</td>
<td>0.0000061%</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>48,783,530</td>
<td>100%</td>
</tr>
</tbody>
</table>
Caveat coercitor

* An attacker can coerce a voter:
  - just demands her credential, and votes on her behalf
  - or, persuades her to use a corrupt ballot forming applet
  - or, installs malware on her machine, etc

But the system will receive multiple ballots for you with different votes. They will not be counted, but the fact will be published.
  - The most the coercer can achieve is forced abstention.
  - The degree of coercion will be published, and is verifiable.
<table>
<thead>
<tr>
<th>Attack</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker persuades you to use a corrupt applet that leaks your vote,</td>
<td>Be careful to use a safe applet. You can check your</td>
</tr>
<tr>
<td>or submits his preference instead of yours.</td>
<td>ballot on another computer (expert mode).</td>
</tr>
<tr>
<td>Attacker steals your credential (unknown to you), or forces you to</td>
<td>Vote normally.</td>
</tr>
<tr>
<td>reveal your credential (known to you).</td>
<td></td>
</tr>
<tr>
<td>Attacker tries to disrupt the election by making it appear as if</td>
<td>Attacker needs to steal or coerce a large number of</td>
</tr>
<tr>
<td>there were lots of coercion.</td>
<td>voters.</td>
</tr>
</tbody>
</table>
What happens if the table looks more like this?

<table>
<thead>
<tr>
<th>n</th>
<th>m</th>
<th>Number of credentials having n ballots corresp. to m different votes</th>
<th>Percentage of ballots</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>15,852,963</td>
<td>32%</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2,128,347</td>
<td>4.3%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>13,105,913</td>
<td>27%</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1,748,362</td>
<td>3.6%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>9,832,472</td>
<td>20%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>8,219</td>
<td>0.017%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,755</td>
<td>2</td>
<td>7</td>
<td>0.0000014%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>48,783,530</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>
The system distinguishes the following 3 cases, but not the subcases.

if $n = 1$ and $m = 1$
- voter cast for one candidate
- voter knowingly abstained and attacker obtained her credentials and cast for one candidate

if $n > 1$ and $m = 1$
- voter cast multiple ballots for one candidate
- attacker obtained voter’s credentials and each cast ballots for the same candidate
- voter knowingly abstained and attacker obtained her credentials and cast ballots for one candidate

if $n > 1$ and $m > 1$
- voter cast multiple ballots for several different candidates
- voter cast multiple ballots for one candidate, attacker obtained her credentials and cast for another candidate
- voter and attacker each cast for several different candidates
- voter knowingly abstained and attacker cast votes on her behalf for multiple different candidates
Conclusions

Idea of *Caveat Coercitor*

- Reduce security requirements
  - coercion proof $\leadsto$ coercion evidence
- Increase usability
  - users judge security for themselves
  - mitigations for threats
  - election recoverability