Caveat Coercitor

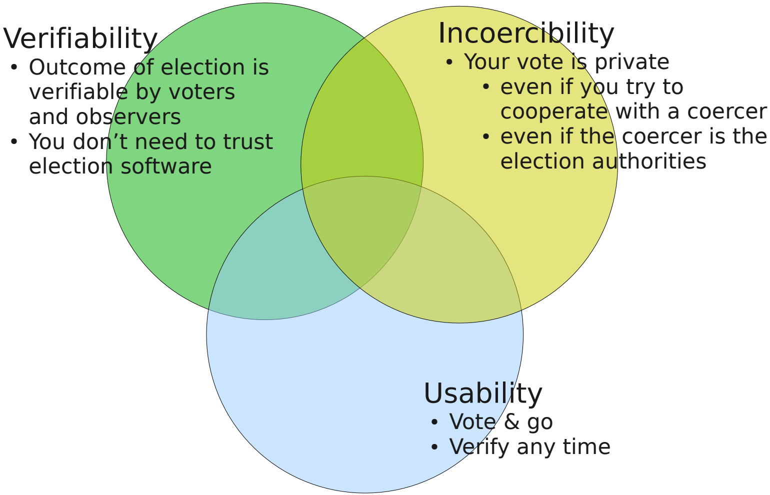
Mark Ryan and Peter Y. A. Ryan
University of Birmingham University of Luxembourg

TVS/SerTVS meeting
University of Birmingham
6th April 2011

Outline

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

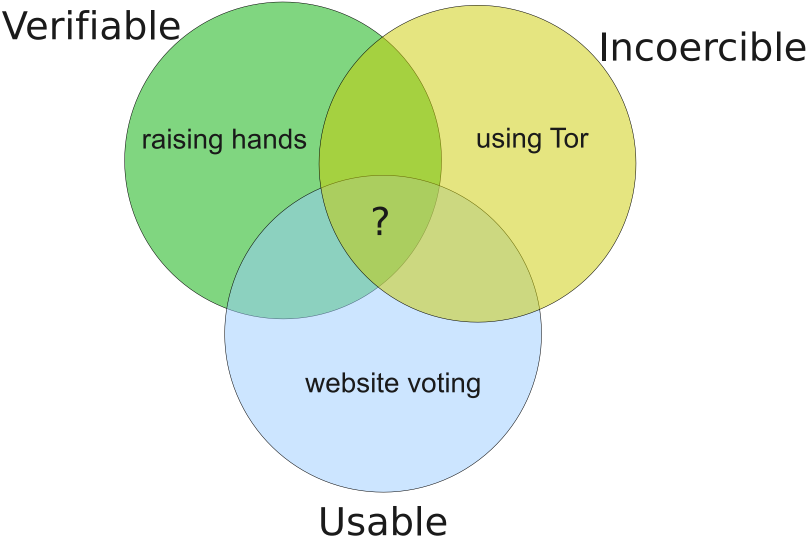
Incoercibility

- 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
Incoercible
raising hands
using Tor
?
website voting
Usable

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*”)

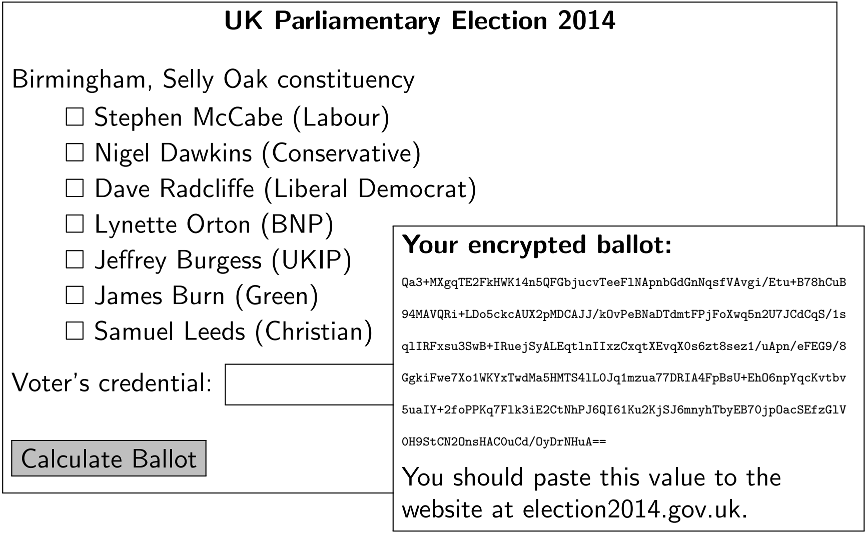
CC: What the voter does

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)

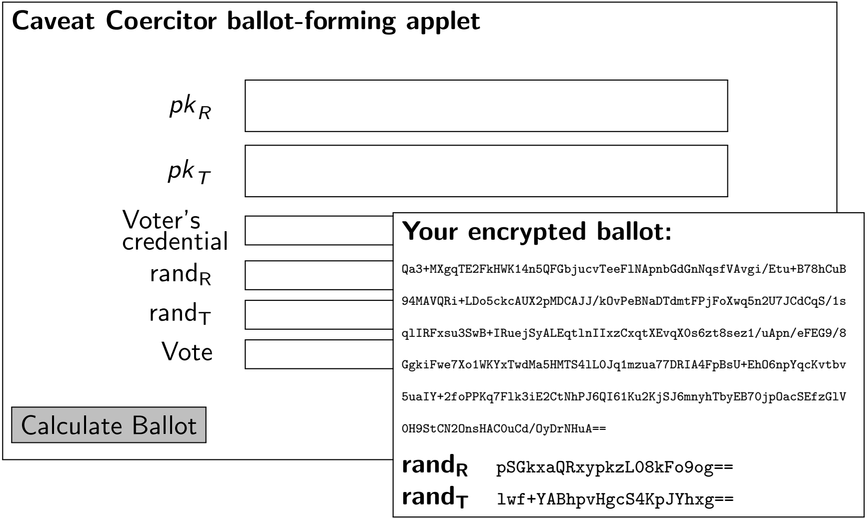
Voter's credential:

Your encrypted ballot:

```
Qa3+MXgqTE2FkHWK14n5QFGbjucvTeeFlNApnbGdGnNqsfVAvgi/Etu+B78hCuB
94MAVQRi+LDo5ckcAUX2pMDCAJJ/kOvPeBNaDTdmtFPjFoXwq5n2U7JCdCqS/1s
qlIRFxsu3SwB+IRuejSyALEqtlnIIxzCxqtXEvqX0s6zt8sez1/uApn/eFEG9/8
GgkiFwe7Xo1WKYxTwdMa5HMTS4lL0Jq1mzua77DRIA4FpBsU+EhO6npYqcKvtbv
5uaIY+2foPPKq7Flk3iE2CtNhPJ6QI61Ku2KjSJ6mnyhTbyEB70jp0acSEfzGlV
0H9StCN2OnsHAC0uCd/OyDrNHuA==
```

You should paste this value to the website at election2014.gov.uk.

Ballot formation applet for experts



Caveat Coercitor ballot-forming applet

pk_R

pk_T

Voter's credential

rand_R

rand_T

Vote

Your encrypted ballot:

```
Qa3+MXgqTE2FkHWK14n5QFGbjucvTeeFlNApnbGdGnNqsfVAvgi/Etu+B78hCuB
94MAVQRi+LDo5ckcAUX2pMDCAJJ/kOvPeBNaDTdmtFPjFoXwq5n2U7JCdCqS/1s
qlIRFxsu3SwB+IRuejSyALEqtlnIIxzCxqtXEvqXOs6zt8sez1/uApn/eFEG9/8
GgkiFwe7Xo1WKYxTwdMa5HMTS4lLOJq1mzua77DRIA4FpBsU+EhO6npYqcKvtbv
5uaIY+2foPPKq7Flk3iE2CtNhPJ6QI61Ku2KjSJ6mnyhTbyEB7OjpOacSEfzGlV
OH9StCN2OnsHACOuCd/OyDrNHuA==
```

rand_R pSGkxaQRxypkzL08kFo9og==

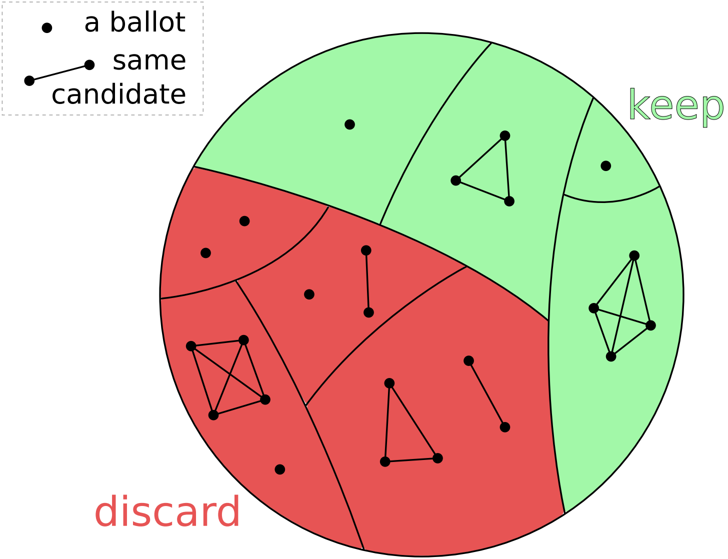
rand_T lwf+YABhpvHgcS4KpJYhxg==

CC: What the system does

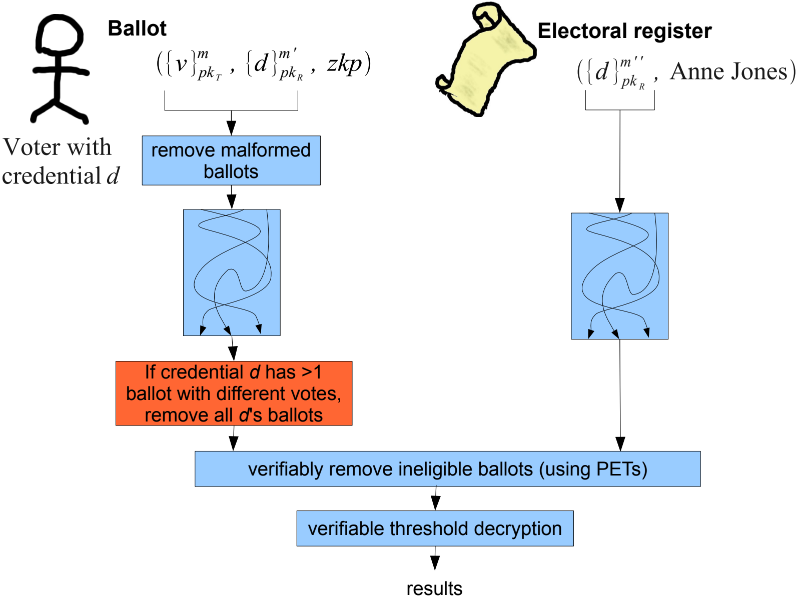
A ballot has the form $\left(\{v\}^{m}_{pk_T}, \{d\}^{M'}_{pk_R}, zkp\right)$, where $\{\cdot\}^{\cdot}_{\cdot}$ is randomised encryption that supports re-encryption, plaintex equivalence testing, and verifiable threshold decryption (e.g. ElGamal).
On receipt of the ballots, the system:

- 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.


a ballot
same candidate
keep
discard

Caveat Coercitor (based on JCJ-Civitas)



Voter with credential d

Ballot

$(\{v\}^{m}_{pk_T}, \{d\}^{m'}_{pk_R}, zkp)$

remove malformed ballots

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

Electoral register

$(\{d\}^{m''}_{pk_R}, \text{Anne Jones})$

verifiably remove ineligible ballots (using PETs)

verifiable threshold decryption

results

Coercion evidence

n	*m*	*Number of credentials having n ballots corresp. to m different votes*	*Percentage of ballots*
1	1	40,485,324	83%
2	1	2,128,347	4.3%
	2	2,654,913	5.4%
3	1	1,748,362	3.6%
	2	549,472	1.1%
	3	3,842	0.0079%
⋮			
1,755	2	3	0.0000061%
⋮			
Total		48,783,530	100%

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.

Possible attacks

Attack	Mitigation
Attacker persuades you to use a corrupt applet that leaks your vote, or submits his preference instead of yours.	Be careful to use a safe applet. You can check your ballot on another computer (expert mode).
Attacker steals your credential (unknown to you), or forces you to reveal your credential (known to you).	Vote normally.
Attacker tries to disrupt the election by making it appear as if there were lots of coercion.	Attacker needs to steal or coerce a large number of voters.

Coercion resistance & coercion evidence

What happens if the table looks more like this?

n	*m*	*Number of credentials having n ballots corresp. to m different votes*	*Percentage of ballots*
1	1	15,852,963	32%
2	1	2,128,347	4.3%
	2	13,105,913	27%
3	1	1,748,362	3.6%
	2	9,832,472	20%
	3	8,219	0.017%
⋮			
1,755	2	7	0.0000014%
⋮			
Total		48,783,530	100%

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 knowlingly 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 $\rightsquigarrow$ coercion evidence
- Increase usability
  - users judge security for themselves
  - mitigations for threats
  - election recoverability