Using Prêt à Voter in Victorian State Elections

EVT     August 2012

Craig Burton¹  Chris Culnane²  James Heather²
Thea Peacock³  Peter Y. A. Ryan³  Steve Schneider²
Sriram Srinivasan²  Vanessa Teague⁴  Roland Wen⁵  Zhe Xia²
Structure of talk

- Voting in the State of Victoria, Australia
- VEC’s motivation for e-voting
- Introducing the Prêt a Voter voter-verifiable system
- Adapting to the VEC requirements: practical challenges
- Conclusion
Legislative Assembly (Lower House)

- Full preferential voting: number the candidates in order of preference.

Legislative Council (Upper House)

- ATL: select exactly one choice;  or
- BTL: number the candidates in order of preference

VEC’s motivation for electronic voting

- VEC was an early adopter of e-voting (2006)
- **flexibility:** for remote (but supervised) voting including overseas, out of state, out of district
- **accessibility:** supports voters with disabilities. Electronic voting machines also handle foreign languages. Complexity of ballots means need for help to avoid malformed ballots – but human help loses privacy
- **usability:** to reduce (accidental) informal ballots
- **BUT:** proprietary system not open to inspection; lack of verifiability; issues with integration with VEC processes
- **WANT** e-voting but recognise the need for verifiability
Context of this project

- **Australian elections**: solution needs to be able to handle STV and preferential voting. Prêt à Voter judged to be the most appropriate voter-verifiable system able to support this.

- **usability vs security**: what can you ask and expect voters to do?

- **scalability**: issues to be resolved for us to scale up to a state election.

- **pragmatics**: scanning (including OCR) and printing.

- **integrity and trust**: the electorate must have confidence in the solution.
Prêt à Voter

• A voter-verifiable voting system

• Verifiability: voters, independent checkers can verify stages of the election

• Integrity: evidence provided that the result is correct

• Privacy: have to trust some elements of the system, but aim to minimize this
Voting with Prêt à Voter

- Place X or preferences against desired candidate. (*candidates in random order*)
- Separate left hand side.
- Destroy left hand side.
- Cast (scan) vote.
- Take receipt home.

<table>
<thead>
<tr>
<th></th>
<th>4. Diane</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Bob</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5. Elaine</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3. Crystal</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

#1726
Publish the ballots cast

- Voter receipts prevent election officials from altering or removing votes.
- Voters confirm inclusion of their vote
Tallying the votes

Public bulletin board of votes cast.

2 4 3 5 1
#1665

5 3 4 1 2
#0809

4 5 3 1 2
#2197

3 5 2 1 4
#1726
Tallying the votes

Public bulletin board of votes cast.

Public list of votes, shuffled and decrypted.

Votes need to be decrypted to be tallied.

Alice 4
Bob 5
Crystal 1
Diane 3
Elaine 2

etc

etc

etc
Tallying the votes

Public bulletin board of votes cast.

Public list of votes, shuffled and decrypted.

The links are not published (receipts are not linked to votes)
Tallying

When the votes are cast:

• Publish the votes cast (newspaper, or web bulletin board) – these should match the receipts, and voters can check.
• Mix up the votes (see next slide), so resulting votes are not linked to input votes (which correspond to receipts):
  • Decrypt the mixed votes
  • Publish the resulting votes.
• Count the votes.
Re-encryption mixnets with proofs (Chaum; Park et al.; Sako and Kilian)

- Re-encryption mixing: \( \{c, r_1\} \rightarrow \{c, r_2\} \) are different encryptions of \( c \)
Re-encryption mixnets with proofs (Chaum; Park et al.; Sako and Kilian)

- Tellers provide `proofs of shuffles`: that the set of encrypted values is not changed from one stage to the next.
- These proofs can be independently checked.
End-to-end Verifiability for Prêt à Voter

Voters → Ballot Casting → Encrypted Votes (Verify by receipts)

Encrypted Votes → Ballot Shuffling by mixnet → Encrypted Votes (Verify by checking proofs)

Encrypted Votes → Decrypt and Count → Results (Verify by public information)

End-to-end verifiability
Practical Challenges
Practical challenges

• In practice in Victorian State elections there are typically around 35+ BTL candidates

• Prêt à Voter requires those candidates to be in a random order on each ballot

• Significant cryptography required to create the ballot forms

• Presenting 35+ spaces for voters to write preferences in a single column will require a long ballot form.

• Difficult for voters to find their choices by hand; issues around the order candidates are presented to voters

• Accessibility issues are compounded
Adapting Prêt à Voter: Front end

- **Solution**: Use an offline Electronic Ballot Marker to assist the voter to complete the ballot.
- It will capture the voter’s preferences in a user-friendly way, and will print the preferences on the ballot form.
- Presents the candidates in the given fixed order
-Captures the voters preferences via touch screen
-Prints the preferences onto the ballot form in the appropriate permutation
- Voter confirms selection before scanning.
- Alerts voter if ballot not well formed
- Can have accessibility plug-ins (vision/mobility impaired) and offer different languages.
- **NB**: does lose the attractive feature of Prêt à Voter that no device learns the vote. Seems unavoidable.
End-to-end Verifiability for Prêt à Voter with EBM

Voters → Construct Ballot with EBM → Printed Vote → Ballot Casting → Encrypted Votes → Ballot Shuffling by mixnet → Encrypted Votes → Decrypt and Count → Results

Verify by checking EMB printing → Verify by receipts → Verify by checking proofs → Verify by public information

End-to-end verifiability
VEC Ballot Form
Ballot form gives the permutation

<table>
<thead>
<tr>
<th>No. 1</th>
<th>Legislative Assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>Donna</td>
</tr>
<tr>
<td>(</td>
<td>Alice</td>
</tr>
<tr>
<td>(</td>
<td>Charlie</td>
</tr>
<tr>
<td>(</td>
<td>Bob</td>
</tr>
<tr>
<td></td>
<td>Legislative Council</td>
</tr>
<tr>
<td></td>
<td>Above the Line (ATL)</td>
</tr>
<tr>
<td>[]</td>
<td>Lib Dem</td>
</tr>
<tr>
<td>[]</td>
<td>Labour</td>
</tr>
<tr>
<td>[]</td>
<td>Green</td>
</tr>
</tbody>
</table>

Serial No. 1
(Donna, Alice, Charlie, Bob),
(Lib Dem, Labour, Green),
(Steve, Vanessa, Craig, Peter Chris, Thea, James)
Ballot form gives the permutation

<table>
<thead>
<tr>
<th>No. 1</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Legislative Council Below the Line (BTL)</td>
</tr>
<tr>
<td>( )</td>
<td>Steve</td>
</tr>
<tr>
<td>( )</td>
<td>Vanessa</td>
</tr>
<tr>
<td>( )</td>
<td>Craig</td>
</tr>
<tr>
<td>( )</td>
<td>Peter</td>
</tr>
<tr>
<td>( )</td>
<td>Chris</td>
</tr>
<tr>
<td>( )</td>
<td>Thea</td>
</tr>
<tr>
<td>( )</td>
<td>James</td>
</tr>
</tbody>
</table>
A VEC ballot example

The front side

The back side
Victorian Voter Experience
1. Language selection and training

Language:
- English [X]
- French [ ]
- Chinese [ ]

Training:
- Yes [X]
- No [ ]
2. Scan candidate QR code (device obtains permutation)
3a. Construct vote via voting device (LA + LC-ATL)

LA:
Alice: 4
Bob: 1
Charlie: 3
Donna: 2

LC-ATL:
Green [ ]
Labour [X]
Lib Dem [ ]
3b. Construct vote via voting device (LA + LC-BTL)

LC-BTL:
Chris: 6
Craig: 1
James: 7
Peter: 2
Steve: 3
Thea: 4
Vanessa: 5
3c. Vote casting for blind voters

<table>
<thead>
<tr>
<th>No. 1</th>
<th>Legislative Assembly</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>()</td>
<td>Donna</td>
<td></td>
</tr>
<tr>
<td>()</td>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>()</td>
<td>Charlie</td>
<td></td>
</tr>
<tr>
<td>()</td>
<td>Bob</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Legislative Council</strong></td>
<td></td>
</tr>
<tr>
<td>[ ]</td>
<td><strong>Above the Line (ATL)</strong></td>
<td></td>
</tr>
<tr>
<td>[ ]</td>
<td>Lib Dem</td>
<td></td>
</tr>
<tr>
<td>[ ]</td>
<td>Labour</td>
<td></td>
</tr>
<tr>
<td>[ ]</td>
<td>Green</td>
<td></td>
</tr>
</tbody>
</table>

You have voted 4 for Alice. Now please vote for Bob.
4a. Overprint on ballot form (LA + LC-ATL)

<table>
<thead>
<tr>
<th>No. 1</th>
<th>Legislative Assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2)</td>
<td>Donna</td>
</tr>
<tr>
<td>(4)</td>
<td>Alice</td>
</tr>
<tr>
<td>(3)</td>
<td>Charlie</td>
</tr>
<tr>
<td>(1)</td>
<td>Bob</td>
</tr>
</tbody>
</table>

**Legislative Council Above the Line (ATL)**

| [ ]   | Lib Dem                     |
| [X]   | Labour                      |
| [ ]   | Green                       |

<table>
<thead>
<tr>
<th>No. 1</th>
<th>Legislative Council Below the Line (BTL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>Steve</td>
</tr>
<tr>
<td>( )</td>
<td>Vanessa</td>
</tr>
<tr>
<td>( )</td>
<td>Craig</td>
</tr>
<tr>
<td>( )</td>
<td>Peter</td>
</tr>
<tr>
<td>( )</td>
<td>Chris</td>
</tr>
<tr>
<td>( )</td>
<td>Thea</td>
</tr>
<tr>
<td>( )</td>
<td>James</td>
</tr>
</tbody>
</table>

Front Side

Back Side (empty)
# 4b. Overprint on ballot form

(LA + LC-BTL)

<table>
<thead>
<tr>
<th>Ballot form</th>
<th>Serial number: 1</th>
</tr>
</thead>
</table>

**Legislative Assembly**

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2)</td>
<td>Donna</td>
</tr>
<tr>
<td>(4)</td>
<td>Alice</td>
</tr>
<tr>
<td>(3)</td>
<td>Charlie</td>
</tr>
<tr>
<td>(1)</td>
<td>Bob</td>
</tr>
</tbody>
</table>

**Legislative Council Above the Line (ATL)**

<table>
<thead>
<tr>
<th></th>
<th>Party</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lib Dem</td>
</tr>
<tr>
<td></td>
<td>Labour</td>
</tr>
<tr>
<td></td>
<td>Green</td>
</tr>
</tbody>
</table>

**Legislative Council Below the Line (BTL)**

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3)</td>
<td>Steve</td>
</tr>
<tr>
<td>(5)</td>
<td>Vanessa</td>
</tr>
<tr>
<td>(1)</td>
<td>Craig</td>
</tr>
<tr>
<td>(2)</td>
<td>Peter</td>
</tr>
<tr>
<td>(6)</td>
<td>Chris</td>
</tr>
<tr>
<td>(4)</td>
<td>Thea</td>
</tr>
<tr>
<td>(7)</td>
<td>James</td>
</tr>
</tbody>
</table>

Front Side (ATL empty)  
Back Side
5. Shred the names

<table>
<thead>
<tr>
<th>Legislative Assembly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
</tr>
<tr>
<td>Bob</td>
</tr>
<tr>
<td>Charlie</td>
</tr>
<tr>
<td>Donna</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Legislative Council Above the Line (ATL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lib Dem</td>
</tr>
<tr>
<td>Labour</td>
</tr>
<tr>
<td>Green</td>
</tr>
</tbody>
</table>

Front side: LA + LC-ATL candidates
Back side: LC-BTL candidates
6a. Submit vote (LA + LC-ATL)

- No. 1
- (2)
- (4)
- (3)
- (1)
- [ ]
- [X]
- [ ]
- Front
- Back

① Scan

② Display to the voter

③ No.1: {2,4,3,1}, [2], {}

Submit to WBB

Bulletin Board

No. 1
(2) ( )
(4) ( )
(3) ( )
(1) ( )
[ ] ( )
[X] ( )
[ ] ( )
6b. Submit vote (LA + LC-BTL)

① Scan
② Display to the voter
③ Satisfy/modify
④ Submit to WBB

Bulletin Board

No. 1: {2,4,3,1}, [], {3,5,1,2,6,4,7}
Overprinted Signature

① {No.1: {2, 4, 3, 1}, [2], {}}_SK(WBB)
<table>
<thead>
<tr>
<th>No. 1</th>
<th>No. 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2)</td>
<td>(3)</td>
</tr>
<tr>
<td>(4)</td>
<td>(5)</td>
</tr>
<tr>
<td>(3)</td>
<td>(1)</td>
</tr>
<tr>
<td>(1)</td>
<td>(2)</td>
</tr>
<tr>
<td>[ ]</td>
<td>(6)</td>
</tr>
<tr>
<td>[ ]</td>
<td>(4)</td>
</tr>
</tbody>
</table>

{No.1: {2, 4, 3, 1}, [], {3, 5, 1, 2, 6, 4, 7}}_SK(WBB)

① Overprinted Signature

② overprint
8a. WBB check later (LA + LC-ATL)
8b. WBB check later (LA + LC-BTL)

<table>
<thead>
<tr>
<th></th>
<th>No. 1</th>
<th></th>
<th>Bulletin Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>[ ]</td>
<td></td>
<td>[ ]</td>
<td></td>
</tr>
<tr>
<td>[ ]</td>
<td></td>
<td>[ ]</td>
<td></td>
</tr>
<tr>
<td>[ ]</td>
<td></td>
<td>[ ]</td>
<td></td>
</tr>
</tbody>
</table>

receipt

No. 1
(2) (3)
(4) (5)
(3) (1)
(1) (2)
[ ] (6)
[ ] (4)
[ ] (7)
Adapting Prêt à Voter: Processing the votes

• We use Douglas Wikström’s implementation of a re-encryption mixnet: the Verificatum system.

• This provides shuffles, re-encryptions and proofs.

• It also provides the final decryption step following the mix, to produce a list of plaintext votes.

• Given the large numbers of candidates, each preference list is compressed into a small number of ciphertexts to optimise the mixing process, and expanded at the other end. These steps are also verifiable. [Technical details in the paper]
## Implementation Timings

<table>
<thead>
<tr>
<th>Processing stage</th>
<th>Time taken</th>
<th>Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher generation</td>
<td>39hrs 34mins</td>
<td>1.4 seconds per ballot</td>
</tr>
<tr>
<td>Mixing ATL</td>
<td>2hrs 0mins</td>
<td>12 ballots per second</td>
</tr>
<tr>
<td>Decryption ATL</td>
<td>12mins 9s</td>
<td>120 ballots per second</td>
</tr>
<tr>
<td>Mixing BTL</td>
<td>1hr 33mins</td>
<td>2 ballots per second</td>
</tr>
<tr>
<td>Decryption BTL</td>
<td>9mins 27sec</td>
<td>18 ballots per second</td>
</tr>
<tr>
<td>Reconstructing BTL</td>
<td>57mins 10sec</td>
<td>3 ballots per second</td>
</tr>
</tbody>
</table>

**100,000 ballots:**
38 candidates, 8 parties, 90000 ATL + 10000 BTL votes
Distributed Ballot Generation

- Servers inject randomness, and re-encrypt with a different key for the two parts:

  \[(PKp\{c,r1\}, PKm\{c,r1'\}) \rightarrow (PKp\{c,r2\}, PKm\{c,r2'\})\]
Distributed Ballot Generation

Candidate list encrypted with PKm

Provably same candidate list encrypted with PKp

- Servers publish proofs of shuffle
- PKm and PKp are threshold keys
Print on Demand: step 1

- Printer generates a blinding factor $b_i$ for each candidate.
- Encrypts them with $PKp$.
- Sends them to the ballot servers as a ballot request, with a proof of knowledge (ZKP).
Print on Demand: step 2

Ballot server selects an unused ballot: #N

PKp(c_1)
PKp(c_2)
PKp(c_3)
PKp(c_4)

• Bulletin Board
Print on Demand: step 2

- Ballot server selects an unused ballot: #N
- Combines the blinding factors with the encrypted names

Ballot #N

\[ PK_P(c_1+b_1) \]
\[ PK_P(c_2+b_2) \]
\[ PK_P(c_3+b_3) \]
\[ PK_P(c_4+b_4) \]
Print on Demand: step 2

- Ballot server selects an unused ballot: #N
- Combines the blinding factors with the encrypted names
- (Threshold) decrypts the blinded names
Print on Demand: step 3

- Blinded candidate names returned to the printer
Print on Demand: step 4

Ballot #N

- Ballot printer
- Printer removes blindings on names

\[ c_1 + b_1 \]
\[ c_2 + b_2 \]
\[ c_3 + b_3 \]
\[ c_4 + b_4 \]
Print on Demand: step 4

Ballot #N

- Printer removes blindings on names

Ballot printer

c_1
c_2
c_3
c_4
Ballot printer

- Printer removes blindings on names
- Printer can then print ballot form

Ballot #N

- c_1
- c_2
- c_3
- c_4
Auditing printed ballots

• If a printed ballot is challenged...

• ... the ballot servers can threshold decrypt the blinding factors PKp(b_i) provided by the printer,
  ... which enables the c_i + b_i values to be unblinded and checked against the printed ballot

• ... or can threshold decrypt the candidate names Kp(c_i) directly, and check against the printed ballot
• Usability, accessibility, and remote voting, while retaining assurance in the system, are key drivers.

• Prêt à Voter can be customised to the VEC requirements. The main new design feature is the EBM, which introduces fresh challenges. Scaling up also raises issues with processing the votes

• A demonstrator is currently being implemented for evaluation, with a view to VEC trialling it next year

• The system can handle the scale of Australian state elections

• Verifiability comes from the ability to check the information published by the system. The code is also open to inspection, though it’s the output of the code that is verified