Prêt à Voter

Advances in Verifiable Voting Systems

Peter Y A Ryan

Université de Luxembourg

Outline

- The problem.
- Brief history of verifiable voting
- Voter-verifiability.
- Overview of Prêt à Voter.
- OpenVeto
- OpenVote
- Conclusions.

Secure, distributed computing

- At one level, secure, verifiable elections can be regarded as a special case of secure, distributed computation.
- But real elections are different:
  - Voters typically won't have computation power or expertise.
  - Elections need to be scalable
  - Etc....

The Problem

- From the dawn of democracy it was recognised that people would be tempted to try to corrupt the outcome of elections.
- Highly adversarial: system trying to cheat voters, voters trying to cheat the system, coercers trying to influence voters, voters trying to fool coercers etc.
- In the US they have been using technological devices for voting for over a century: e.g., level machines since 1897, punch cards, optical scans, touch screen etc. Prompted by high instance of fraud with paper ballots.
- All have problems, see “Steal this Vote” Andrew Gumbel.

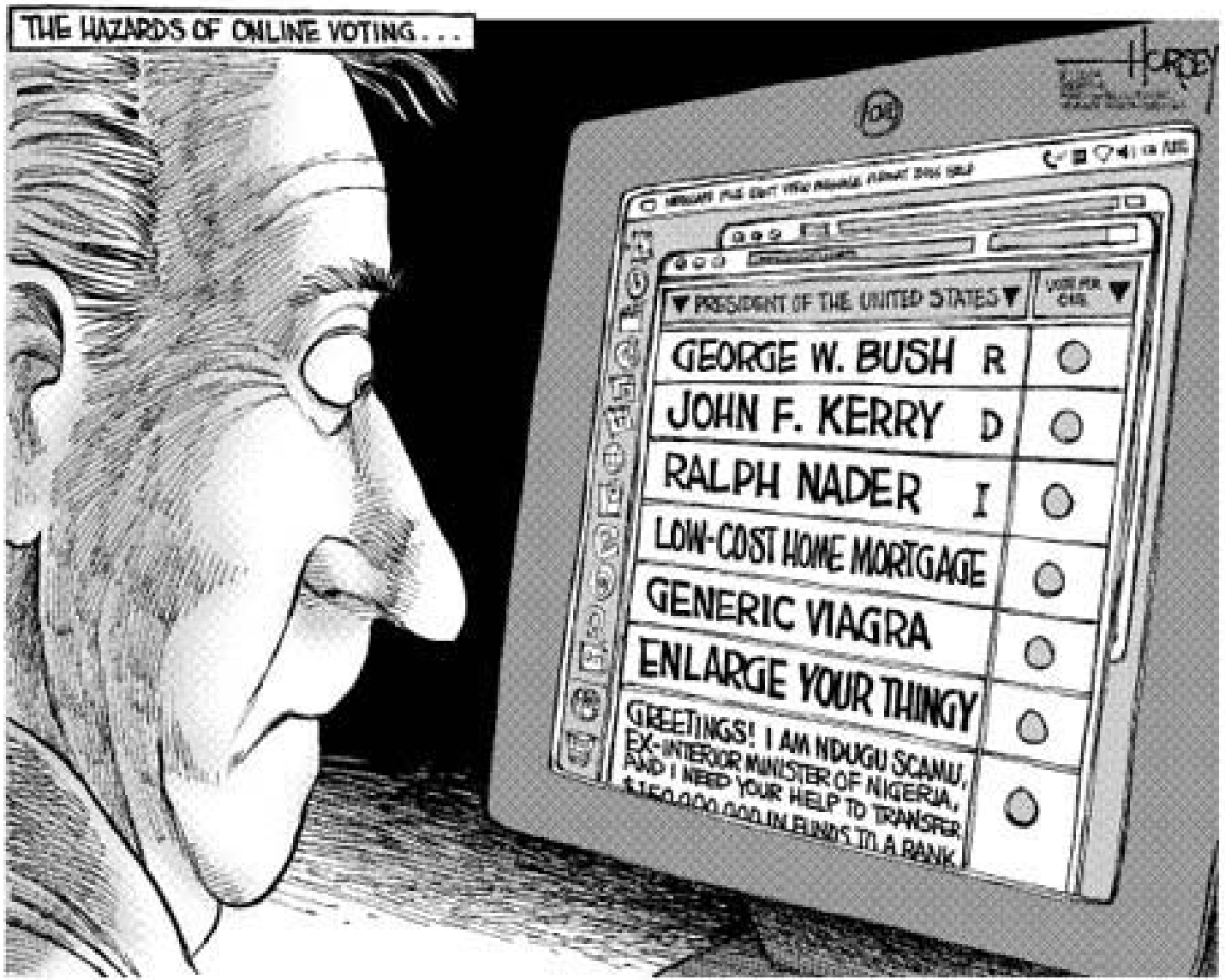
“The Computer Ate my Vote”

- In the 2004 US presidential election, ~30% of the electorate used DRE, touch screen devices.
- Aside from the “thank you for your vote for Kerry, have a nice day” what assurance do they have that their vote will be accurately counted?
- What do you do if the vote recording and counting process is called into question?
- Voter Verifiable Paper Audit Trail (VVPAT) and “Mercuri method”. But paper trails are not infallible either.

Remote vs Supervised

- Important to draw a clear distinction between supervised and remote voting.
- In the former the voter casts their vote in enforced isolation, e.g., in a booth in a polling station.
- Remote voting, e.g., internet, postal etc. such isolation cannot be enforced.
- Hence dangers of coercion.

Hazards of e-voting!



Technical Requirements

- Key requirements:
  - Integrity/accuracy: count accurately reflects votes cast.
  - Ballot secrecy: the way a voter cast their vote should only be known to the voter.
  - Coercion resistance: there should be no way for the voter to prove to a coercer which way they voted, even if the voter is prepared to cooperate with the coercer.
  - Voter verifiability: the voter should be able to confirm that their vote is accurately included in the count.
  - Universal verifiability: everyone should be able to verify the count.
  - Availability: all eligible voters should be able to cast their vote without hindrance throughout the voting period.
  - Ease of use, public trust, cost effective, scalable etc. etc…..

Assumptions

- For the purposes of the talk we will make many sweeping assumptions, e.g.,:
  - An accurate electoral register is maintained.
  - Mechanisms are in place to ensure that voters can be properly authenticated.
  - Mechanisms are in place to prevent double voting.
  - Existence of a secure Web Bulletin Board.
  - Crypto algorithms we use are sufficiently secure.
  - Etc.

Brief history

- 1982: Chaum: anonymising mixes and suggest application to voting.
- 1994(?) Benaloh: receipt-free scheme.
- 2003/4: Chaum: visual crypto scheme
- Neff: VoteHere and MarkPledge.
- Summer 2004-present: Ryan: Prêt à Voter
- 2005: Chaum: PunchScan (~PaV^2)
- 2007: Chaum: Scantegrity II
- 2007: Rivest: ThreeBallot
- 2008: Scantegrity II

Internet

- 2001: Chaum: SureVote-code voting.
- 2005: Juels et al: definition of coercion-resistance and token mechanism.
- 2007: Clarkson et al: Civitas
- 2008: Adida: Helios
- 2008/9: Ryan/Teague: Pretty Good Democracy.

Voter-verifiability in a nutshell

- Voters are provided with an encrypted “receipt”.
- Copies of the receipts are posted to a secure web bulletin board. Voters can verify that their receipt is correctly posted.
- A (universally) verifiable tabulation is performed on the receipts.
- Checks are performed at each stage to detect any attempt to decouple the encryption of the receipt from the decryption performed by the tellers.
- But, proofs provided to the voter of correct encryption of their vote must not be transferable.

Design philosophy

- Verify the election, not the system!
- Assurance of accuracy should be based on maximal transparency and auditability
- Not on claims of correctness of code etc.
- End-to-end verifiability.
- Software independence.
- As simple and understandable as possible.

Prêt à Voter

- Ballot forms encode the vote in familiar form (e.g. a ✘ against the chosen candidate).
- The candidate list is (independently) randomised for each ballot form.
- Information defining the candidate list is buried cryptographically in an “onion” value printed on each ballot form.
- An excess number of forms are generated to allow for random auditing, before, during and after the election.

Prêt à Voter

- Each ballot form has a unique, secret, random seed *s* buried cryptographically with threshold public keys of a number of tellers in an “onion” printed on the form.
- For each form, a permutation of the candidate list is computed as a publicly known function of this seed.
- The seed can only be extracted by the collective actions of tellers, or suitable subset if a threshold scheme is used.

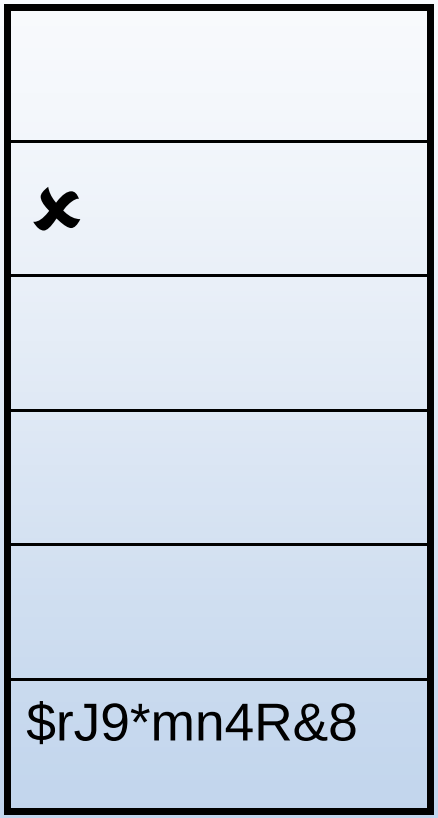
Typical Ballot Sheet

Obelix	
Asterix	
Idefix	
Panormix	
Geriatrix	
	$rJ9*mn4R&8

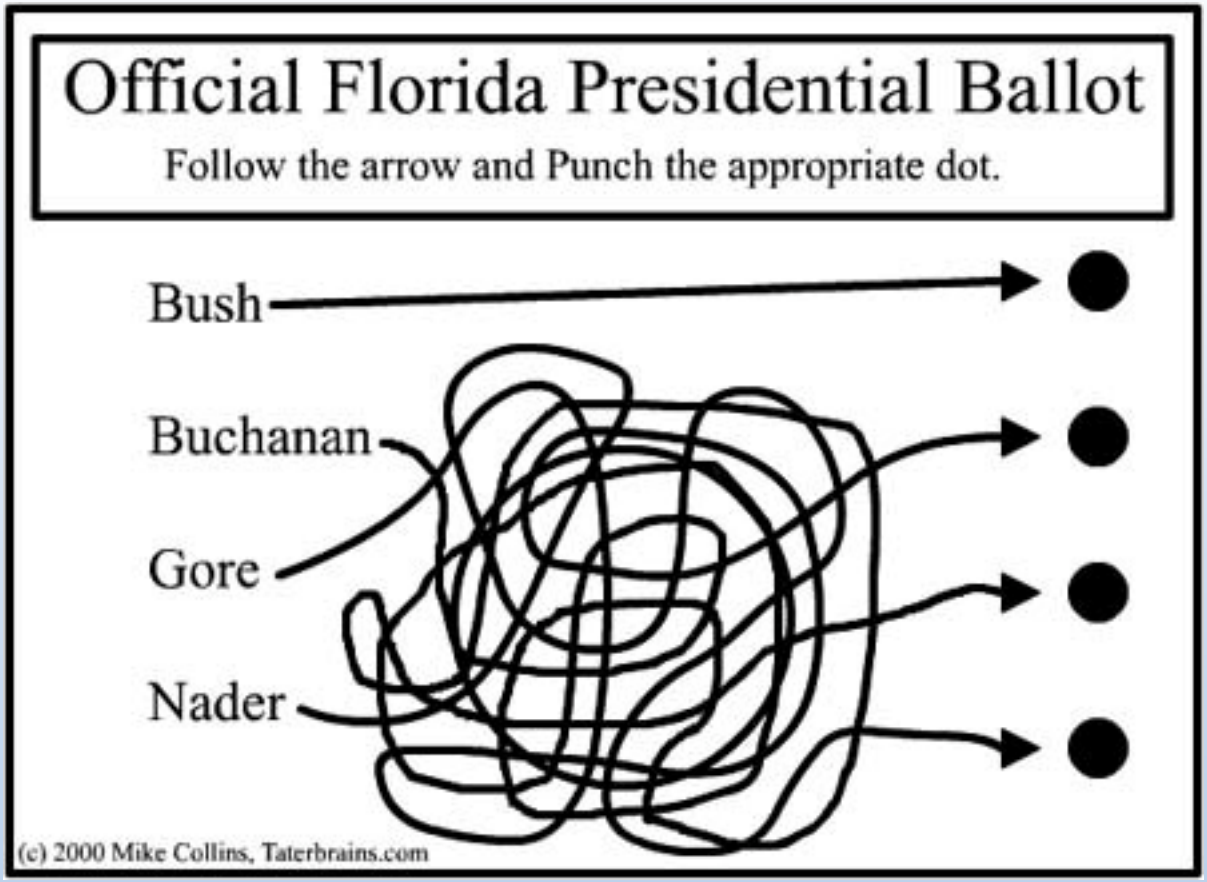
Voter marks their choice

Obelix	
Asterix	✘
Idefix	
Panoramix	
Geriatrix	
	$rJ9*mn4R&8

Voter's Ballot Receipt



Florida 2000



The voting “ceremony”

- Can be varied, but possible scenario:
  - Voter enters the polling station and takes a ballot form at random, sealed in an envelope.
  - The voter goes to a booth, extracts the ballot form and makes their mark.
  - LH strip is destroyed/discarded.
  - The voter leaves the booth with the RH strip, which forms the receipt, and registers with an official.
  - A digital copy, *(r, Onion),* of the receipt is made. The receipt is digitally signed and franked. Additionally, a paper audit copy can be made.
  - The voter exits the polling station with their receipt.

Checking and tabulation

- Digital copies of the receipts are posted to the Web Bulletin Board.
- The voters, or proxies, can visit the WBB and confirm that their receipt appears correctly.
- A verifiable, anonymising tabulation is performed on posted receipts.

Remarks

- The receipt reveals nothing about the vote
- Voter experience simple and familiar.
- No need for voters to have personal keys or computing devices.
- Votes are not directly encrypted, rather the frame of reference, i.e., the candidate list, is randomised and information defining the frame is encrypted.
- Vote casting can be in the presence of an official (à la Française).
- Voters could be allowed to audit ballots of their choice.
- An encrypted paper audit trail can be incorporated.
- Works for ranked, STV etc.

But s**t happens…

- All this is fine as long as we are happy to trust “The Authority”, the tellers etc.
- But we want to trust nobody, just trust in cryptography.
- For the accuracy requirement:
  - Ballot forms may be incorrectly constructed, leading to incorrect encoding of the vote.
  - Ballot receipts could be corrupted before they are entered in the tabulation process.
  - Tellers may perform the decryption incorrectly.
- We now discuss the (fault) detection mechanisms.

Open audit

- A random selection of ballots are audited for well-formedness.
- Voters, or proxies, check that receipts are correctly posted on the BB.
- A verifiable, anonymous tabulation is performed on posted receipts, using partial random checking or Neff style ZK proofs.

OpenVeto (Hoa, Zielinski)

- Boardroom style: only open authenticated channels-no private channels.
- Suppose n members P_i.
- Choose large prime p and a generator g of a subgroup order q of Z_p^*, q|(p-1). In which taking discrete logs is intractable. i.e. Usual El Gamal setting.

OpenVeto

- P_i chooses x_i at random and broadcasts:

  $\alpha_i := g^{xi}$ plus ZK proof of knowledge of x_i.
- After all have broadcast, each checks the ZK proofs and Pi computes:

$$\beta_i := \Pi_{j=1}^{i-1} \alpha_j / \Pi_{j=i+1}^{n} \alpha_j$$

- P_i now broadcasts:

  $\lambda_i := \beta_i^{xi}$ + ZK proof if no veto

  $:= \beta_i^{ri}$ + ZK proof random r_i if veto

The outcome

- All compute:

$$\Pi_{j=1}{}^{n} \lambda_i$$

$$=1 \text{ if no veto}$$

$$\neq 1 \text{ if } >0 \text{ veto}$$

- If all choose $r_i=x_i$, the terms in the exponent cancel out.
- Note: could use crypto commitments in place of ZK proofs, but extra rounds.

OpenVote (Hoa, Ryan, Zielinski)

- Again Pi broadcasts:

  $\alpha_i := g^{xi}$ plus ZK proof of knowledge of x_i.
- After all have broadcast each checks the ZK proofs and Pi computes:

$$\beta_i = g^{yi} := \Pi_{j=1}^{i-1} \alpha_j / \Pi_{j=i+1}^{n} \alpha_j$$

- But now

$$\lambda_i := \beta_i^{xi} g^{vi} + + \text{ ZK proof } vi=0 \vee 1$$

with vi=1 for Yes, vi=0 for No.

OpenVote

- ZK proof: need to show that vi = 0 or 1 without revealing which.
- Form the ElGamal encryption of g^{vi} with PK $\beta_i=g^{yi}$ and randomisation xi:
- $Z_i := (\alpha_i, \lambda_i) = (g^{xi}, g^{yi \cdot xi} g^{vi})$
- Now standard Cramer et al technique to prove encryption of 1 or g.

Tabulation

- All can compute:
- $\Pi_{j=1}^{n} \alpha_i = g^{\Sigma vi}$
- Σv_i is the number of “Yes” votes.
- Can be extended to handle >2 candidates using trick due to Baudrot et al, using a super-increasing sequence to encode the candidates: 2^0, 2^k, 2^{2k}, ...with $2^k > v$, the number of voters.

Conclusions

- Secure voting systems a dynamic area of research.
- Lots of open questions!
- Highly socio-technical in nature.
- Illustrates the use of modern crypto,

Future work

- On the current model:
  - Determine exact requirements.
  - Formal analysis and proofs.
  - Construct threat and trust models.
  - Investigate error handling and recovery strategies.
  - Develop a full, socio-technical systems analysis.
  - Develop prototypes and run trials, e.g., e-voting games!
  - Investigate public understanding, acceptance and trust.

Future work

- Beyond the current scheme:
  - Finalise remote, coercion resistant version (using “capabilities”).
  - Re-encryption mixes for general electoral methods.
  - Establish minimal assumptions.
  - Alternative sources of seed entropy: Voters, optical fibres in the paper, quantum…?
  - Alternative robust mixes.
  - Quantum variants.
  - Unconditional schemes?

Thanks!

Merci de votre attention!

VoteId 2009 in Luxembourg 7-8 September.

EVT/WOTE Montreal

We have post-doc positions in Luxembourg.