

Alternative Schemes

Peter Y A Ryan

University of Luxembourg

Rivest's ThreeBallot scheme

- Voter-verifiability without cryptography!
- In this scheme, each voter casts three ballots.
- Each ballot carries a unique, random serial number.
- One vote for “unselected” candidates, two for selected candidate, distributed randomly across the three ballot forms.
- The ballots are separated and all three are cast, a copy of one chosen at random is retained as the voter's receipt.

ThreeBallot

Anne		Anne		Anne	X
Bob	X	Bob		Bob	
Charles	X	Charles		Charles	
Dave		Dave	X	Dave	X
Edward		Edward		Edward	X
	500439456		738201956		9443823451

Blank Three Ballot form

Asterix		Asterix		Asterix	
Idefix		Idefix		Idefix	
Obelix		Obelix		Obelix	
Panoramix		Panoramix		Panoramix	
28488174		722408712		239961634	

Initialised Form

Asterix		Asterix	x	Asterix	
Idefix		Idefix		Idefix	x
Obelix	x	Obelix		Obelix	
Panoramix		Panoramix		Panoramix	x
28488174		722408712		239961634	

A vote for Obelix...

Asterix		Asterix	x	Asterix	
Idefix		Idefix		Idefix	x
Obelix	x	Obelix	x	Obelix	
Panoramix		Panoramix		Panoramix	x
28488174		722408712		239961634	

The receipt

Asterix	
Idefix	x
Obelix	
Panoramix	x
239961634	

Tabulation

- Suppose that n voters participate. All $3n$ ballots are shuffled and posted to a WBB.
- Tabulation is beautifully straightforward: votes for each candidate added up and n deducted from each total.
- Voter-verification: voters can check that their receipt ballot is correctly posted.

Discussion

- Tricky user interface.
- Enforcing the voting rules without jeopardising ballot secrecy is extremely difficult.
- Reconstruction attacks.
- Coercer chosen pattern attack.
- Etc....
- Probably not a viable scheme as it stands, but demonstrates the possibility of voter-verifiable, secret elections without cryptography.

Scantegrity II



David Chaum
Aleksander Essex
Peter Y. A. Ryan

Richard Carback
Stefan Popoveniuc
Emily Shen

Jeremy Clark
Ronald L. Rivest
Alan T. Sherman

Scantegrity II

- It is a traditional (US) optical scan system
 - which gives verifiably cast as intended
- *With a twist:*
 - which gives verifiably collected as cast, and verifiably counted as collected

The twist

- Invisible ink confirmation codes
- Web site for election, which posts:
 - Confirmation codes on cast ballots
 - Cryptographic commitments linking codes to candidates
- Web service so voters can check that their ballot id + confirmation codes are correct
- Protest protocol to fix errors
- Verifiable counting procedure

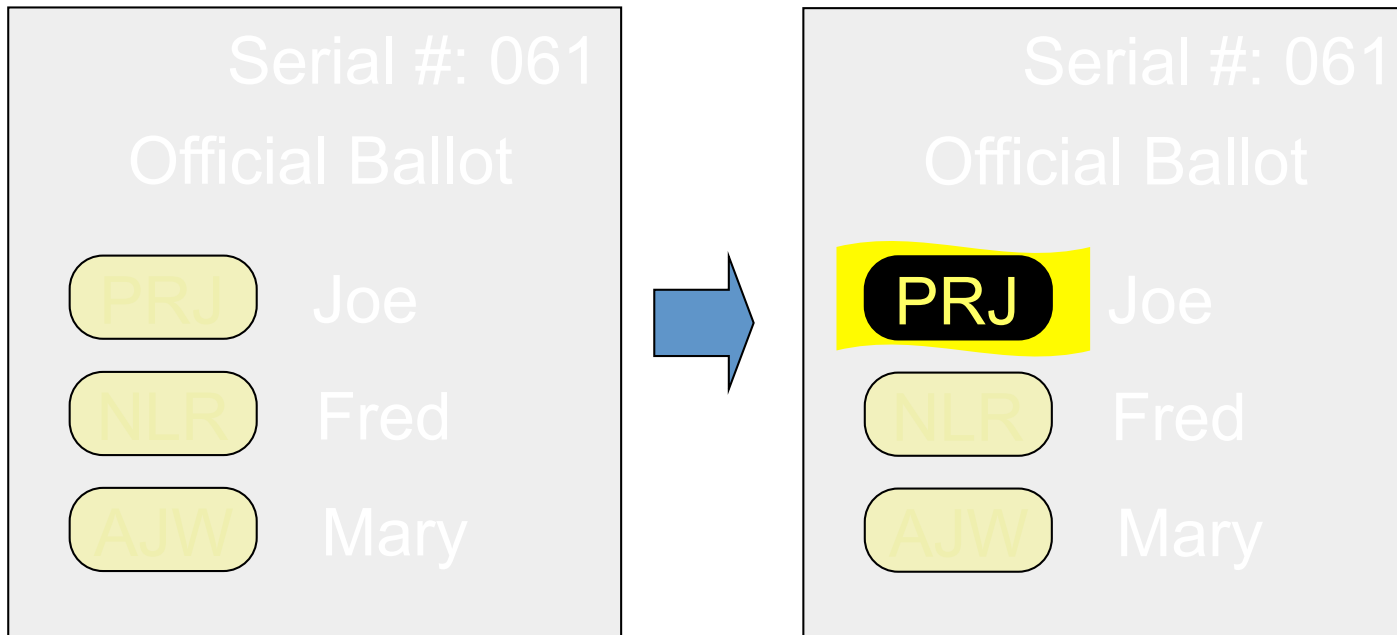
Ballot form

Serial #: 061

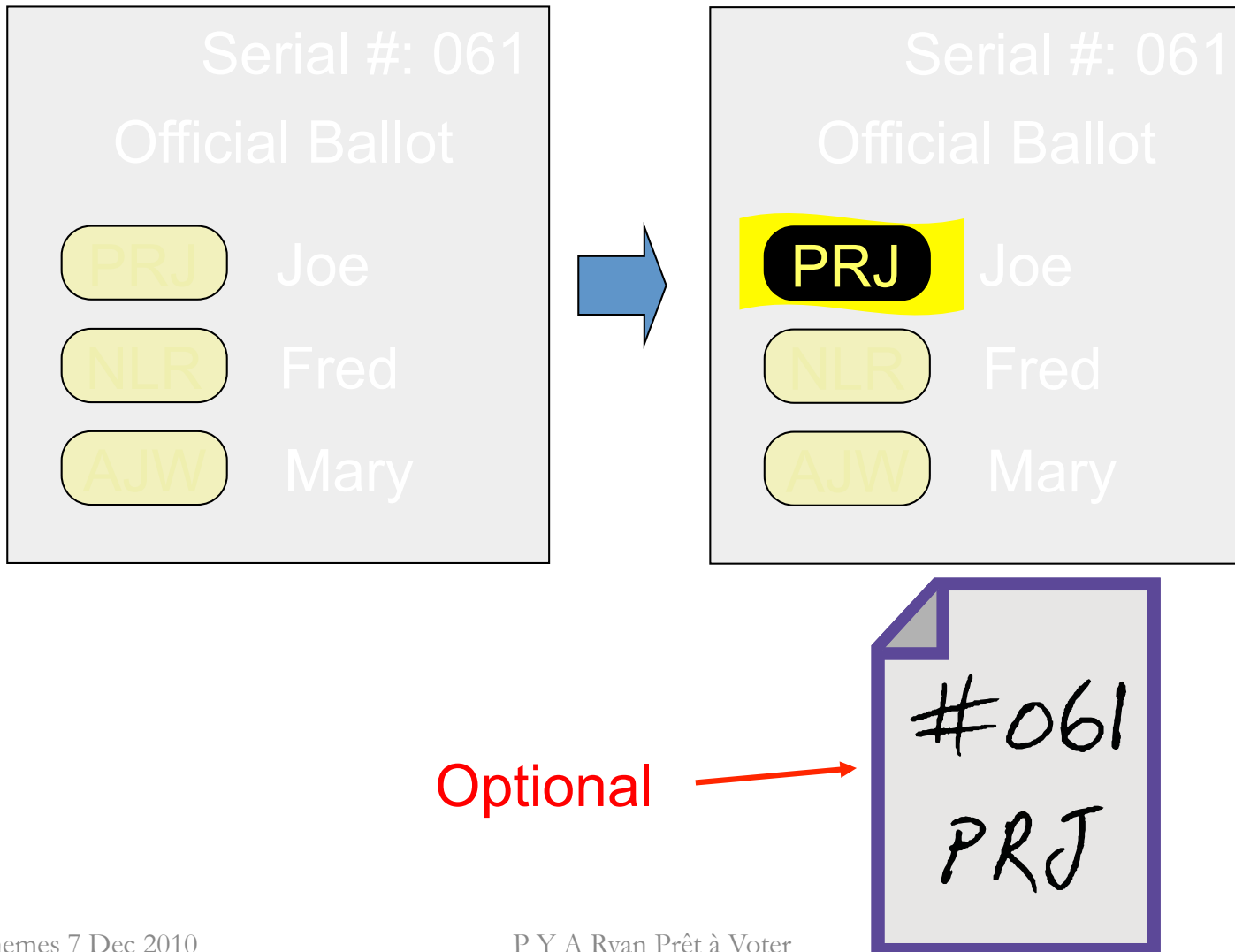
Official Ballot

PRJ	Joe
NLR	Fred
AJW	Mary

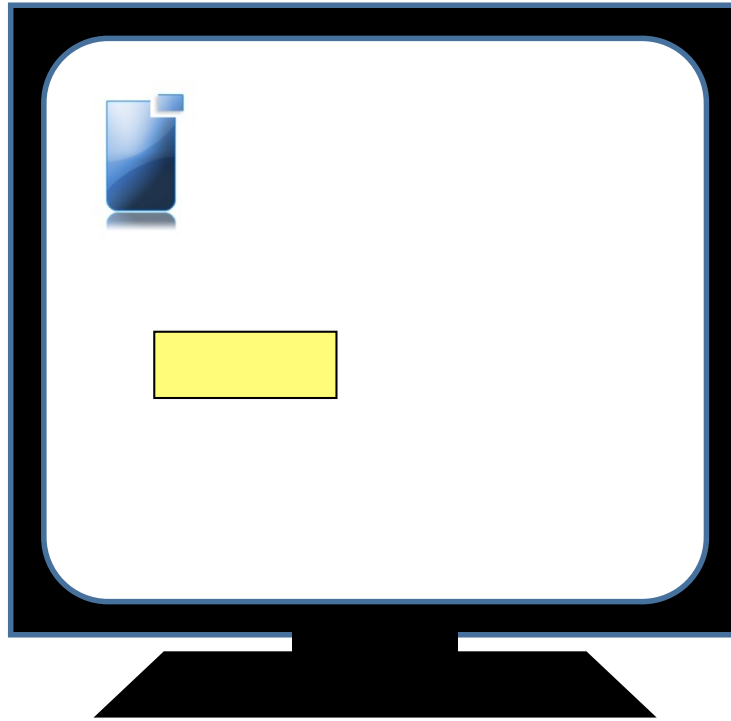
Mark



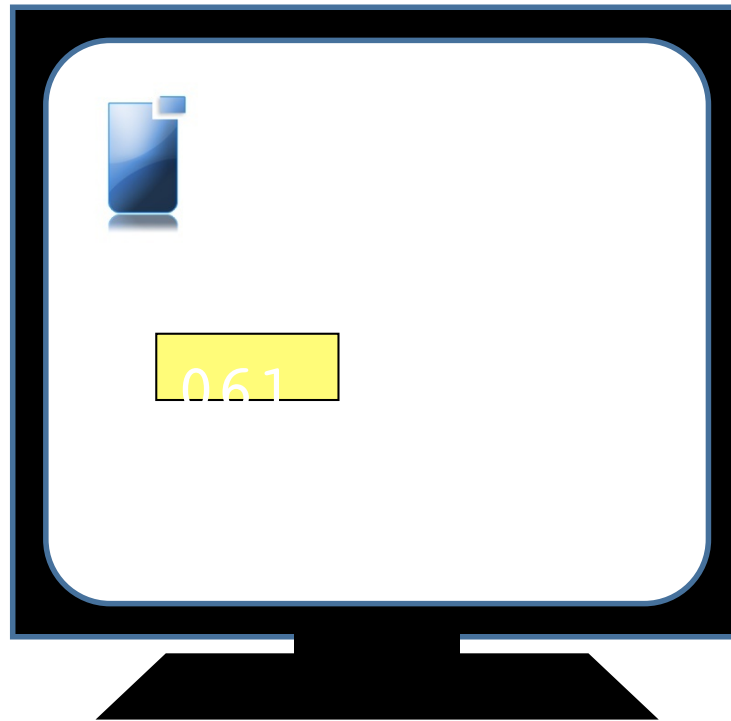
Mark & optionally note



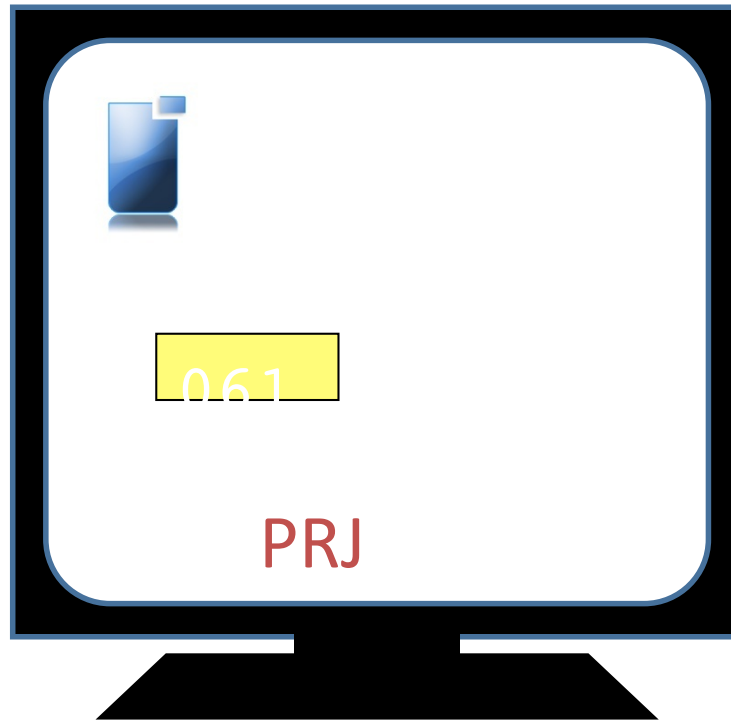
Optional checking of posted serial number & "confirmation codes" online, e.g. from home



Optional checking of posted serial number & "confirmation codes" online, e.g. from home



Optional checking of posted serial number & "confirmation codes" online, e.g. from home



Wombat

- New kiosk voting system developed and implemented by Tel Aviv University and IDC Herzliya.
- Device creates and prints an ElGamal encryption of the vote and plaintext.
- Voter can challenge or cast.
- Plaintext ballots are cast and counted conventionally.

Wombat

- Student election at IDC last week.
- Very minor discrepancies between the electronic and hand counts.

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

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

- P_i now broadcasts:
 $\lambda_i := \beta_i^{x_i}$ + ZK proof if no veto
 $:= \beta_i^{r_i}$ + ZK proof random r_i if veto

The outcome

- All compute:

$$\prod_{j=1}^n \lambda_j$$

=1 if no veto

≠1 if >0 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 P_i broadcasts:

$\alpha_i := g^{x_i}$ plus ZK proof of knowledge of x_i .

- After all have broadcast each checks the ZK proofs and P_i computes:

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

- But now broadcast:

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

with $v_i=1$ for Yes, $v_i=0$ for No.

OpenVote

- ZK proof: need to show that $v_i = 0$ or 1 without revealing which.
- Form the ElGamal encryption of g^{v_i} with PK $\beta_i = g^{y_i}$ and randomisation x_i :
- $z_i := (\alpha_i, \lambda_i) = (g^{x_i}, g^{y_i \cdot x_i} g^{v_i})$
- Now use the Cramer, Damgård, Schoenmakers technique to prove that z_i is an encryption of 1 or g .

Tabulation

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