An Electronic Voting Protocol (revisited)

Paul Valiant

extending work by Dale Neal & Garrett Smith

Source Paper

- "An Anonymous Electronic Voting Protocol for Voting Over the Internet", Indrajit Ray, Indrakshi Ray, Natarajan Narasimhamurthi

The Problem

- We want a protocol for voting over the internet that has all the salient features of voting in person
- These properties can be grouped under the categories **Accuracy**, **Democracy**, **Privacy**, and **No Unauthorized Proxy**

Desirable Properties

- Accuracy
  - A cast vote cannot be altered
  - An invalid vote is not counted
  - Each voter can verify that his/her vote is counted
- Democracy
  - Only an eligible voter can participate
  - Each voter can cast only one vote

Desirable Properties (II)

- Privacy
  - A ballot cannot be linked back to the voter who cast it
  - (No vote buying) A voter cannot prove to someone else what his/her vote is
- No Unauthorized Proxy
  - If a voter decides not to cast his/her ballot, no party can take advantage of this and cast a forged ballot

Desirable Properties (III)

- In principle, if some property of the election is compromised, some authority should be able to detect and prove it.
- At worst, some consortium of people should be able to prove it without compromising their own privacy
  - One breach of this occurs in a protocol violation; I may have a hard time proving to someone that a server is ignoring me without giving up some privacy.

Assumptions

- Any two parties can arrange a secure communication channel
- Additionally, a voter can send secure, anonymous messages (votes) to a server
- Certain systems that do not interact with voters in the voting process are secure
  - The voter registry that knows the names of all registered voters is secure

The Protocol

1. Ballot distribution: BD → V: $\{y,sig_{BD}\{h(y)\}\}_V$, $sig_{BD}\{h(\text{voter certificate})\}$
   - y is the ballot number
   - h is a one-way permutation
2. Generating a voter mark: m=h(y)
3. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V,\text{voter certificate}, sig_{BD}\{h(\text{voter certificate})\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
4. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
   d. Public FTP site → V: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
5. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

The Revised Protocol (I)

1. Ballot distribution: BD → V: $\{y,sig_{BD}\{h(y)\}\}_V$, $sig_{BD}\{h(\text{voter certificate})\}$
   - y is the ballot number
   - h is a one-way permutation
2. Generating a voter mark: m=h(y)
3. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V,\text{voter certificate}, sig_{BD}\{h(\text{voter certificate})\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
4. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
   d. Public FTP site → V: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
5. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

To the extent that we can verify a y-m pair, we can identify people's votes. This should not happen.

Clarification: Suppose that given y, an authority could construct m. This would violate privacy. An alternative interpretation is that m is produced (from y) by some method known only to the voter. Here the voter could be expected to demonstrate that he produced m. We propose instead that the voter picks a random y, then uses h to generate m. He can prove that he generated m by exhibiting y (no one else can invert the permutation). This construction has the desired property that no one knows the origin of m until the voter chooses to reveal it.

We note that from the perspective of a protocol, the above outlined procedure appears as if the voter just choose a random m, as in the next slide.

The Revised Protocol (II)

1. Ballot distribution: BD → V: $\{y,sig_{BD}\{h(y)\}\}_V$, $sig_{BD}\{h(\text{voter certificate})\}$
   - y is the ballot number
   - h is a one-way permutation
2. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V,\text{voter certificate}, sig_{BD}\{h(\text{voter certificate})\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
3. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
   d. Public FTP site → V: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

Let m be random. y is now useless

The Revised Protocol (III)

1. Ballot distribution: BD → V:$sig_{BD}\{h(\text{voter certificate})\}$
   - h is a one-way permutation
2. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V, \text{voter certificate}, sig_{BD}\{h(\text{voter certificate})\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
3. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
   d. Public FTP site → V: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

The voter certificate identifies the voter as eligible for the election. It is signed by the Registration authority.

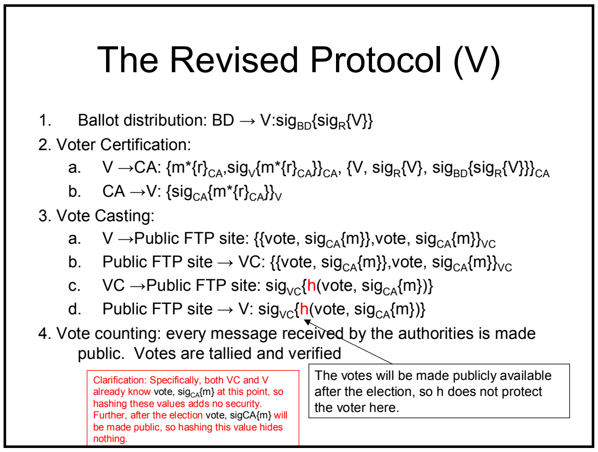
Note: Here we make a best effort to clarify something that is vaguely specified in the paper.

The Revised Protocol (IV)

1. Ballot distribution: BD → V:$sig_{BD}\{h(sig_R\{V\})\}$
   - h is a one-way permutation
2. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V, sig_R\{V\}_R, sig_{BD}\{h(sig_R\{V\})\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
3. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\},h(vote, sig_{CA}\{m\})\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
   d. Public FTP site → V: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

The function h(x) does not add any protection if the parties already know x

Clarification: We mean specifically that for any run of the protocol with the above marked functions h, there is a corresponding run of the protocol without it, because all the involved parties can both apply h, or "invert" it when they already know the inverted value.

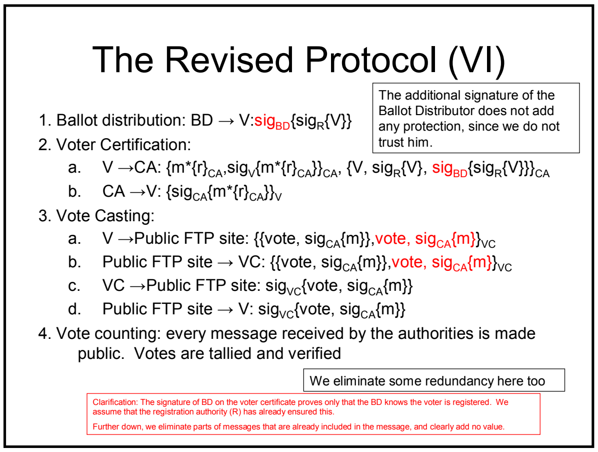


The Revised Protocol (V)

1. Ballot distribution: BD → V:$sig_{BD}\{sig_R\{V\}\}$
2. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V, sig_R\{V\}, sig_{BD}\{sig_R\{V\}\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
3. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\},vote, sig_{CA}\{m\}\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\},vote, sig_{CA}\{m\}\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
   d. Public FTP site → V: $sig_{VC}\{h(vote, sig_{CA}\{m\})\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

Clarification: Specifically, both VC and V already know vote, $sig_{CA}\{m\}$ at this point, so hashing these values adds no security. Further, after the election vote, sigCA{m} will be made public, so hashing this value hides nothing.

The votes will be made publicly available after the election, so h does not protect the voter here.



The Revised Protocol (VI)

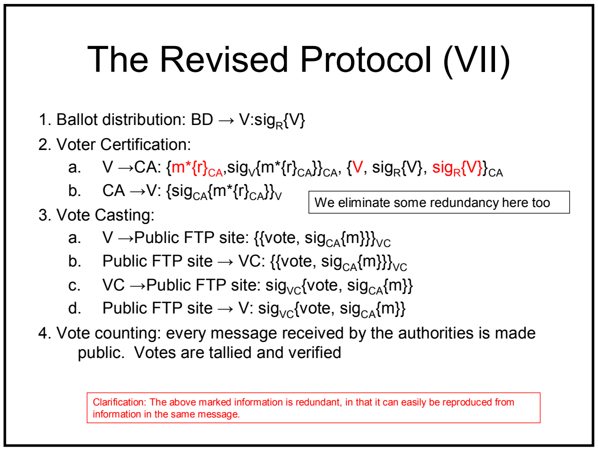
The additional signature of the Ballot Distributor does not add any protection, since we do not trust him.

1. Ballot distribution: BD → V:$sig_{BD}\{sig_R\{V\}\}$
2. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V, sig_R\{V\}, sig_{BD}\{sig_R\{V\}\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
3. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\},vote, sig_{CA}\{m\}\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\},vote, sig_{CA}\{m\}\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{vote, sig_{CA}\{m\}\}$
   d. Public FTP site → V: $sig_{VC}\{vote, sig_{CA}\{m\}\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

We eliminate some redundancy here too

Clarification: The signature of BD on the voter certificate proves only that the BD knows the voter is registered. We assume that the registration authority (R) has already ensured this.

Further down, we eliminate parts of messages that are already included in the message, and clearly add no value.

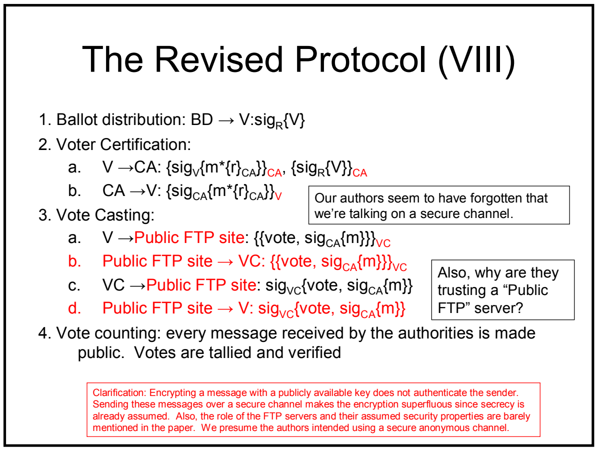


The Revised Protocol (VII)

1. Ballot distribution: BD → V:$sig_R\{V\}$
2. Voter Certification:
   a. V →CA: $\{m^*\{r\}_{CA},sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{V, sig_R\{V\}, sig_R\{V\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
3. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\}\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\}\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{vote, sig_{CA}\{m\}\}$
   d. Public FTP site → V: $sig_{VC}\{vote, sig_{CA}\{m\}\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

We eliminate some redundancy here too

Clarification: The above marked information is redundant, in that it can easily be reproduced from information in the same message.



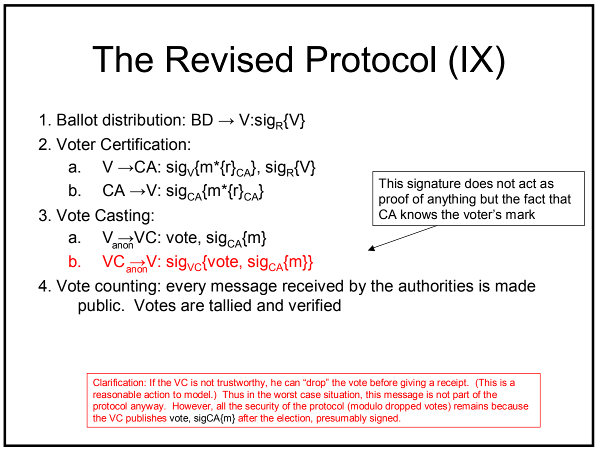
The Revised Protocol (VIII)

1. Ballot distribution: BD → V:$sig_R\{V\}$
2. Voter Certification:
   a. V →CA: $\{sig_V\{m^*\{r\}_{CA}\}\}_{CA}$, $\{sig_R\{V\}\}_{CA}$
   b. CA →V: $\{sig_{CA}\{m^*\{r\}_{CA}\}\}_V$
3. Vote Casting:
   a. V →Public FTP site: $\{\{vote, sig_{CA}\{m\}\}\}_{VC}$
   b. Public FTP site → VC: $\{\{vote, sig_{CA}\{m\}\}\}_{VC}$
   c. VC →Public FTP site: $sig_{VC}\{vote, sig_{CA}\{m\}\}$
   d. Public FTP site → V: $sig_{VC}\{vote, sig_{CA}\{m\}\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

Our authors seem to have forgotten that we're talking on a secure channel.

Also, why are they trusting a "Public FTP" server?

Clarification: Encrypting a message with a publicly available key does not authenticate the sender. Sending these messages over a secure channel makes the encryption superfluous since secrecy is already assumed. Also, the role of the FTP servers and their assumed security properties are barely mentioned in the paper. We presume the authors intended using a secure anonymous channel.

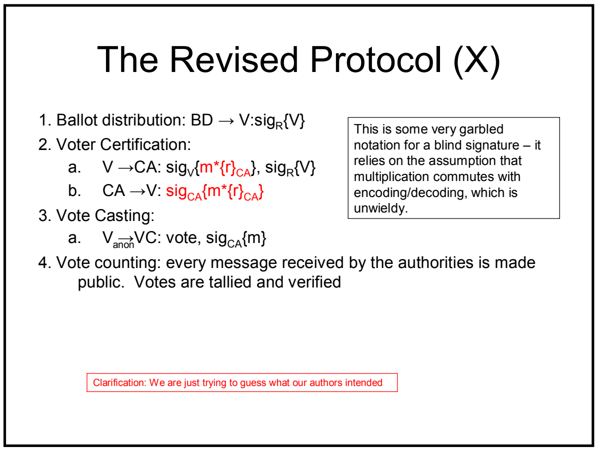


The Revised Protocol (IX)

1. Ballot distribution: BD → V:$sig_R\{V\}$
2. Voter Certification:
   a. V →CA: $sig_V\{m^*\{r\}_{CA}\}$, $sig_R\{V\}$
   b. CA →V: $sig_{CA}\{m^*\{r\}_{CA}\}$
3. Vote Casting:
   a. $V \xrightarrow[anon]{} VC$: vote, $sig_{CA}\{m\}$
   b. $VC \xrightarrow[anon]{} V$: $sig_{VC}\{vote, sig_{CA}\{m\}\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

This signature does not act as proof of anything but the fact that CA knows the voter's mark

Clarification: If the VC is not trustworthy, he can "drop" the vote before giving a receipt. (This is a reasonable action to model.) Thus in the worst case situation, this message is not part of the protocol anyway. However, all the security of the protocol (modulo dropped votes) remains because the VC publishes vote, sigCA{m} after the election, presumably signed.

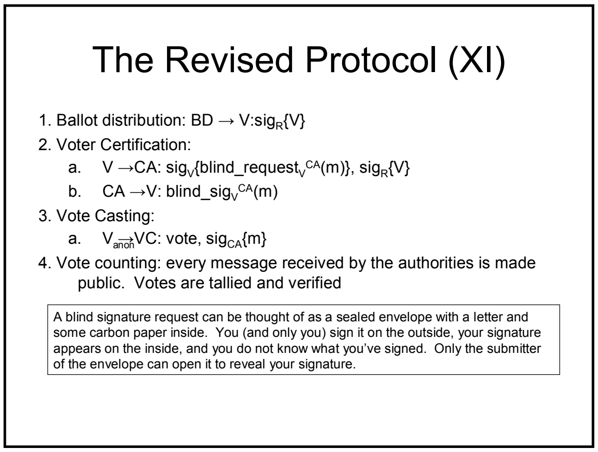


The Revised Protocol (X)

1. Ballot distribution: BD → V:$sig_R\{V\}$
2. Voter Certification:
   a. V →CA: $sig_V\{m^*\{r\}_{CA}\}$, $sig_R\{V\}$
   b. CA →V: $sig_{CA}\{m^*\{r\}_{CA}\}$
3. Vote Casting:
   a. $V \xrightarrow[anon]{} VC$: vote, $sig_{CA}\{m\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

This is some very garbled notation for a blind signature – it relies on the assumption that multiplication commutes with encoding/decoding, which is unwieldy.

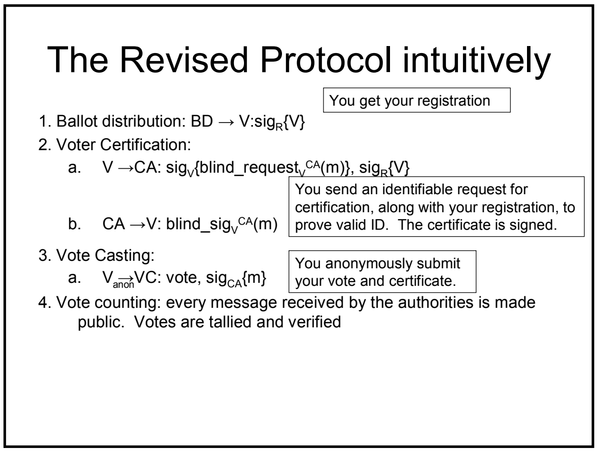
Clarification: We are just trying to guess what our authors intended



The Revised Protocol (XI)

1. Ballot distribution: BD → V:$\mathrm{sig}_R\{V\}$
2. Voter Certification:
   a. V →CA: $\mathrm{sig}_V\{\mathrm{blind\_request}_V^{CA}(m)\}$, $\mathrm{sig}_R\{V\}$
   b. CA →V: $\mathrm{blind\_sig}_V^{CA}(m)$
3. Vote Casting:
   a. $V \xrightarrow[anon]{} VC$: vote, $\mathrm{sig}_{CA}\{m\}$
4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified

A blind signature request can be thought of as a sealed envelope with a letter and some carbon paper inside. You (and only you) sign it on the outside, your signature appears on the inside, and you do not know what you've signed. Only the submitter of the envelope can open it to reveal your signature.



The Revised Protocol intuitively

1. Ballot distribution: BD → V:$\mathrm{sig}_R\{V\}$

You get your registration

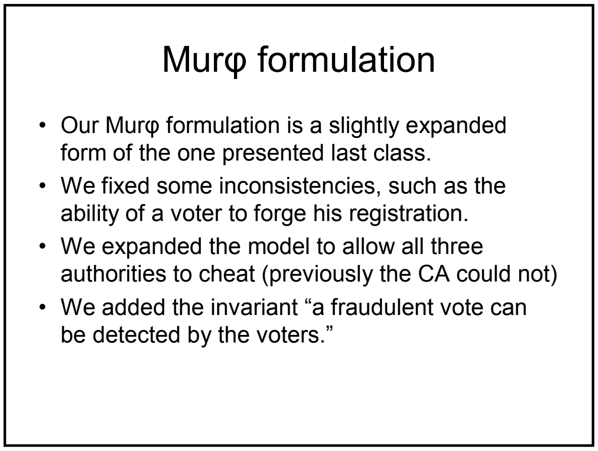
2. Voter Certification:
   a. V →CA: $\mathrm{sig}_V\{\mathrm{blind\_request}_V^{CA}(m)\}$, $\mathrm{sig}_R\{V\}$
   b. CA →V: $\mathrm{blind\_sig}_V^{CA}(m)$

You send an identifiable request for certification, along with your registration, to prove valid ID. The certificate is signed.

3. Vote Casting:
   a. $V \xrightarrow[anon]{} VC$: vote, $\mathrm{sig}_{CA}\{m\}$

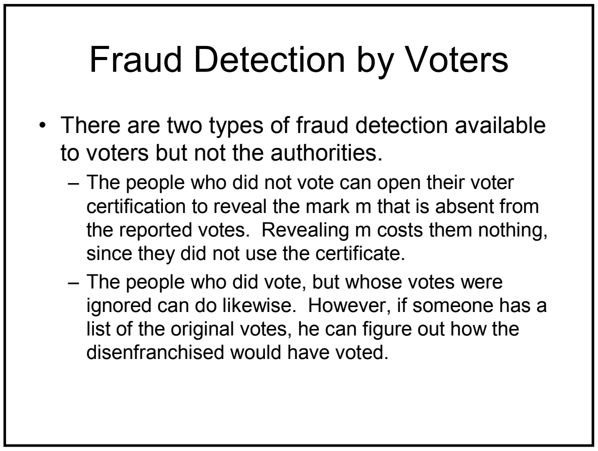
You anonymously submit your vote and certificate.

4. Vote counting: every message received by the authorities is made public. Votes are tallied and verified



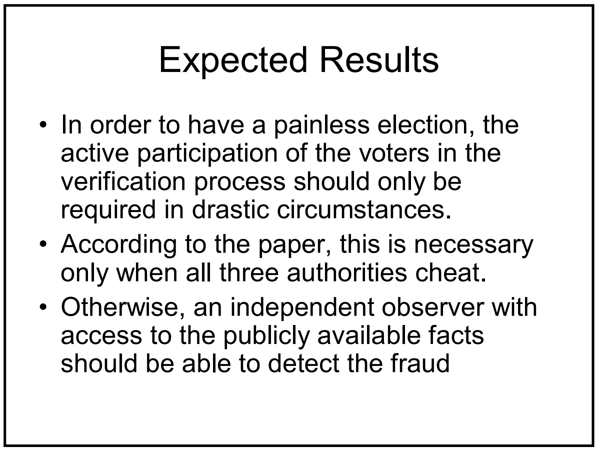
Murφ formulation

- Our Murφ formulation is a slightly expanded form of the one presented last class.
- We fixed some inconsistencies, such as the ability of a voter to forge his registration.
- We expanded the model to allow all three authorities to cheat (previously the CA could not)
- We added the invariant "a fraudulent vote can be detected by the voters."



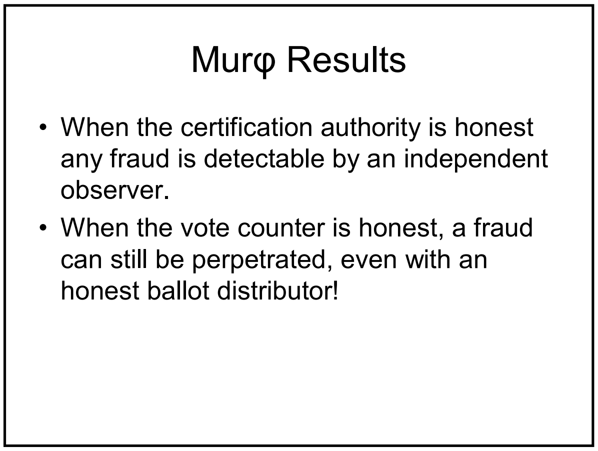
Fraud Detection by Voters

- There are two types of fraud detection available to voters but not the authorities.
  – The people who did not vote can open their voter certification to reveal the mark m that is absent from the reported votes. Revealing m costs them nothing, since they did not use the certificate.
  – The people who did vote, but whose votes were ignored can do likewise. However, if someone has a list of the original votes, he can figure out how the disenfranchised would have voted.



Expected Results

- In order to have a painless election, the active participation of the voters in the verification process should only be required in drastic circumstances.
- According to the paper, this is necessary only when all three authorities cheat.
- Otherwise, an independent observer with access to the publicly available facts should be able to detect the fraud



Murφ Results

- When the certification authority is honest any fraud is detectable by an independent observer.
- When the vote counter is honest, a fraud can still be perpetrated, even with an honest ballot distributor!

Fraud!

- With the cooperation of the CA, a dishonest voter obtains a signed mark, which he then votes with.
- Meanwhile, an unsuspecting voter completes the protocol up until the registration stage, but does not vote. The CA publishes his/her submitted registration info, pretending that the fraudulent voter is associated with it.

With Voter Verification

- However, Murφ confirms that the voters can still detect fraud in these cases.

Thoughts

- This whole analysis rests on the assumption that the agents follow a reasonable version of the protocol.
- Instead, what if the vote counter changed everyone’s vote to “Bush”?
- Would a receipt help? No, because the vote counter could forge whatever receipt he wants, and still change your vote.

Thoughts (II)

- What about selling your vote? Can you prove a link between your voter mark and what you submitted to the CA?
- Yes! Because you are the (only) one who can open the blinded signature form you sent to the CA.