Voting
Lecture 22
Requirements
Requirements

- Integrity/End-to-End verifiability
Requirements

- Integrity/End-to-End verifiability
  - Collected as cast: Each voter should be convinced that their vote was collected correctly
Requirements

- Integrity/End-to-End verifiability
  - Collected as cast: Each voter should be convinced that their vote was collected correctly
  - Counted as collected: Tallying is publicly verifiable
Requirements

- Integrity/End-to-End verifiability
  - Collected as cast: Each voter should be convinced that their vote was collected correctly
  - Counted as collected: Tallying is publicly verifiable

- Secrecy
Requirements

- **Integrity/End-to-End verifiability**
  - Collected as cast: Each voter should be convinced that their vote was collected correctly
  - Counted as collected: Tallying is publicly verifiable

- **Secrecy**
  - Honest voters’ votes are not revealed by the system (beyond what the tally reveals)
Requirements

- **Integrity/End-to-End verifiability**
  - Collected as cast: Each voter should be convinced that their vote was collected correctly
  - Counted as collected: Tallying is publicly verifiable

- **Secrecy**
  - Honest voters’ votes are not revealed by the system (beyond what the tally reveals)
  - Incoercibility: Even corrupt voters should not be able to convince an adversary about their vote (i.e., no vote-buying/selling)
A Voting Architecture
A Voting Architecture

Produce a public list which encodes all the votes cast
A Voting Architecture

- Produce a public list which encodes all the votes cast
- Individual voters can verify that their vote is correctly captured in this list
A Voting Architecture

- Produce a public list which encodes all the votes cast

- Individual voters can verify that their vote is correctly captured in this list

- Based on a receipt (and other knowledge) from the polling booth
A Voting Architecture

- Produce a public list which encodes all the votes cast
  - Individual voters can verify that their vote is correctly captured in this list
  - Based on a receipt (and other knowledge) from the polling booth
- Tallying is done on this list
A Voting Architecture

- Produce a public list which encodes all the votes cast
  - Individual voters can verify that their vote is correctly captured in this list
    - Based on a receipt (and other knowledge) from the polling booth
  - Tallying is done on this list
    - Publicly verifiable that the posted votes are correctly tabulated
A Voting Architecture

- Produce a public list which encodes all the votes cast

  - Individual voters can verify that their vote is correctly captured in this list

  - Based on a receipt (and other knowledge) from the polling booth

- Tallying is done on this list

  - Publicly verifiable that the posted votes are correctly tabulated
A Voting Architecture

- Produce a public list which encodes all the votes cast
  - Individual voters can verify that their vote is correctly captured in this list
    - Based on a receipt (and other knowledge) from the polling booth
  - Tallying is done on this list
    - Publicly verifiable that the posted votes are correctly tabulated

Front-End
- Ballot Preparation
A Voting Architecture

- Produce a public list which encodes all the votes cast
  - Individual voters can verify that their vote is correctly captured in this list
    - Based on a receipt (and other knowledge) from the polling booth
  - Tallying is done on this list
    - Publicly verifiable that the posted votes are correctly tabulated
- Front-End
  - Ballot Preparation
  - Vote capturing/Receipt issue
A Voting Architecture

- Produce a public list which encodes all the votes cast
  - Individual voters can verify that their vote is correctly captured in this list
    - Based on a receipt (and other knowledge) from the polling booth
  - Tallying is done on this list
- Publicly verifiable that the posted votes are correctly tabulated

Front-End
- Ballot Preparation
- Vote capturing/Receipt issue
- Verification
A Voting Architecture

- Produce a public list which encodes all the votes cast
  - Individual voters can verify that their vote is correctly captured in this list
  - Based on a receipt (and other knowledge) from the polling booth
- Tallying is done on this list
  - Publicly verifiable that the posted votes are correctly tabulated

Front-End

- Ballot Preparation
- Vote capturing/Receipt issue
- Verification

Back-End
A Voting Architecture

- Produce a public list which encodes all the votes cast
  - Individual voters can verify that their vote is correctly captured in this list
    - Based on a receipt (and other knowledge) from the polling booth
  - Tallying is done on this list
    - Publicly verifiable that the posted votes are correctly tabulated

Front-End
- Ballot Preparation
- Vote capturing/Receipt issue
- Verification

Back-End
- Tallying/Verification
Use MPC?
Use MPC?

- Impractical
Use MPC?

- Impractical

- In the front-end, want voters not to have to do crypto, and arrive/leave one by one
Use MPC?

- Impractical
  - In the front-end, want voters not to have to do crypto, and arrive/leave one by one
  - OK in the back-end, but needs to be very efficient if a large election
Use MPC?

- Impractical
  - In the front-end, want voters not to have to do crypto, and arrive/leave one by one
  - OK in the back-end, but needs to be very efficient if a large election
  - Doesn't account for incoercibility (unless security requirement augmented)
Incoercibility
Incoercibility

Coercion: voters can get rewards from adversary by following adversary’s instructions in a detectable fashion
Incoercibility

Coercion: voters can get rewards from adversary by following adversary’s instructions in a detectable fashion

What is not coercion?
Incoercibility

Coercion: voters can get rewards from adversary by following adversary's instructions in a detectable fashion

What is not coercion?

   e.g. Adversary rewards the entire set of voters if all votes are for candidate A
Incoercibility

- **Coercion**: voters can get rewards from adversary by following adversary's instructions in a detectable fashion

- **What is not coercion?**
  - e.g. Adversary rewards the entire set of voters if all votes are for candidate A
  - Is coercion: Voters cannot behave arbitrarily and still collect the reward
Incoercibility

- Coercion: voters can get rewards from adversary by following adversary’s instructions in a detectable fashion.

- What is not coercion?

  - e.g. Adversary rewards the entire set of voters if all votes are for candidate A.

  - Is coercion: Voters cannot behave arbitrarily and still collect the reward.

  - But unavoidable coercion (even in an Ideal world).
Incoercibility

Coercion: voters can get rewards from adversary by following adversary’s instructions in a detectable fashion

What is not coercion?

- e.g. Adversary rewards the entire set of voters if all votes are for candidate A
  - Is coercion: Voters cannot behave arbitrarily and still collect the reward
  - But unavoidable coercion (even in an Ideal world)

We need to protect against further coercion than is possible in the Ideal world
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:

∀ and
∃ and s.t.
∀
IDEAL/c ≈ REAL/c
and
IDEAL/u ≈ REAL/u
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:

\[ \forall \text{ and } \exists \text{ and s.t. } \]

\[ \text{IDEAL/c} \approx \text{REAL/c} \]
and
\[ \text{IDEAL/u} \approx \text{REAL/u} \]

Hence \text{REAL/c} and \text{REAL/u} only as distinguishable as \text{IDEAL/c} and \text{IDEAL/u}
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:

\[ \forall \text{ and } \exists \text{ and } s.t. \]

\[ \forall \text{ IDEAL/c } \approx \text{ REAL/c } \text{ and } \text{ IDEAL/u } \approx \text{ REAL/u } \]

Hence REAL/c and REAL/u only as distinguishable as IDEAL/c and IDEAL/u i.e., if coercion can be (somewhat) simulated in Ideal, it can be (somewhat) simulated in Real too.
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:

\[ \forall \text{ and } \exists \text{ and } \text{s.t.} \]

\[ \forall \text{ IDEAL/c } \approx \text{ REAL/c} \text{ and } \text{IDEAL/u } \approx \text{ REAL/u} \]

Hence REAL/c and REAL/u only as distinguishable as IDEAL/c and IDEAL/u i.e., if coercion can be (somewhat) simulated in Ideal, it can be (somewhat) simulated in Real too

Definition says nothing about the existence/choice of the Ideal coercion simulator.
Defining Incoercibility

Real as incoercible (and secure) as Ideal if:

\[ \forall \text{ and } \exists \text{ and s.t. } \forall \]

\[ \text{IDEAL/c} \approx \text{REAL/c} \text{ and } \text{IDEAL/u} \approx \text{REAL/u} \]

Hence REAL/c and REAL/u only as distinguishable as IDEAL/c and IDEAL/u i.e., if coercion can be (somewhat) simulated in Ideal, it can be (somewhat) simulated in Real too

Definition says nothing about the existence/choice of the Ideal coercion simulator

Meaningful only if Real/u simulator is realistic
e-Voting: First Try
e-Voting: First Try

Front-end:
e-Voting: First Try

Front-end:

Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext.
e-Voting: First Try

Front-end:

- Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext

- The encrypted vote is publicly posted
e-Voting: First Try

Front-end:

- Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext
- The encrypted vote is publicly posted

Back-end:
e-Voting: First Try

Front-end:
- Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext
- The encrypted vote is publicly posted

Back-end:
- A mix-net shuffles, decrypts the set of votes. Publicly tallied
e-Voting: First Try

Front-end:
- Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext
- The encrypted vote is publicly posted

Back-end:
- A mix-net shuffles, decrypts the set of votes. Publicly tallied
- Each candidate/observer can have a mix-net server
e-Voting: First Try

Front-end:
- Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext
- The encrypted vote is publicly posted

Back-end:
- A mix-net shuffles, decrypts the set of votes. Publicly tallied
  - Each candidate/observer can have a mix-net server
  - Public proofs given to each other (or to the public at large, using Fiat-Shamir heuristics)
e-Voting: First Try

Front-end:
- Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext
- The encrypted vote is publicly posted

Back-end:
- A mix-net shuffles, decrypts the set of votes. Publicly tallied
  - Each candidate/observer can have a mix-net server
  - Public proofs given to each other (or to the public at large, using Fiat-Shamir heuristics)

Requires voters to use/trust computational devices
e-Voting: First Try

Front-end:
- Voters encrypt their votes using a threshold encryption scheme (with the decryption key shared among authorities/candidates), and submit the vote; receives a receipt showing the ciphertext
- The encrypted vote is publicly posted

Back-end:
- A mix-net shuffles, decrypts the set of votes. Publicly tallied
  - Each candidate/observer can have a mix-net server
  - Public proofs given to each other (or to the public at large, using Fiat-Shamir heuristics)

Provide encryption devices that have been “verified” by the public?
(Perception of) threats: difficulty in verifying devices, substituting devices...
Challenge
Challenge

Keep it simple for the voter
Challenge

- Keep it simple for the voter
- No crypto to ensure vote collected as cast
Challenge

- Keep it simple for the voter
- No crypto to ensure vote collected as cast
- Public list will contain information that proves to the voter that the vote collected is as cast
Challenge

- Keep it simple for the voter
  - No crypto to ensure vote collected as cast
- Public list will contain information that proves to the voter that the vote collected is as cast
- Should not allow voter to prove to a vote-buyer how the vote was cast
Challenge

- Keep it simple for the voter
  - No crypto to ensure vote collected as cast
  - Public list will contain information that proves to the voter that the vote collected is as cast
  - Should not allow voter to prove to a vote-buyer how the vote was cast
  - e.g., not OK to let the voter submit (multiple rerandomized) ciphertexts and get them decrypted later
Prêt à Voter
Prêt à Voter

Ballot has two parts
Prêt à Voter

Ballot has two parts

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Carol</td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>Barack</td>
<td>X</td>
</tr>
</tbody>
</table>

ahdf87
Prêt à Voter

Ballot has two parts

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Carol</td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>Barack</td>
<td>X</td>
</tr>
</tbody>
</table>

ahdf87
Ballot has two parts

Left-hand side: Candidate list
Prêt à Voter

Ballot has two parts

Left-hand side: Candidate list

Right-hand side: Vote-mark and encrypted candidate list (and a serial number)
Prêt à Voter

- Ballot has two parts
  - Left-hand side: Candidate list
  - Right-hand side: Vote-mark and encrypted candidate list (and a serial number)

- Right-hand part has enough information for tallying. Will be posted publicly. Also serves as receipt.
Prêt à Voter

- Ballot has two parts
  - Left-hand side: Candidate list
  - Right-hand side: Vote-mark and encrypted candidate list (and a serial number)

- Right-hand part has enough information for tallying. Will be posted publicly. Also serves as receipt.

- Auditing assures that w.h.p the two parts are consistent
Prêt à Voter

Ballot has two parts

- Left-hand side: Candidate list
- Right-hand side: Vote-mark and encrypted candidate list (and a serial number)

Right-hand part has enough information for tallying. Will be posted publicly. Also serves as receipt.

Auditing assures that w.h.p the two parts are consistent

Voter retains a copy of the right-hand part (with a digital signature, possibly verified by helpers outside the booth, to prevent false claims) as a receipt to verify the publicly posted vote. Left-hand part must be destroyed before leaving the polling-booth.
Prêt à Voter

<table>
<thead>
<tr>
<th></th>
<th>Carol</th>
<th>Alice</th>
<th>Barack</th>
<th>ahdf87</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Prêt à Voter

Tallying: combine vote-mark and encrypted candidate list into an encrypted vote
Prêt à Voter

Tallying: combine vote-mark and encrypted candidate list into an encrypted vote

Candidate list is cyclically permuted by s positions
Prêt à Voter

- Tallying: combine vote-mark and encrypted candidate list into an encrypted vote
- Candidate list is cyclically permuted by $s$ positions
- Encryption encodes $s$
Prêt à Voter

- Tallying: combine vote-mark and encrypted candidate list into an encrypted vote
  - Candidate list is cyclically permuted by $s$ positions
  - Encryption encodes $s$
  - Homomorphically add vote-mark position to encryption of $s$, to get encryption of candidate's index

<table>
<thead>
<tr>
<th></th>
<th>Carol</th>
<th>Alice</th>
<th>Barack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ahdf87</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prêt à Voter

Tallying: combine vote-mark and encrypted candidate list into an encrypted vote

- Candidate list is cyclically permuted by $s$ positions
- Encryption encodes $s$
- Homomorphically add vote-mark position to encryption of $s$, to get encryption of candidate’s index

Additive homomorphism: Use Paillier, or El Gamal with messages in the exponent (since only a few messages possible)
Prêt à Voter

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Carol</td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>Barack</td>
<td>X</td>
</tr>
</tbody>
</table>

ahdf87
Prêt à Voter

Counted as collected: ensured by the mix-net

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Carol</td>
<td>Carol</td>
</tr>
<tr>
<td>Alice</td>
<td>Alice</td>
</tr>
<tr>
<td>Barack</td>
<td>Barack</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>ahdf87</td>
</tr>
</tbody>
</table>
Prêt à Voter

Counted as collected: ensured by the mix-net

To ensure collected as cast, need to ensure that the ballot papers are correctly formed
Prêt à Voter

- Counted as collected: ensured by the mix-net
- To ensure collected as cast, need to ensure that the ballot papers are correctly formed
- Auditing: before voting, select a random subset of ballots and have them decrypted
Prêt à Voter

- Counted as collected: ensured by the mix-net

- To ensure collected as cast, need to ensure that the ballot papers are correctly formed

- Auditing: before voting, select a random subset of ballots and have them decrypted

- If no errors found in a large random sample (say half the ballots) probability of more than a few bad ballots is very small ($\approx 2^{-t}$ probability that more than $t$ bad)
Prêt à Voter

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Carol</td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>Barack</td>
<td>ahdfl87</td>
</tr>
</tbody>
</table>
Prêt à Voter

For secrecy, need to ensure LHS of ballot-paper remains secret (till voting) and encryption in the RHS is honest (i.e., randomly generated)
Prêt à Voter

- For secrecy, need to ensure LHS of ballot-paper remains secret (till voting) and encryption in the RHS is honest (i.e., randomly generated)

- A trusted/audited ballot-sheet printer with an encryption key pair
Prêt à Voter

For secrecy, need to ensure LHS of ballot-paper remains secret (till voting) and encryption in the RHS is honest (i.e., randomly generated)

- A trusted/audited ballot-sheet printer with an encryption key pair
- Use MPC (among candidates/trustees) to encrypt a random rotation twice: one ciphertext using printer’s PK (in the left-hand side) and one using the mix-net’s PK
For secrecy, need to ensure LHS of ballot-paper remains secret (till voting) and encryption in the RHS is honest (i.e., randomly generated)

A trusted/audited ballot-sheet printer with an encryption key pair

Use MPC (among candidates/trustees) to encrypt a random rotation twice: one ciphertext using printer’s PK (in the left-hand side) and one using the mix-net’s PK

At the polling-booth the printer decrypts the left-hand ciphertext, and prints the candidate names in order

Prêt à Voter

Carol
Alice
Barack

x5qu0d  ahdf87
Prêt à Voter

For secrecy, need to ensure LHS of ballot-paper remains secret (till voting) and encryption in the RHS is honest (i.e., randomly generated)

A trusted/audited ballot-sheet printer with an encryption key pair

Use MPC (among candidates/trustees) to encrypt a random rotation twice: one ciphertext using printer’s PK (in the left-hand side) and one using the mix-net’s PK

At the polling-booth the printer decrypts the left-hand ciphertext, and prints the candidate names in order

<table>
<thead>
<tr>
<th>Carol</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>Barack</td>
<td></td>
</tr>
<tr>
<td>x5qu0d</td>
<td>ahd87</td>
</tr>
</tbody>
</table>
Prêt à Voter

For secrecy, need to ensure LHS of ballot-paper remains secret (till voting) and encryption in the RHS is honest (i.e., randomly generated)

A trusted/audited ballot-sheet printer with an encryption key pair

Use MPC (among candidates/trustees) to encrypt a random rotation twice: one ciphertext using printer’s PK (in the left-hand side) and one using the mix-net’s PK

At the polling-booth the printer decrypts the left-hand ciphertext, and prints the candidate names in order

Can be audited by the voter: choose one of (say) two ballot sheets for auditing later; printer’s key kept shared among auditors who can audit sheets selected by the voters
Threats/Remedies
Threats/Remedies

Chain voting: One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet.
Threats/Remedies

Chain voting: One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet.

Officials should ensure ballot-sheet turned in is the same as ballot-sheet given.
Threats/Remedies

- **Chain voting:** One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet.
  - Officials should ensure ballot-sheet turned in is the same as ballot-sheet given.
- **Randomization attack:** Coercer can ask voters to mark the first candidate, thereby ensuring they vote randomly.
Threats/Remedies

**Chain voting:** One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet.

Officials should ensure ballot-sheet turned in is the same as ballot-sheet given.

**Randomization attack:** Coercer can ask voters to mark the first candidate, thereby ensuring they vote randomly.

Comparable to coercing to not cast a vote (allowed in Ideal).
Threats/Remedies

Chain voting: One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet

Officials should ensure ballot-sheet turned in is the same as ballot-sheet given

Randomization attack: Coercer can ask voters to mark the first candidate, thereby ensuring they vote randomly

Comparable to coercing to not cast a vote (allowed in Ideal)

Discarded receipt attack: If corrupt election authority learns that a receipt was discarded, can safely change the collected vote
Threats/Remedies

**Chain voting:** One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet

- Officials should ensure ballot-sheet turned in is the same as ballot-sheet given

**Randomization attack:** Coercer can ask voters to mark the first candidate, thereby ensuring they vote randomly

- Comparable to coercing to not cast a vote (allowed in Ideal)

**Discarded receipt attack:** If corrupt election authority learns that a receipt was discarded, can safely change the collected vote

**Retained left-hand part:** can be used to sell votes
Threats/Remedies

Chain voting: One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet

Officials should ensure ballot-sheet turned in is the same as ballot-sheet given

Randomization attack: Coercer can ask voters to mark the first candidate, thereby ensuring they vote randomly

Comparable to coercing to not cast a vote (allowed in Ideal)

Discarded receipt attack: If corrupt election authority learns that a receipt was discarded, can safely change the collected vote

Retained left-hand part: can be used to sell votes

Ensure it is destroyed. Also make decoys available
Threats/Remedies

Chain voting: One ballot-sheet smuggled out and marked. Then repeatedly coerce voters to use the marked ballot-sheet and return with a blank ballot-sheet.

Officials should ensure ballot-sheet turned in is the same as ballot-sheet given.

Randomization attack: Coercer can ask voters to mark the first candidate, thereby ensuring they vote randomly.

Comparable to coercing to not cast a vote (allowed in Ideal).

Discarded receipt attack: If corrupt election authority learns that a receipt was discarded, can safely change the collected vote.

Retained left-hand part: can be used to sell votes.

Ensure it is destroyed. Also make decoys available.

Printer's key known: Attack if also (LHS, RHS) pairing known.
Some Other Schemes
Some Other Schemes

Several schemes
Some Other Schemes

- Several schemes
- Few security definitions/proofs
Some Other Schemes

- Several schemes
- Few security definitions/proofs
- Punchscan
Some Other Schemes

- Several schemes
- Few security definitions/proofs
- Punchscan
- Two-layer ballot-sheet
Some Other Schemes

- Several schemes
- Few security definitions/proofs
- Punchscan
- Two-layer ballot-sheet
Some Other Schemes

- Several schemes
- Few security definitions/proofs
- Punchscan
- Two-layer ballot-sheet
- Scratch-and-Vote
Some Other Schemes

- Several schemes
- Few security definitions/proofs
- Punchscan
- Two-layer ballot-sheet
- Scratch-and-Vote
Some Other Schemes

- Several schemes
- Few security definitions/proofs
- Punchscan
- Two-layer ballot-sheet
- Scratch-and-Vote
- Punchscan variant
Some Other Schemes

- Several schemes
- Few security definitions/proofs
- Punchscan
- Two-layer ballot-sheet
- Scratch-and-Vote
- Punchscan variant
- To audit a ballot-sheet, scratch off and obtain randomness used in encryption
Back-Ends
Back-Ends

Efficient (and publicly verifiable) MPC for tallying encrypted votes
Back-Ends

Efficient (and publicly verifiable) MPC for tallying encrypted votes

Using mix-nets: Shuffle, decrypt and tally
Back-Ends

- Efficient (and publicly verifiable) MPC for tallying encrypted votes
- Using mix-nets: Shuffle, decrypt and tally
- Using homomorphic counters: Tally and decrypt
Back-Ends

Efficient (and publicly verifiable) MPC for tallying encrypted votes

Using mix-nets: Shuffle, decrypt and tally

Using homomorphic counters: Tally and decrypt

A single counter that is the concatenation of counters for each candidate
Back-Ends

- Efficient (and publicly verifiable) MPC for tallying encrypted votes
- Using mix-nets: Shuffle, decrypt and tally
- Using homomorphic counters: Tally and decrypt
  - A single counter that is the concatenation of counters for each candidate
  - To add to a counter for a candidate, must add after appropriately shifting
Back-Ends

- Efficient (and publicly verifiable) MPC for tallying encrypted votes
- Using mix-nets: Shuffle, decrypt and tally
- Using homomorphic counters: Tally and decrypt
  - A single counter that is the concatenation of counters for each candidate
  - To add to a counter for a candidate, must add after appropriately shifting
  - In Prêt à Voter, information on RHS: encryptions of the shifted value to be added for each possible mark
Other Issues
Other Issues

- Dispute resolution (without compromising voter’s privacy)
Other Issues

- Dispute resolution (without compromising voter's privacy)
- Subliminal channels from polling booth to the adversary that facilitate coercion
Other Issues

- Dispute resolution (without compromising voter’s privacy)
- Subliminal channels from polling booth to the adversary that facilitate coercion
  - Coerced voters could be asked to bring along a “verifier” (implemented as scratch cards etc.) to which they should “prove” that they are voting as promised
Other Issues

- Dispute resolution (without compromising voter’s privacy)
- Subliminal channels from polling booth to the adversary that facilitate coercion
  - Coerced voters could be asked to bring along a “verifier” (implemented as scratch cards etc.) to which they should “prove” that they are voting as promised
- Aggravated by allowing voters to audit at the polling-booth
Other Issues

- Dispute resolution (without compromising voter’s privacy)
- Subliminal channels from polling booth to the adversary that facilitate coercion
- Coerced voters could be asked to bring along a “verifier” (implemented as scratch cards etc.) to which they should “prove” that they are voting as promised
- Aggravated by allowing voters to audit at the polling-booth
- Internet voting?
Other Issues

- Dispute resolution (without compromising voter's privacy)
- Subliminal channels from polling booth to the adversary that facilitate coercion
  - Coerced voters could be asked to bring along a "verifier" (implemented as scratch cards etc.) to which they should "prove" that they are voting as promised
- Aggravated by allowing voters to audit at the polling-booth

Internet voting?

- Coercion is hard to prevent, but can be mitigated by allowing voters to change votes any time
Voting Schemes
Voting Schemes

“Standard” (a.k.a plurality rule or First Past the Pole): each voter has a single vote and candidate with most votes win.
Voting Schemes

“Standard” (a.k.a plurality rule or First Past the Pole): each voter has a single vote and candidate with most votes wins.

Approval voting: a voter can vote for arbitrary number of candidates; candidate with most votes wins.
Voting Schemes

“Standard” (a.k.a plurality rule or First Past the Pole): each voter has a single vote and candidate with most votes win

Approval voting: a voter can vote for arbitrary number of candidates; candidate with most votes win

Condorcet voting: voters provide a full-ranking; defines a “tournament” between candidates, so that A beats B if A appears above B in more rankings than vice versa. If the tournament has a champion who beats everyone else, that candidate wins. Several special rules for handling cycles.
Voting Schemes

“Standard” (a.k.a plurality rule or First Past the Pole): each voter has a single vote and candidate with most votes win

Approval voting: a voter can vote for arbitrary number of candidates; candidate with most votes win

Condorcet voting: voters provide a full-ranking; defines a “tournament” between candidates, so that A beats B if A appears above B in more rankings than vice versa. If the tournament has a champion who beats everyone else, that candidate wins. Several special rules for handling cycles.

Multiple round tallying: Supplementary vote, Instant Run-off elections, Single Transferable Vote
Voting Schemes

“Standard” (a.k.a. plurality rule or First Past the Pole): each voter has a single vote and candidate with most votes win.

Approval voting: a voter can vote for arbitrary number of candidates; candidate with most votes win.

Condorcet voting: voters provide a full-ranking; defines a “tournament” between candidates, so that A beats B if A appears above B in more rankings than vice versa. If the tournament has a champion who beats everyone else, that candidate wins. Several special rules for handling cycles.

Multiple round tallying: Supplementary vote, Instant Run-off elections, Single Transferable Vote.

Front-end and back-end need to be modified.
Summary
Summary

Several proposals for electronic voting
Summary

- Several proposals for electronic voting
- Crypto tools based on homomorphic encryption
Summary

Several proposals for electronic voting

Crypto tools based on homomorphic encryption

Aims to get unprecedented level of confidence from individual voters and public auditors (E2E security)
Summary

- Several proposals for electronic voting
  - Crypto tools based on homomorphic encryption
  - Aims to get unprecedented level of confidence from individual voters and public auditors (E2E security)
  - Challenge: Increases risk of coercion
Summary

Several proposals for electronic voting

Crypto tools based on homomorphic encryption

Aims to get unprecedented level of confidence from individual voters and public auditors (E2E security)

Challenge: Increases risk of coercion

A cyber-physical system with avenue for new protocol techniques and attacks
Summary

- Several proposals for electronic voting
  - Crypto tools based on homomorphic encryption
  - Aims to get unprecedented level of confidence from individual voters and public auditors (E2E security)
  - Challenge: Increases risk of coercion
- A cyber-physical system with avenue for new protocol techniques and attacks
- Few satisfactory security definitions yet (let alone proofs)