

Reversing, Breaking, and Fixing the French Legislative Election E-Voting Protocol

Alexandre Debant and [Lucca Hirschi](#), Inria Nancy

28 mars 2023 @ Real World Crypto

Largest E-voting Election

Largest election?

Largest election?



+1.5MM eligible voters (French citizens resident overseas)



+500k ballots were cast over internet (largest number ever using e-voting)

Largest election?



+1.5MM eligible voters (French citizens resident overseas)



+500k ballots were cast over internet (largest number ever using e-voting)

This election was based on a **new protocol**, **better be sure it is secure!**

(FLEP)

Largest election?



+1.5MM eligible voters (French citizens resident overseas)



+500k ballots were cast over internet (largest number ever using e-voting)

This election was based on a **new protocol**, **better be sure it is secure!**
(FLEP)

Two central **security goals** for e-voting:



Ballot Privacy: an attacker cannot learn the choice of a voter

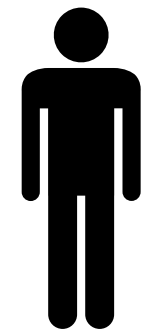
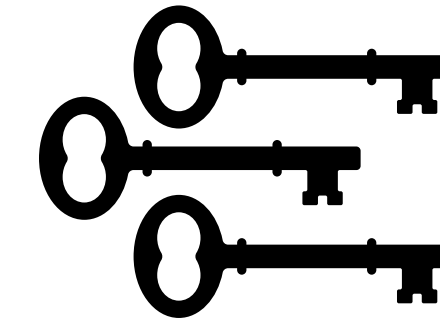


Verifiability: voters must have the guarantee that their ballots are counted

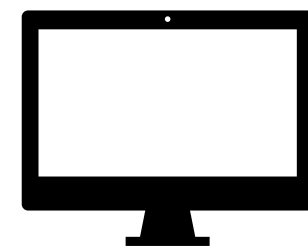
The E-voting Protocol: FLEP

The protocol **roles**

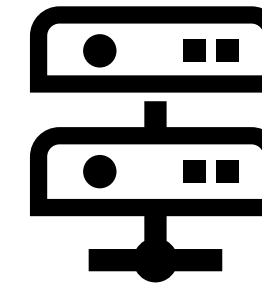
Decryption Trustees



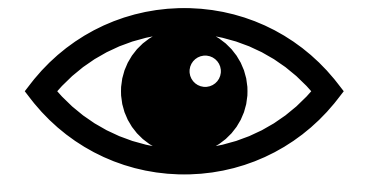
Voter



Voting Client



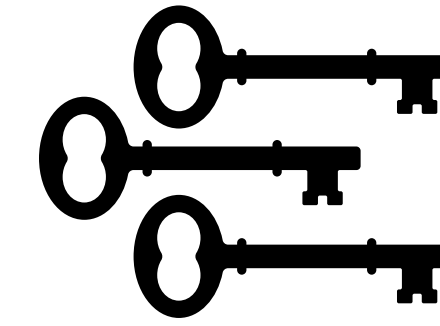
Voting Server



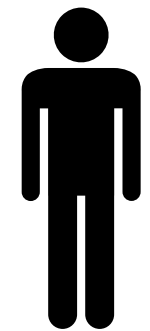
Third-Party

The protocol **roles**

Decryption Trustees

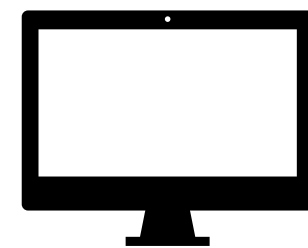


by representatives and officials



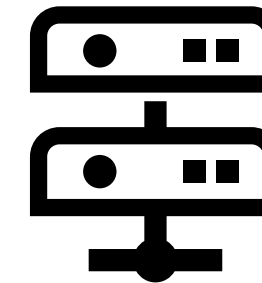
Voter

At home



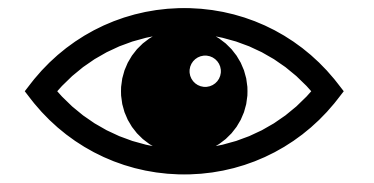
Voting Client

Javascript running in a browser



Voting Server

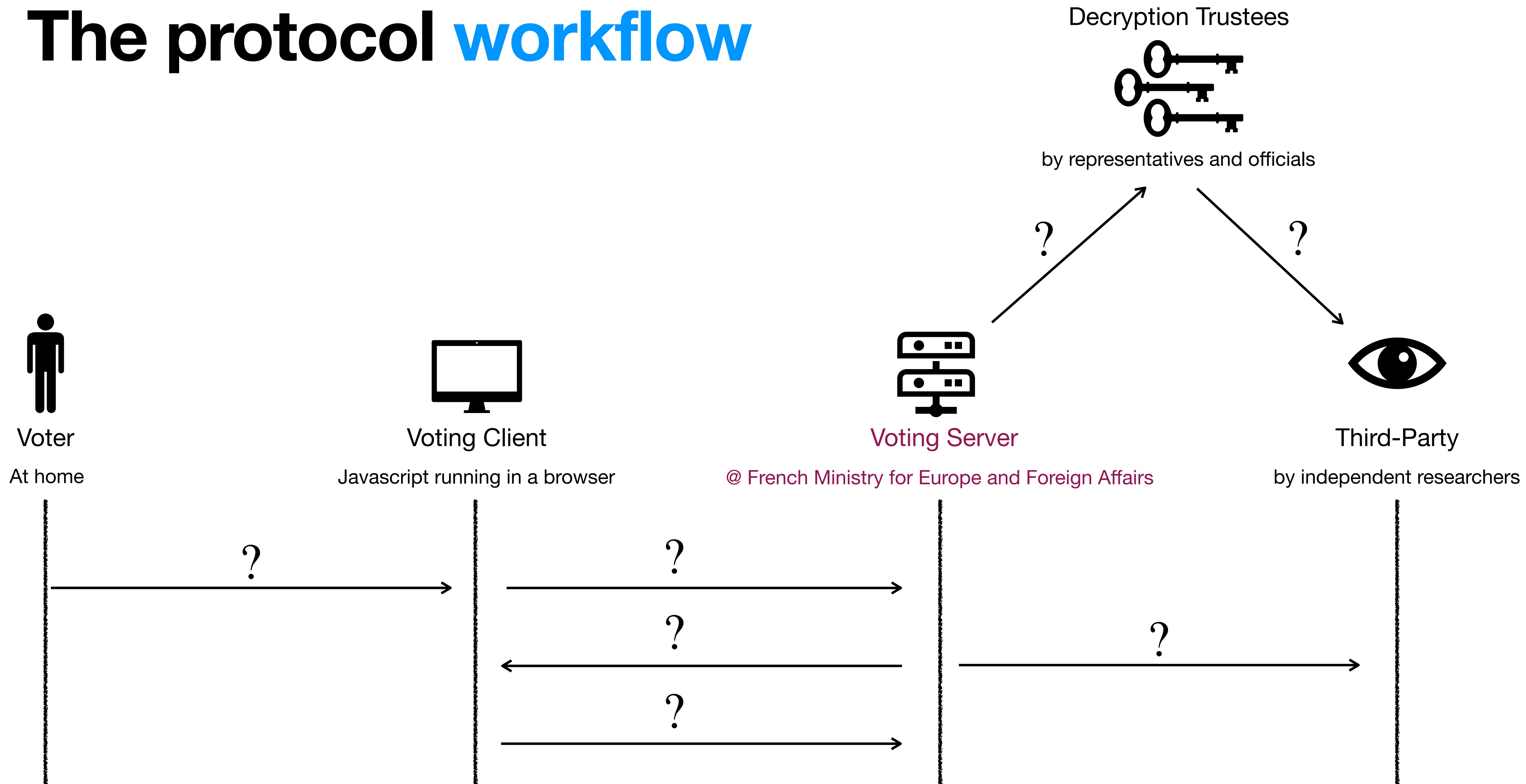
@ French Ministry for Europe and Foreign Affairs



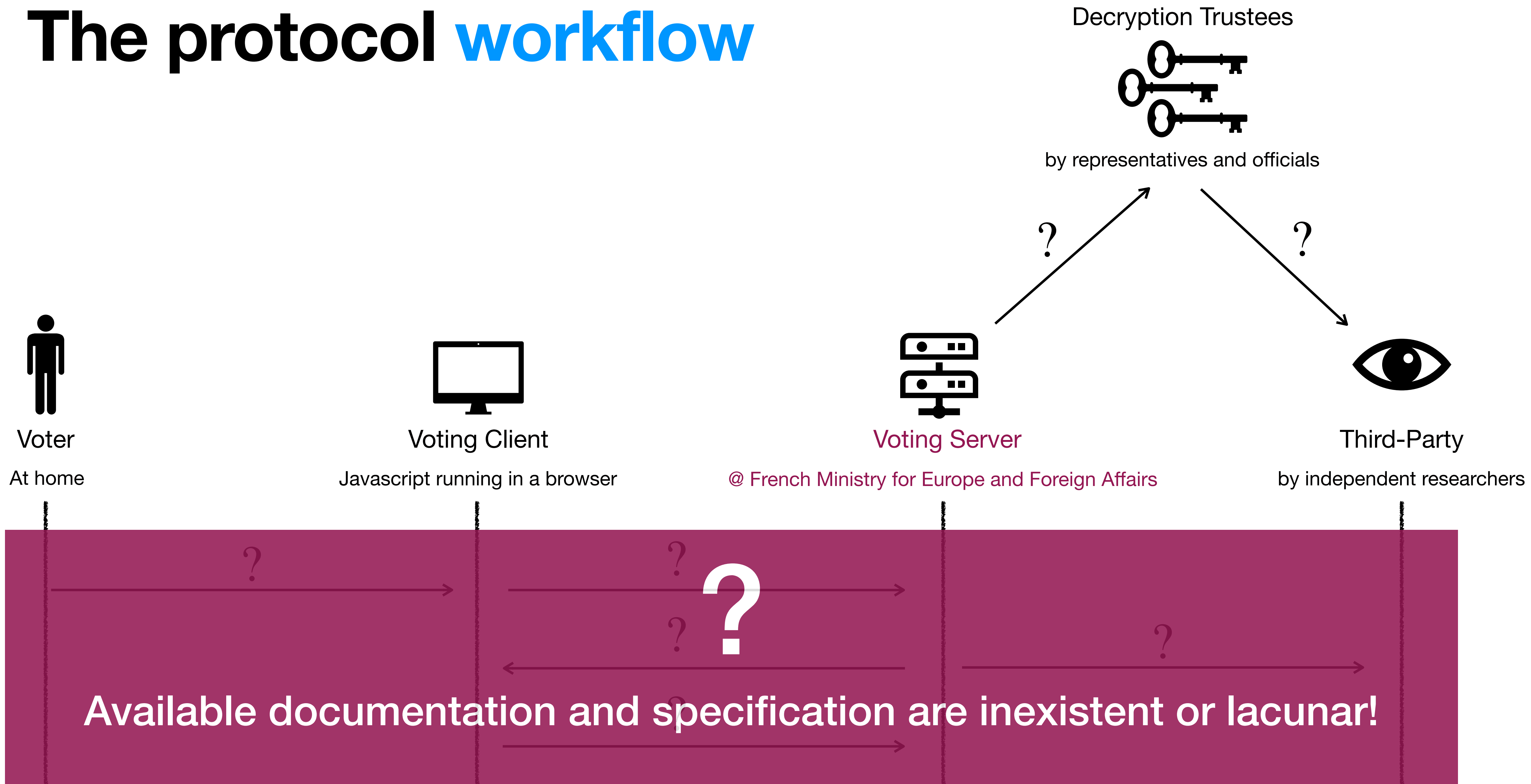
Third-Party

by independent researchers

The protocol workflow

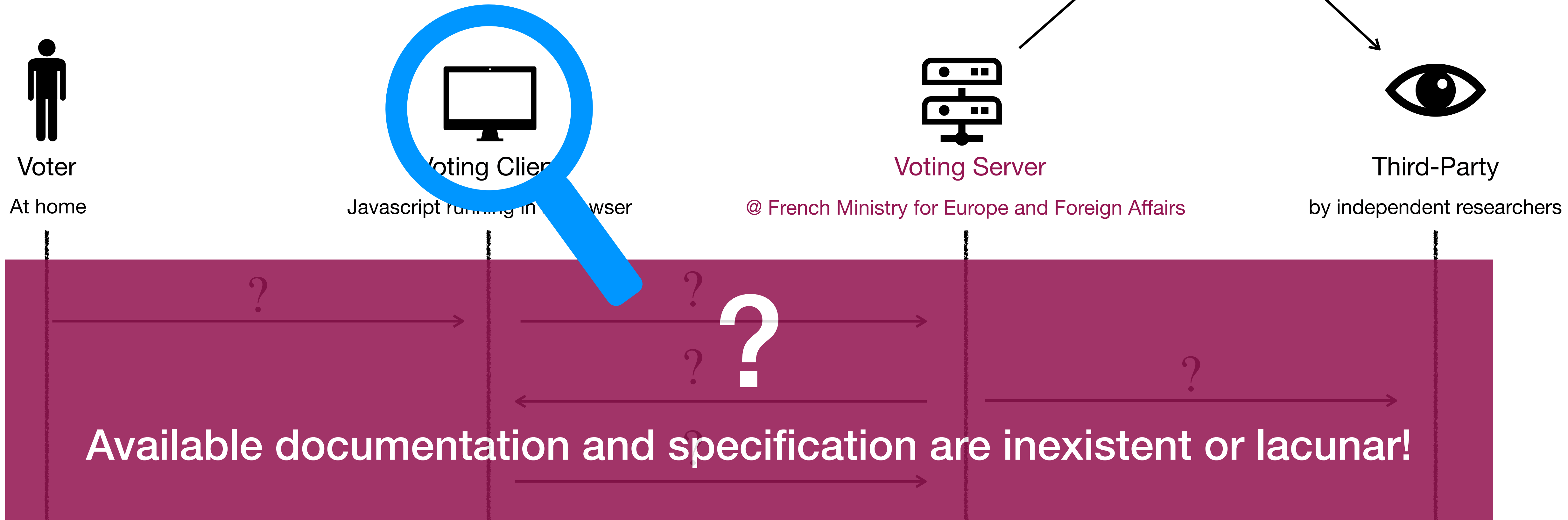


The protocol workflow



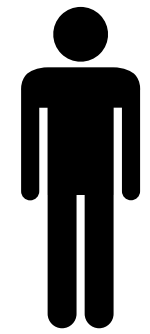
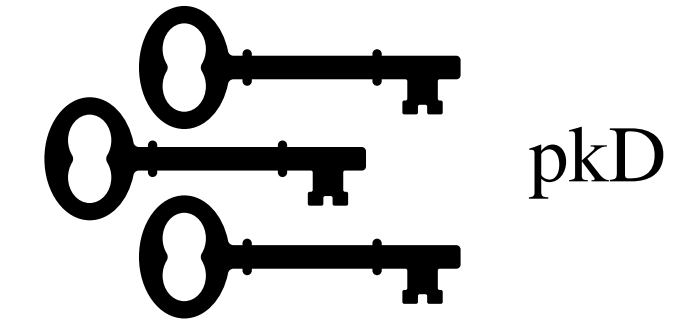
The protocol workflow

Reverse the obfuscated voting client
(Javascript & HTML)

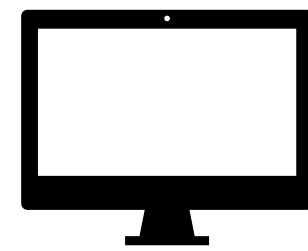


The protocol **flow** (simplified)

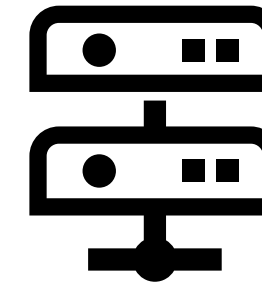
Decryption Trustees



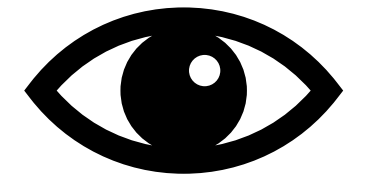
Voter



Voting Client



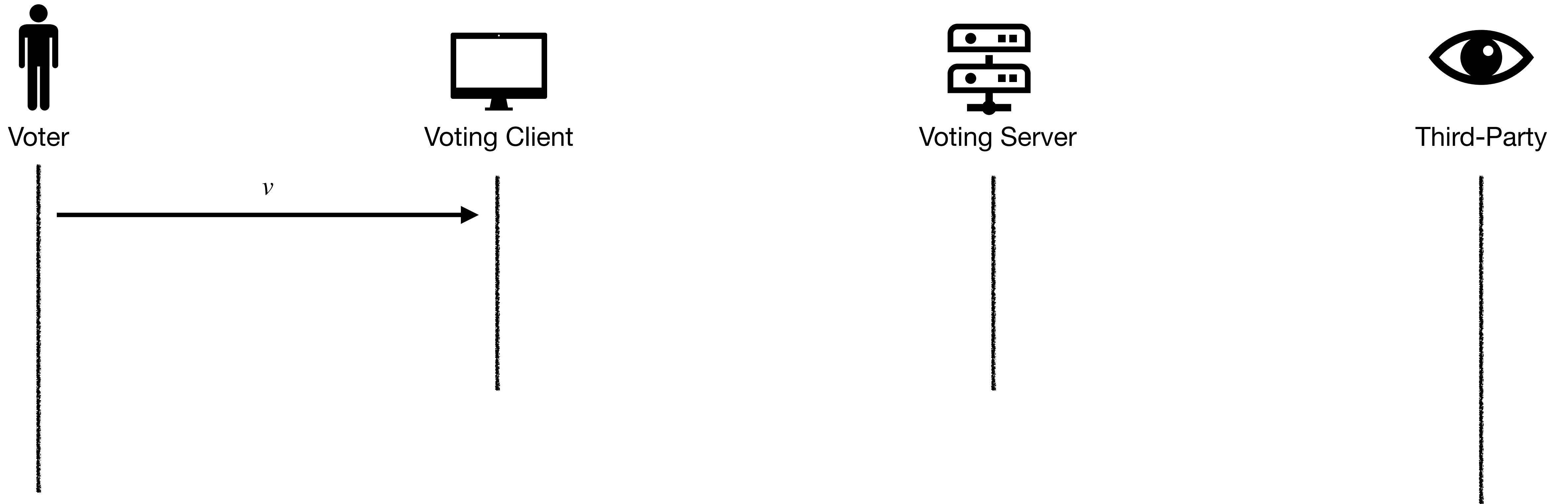
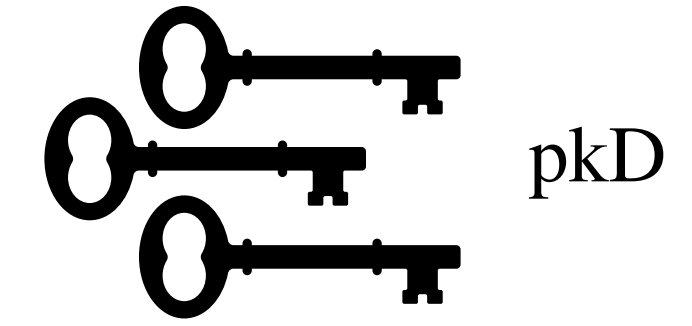
Voting Server



Third-Party

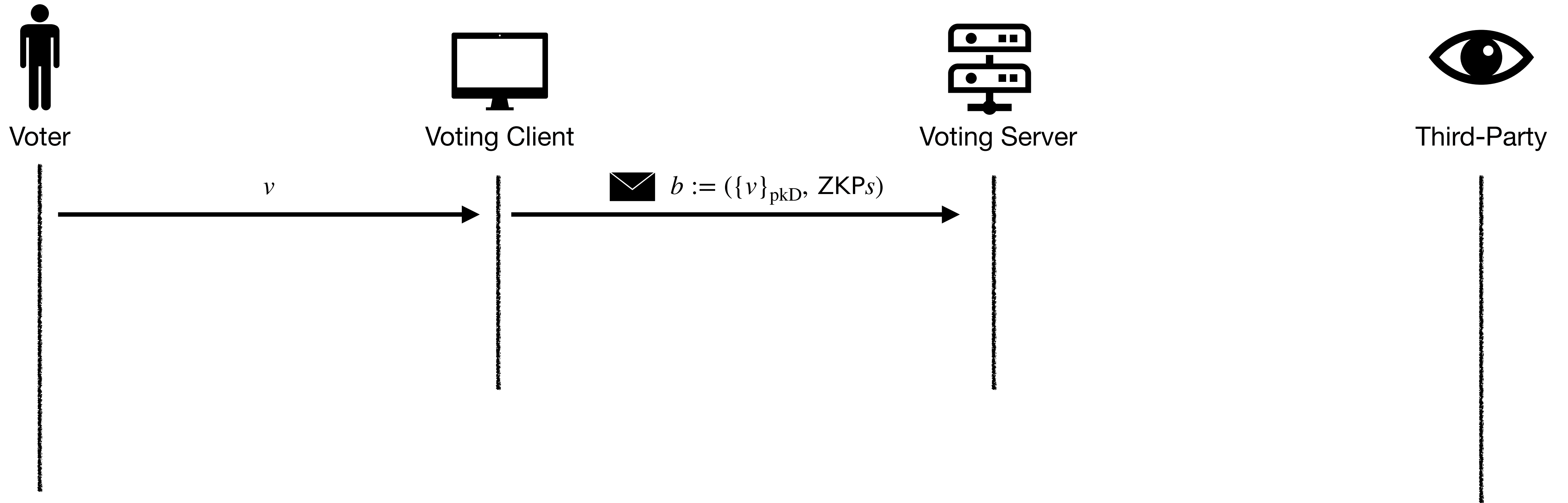
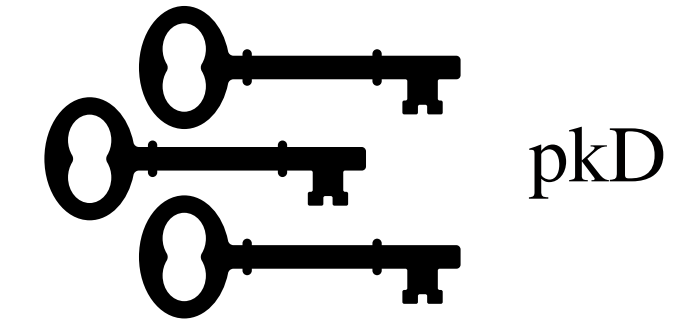
The protocol **flow** (simplified)

Decryption Trustees



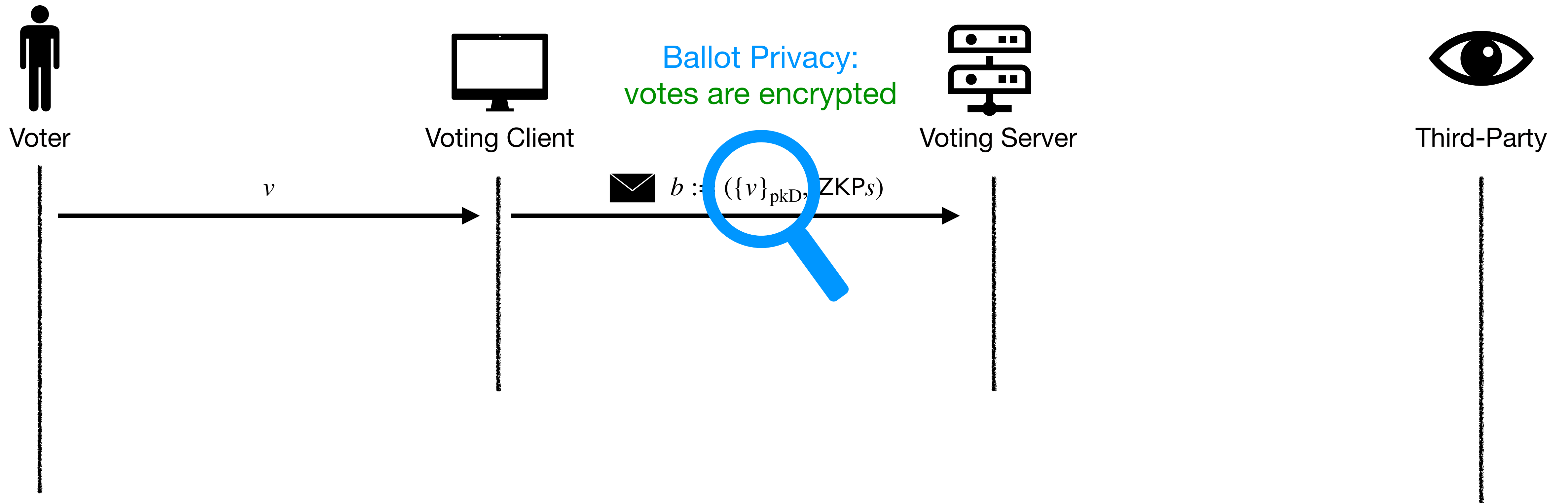
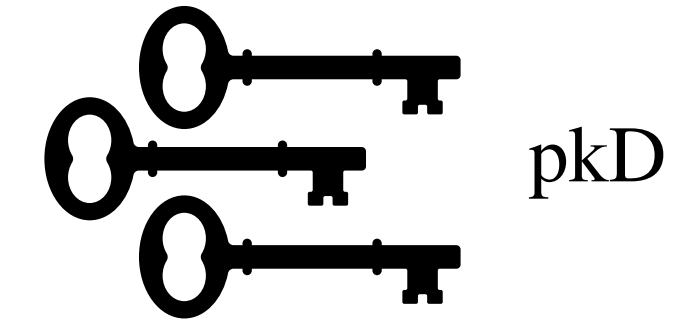
The protocol **flow** (simplified)

Decryption Trustees



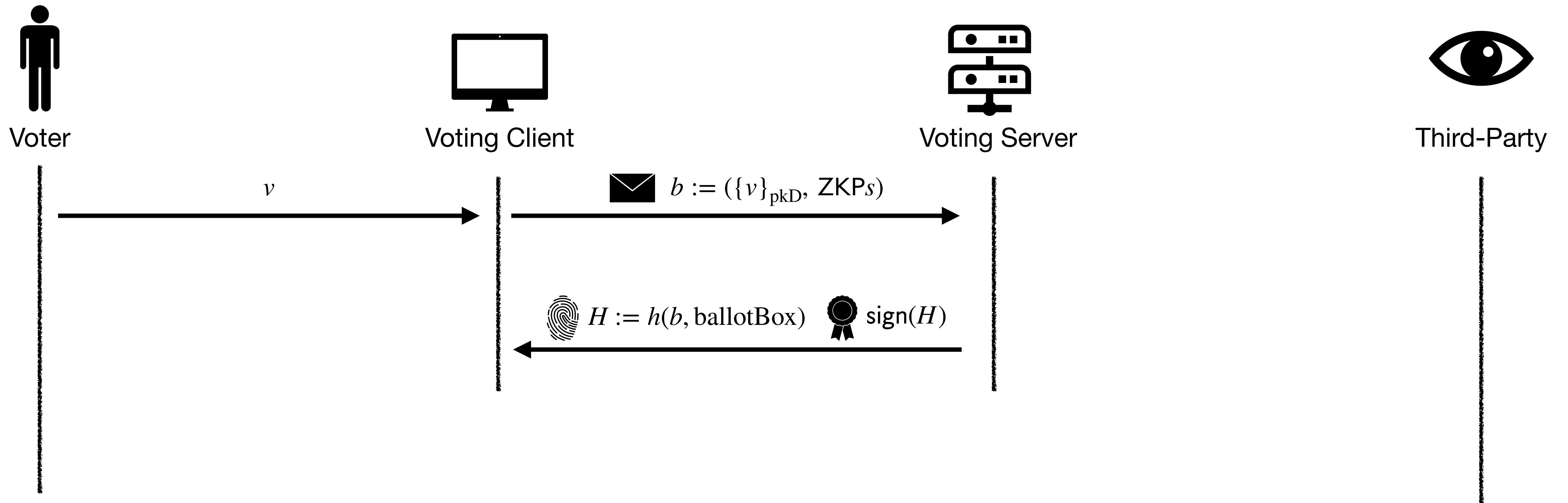
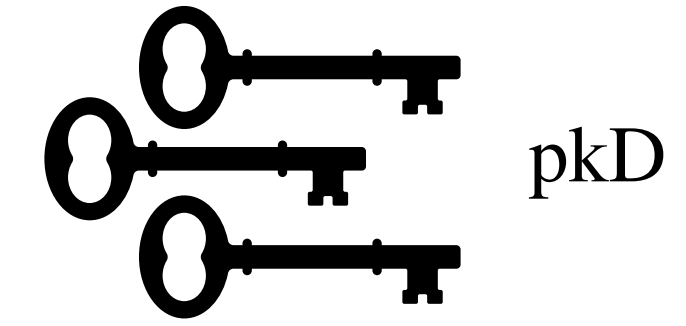
The protocol **flow** (simplified)

Decryption Trustees



The protocol **flow** (simplified)

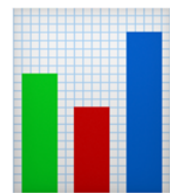
Decryption Trustees



The protocol **flow** (simplified)

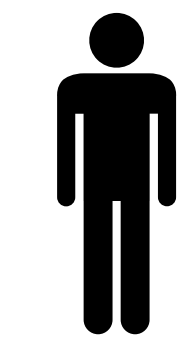
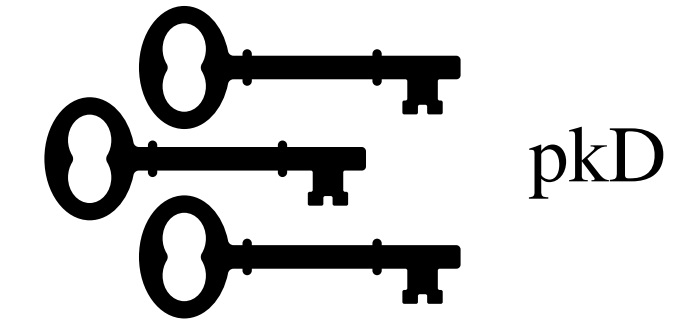


ballotBox for each consular (~city)

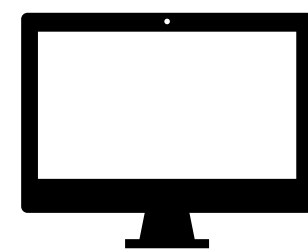


result per ballotBox

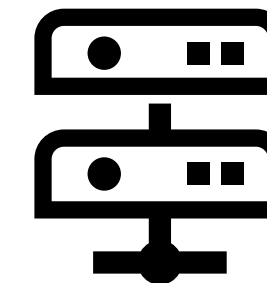
Decryption Trustees



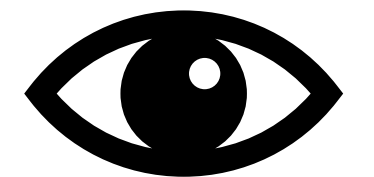
Voter



Voting Client



Voting Server



Third-Party

v

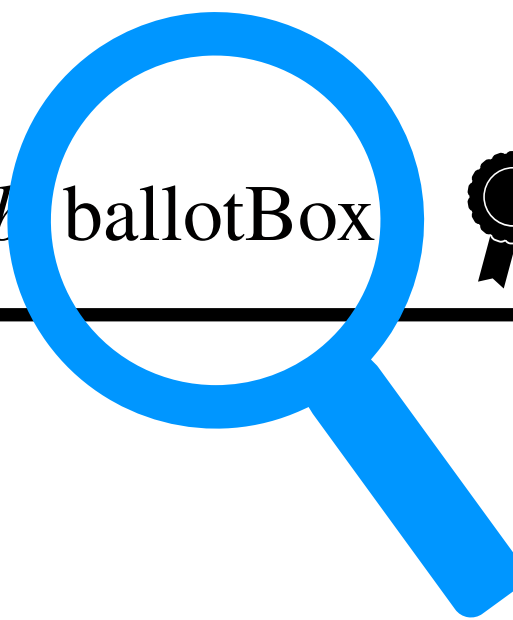
$b := (\{v\}_{pkD}, ZKPs)$



$H := h(\text{ballotBox})$



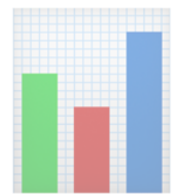
$\text{sign}(H)$



The protocol **flow** (simplified)

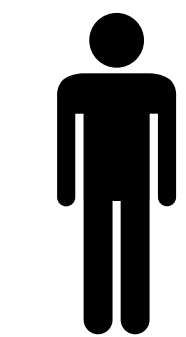
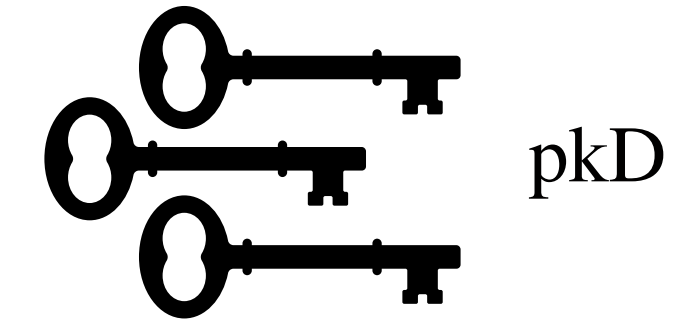


ballotBox for each consular (~city)

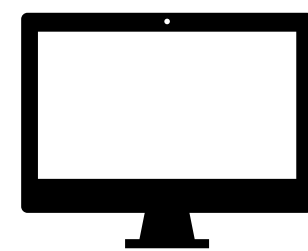


result per ballotBox

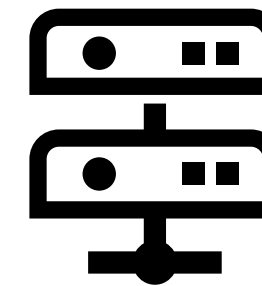
Decryption Trustees



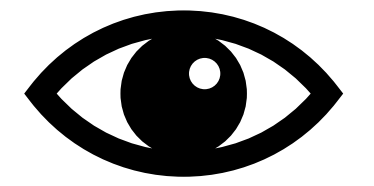
Voter



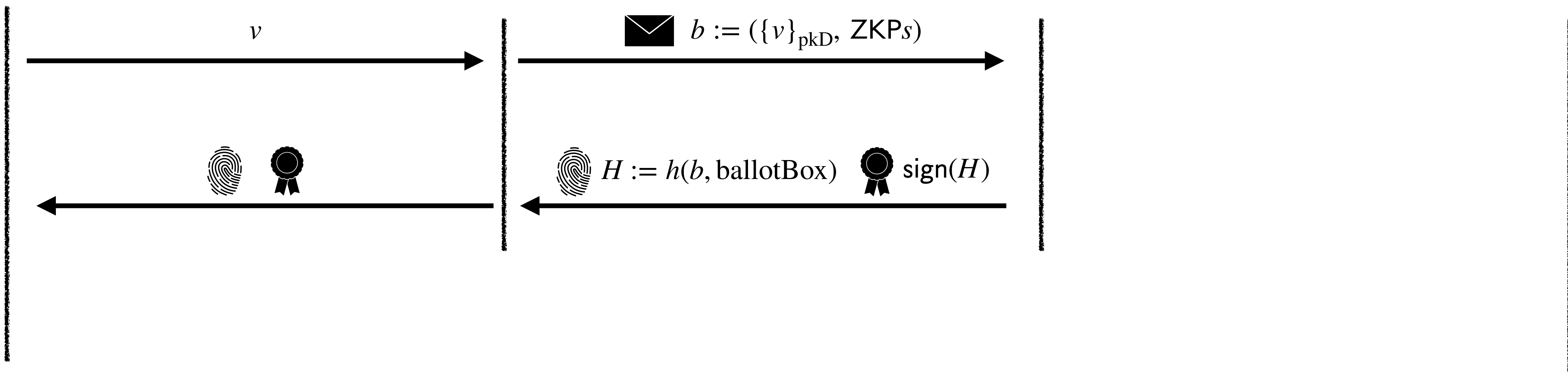
Voting Client



Voting Server



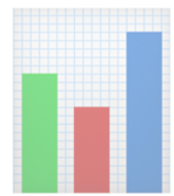
Third-Party



The protocol **flow** (simplified)

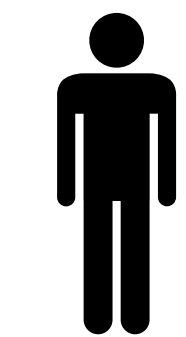
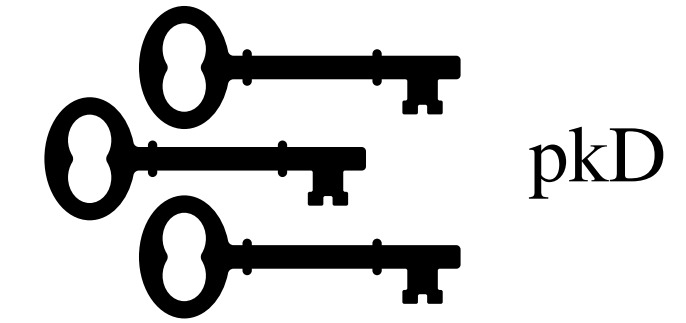


ballotBox for each consular (~city)

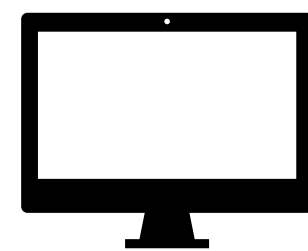


result per ballotBox

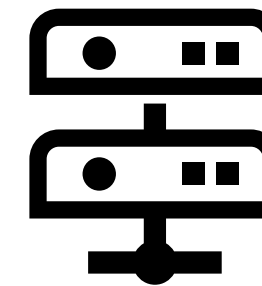
Decryption Trustees



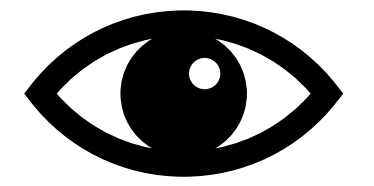
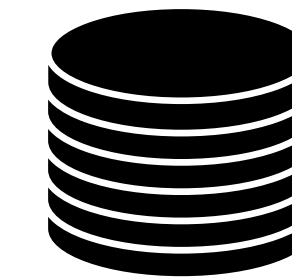
Voter



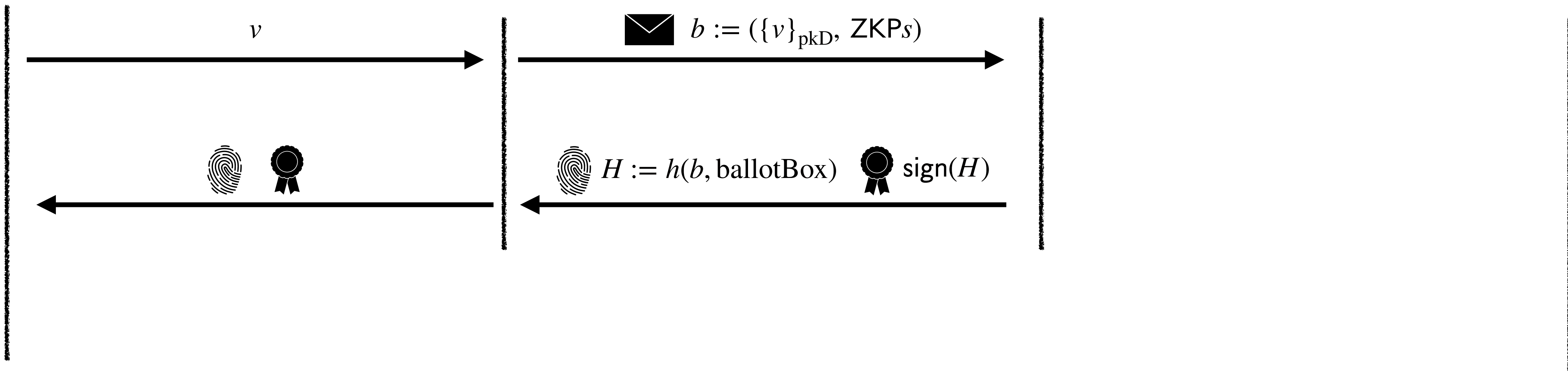
Voting Client



Voting Server



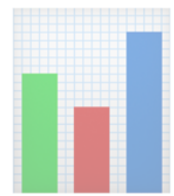
Third-Party



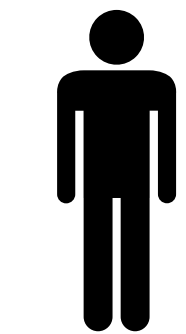
The protocol **flow** (simplified)



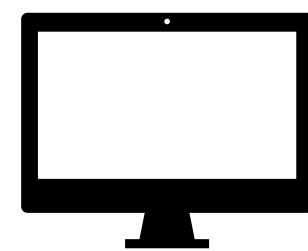
ballotBox for each consular (~city)



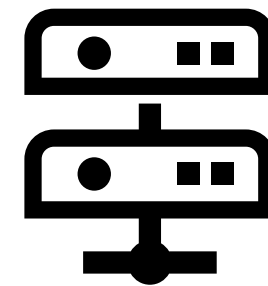
result per ballotBox



Voter

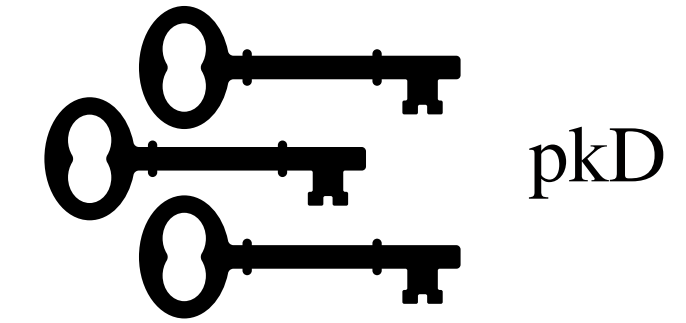


Voting Client

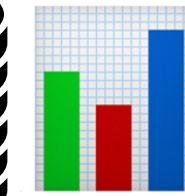
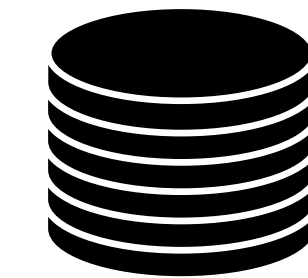


Voting Server

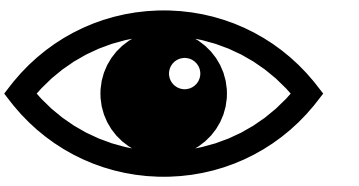
Decryption Trustees



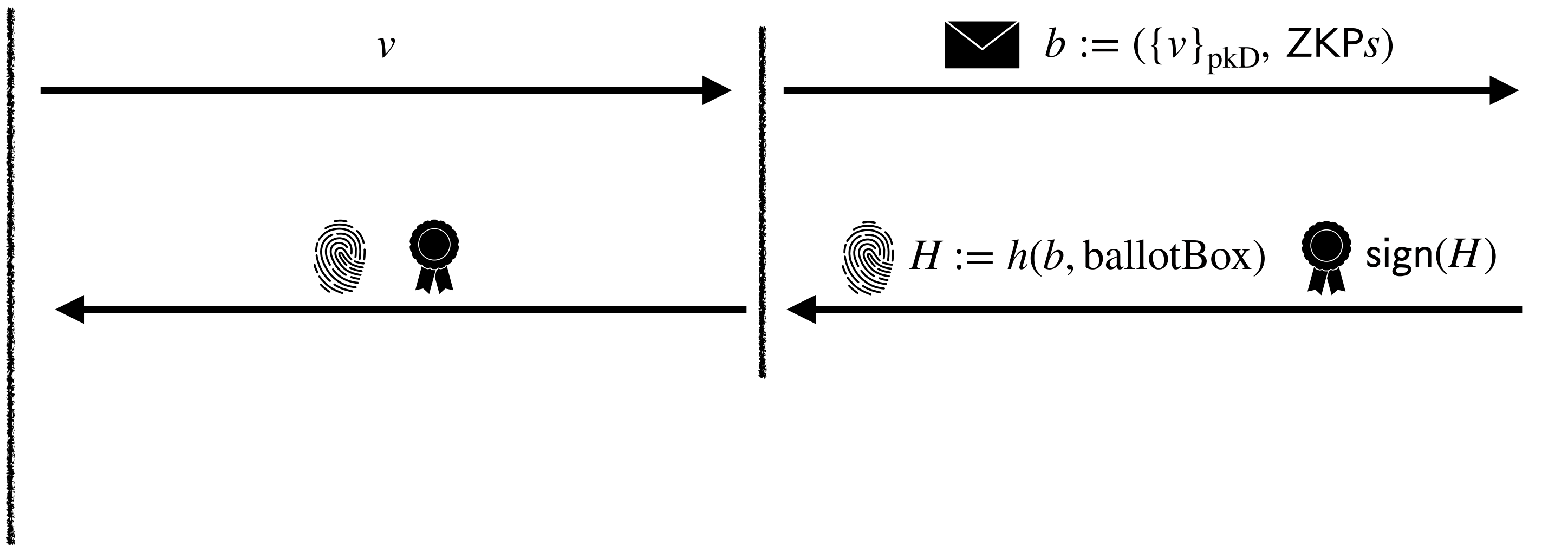
pkD



per ballotBox



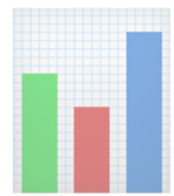
Third-Party



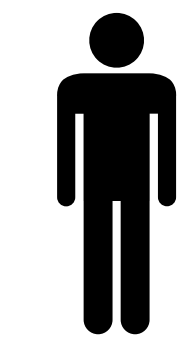
The protocol **flow** (simplified)



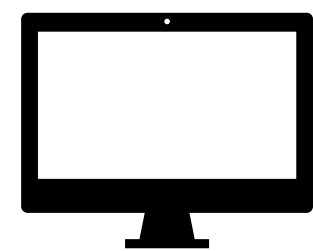
ballotBox for each consular (~city)



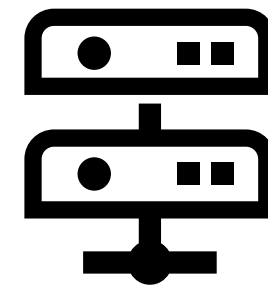
result per ballotBox



Voter

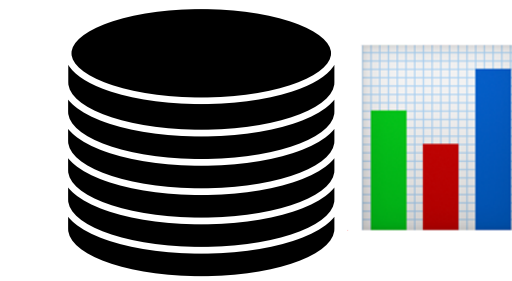
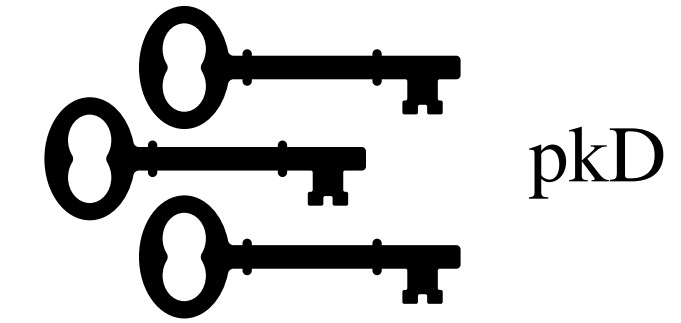


Voting Client

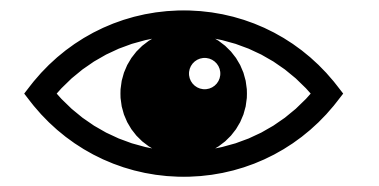


Voting Server

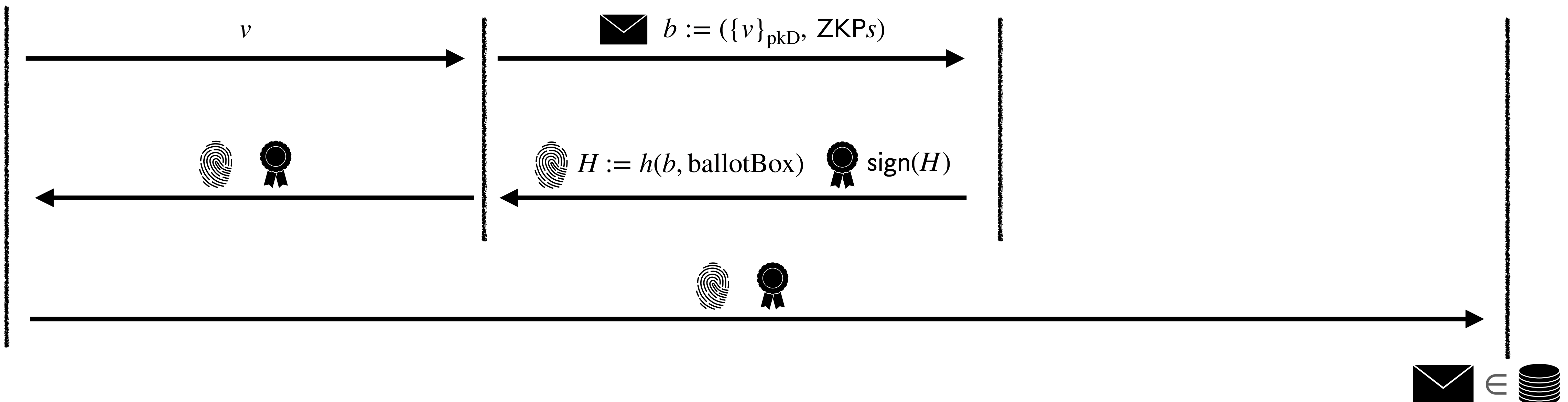
Decryption Trustees



per ballotBox



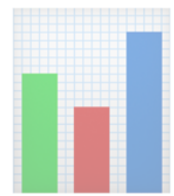
Third-Party



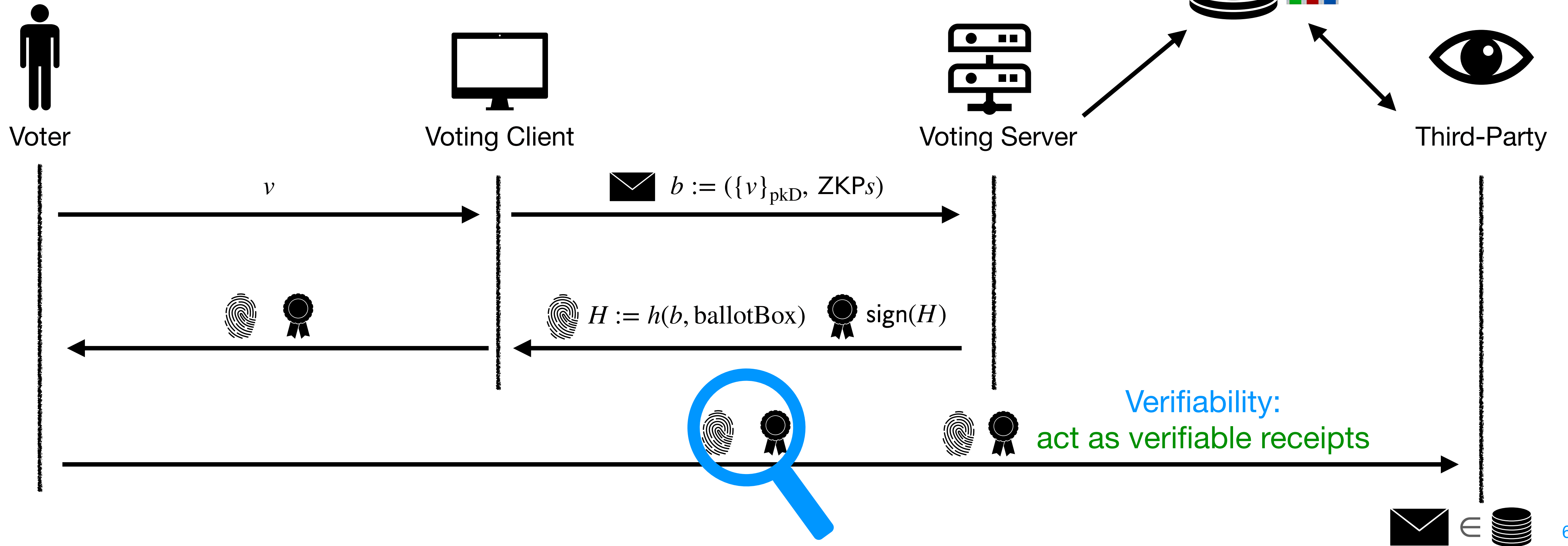
The protocol **flow** (simplified)



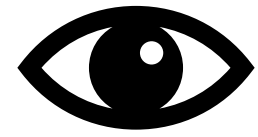
ballotBox for each consular (~city)



result per ballotBox



Security goals and threat models



Security goals and threat models

Security goals

 **Ballot Privacy**: an attacker cannot learn the choice of a voter

 **Verifiability**: voters must have the guarantee that their ballots are counted

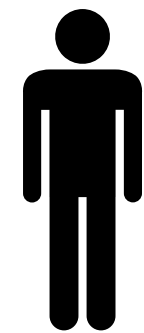
Security goals and threat models

Security goals

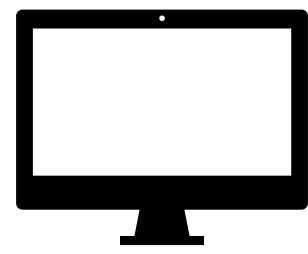
 **Ballot Privacy**: an attacker cannot learn the choice of a voter

 **Verifiability**: voters must have the guarantee that their ballots are counted

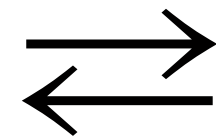
Threat models — **security expectations under**



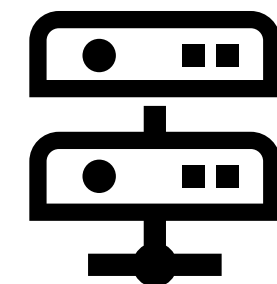
Voter



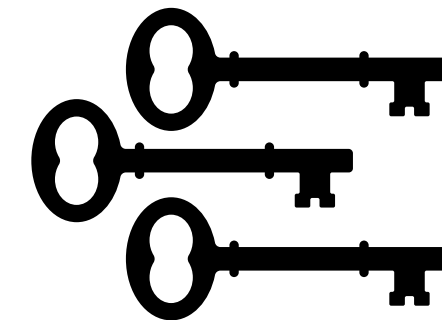
Voting Client



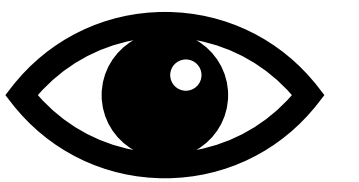
Communication
Channel



Voting Server



Decryption Trustees



Third-Party

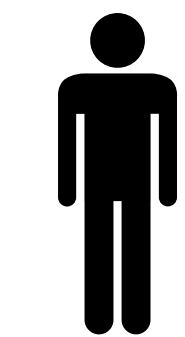
Security goals and threat models

Security goals

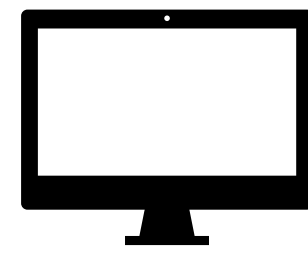
 **Ballot Privacy**: an attacker cannot learn the choice of a voter

 **Verifiability**: voters must have the guarantee that their ballots are counted

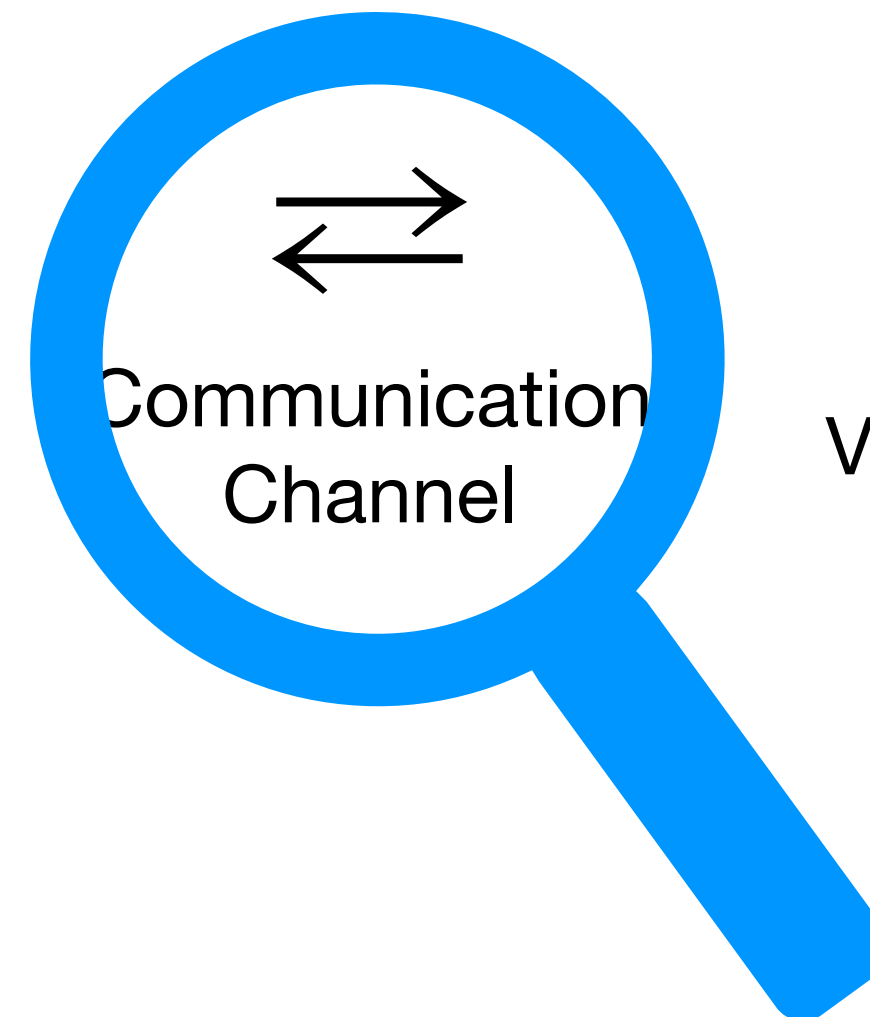
Threat models — **security expectations under**



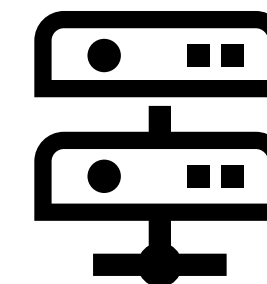
Voter



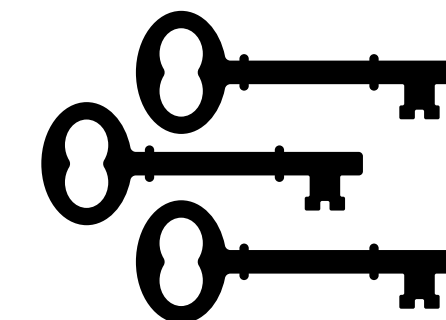
Voting Client



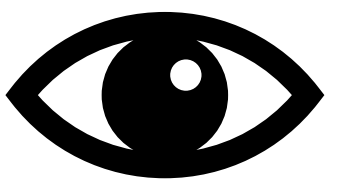
Communication Channel



Voting Server



Decryption Trustees



Third-Party

Plaintext/under TLS: e.g., certificate leak

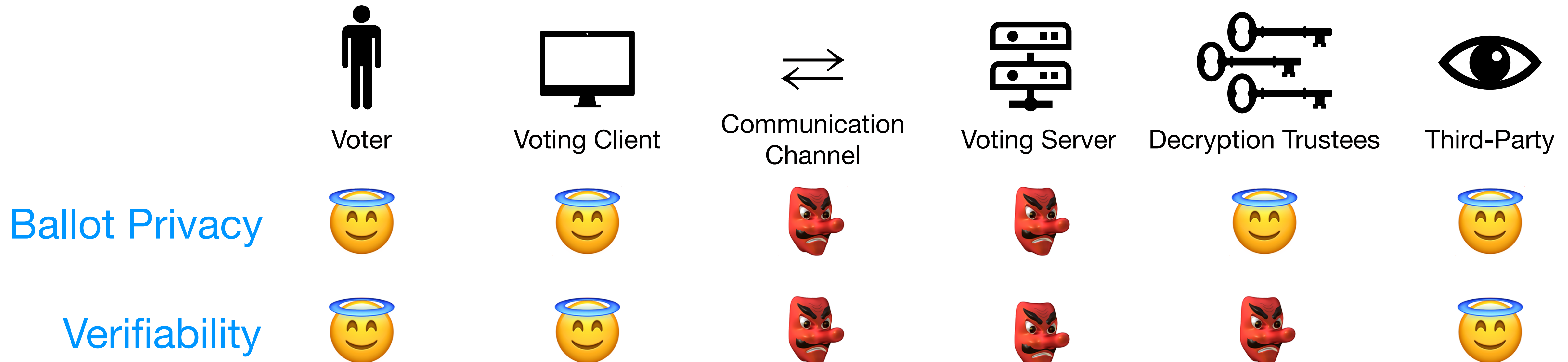
Security goals and threat models

Security goals

 **Ballot Privacy**: an attacker cannot learn the choice of a voter

 **Verifiability**: voters must have the guarantee that their ballots are counted

Threat models — **security expectations under**



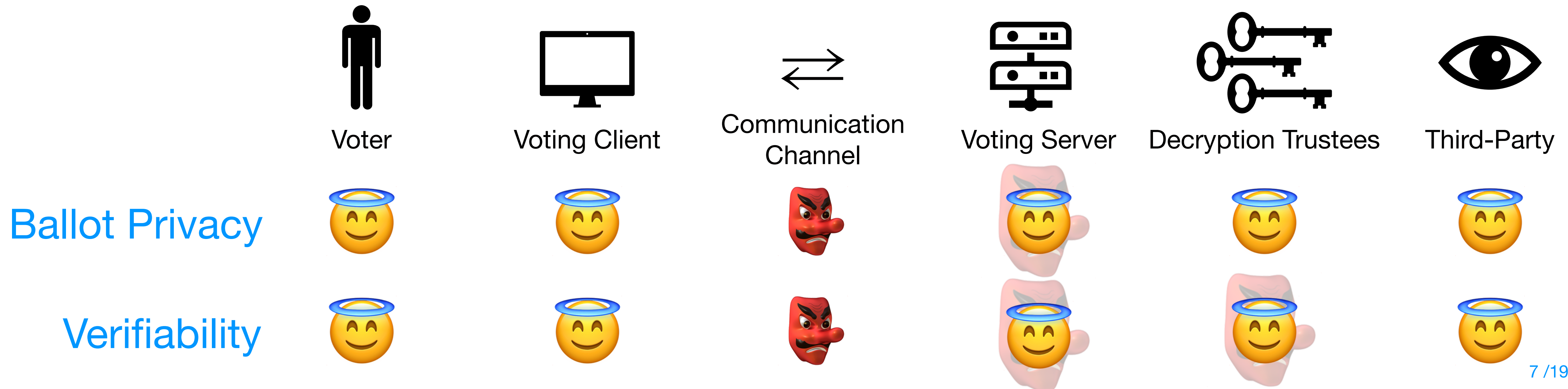
Security goals and threat models

Security goals

 **Ballot Privacy**: an attacker cannot learn the choice of a voter

 **Verifiability**: voters must have the guarantee that their ballots are counted

Threat models — attacks under



Contributions

Contributions

👉 <https://eprint.iacr.org/2022/1653>



First public and comprehensive **specification** of the protocol by reverse



Verifiability and **ballot privacy** can be **attacked** by a channel/server attacker:

- 2 design and implementation **vulnerabilities**
- 6 **attack** variants



Propose **6 fixes**, most of them already implemented for the 2023 election

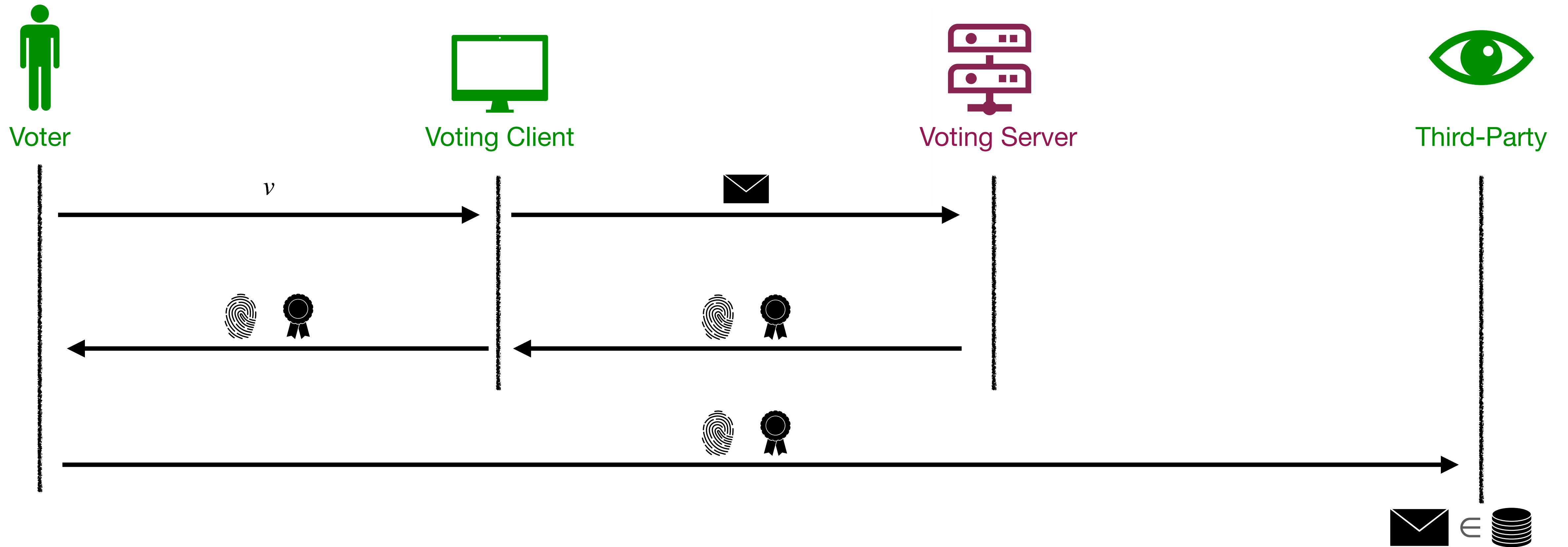


Lessons for future e-voting elections

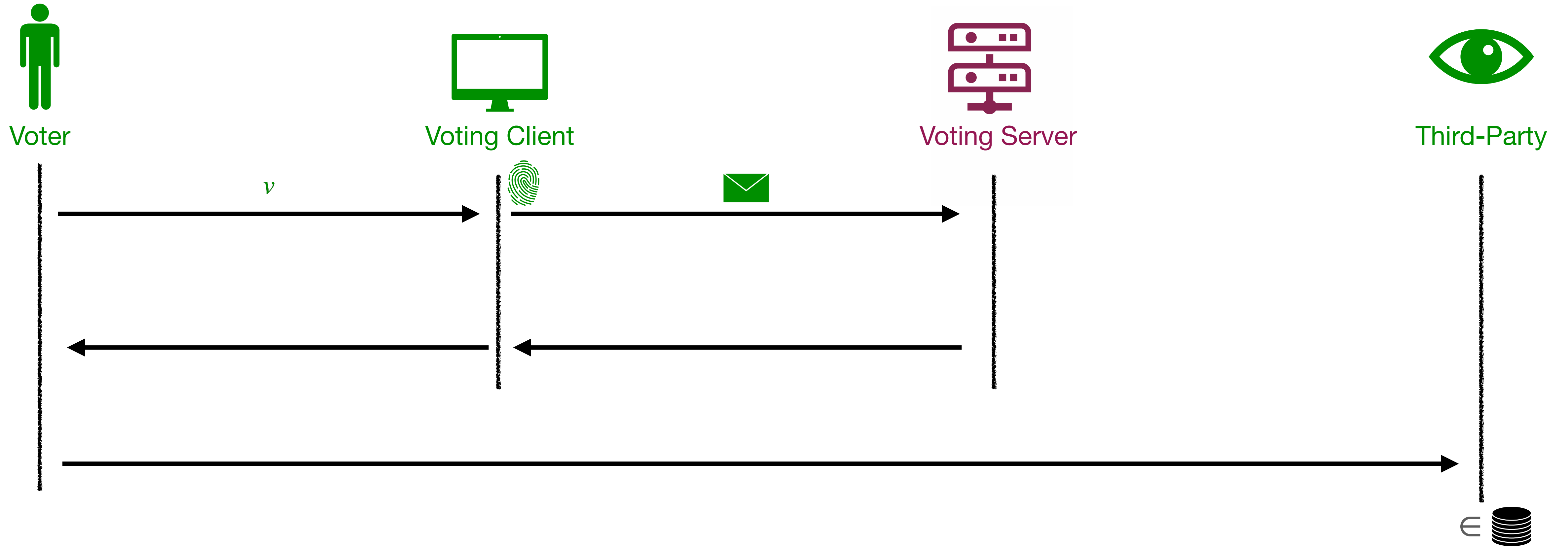
Attacking and Fixing

Election **integrity** and **privacy**

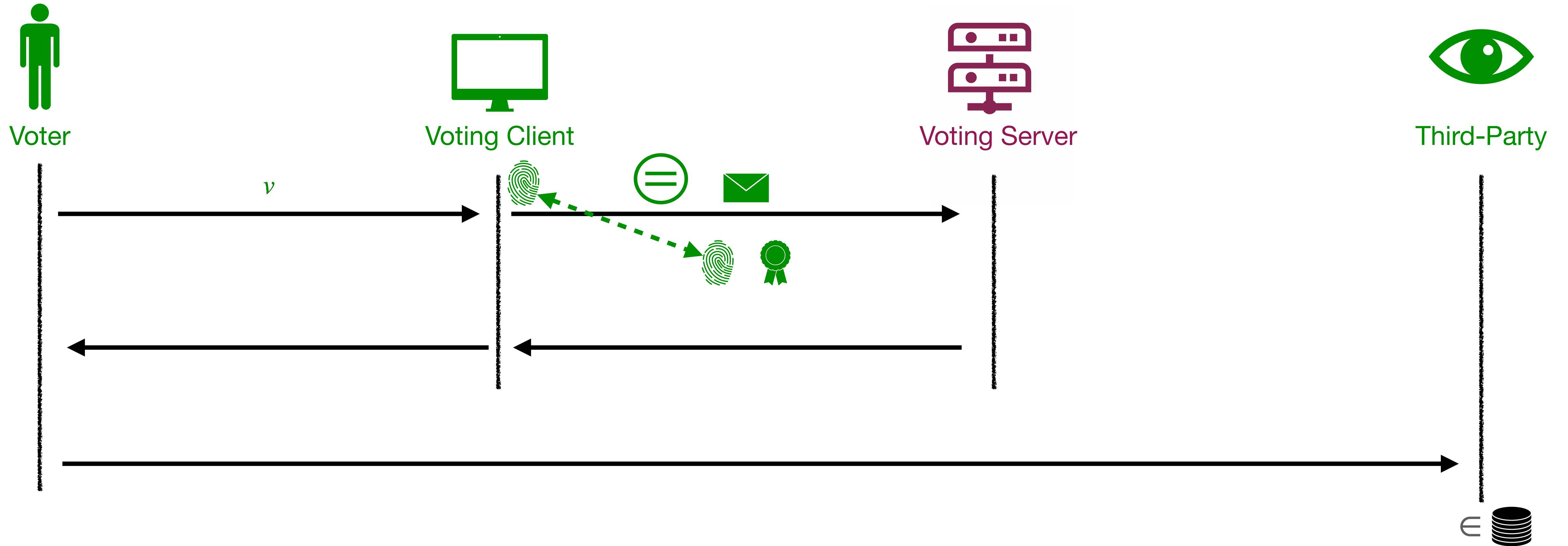
Attacking verifiability and election integrity



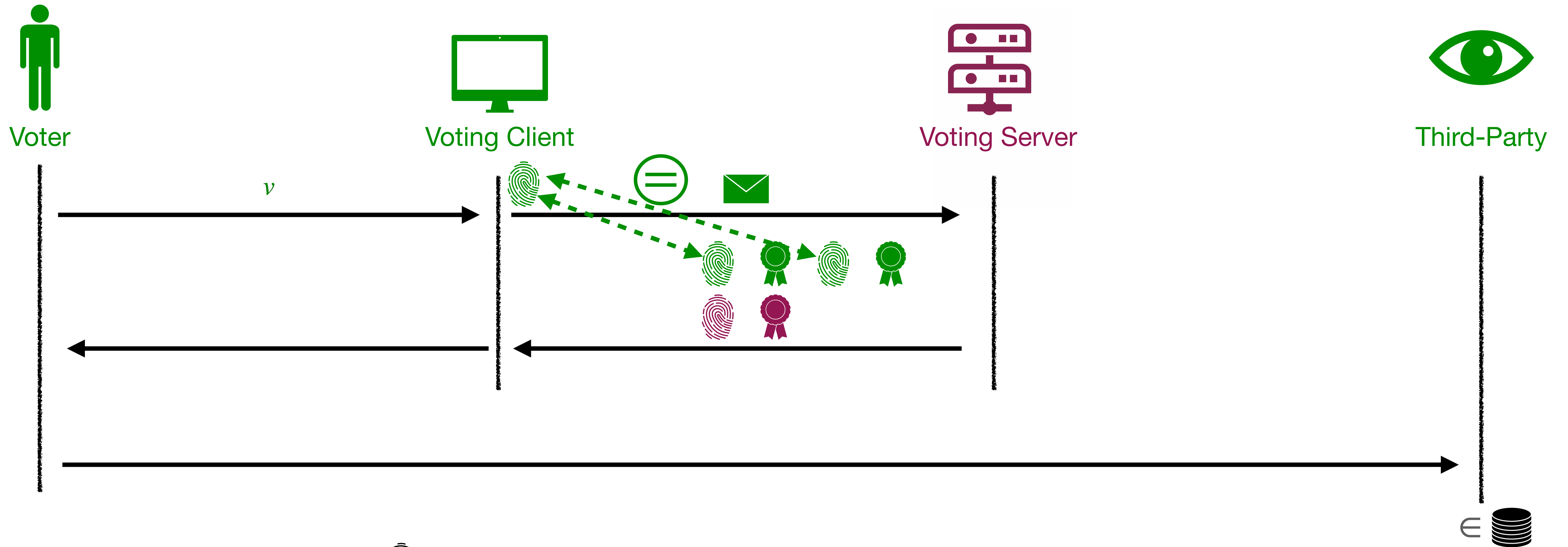
Attacking verifiability and election integrity



Attacking verifiability and election integrity

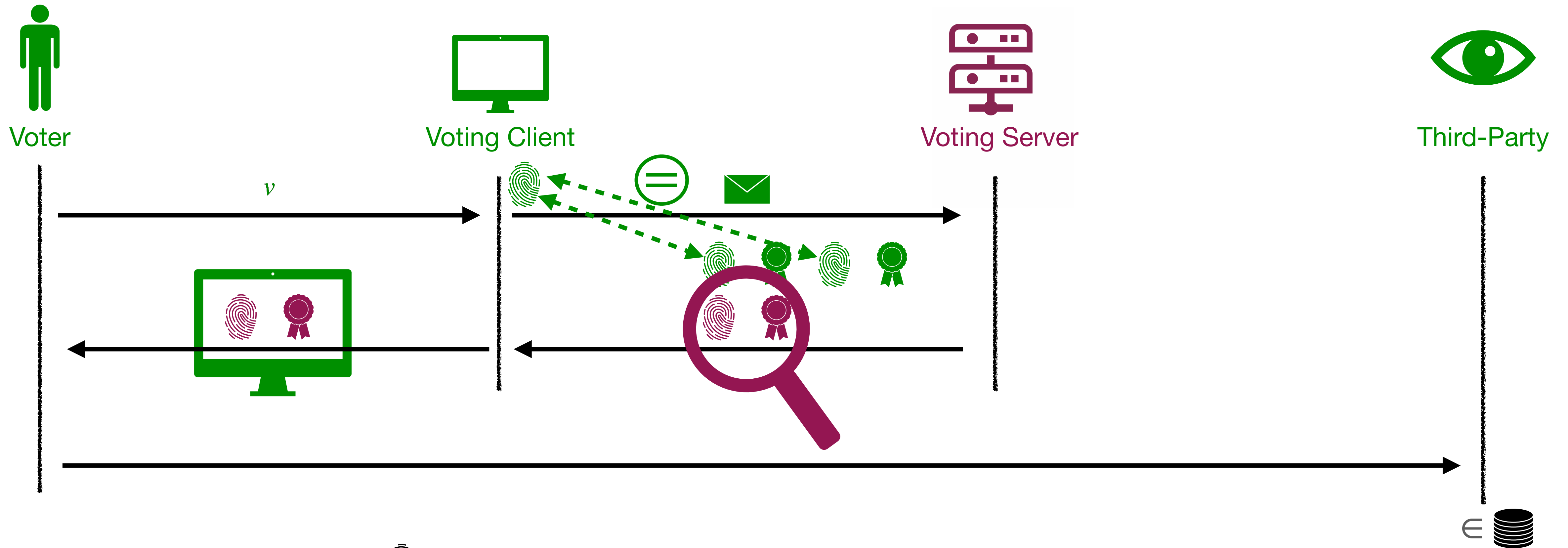





Attacking verifiability and election integrity



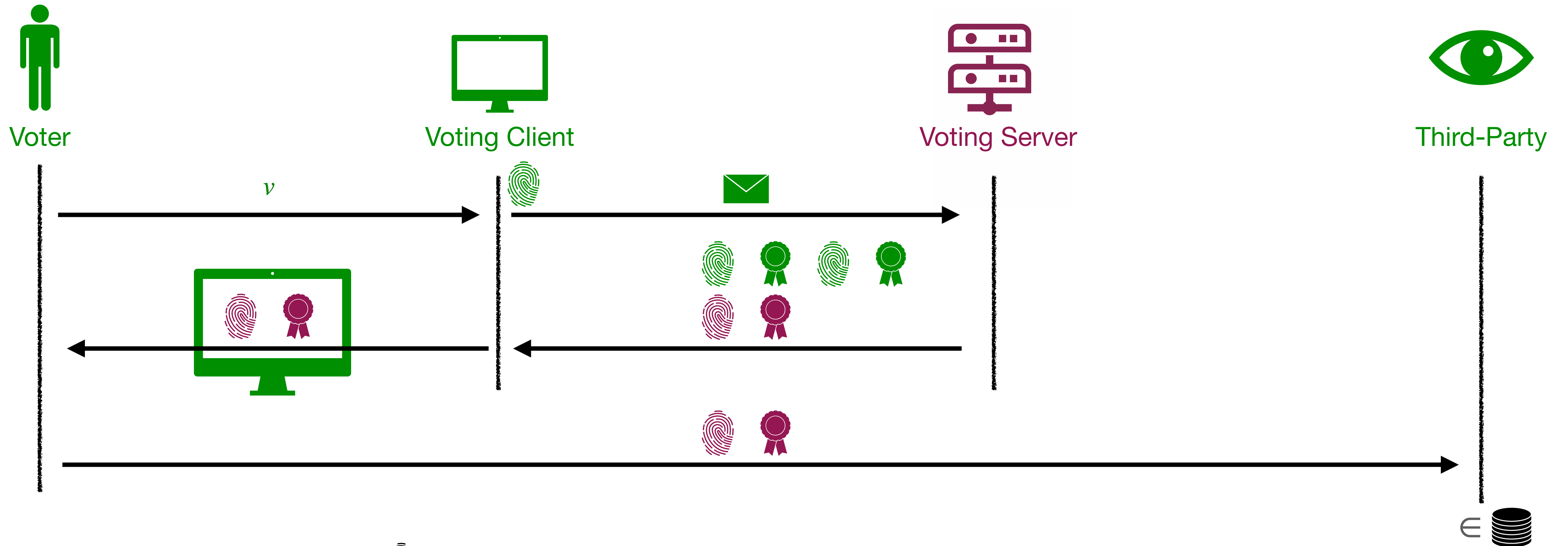
- There are 4 versions of  with various consistency checks in the JavaScript voting client




Attacking verifiability and election integrity



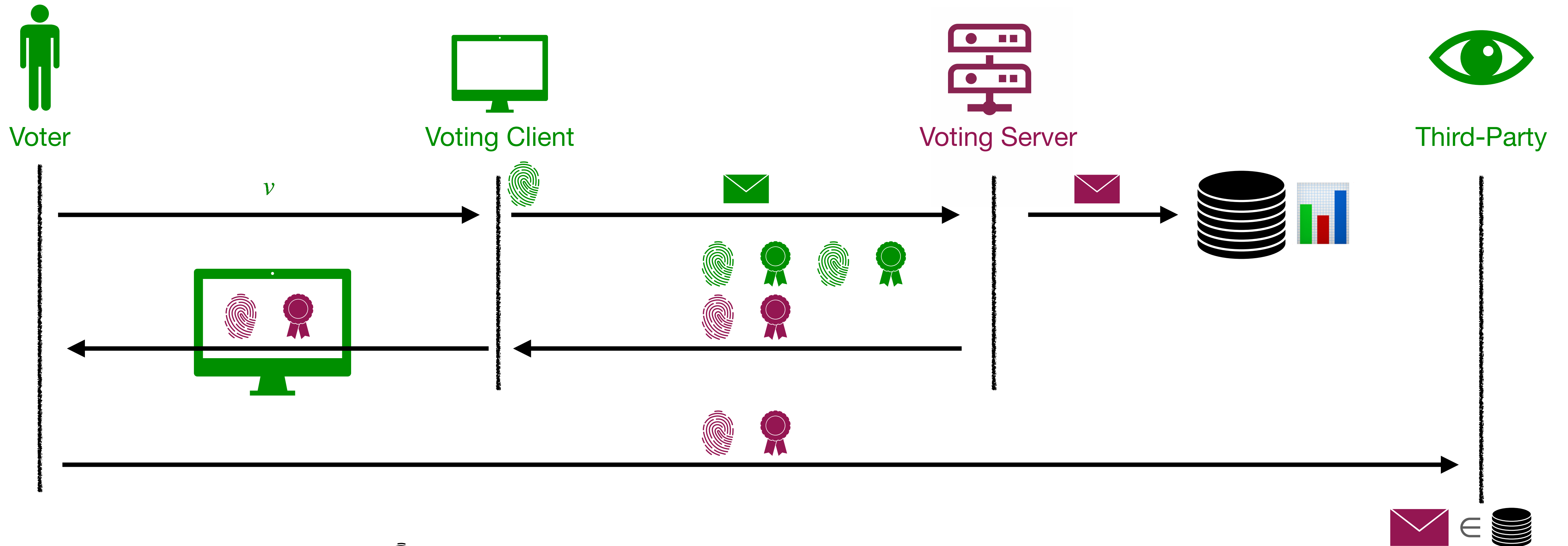
- There are 4 versions of  with various consistency checks in the JavaScript voting client
- **Implementation vulnerability #1** \Rightarrow the   actually displayed to the voter can be **attacker-controlled**




Attacking verifiability and election integrity



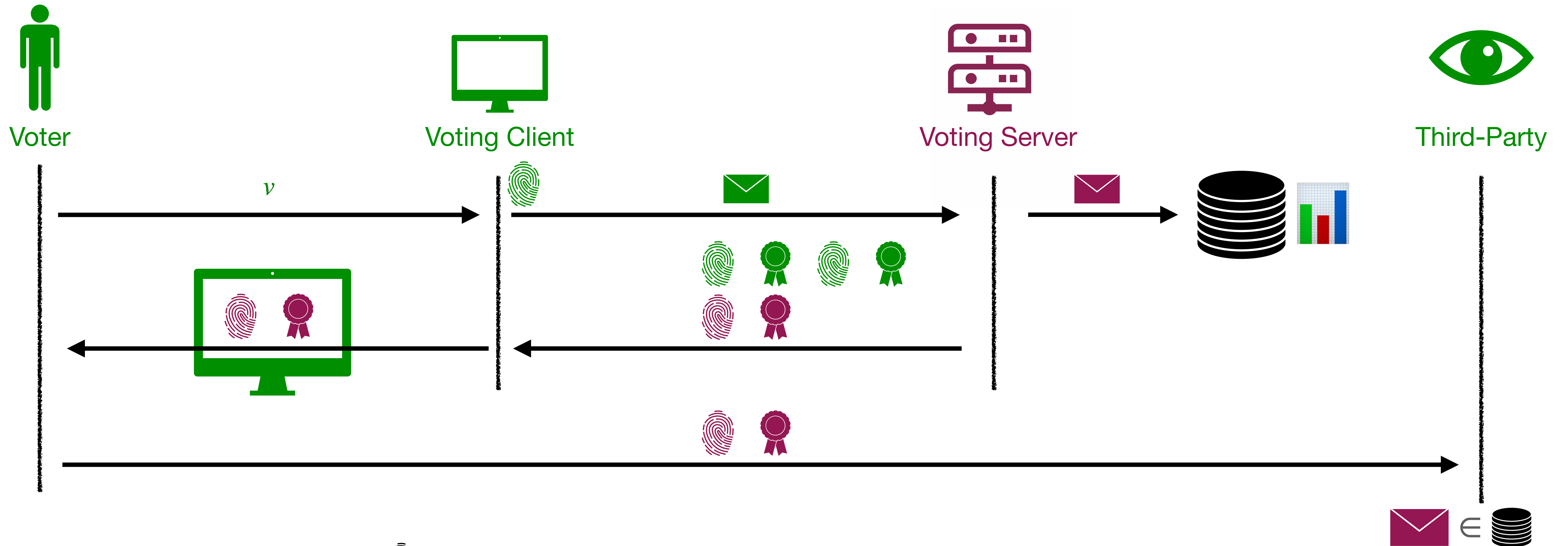
- There are 4 versions of  with various consistency checks in the JavaScript voting client
- **Implementation vulnerability #1** \Rightarrow the   actually displayed to the voter can be **attacker-controlled**




Attacking verifiability and election integrity



- There are 4 versions of  with various consistency checks in the JavaScript voting client
- **Implementation vulnerability #1** \Rightarrow the   actually displayed to the voter can be **attacker-controlled**


Attacking verifiability and election integrity




- There are 4 versions of  with various consistency checks in the JavaScript voting client
- **Implementation vulnerability #1** \Rightarrow the   actually displayed to the voter can be **attacker-controlled**
- **Impact:** channel or server attacker can **stealthily modify the outcome by replacing or dropping ballots**

Attacking ballot privacy (preliminary)


- Design vulnerability #2 \Rightarrow ballots ZKPs do not bind ballotBox


 $b := (\{v\}_{pkD}, ZKPs)$

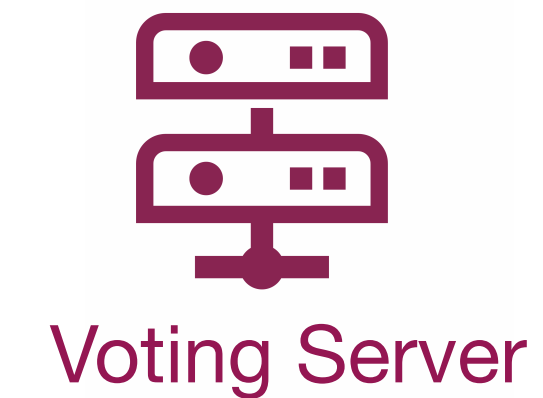
 $H := h(b, \text{ballotBox})$

Attacking ballot privacy (preliminary)

- Design vulnerability #2 \Rightarrow ballots ZKPs do not bind ballotBox


 $b := (\{v\}_{pkD}, ZKPs)$


 $H := h(b, \text{ballotBox})$

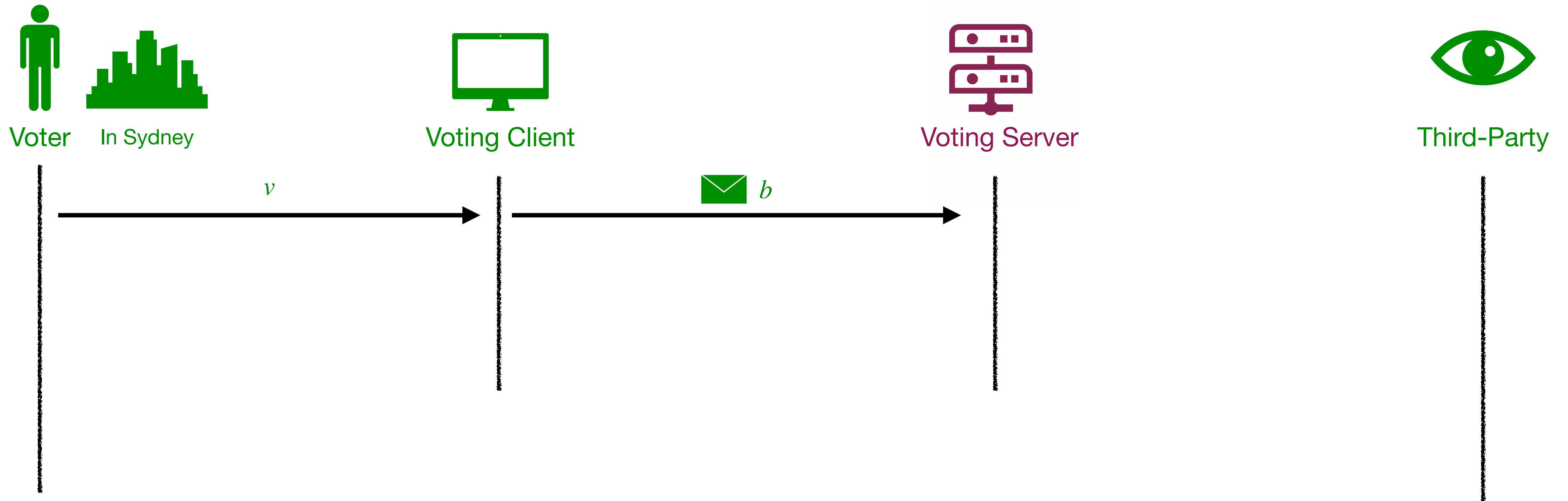


Attacking ballot privacy (preliminary)

- Design vulnerability #2 \Rightarrow ballots ZKPs do not bind ballotBox


 $b := (\{v\}_{pkD}, ZKPs)$


 $H := h(b, \text{ballotBox})$

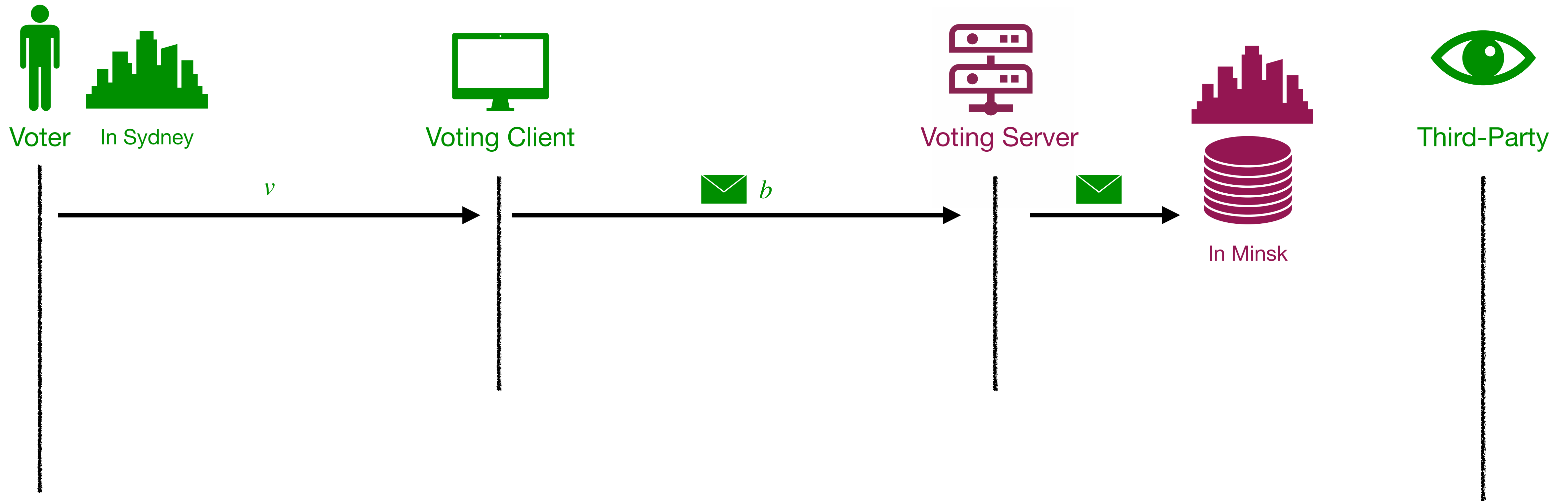


Attacking ballot privacy (preliminary)

- Design vulnerability #2 \Rightarrow ballots ZKPs do not bind ballotBox


 $b := (\{v\}_{pkD}, ZKPs)$


 $H := h(b, \text{ballotBox})$

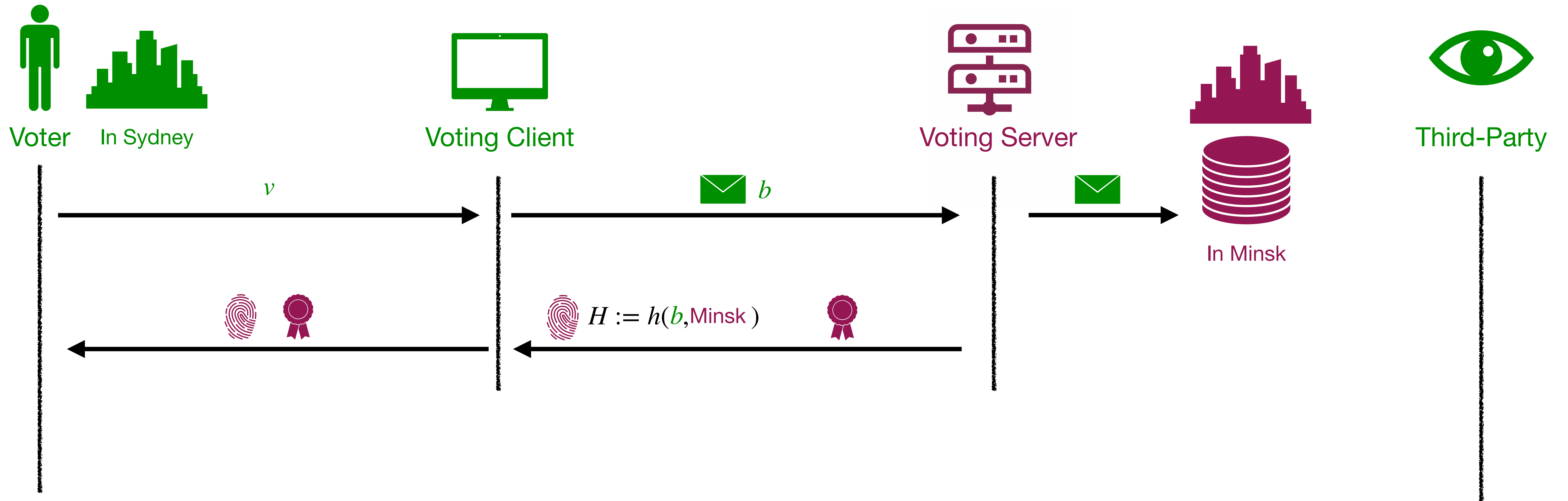


Attacking ballot privacy (preliminary)

- Design vulnerability #2 \Rightarrow ballots ZKPs do not bind ballotBox


 $b := (\{v\}_{pkD}, ZKPs)$


 $H := h(b, \text{ballotBox})$

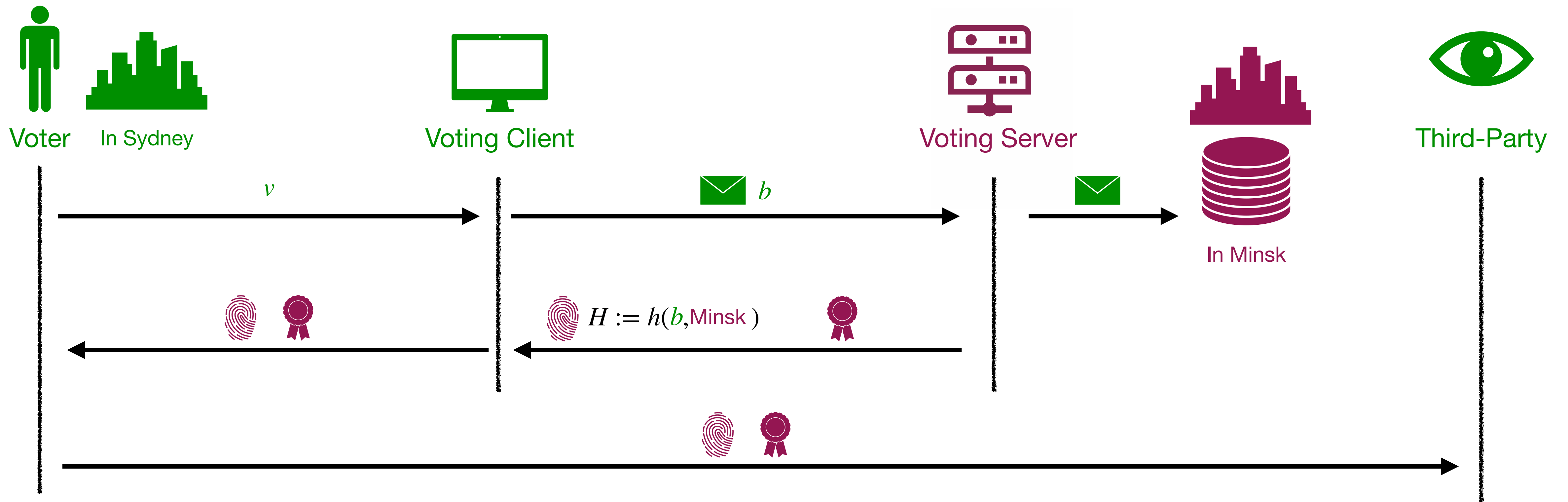


Attacking ballot privacy (preliminary)

- Design vulnerability #2 \Rightarrow ballots ZKPs do not bind ballotBox


 $b := (\{v\}_{pkD}, ZKPs)$


 $H := h(b, \text{ballotBox})$

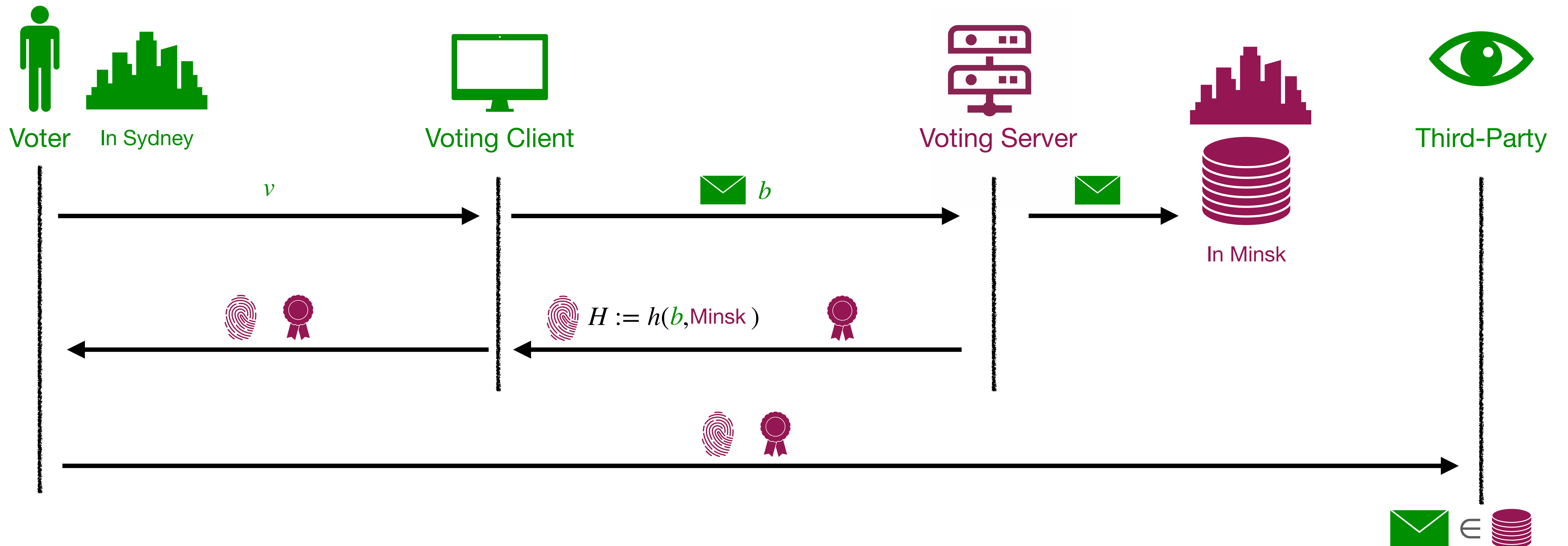


Attacking ballot privacy (preliminary)

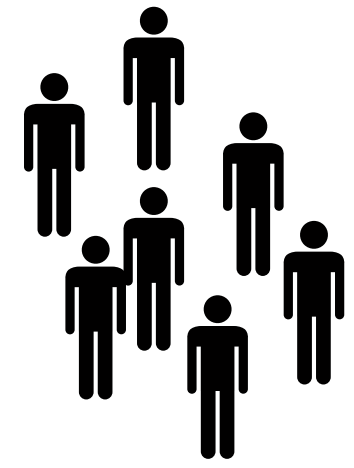
- Design vulnerability #2 \Rightarrow ballots ZKPs do not bind ballotBox

 $b := (\{v\}_{pkD}, ZKPs)$

 $H := h(b, \text{ballotBox})$



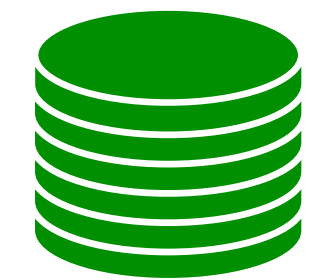
Attacking ballot privacy (simplified)



In Sydney



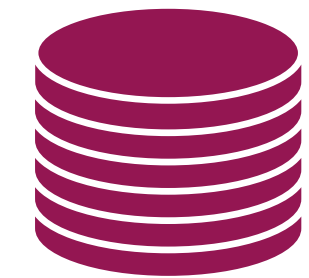
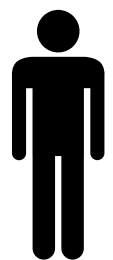
Target Voter



Sydney

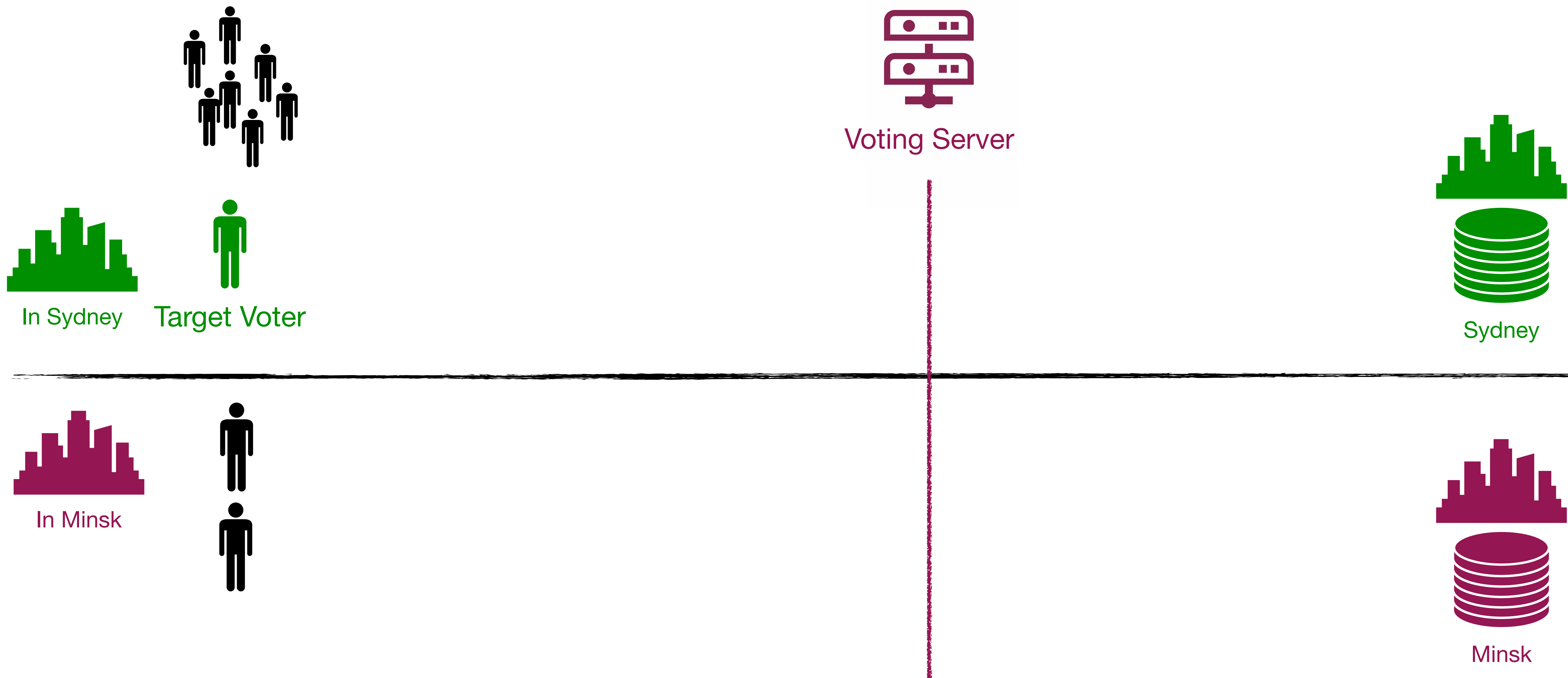


In Minsk

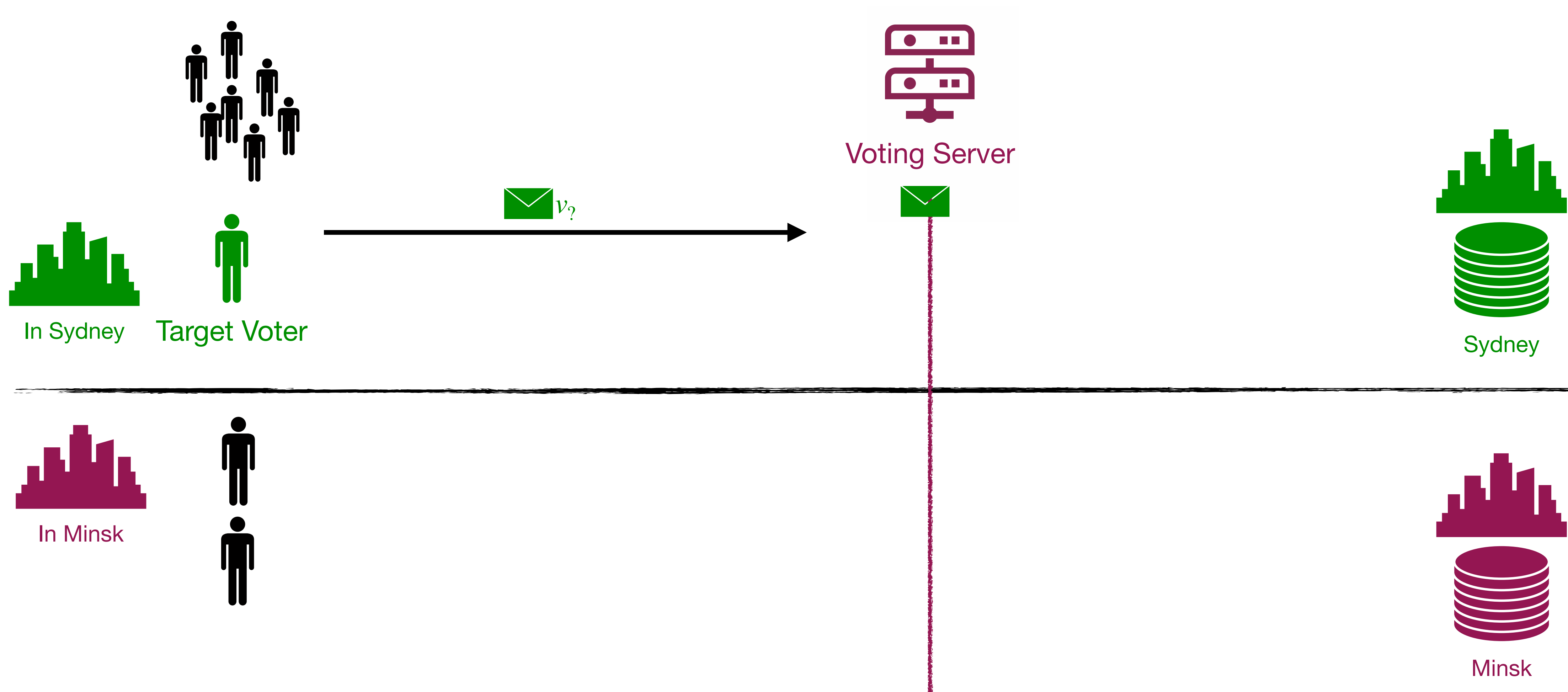


Minsk

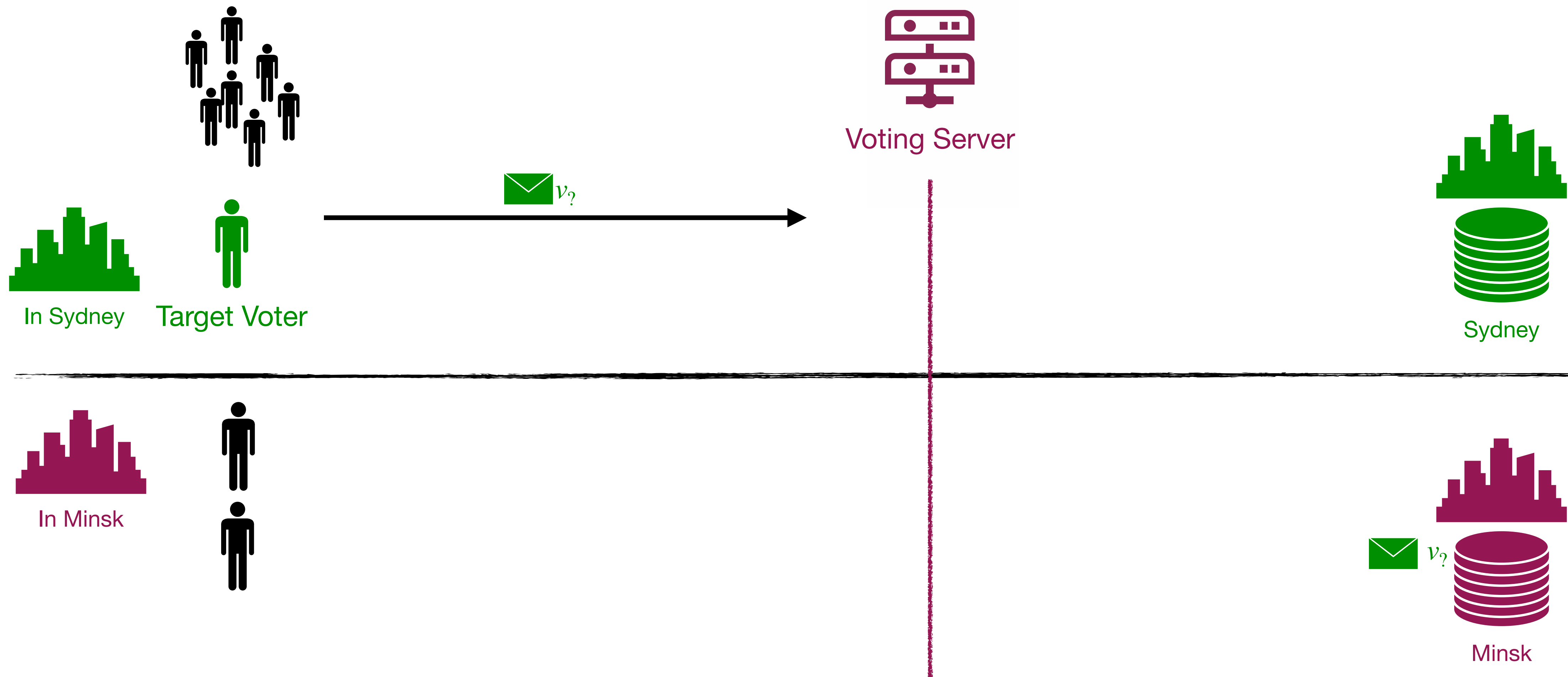
Attacking ballot privacy (simplified)



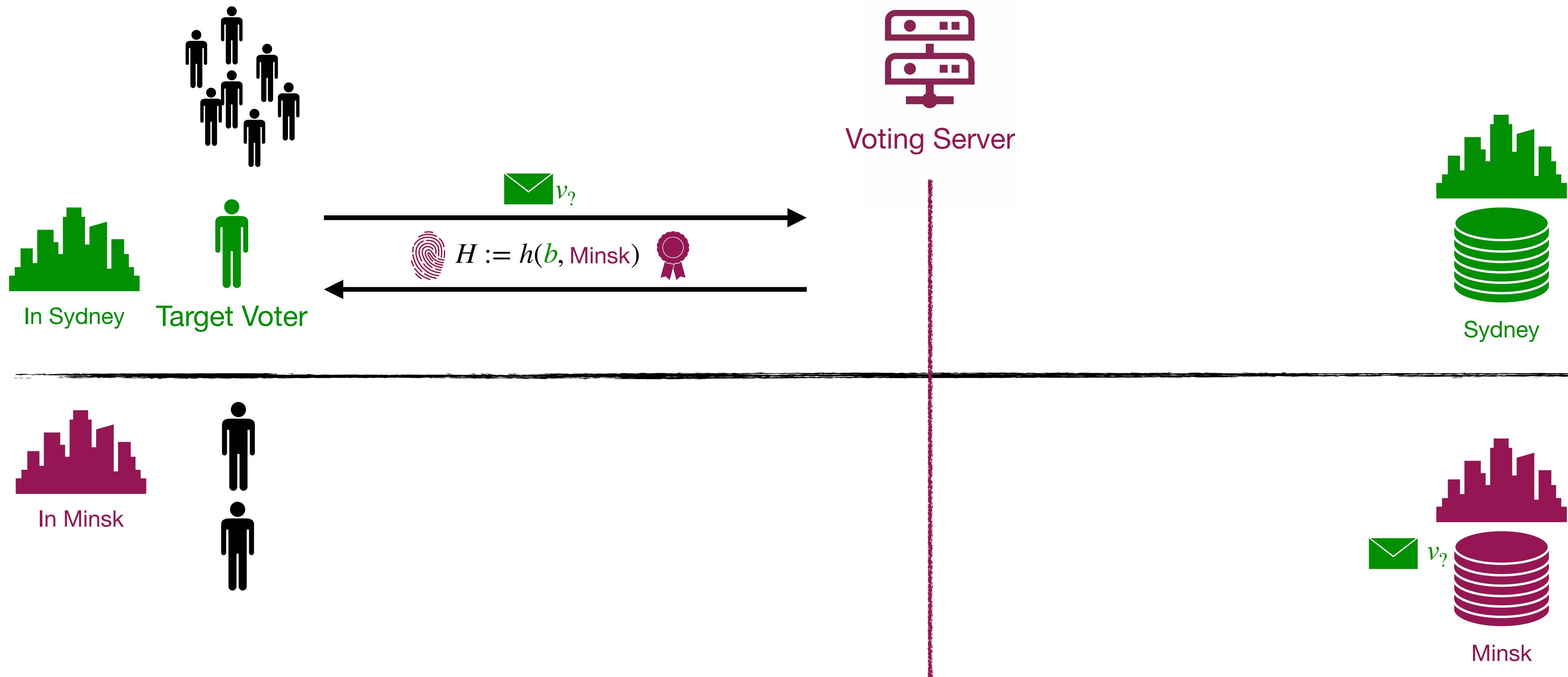
Attacking ballot privacy (simplified)



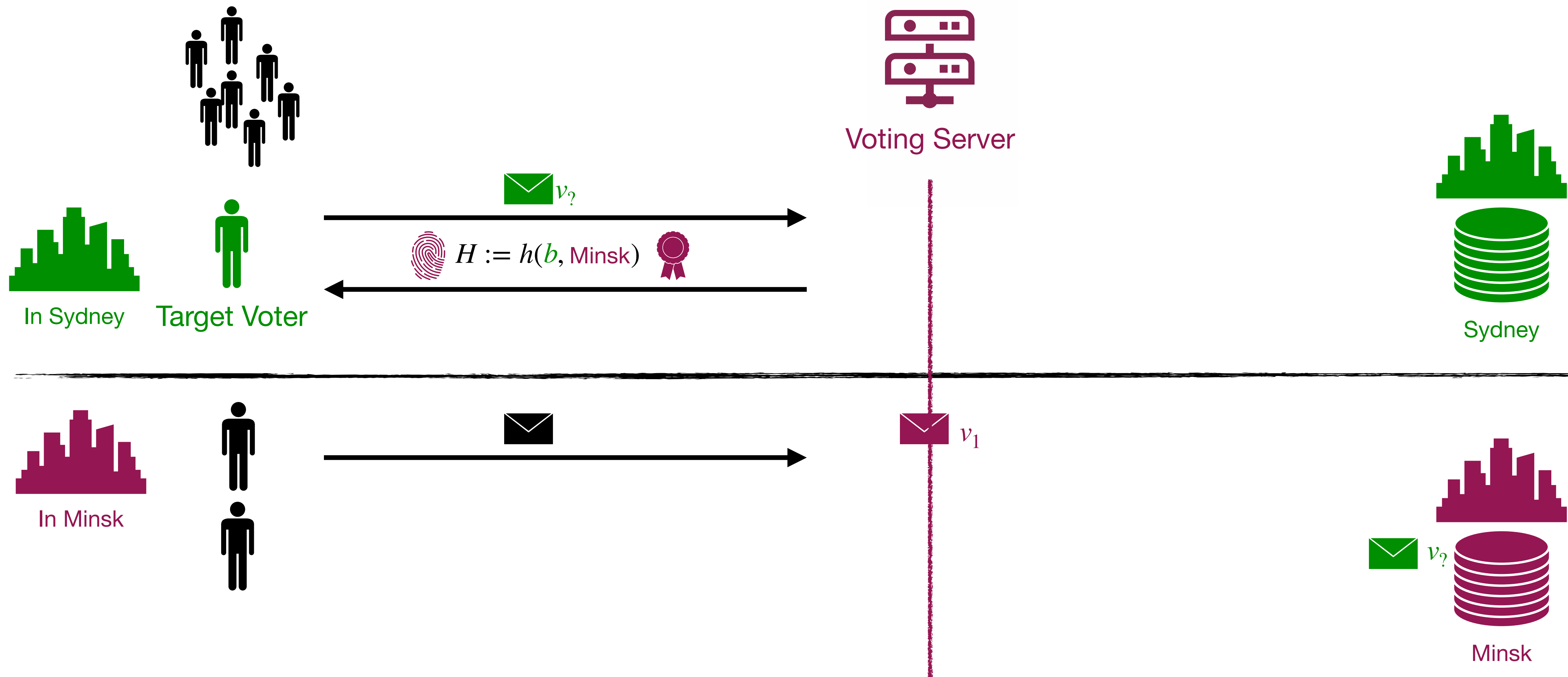
Attacking ballot privacy (simplified)



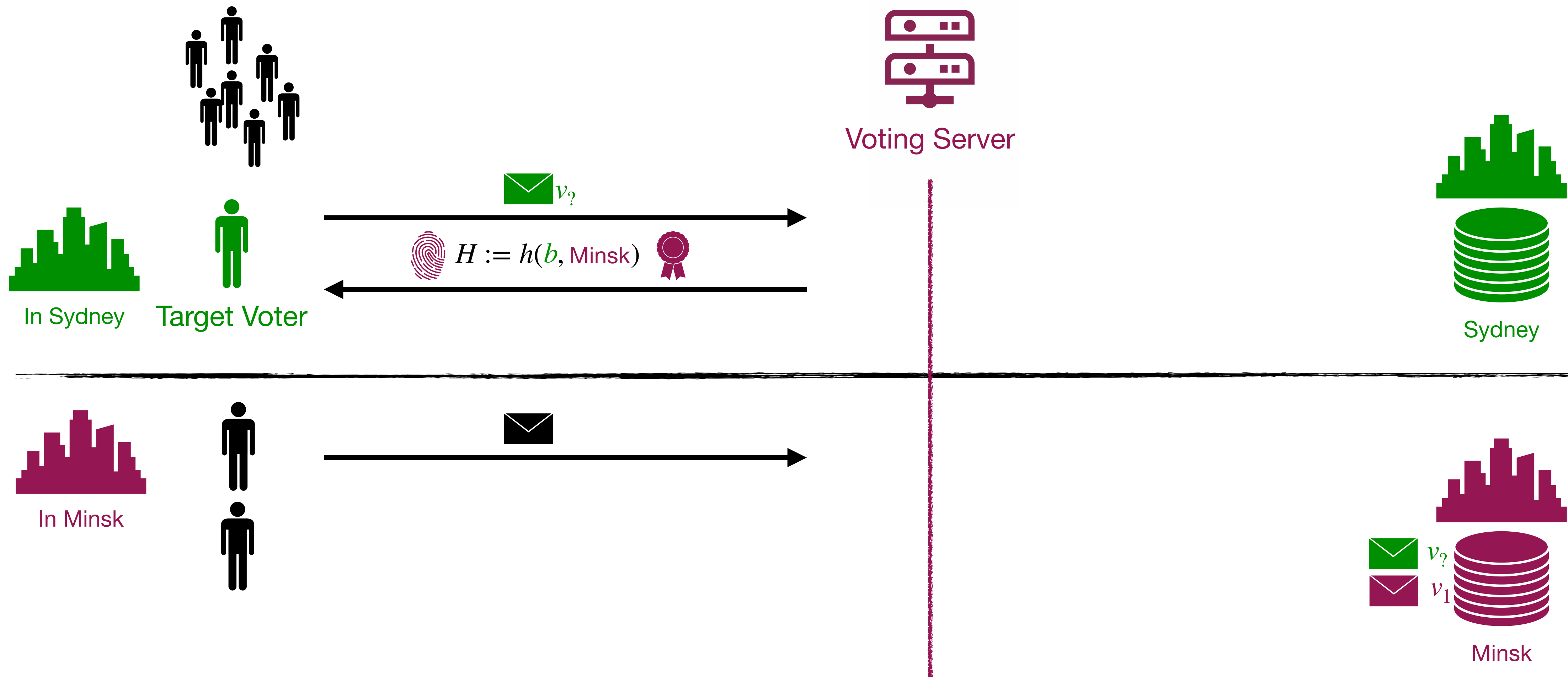
Attacking ballot privacy (simplified)



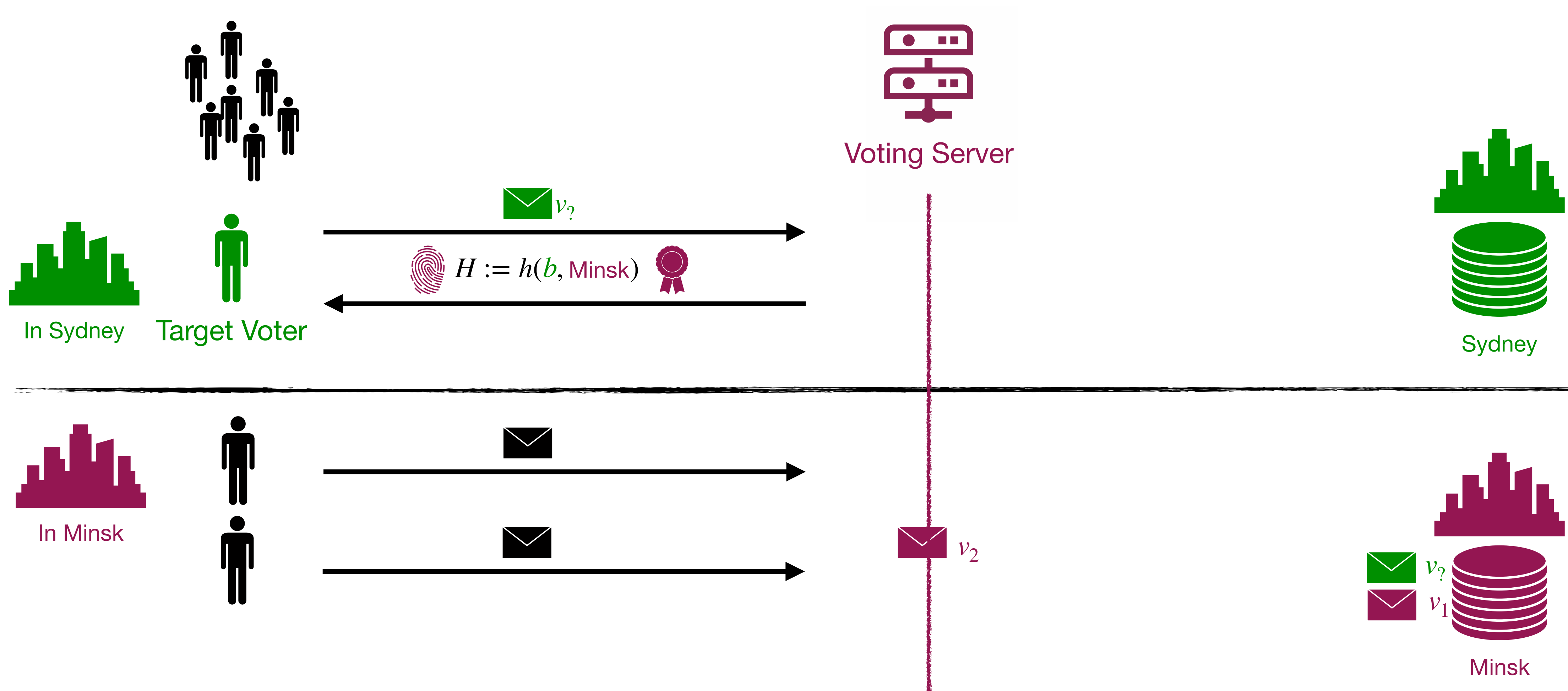
Attacking ballot privacy (simplified)



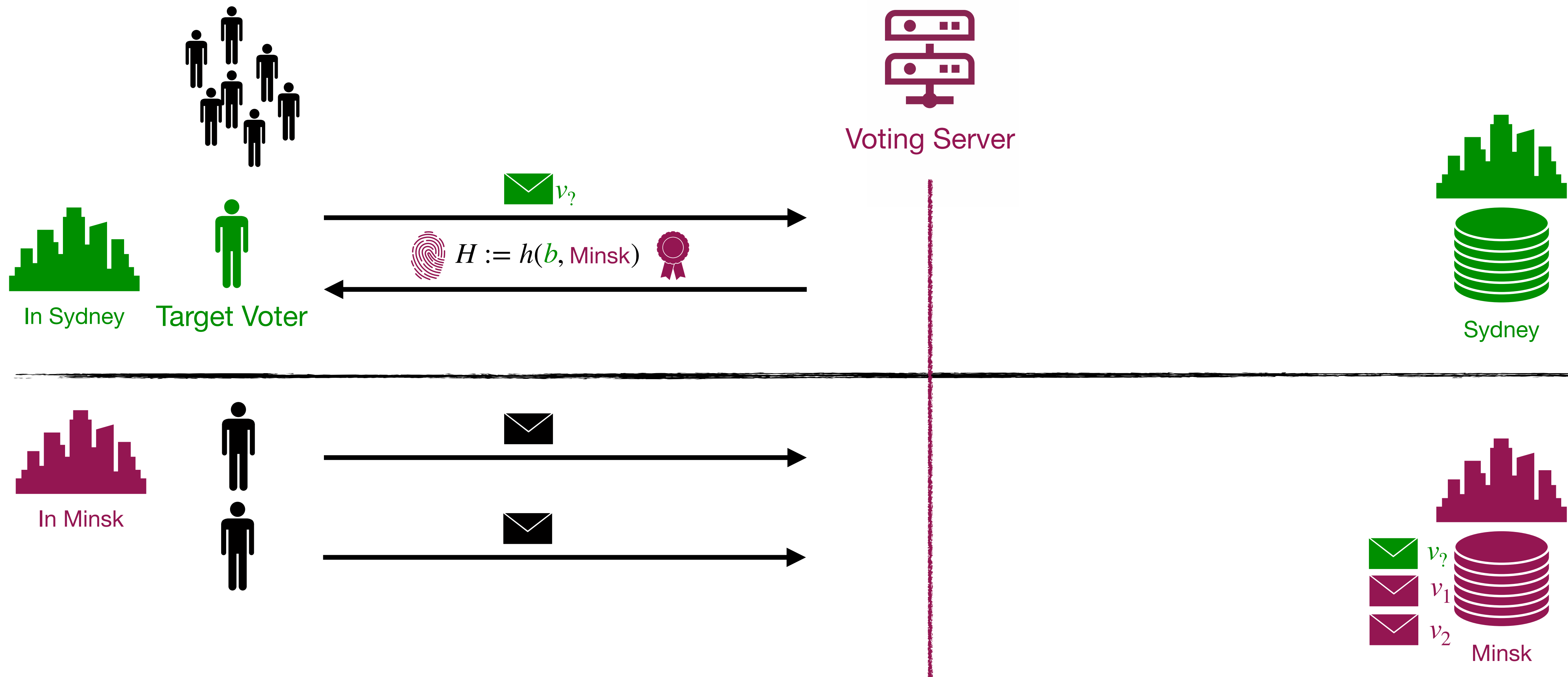
Attacking ballot privacy (simplified)



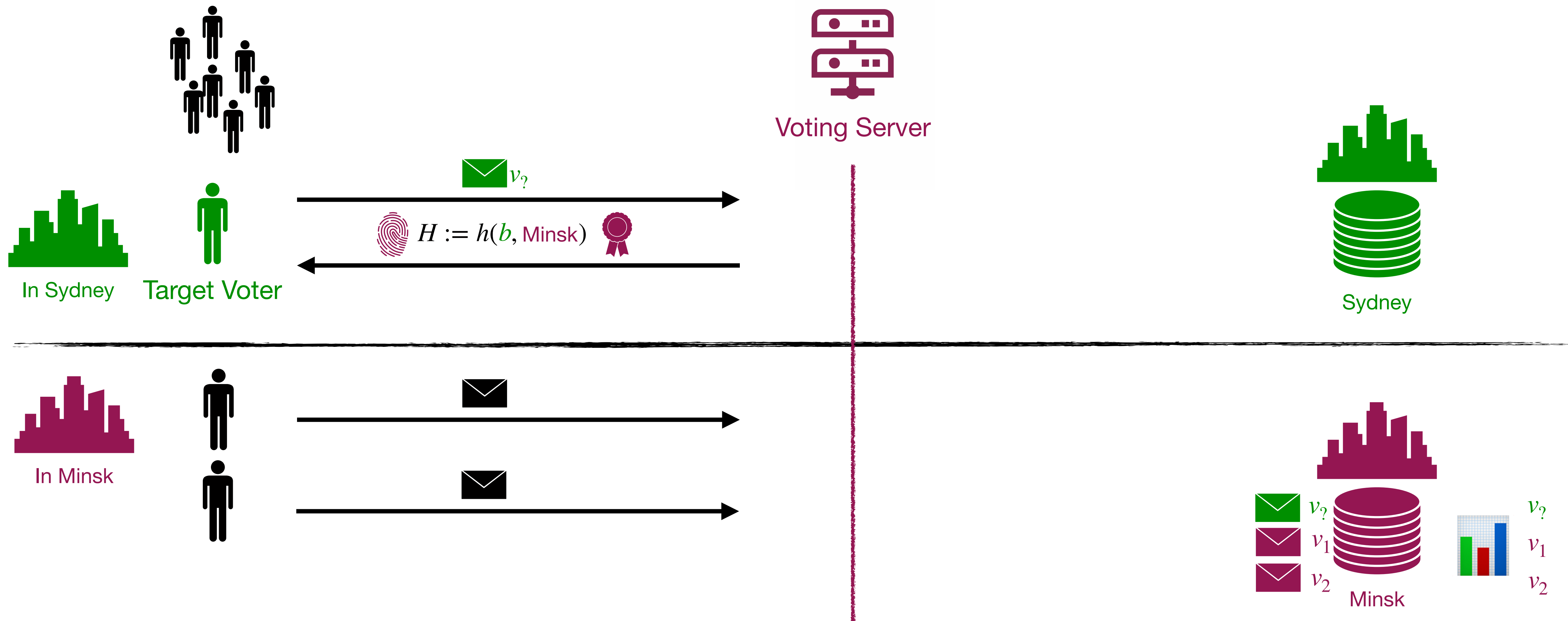
Attacking ballot privacy (simplified)



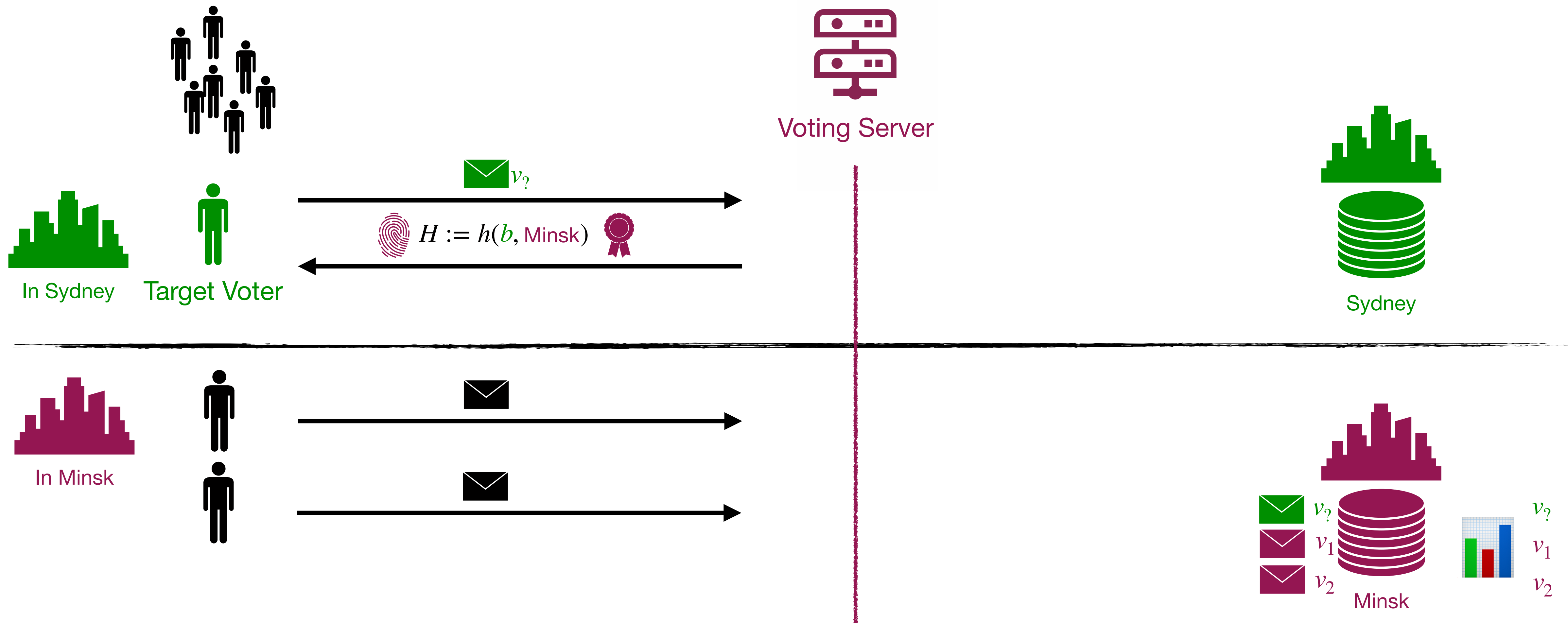
Attacking ballot privacy (simplified)



Attacking ballot privacy (simplified)





Attacking ballot privacy (simplified)



- **Impact:** channel or server attacker can **stealthily learn some target voters' vote** (and perform remote coercion)

Fix the FLEP and future election

We proposed 6 fixes and notably:

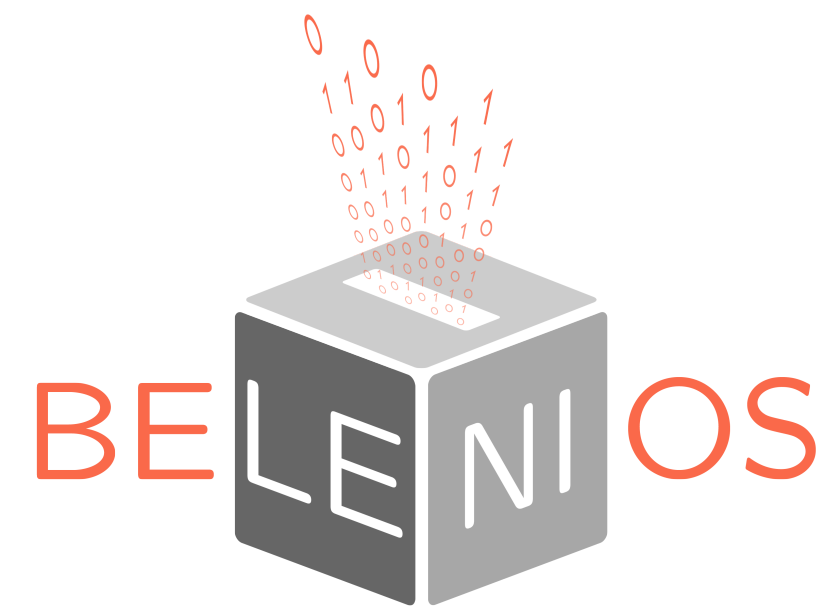
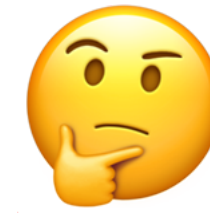
1. Display and check  instead of  ✓/✗ partially done for 2023 election
2. Binds `ballotBox` to the ballot ZKPs ✓ already implemented for 2023
3. Third-Party checks `ballotBox` ✓ already implemented for 2023

(Attacks and fixes were responsibly disclosed to the vendor and stakeholders.)
Special thanks to the ANSSI who have been proactive in this process.

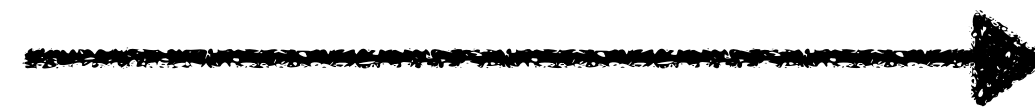
Lessons Learned

Recommendations and research questions

How come?



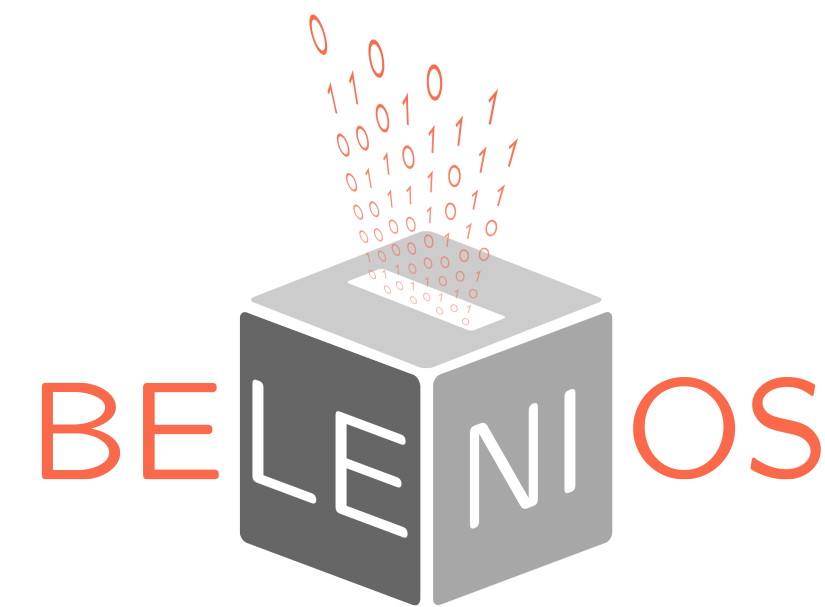
1: Adapt the design



FLEP
Protocol

State-of-art protocol
affected by none of the attacks

How come?



1: Adapt the design



FLEP
Protocol

2: Implement, Deploy, Audit



2022 Election



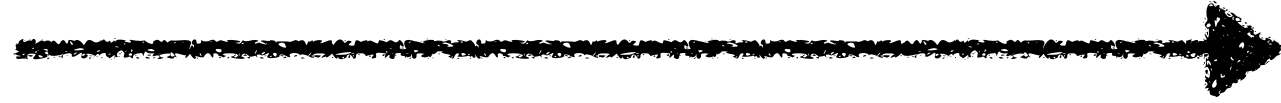
State-of-art protocol
affected by none of the attacks

FLEP 2022
affected by 6 attacks
+ other concerns
not discussed here



1: Adapt the design



1: Adapt the design



Operational constraints as scientific bottleneck

1. State-of-the-art solutions **lack features** for real-world use cases
 - **Multi-ballot-box** for announcing fine-grain results (+ properties./proofs)
 - Downloadable receipts  
2. Distribute trust for the voters authentication is **an open problem** (practical solution)
 - 👉 currently a **single-point-of-trust** for eligibility verification
3. Security by protocol-design versus operational rules
 - 👉 currently decryption quorum rules are not properly **cryptographically enforced**

2: Implement, Deploy, Audit



2: Implement, Deploy, Audit



1. Voting client is **the critical component** (versus focus on securing the server)

- Make it **trustworthy**: open spec and source, audit, etc.
- Make it **monitorable** to allow detecting servers serving modified voting clients (e.g., SPA)
- **Simplify and specify** the voters' journey/tasks and assume no more (we proposed some)

More generally:

any component that needs to be trusted must undergo such process

2. Transparency and Openness

- Clear security objectives and threat models
- Open specification, promote public scrutiny (e.g., bung bounty as in Switzerland)

Conclusion

 <https://eprint.iacr.org/2022/1653>

Conclusion

👉 <https://eprint.iacr.org/2022/1653>

What can go wrong with and what can be learnt from adapting and deploying a proven secure academic e-voting protocol to the real world ?



First public and comprehensive **specification** of the protocol by reverse



Verifiability and **ballot privacy** can be **attacked** by a channel/server attacker:

- 2 design and implementation **vulnerabilities**
- 6 **attack** variants



Propose 6 **fixes**, most of them already implemented for the 2023 election



Lessons for future e-voting elections