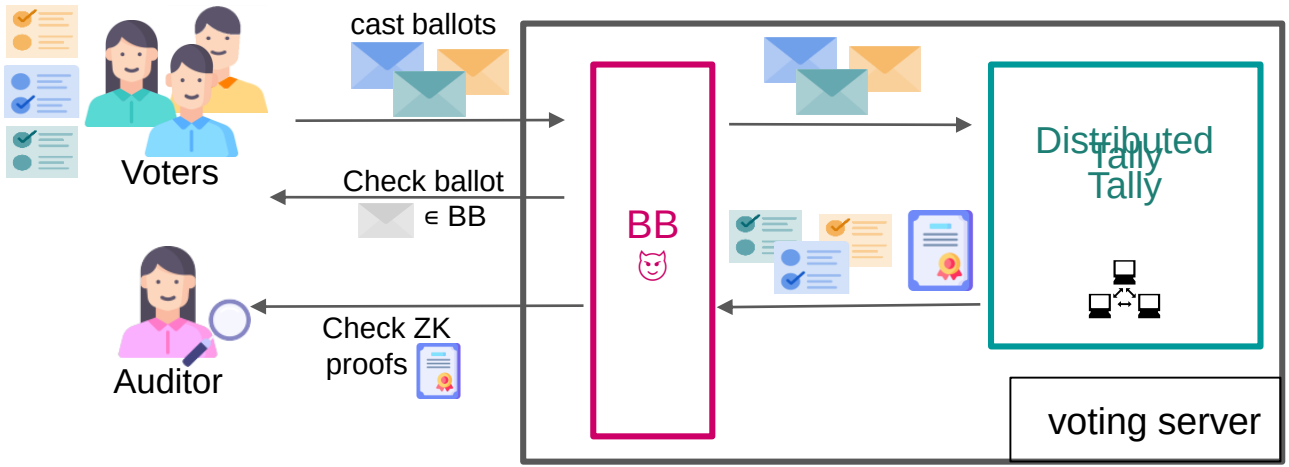
Fixing the Achilles Heel of E-Voting: The Bulletin Board

IEEE Computer Security Foundations Symposium 2021

Lucca Hirschi, Lara Schmid, David Basin

Inria *DFINITY* *ETH Zurich*



Key goals

1. **Vote privacy**. Threat model: 1 out of n tally servers; other tally,voting server ,BB

- Security **proof** of verifiability:
  - assume an **idealized BB** (sometimes implicit): ϵ

stronger

mismatch

2. Weaker **BB requirement** provably sufficient for Verifiability

3. **Design** BB protocol + machine-checked formal proof

- Actual **design**, reference **implementation**, and **deployments**:
  - voting server + BB = centralized server → **single point of trust BB**

1.

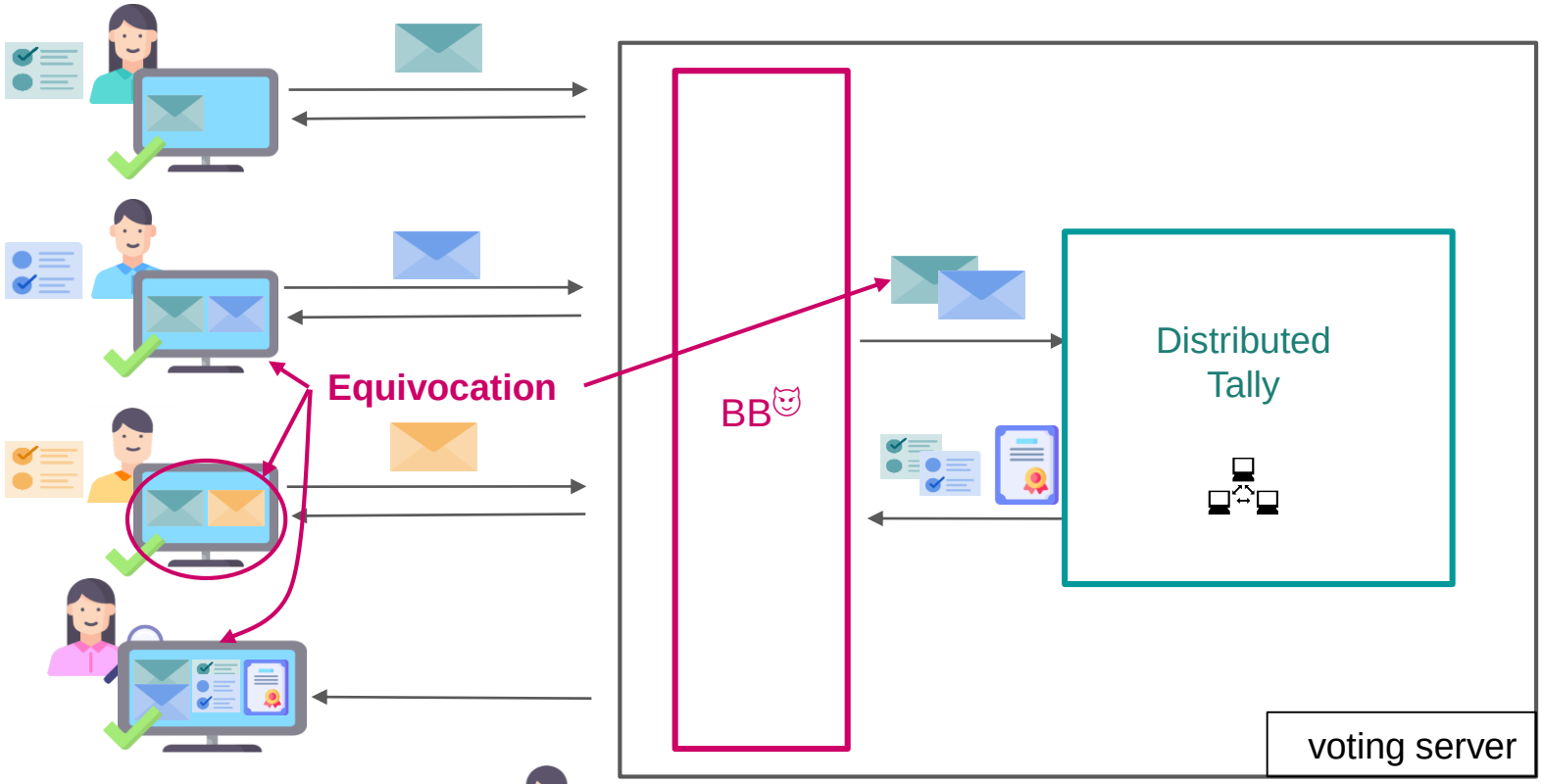
Questions

1. How does BI

Contribution time !

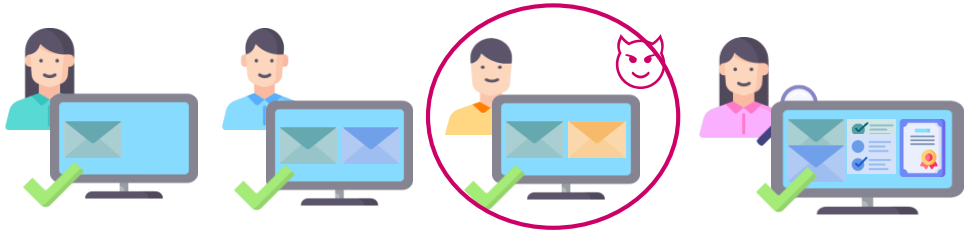
les heel

Attack vector: BB equivocation



Equivocation: BB shows a different content!

Main equivocation attack



😈 **can induce a bias for**

1. **target** a **set** of **voters** who likely do **not** vote for
2. when casts a vote , **remove** from the BB😈, except for
3. proceed honestly with other voters and the auditors

→ Tally has #(target voters) less ballots against

Detection is overwhelmingly unlikely (more in the paper)...

Other equivocation attacks

BB😈 can equivocate on **other data items** towards **different agents**

We found various such equivocation attacks on Civitas and Belenios/Helios:

		Threat Model	Violate	Equivocation (content, reader)	PD?
Civitas	C.1	none (hon. tellers)	IV	possible candidates, voters	✓
	C.2	none (hon. tellers)	IV, UV	(public) credentials, TTs	✓
	C.3	tabulation tellers	IV, UV	ballots on final BB, voters	✗
	C.4	none (hon. tellers)	IV, UV	blocks on final BB, final readers	✓
	C.5	none (hon. tellers)	EV, CR	per-block credentials, TTs	✓
Belenios/Helios	B.1	decryption trustees	IV, UV	ballots on final BB, voters	✗
	B.2	none	IV	ballots on non-final BB, voters	✗

Practical Detection?
i.e., easy fix?
(other than a secure BB)

Fix the mismatch and the e-voting protocols

- Verifiability definitions consider **BB**, we define Verifiability accounting for BB
- New BB requirement: **FA** that is
  - sufficient for verifiability:

    (Verifiability $\wedge$ BB $\vdash$ **FA**) $\Rightarrow$ Verifiability(BB)
  - provably minimal
- New easily deployable BB protocol + machine-checked proof BB $\vdash$ **FA**

One can securely replace the insecure BB (1 server) by our secure BB protocol
$\rightarrow$ effectively weaken trust assumptions:
Verifiability $\rightarrow$ Verifiability

Conclusion

Contributions:

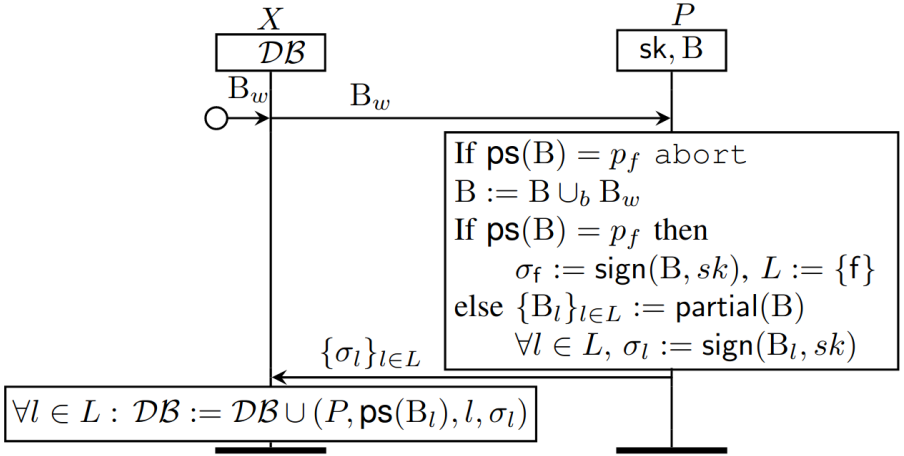
1. Practical attacks on Helios, Belenios, and Civitas
2. New BB requirement that is provably sufficient for verifiability
3. A BB protocol that can be used to weaken trust assumptions & prevent

Future work:

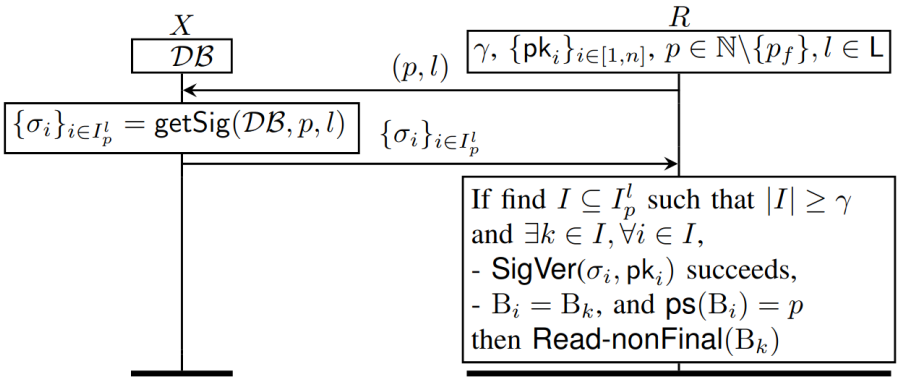
1. Implement our attacks in the wild + user studies
2. Adapt Verifiability to the probabilistic setting (instead of possibilistic)
3. Explore other trade-off threat model versus deployment cost

Backup slides

Our BB protocol design:



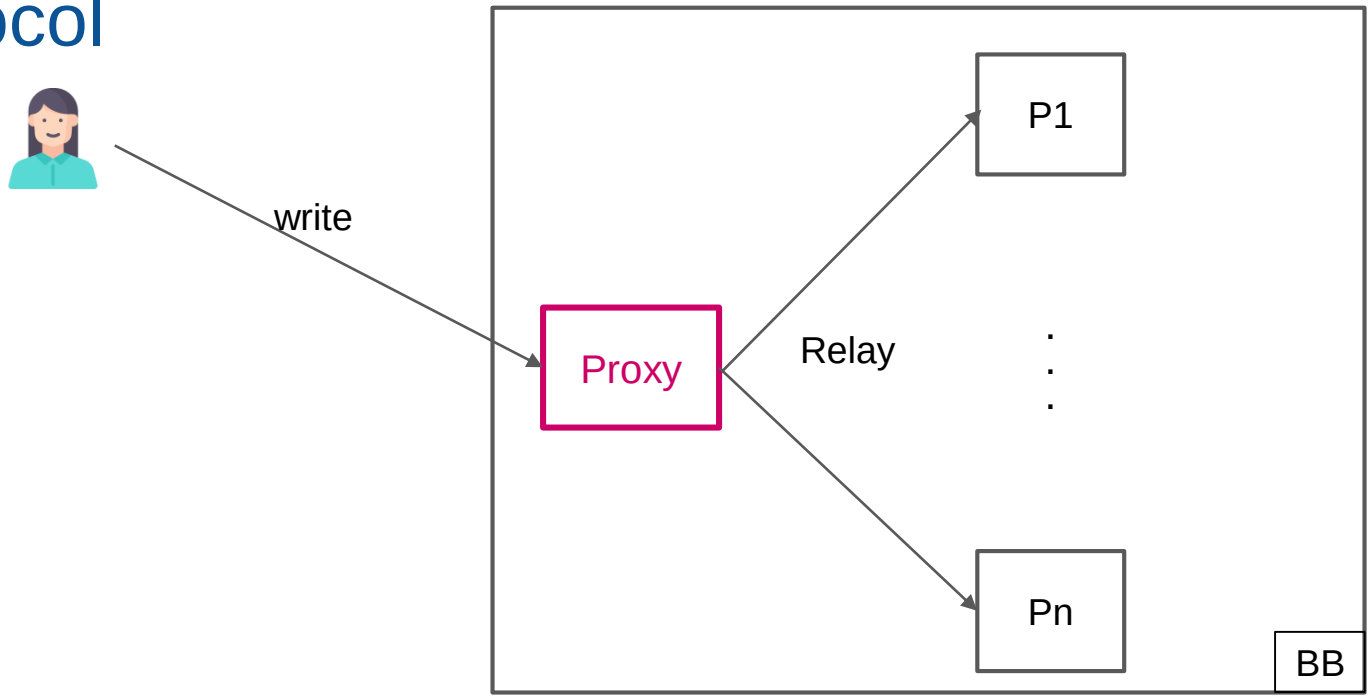
Write to the BB



Read from the BB

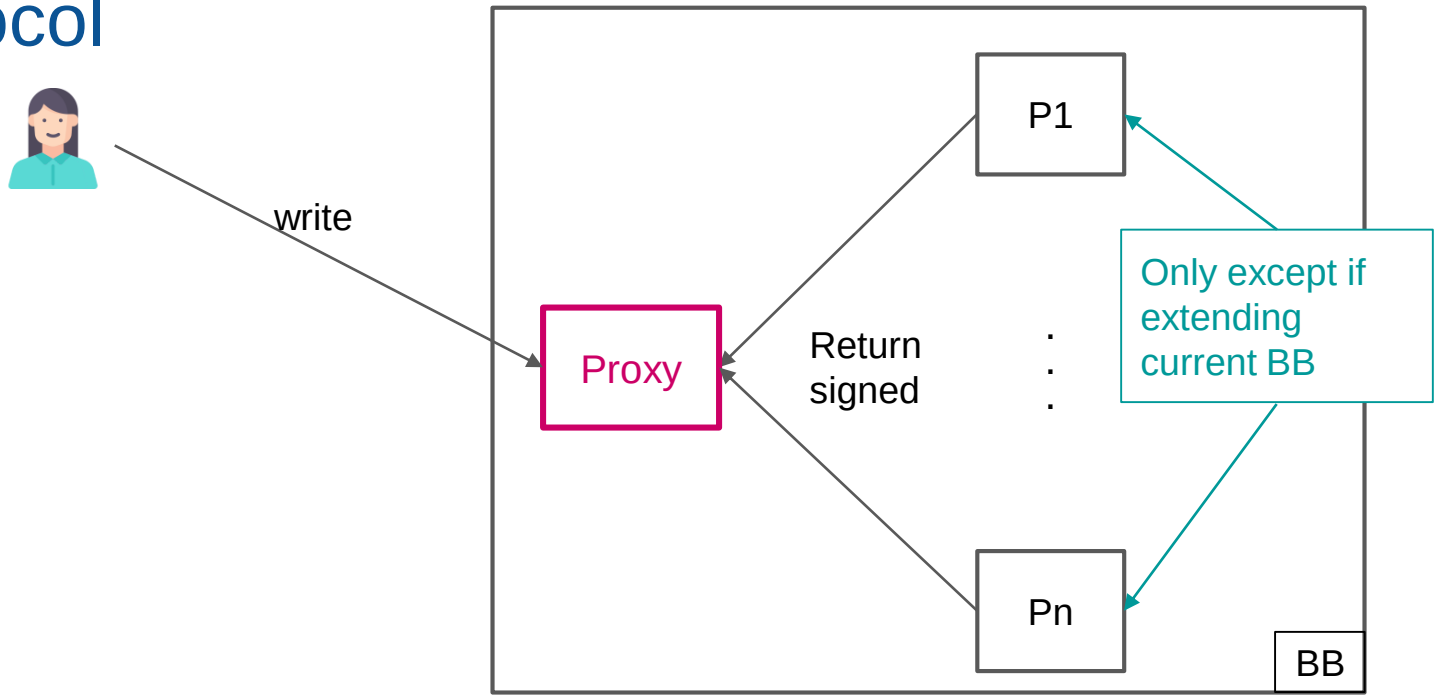
Assuming γ satisfies γ > n−nh / 2 **versus** γ > 2n / 3 (BFT).

BB protocol



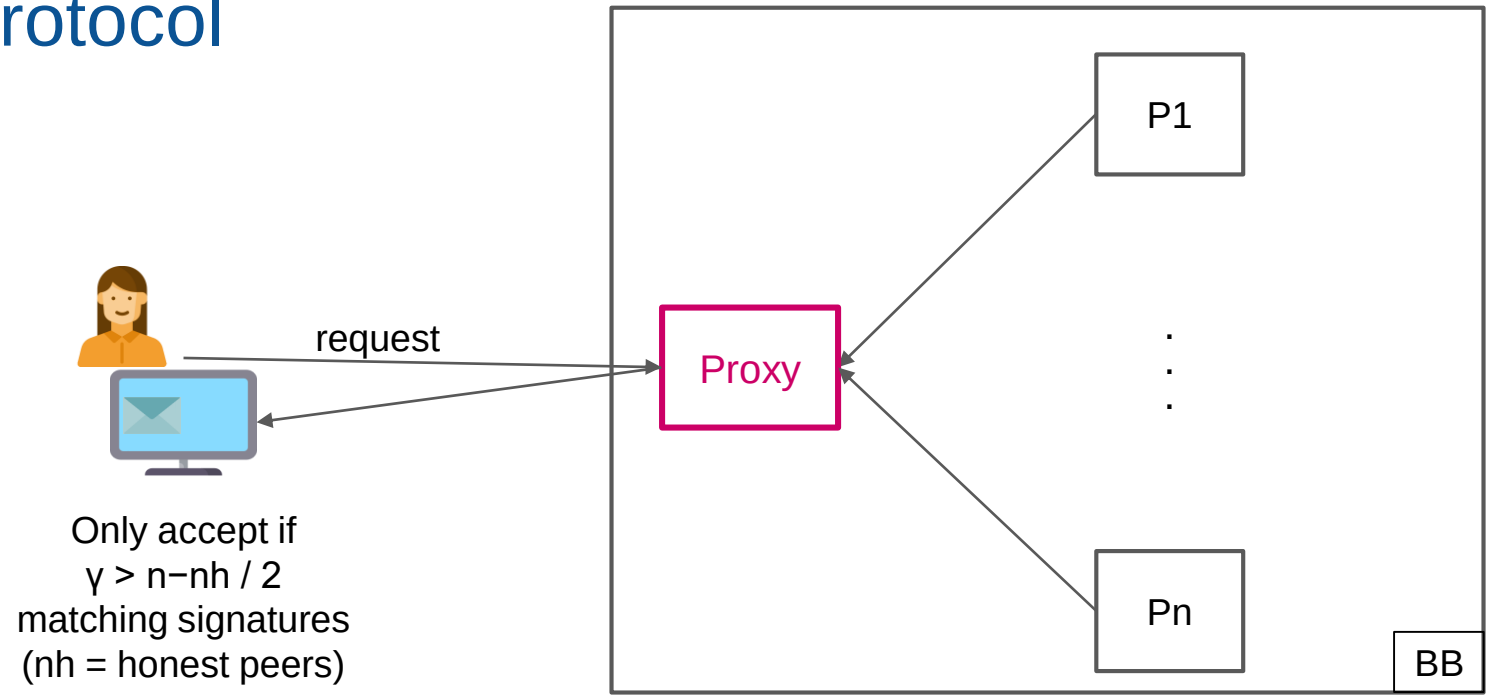
- We were looking for minimal requirements for verifiability (no availability)
  - Readers agree on final state
  - Readers that read in between, can be sure that it will be included in the final state

BB protocol



- We were looking for minimal requirements for verifiability (no availability)
  - Readers agree on final state
  - Readers that read in between, can be sure that it will be included in the final state

BB protocol



- We were looking for minimal requirements for verifiability (no availability)
  - Readers agree on final state
  - Readers that read in between, can be sure that it will be included in the final state

- Permissionless:
  - rely on **economic incentives** ⇒ hard to quantify in the case of elections
  - transaction **costs**
  - often centralized in practice due to **pools**
- Permissioned ledgers: **few distinguished** nodes establish a consensus on data that can be publicly accessed by all other nodes
  - BFT, which requires **strictly stronger trust** assumptions than **our solution**