Stellot

Secure internet voting using distributed networks

Stanislaw Baranski

30.12.2022

Motivation

- Voting is one of the most popular mechanisms for collective decision-making; yet, it’s still something we can not do securely online.
- There are several ways of voting; But, internet voting is the most conventional, cheapest, fastest, and safest (e.g., during the outbreak of COVID-19), and hence, a preferred method for conducting voting.

Internet voting is hard

- Analysis of this area quickly reveals several unsolved issues.
- Secure voting requires four main properties:
  - **Correctness**, all and only eligible votes are counted.
  - **Censorship resistance**, any eligible user that wants to cast a vote can do it.
  - **Privacy**, no one can tell which candidate the voters voted for, or even if they voted at all—preventing preliminary results and guaranteeing freedom of choice.
  - **Coercion resistance**, voters can not prove to anyone how they voted even if they want to—preventing selling votes as there is no way of verifying if they indeed voted on the paid candidate.

Internet voting is hard

- Analysis of this area quickly reveals several unsolved issues.
- Secure voting requires four main properties:
  - **Correctness**, all and only eligible votes are counted.
  - **Censorship resistance**, any eligible user that wants to cast a vote can do it.
  - **Privacy**, no one can tell which candidate the voters voted for, or even if they voted at all—preventing preliminary results and guaranteeing freedom of choice.
  - **Coercion resistance**, voters can not prove to anyone how they voted even if they want to—preventing selling votes as there is no way of verifying if they indeed voted on the paid candidate.

They are hard to satisfy together

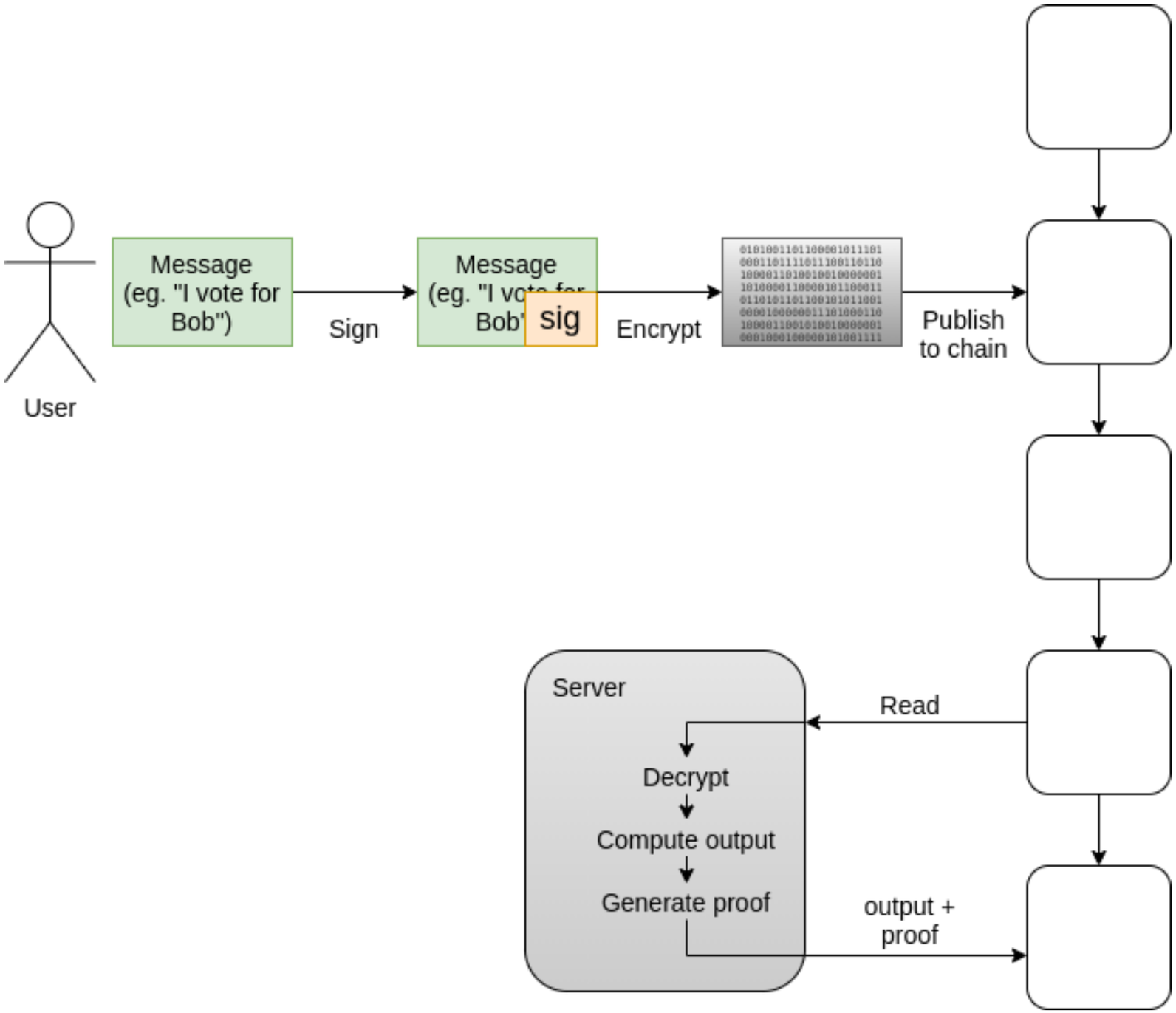
State of the art

- Most of the internet protocols rely on a trusted third party. They differ in what the server can or cannot do. The honesty of the trusted third party determines either anonymity, privacy, or coercion resistance properties.

- Some of them use blockchain for integral and transparent storage (Voatz, Polys, MACI).

- Some are distributing the trusted third party using MPC (Civitas, Swisspost/Scytl, iVoting)

Blockchain + Trusted third party

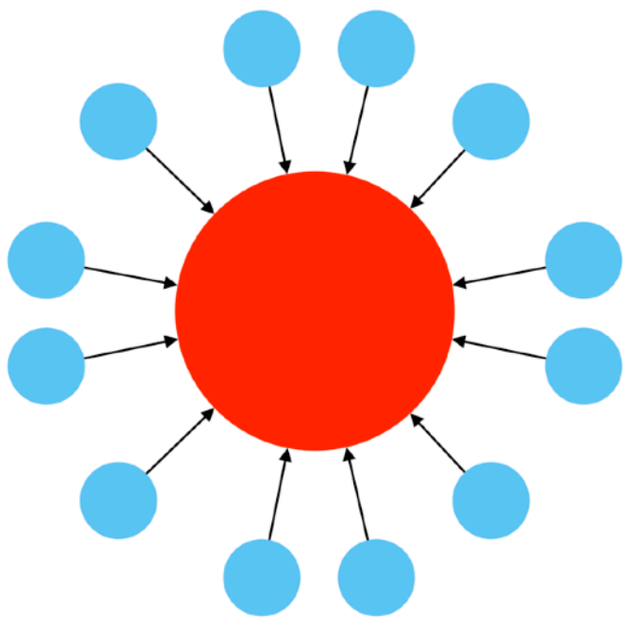


Source: https://vitalik.ca/general/2021/05/25/voting2.html

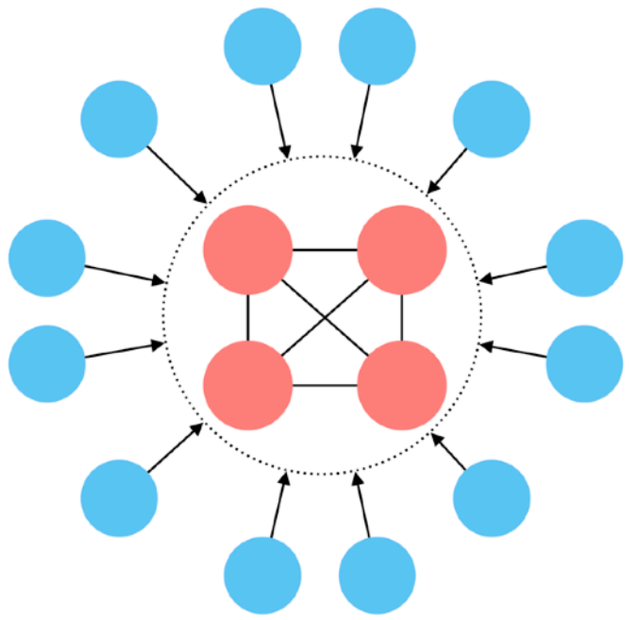
Voter-to-voter trust model

- Most people think about voting in terms of presidential elections. However, voting is used also in small, local votings like housing associations, board members, contests, and all forms of committees.

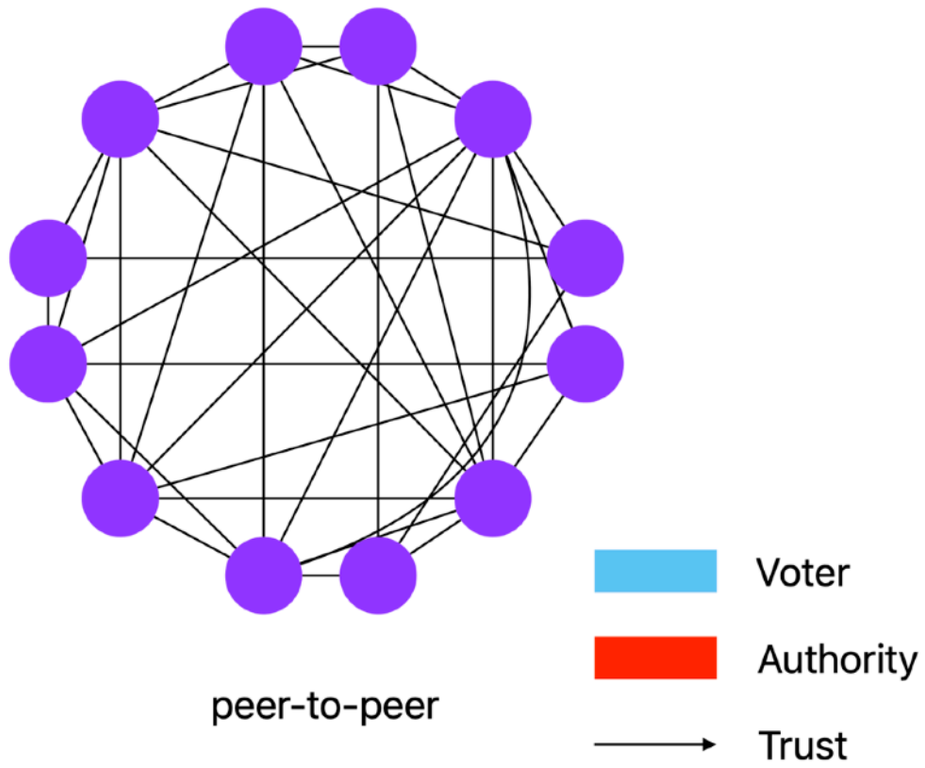
- We want to go even further and conduct the voting on voters' end devices (PC, laptops, or even smartphones) using both blockchain and MPC.



Trusted third party



Distributed third party



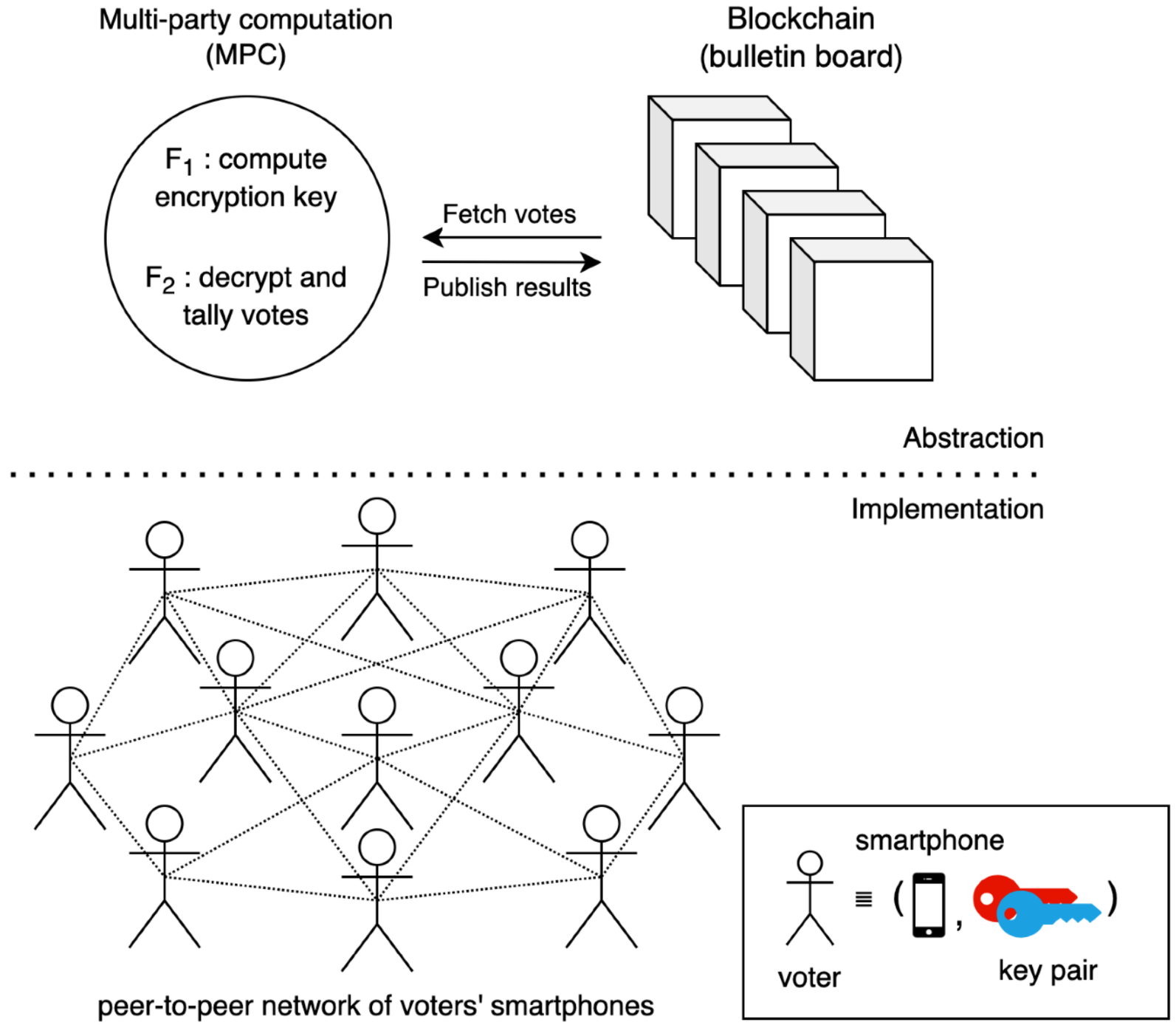
peer-to-peer

Aim

- We **don’t** want to build
  - A large-scale voting system for presidential elections;
  - A voting system for crypto space only;
  - A perfectly secure, coercion-resistant, protocol.
- Rather, we **do** want to build
  - A voting protocol for small to medium size voting like 100 voters;
  - A voting protocol for people, without a crypto background;
  - A decentralised and secure enough protocol that works.

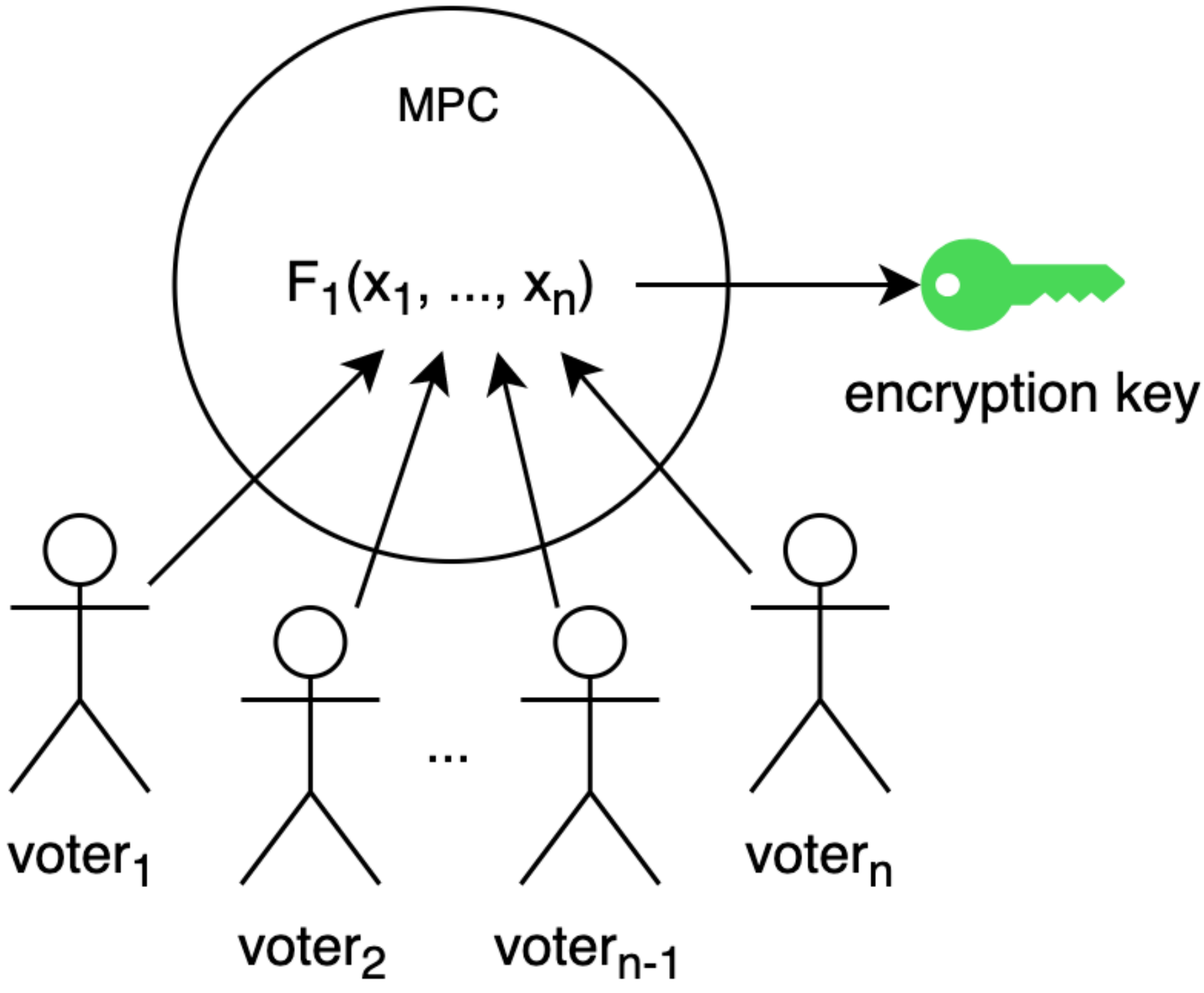
Technical Vision

Architecture



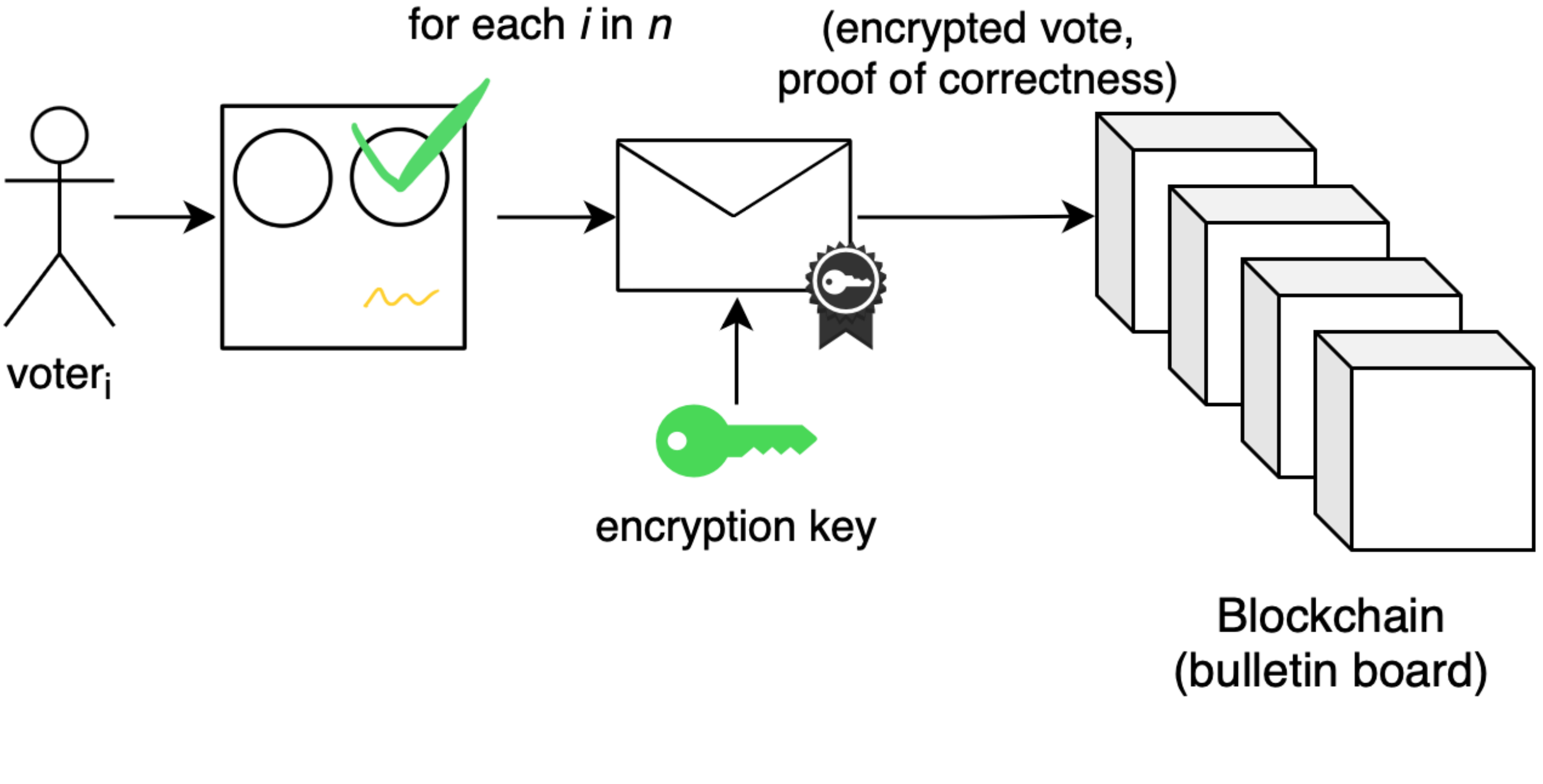
Technical Vision

Setup



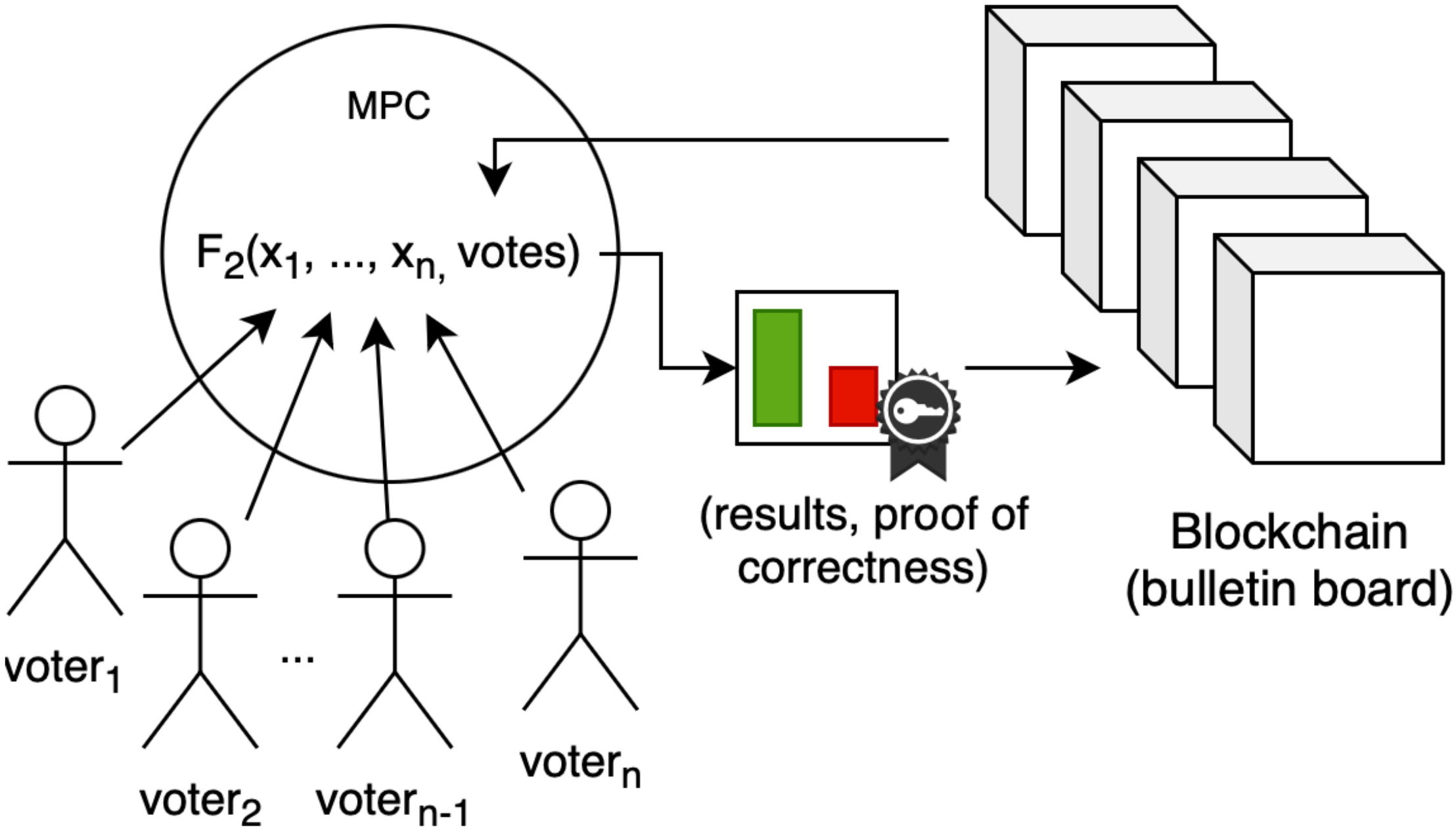
Technical Vision

Voting



Technical Vision

Tally



Objectives

- MPC that supports up to 100 nodes
- Lightweight both in terms of memory and time so it can be executed on voters' laptops or (ideally) smartphones.
- Create voter-to-voter closed PoA blockchain or reuse existing blockchain (and solution stellot.com) for MVP.

Roadmap

- Complete conceptual research on the protocol
    - Find out which privacy scheme is the most lightweight and MPC-friendly. (probably homomorphic encryption using EC-ElGamal, Paillier is too expensive in large MPCs)
    - Find out which MPC protocol suits our trust and liveness requirements. (probably GSZ).
        - Find out which zkSNARK to use on the client side for proving the correctness of encrypted votes. (probably Groth16, or Bulletproofs)
- Implement zkSNARK proof generation for the correctness of encrypted votes
- Implement Distributed Key Generation
- Implement MPC for decrypting and tallying votes
- Implement a smart contract for storing encrypted votes
- Implement p2p network bootstrapping

Team

Grzegorz Barański
Programmer
gbaranski.com

Stanisław Barański
PhD Candidate in Informatics
stan.bar

Lev Soukhanov
PhD Candidate in
Mathematics

Programming ← → Mathematics

Questions?