Evolution of Tezos voting

Stéphane Glondu

SED, Inria Nancy Grand Est

Inria-Nomadic Labs Day
Paris – September 21, 2020

Who am I?

Research engineer at SED Nancy.

Interests:

- Electronic voting (since $\approx$ 2012)
  - With V. Cortier (Pesto) and P. Gaudry (Caramba)
  - Design and implementation of Belenios, an online voting system
  - Transfer towards enterprises (design, evolution)
- Coq ($\approx$ 2007-2012)
- OCaml in Debian (since $\approx$ 2008)

Contract with Nomadic Labs

- Decide what properties are desirable in the context of Tezos
- Design technical specifications for a new voting protocol

Possible properties of voting systems

- Privacy: nobody knows the choices of a voter
- Verifiability
  - Individual: a voter can check that his/her vote has been taken into account
  - Universal: everybody can check that the result is correct
  - Eligibility: only allowed voters vote
- Coercion resistance / receipt freeness
- Accessibility
- ...

Tezos voting: general principles

- Everything is public
- Delegation for voting rights shared with baking rights
- Delegates can propose protocols and vote on them
- Users can change their delegation based on the promise of a delegate

Tezos voting procedure

- Period of 3 months (4 phases of 3 weeks):
  - Proposal phase: approval voting
  - Testing vote: do we test this protocol?
  - Testing phase: fork a test net with the proposed protocol
  - Promotion vote: do we activate the new protocol?
- Votes: yay, nay, pass
- Supermajority (80%) of yays needed
- Adaptative quorum of participation

Possible evolution of Tezos voting

- Dissociate baking rights and voting rights
- Add privacy for those who want it (but no coercion resistance / receipt freeness)
- Keep verifiability properties

Belenios

`https://www.belenios.org/`

- Based on Helios
- Achieves privacy and verifiability (under some trust assumptions)
- Free and open source software
- Public instance, usable by everyone ($\approx$ 500 elections, $\approx$ 20K ballots)

How privacy can be achieved?

- Before the election, a public key with a "virtual" private key is generated by a group of so-called "trustees"
- During the election, ballots are encrypted with this public key
- After the election, ballots are combined into a result that is decrypted

How verifiability can be achieved?

- Each individual ballot stays encrypted and is published on a public bulletin board
- Zero-knowledge proofs are used:
  - to ensure each encrypted ballot is valid
  - to make the decryption of the result verifiable
- Who would be the trustees in Tezos?

Trustees

- Trustees are a group of persons needed
  - before the election (to generate the public key)
  - after the election (to decrypt the result)
  - but can be offline most of the time
- not all of them are necessary for the decryption phase (threshold, but quadratic complexity during setup)
- could be selected with a mechanism similar to bakers; the system would give a reward or a penalty to incentivize good behaviour

Some difficulties

- Trustees must destroy their keys after the election is done
- Dynamic nature of the proposal phase
- Distributed key generation protocol (Pedersen)
- Size of ballots
- Weights