Dolev-Reischuk: Any deterministic protocol requires $\geq O(n^2)$ communication.

Idea: One party p who does not receive any message; adv. who can corrupt all the senders of messages to p.

Q: Can we obtain a sub-quadratic communication protocol, if we assume randomization?

Adaptivity of the adversary:

Static: Adversary corrupts upto t parties at the beginning of the protocol.

Adaptive: Adversary corrupts parties during execution.

- A party can be instantaneously corrupted at the start of a round.

Dolev-Strong: Adaptively secure. n^2t

O(1) : static / adaptive;

Dfinity / ICC: Can delay commits. } $O(n^2)$

Sync HS: "

PBFT: Can delay commits

HotStuff: Can delay commits. $O(n^2)$

Mobile:

ICC:

{ All parties are ranked; everyone knows everyone's rank.

Bitcoin: adaptively-secure.

"a miner useful now is not useful later"

"player-replaceable".

Resource-constrained:

proof-of-stake: more money ⇒ more voting power.

⇒ adversary has minority stake.

Work:

↳ harder: mine a block; poisson distribution

↳ easier: adversary is contrained by resources.

Stake:

↳ Can require all parties to speak at the same time.

↳

2015 - 2017

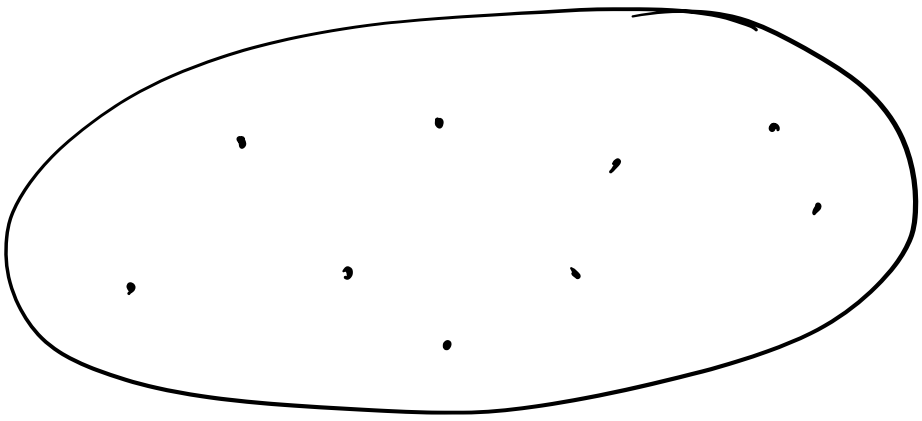
Longest chain paradigm: attack surface higher.

BFT: unexplored.

Algorand: "player-replaceability"

↳ Adaptively-secure.

↳ Subquadratic protocol. (subject DR).



↳ Elect at random a set of k parties. "committee".

↳ Run consensus among committee members. $O(k^2) \Rightarrow O(n\,\mathrm{poly}(k))$ (adaptive)

↳ Announce results to rest of the parties. $O(n\,\mathrm{poly}(k))$.

$t \leq \frac{n}{2}$ $n/3$

$O(n \text{ poly}(k))$.

$k << t$

$k << n$

& $f(i, r)$

- A party will know whether they are in the committee
- Once they announce, everyone can verify this fact

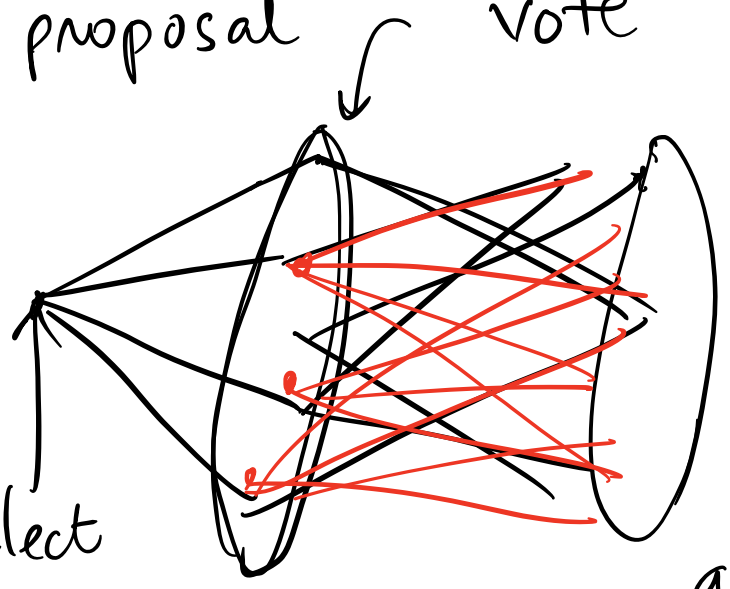
Verifiable Random Functions:

$VRF_{SK_i}(P_i, r)$:

↳ Verifiability

↳ Random function:

↳ Unique.



Only committee members speak.

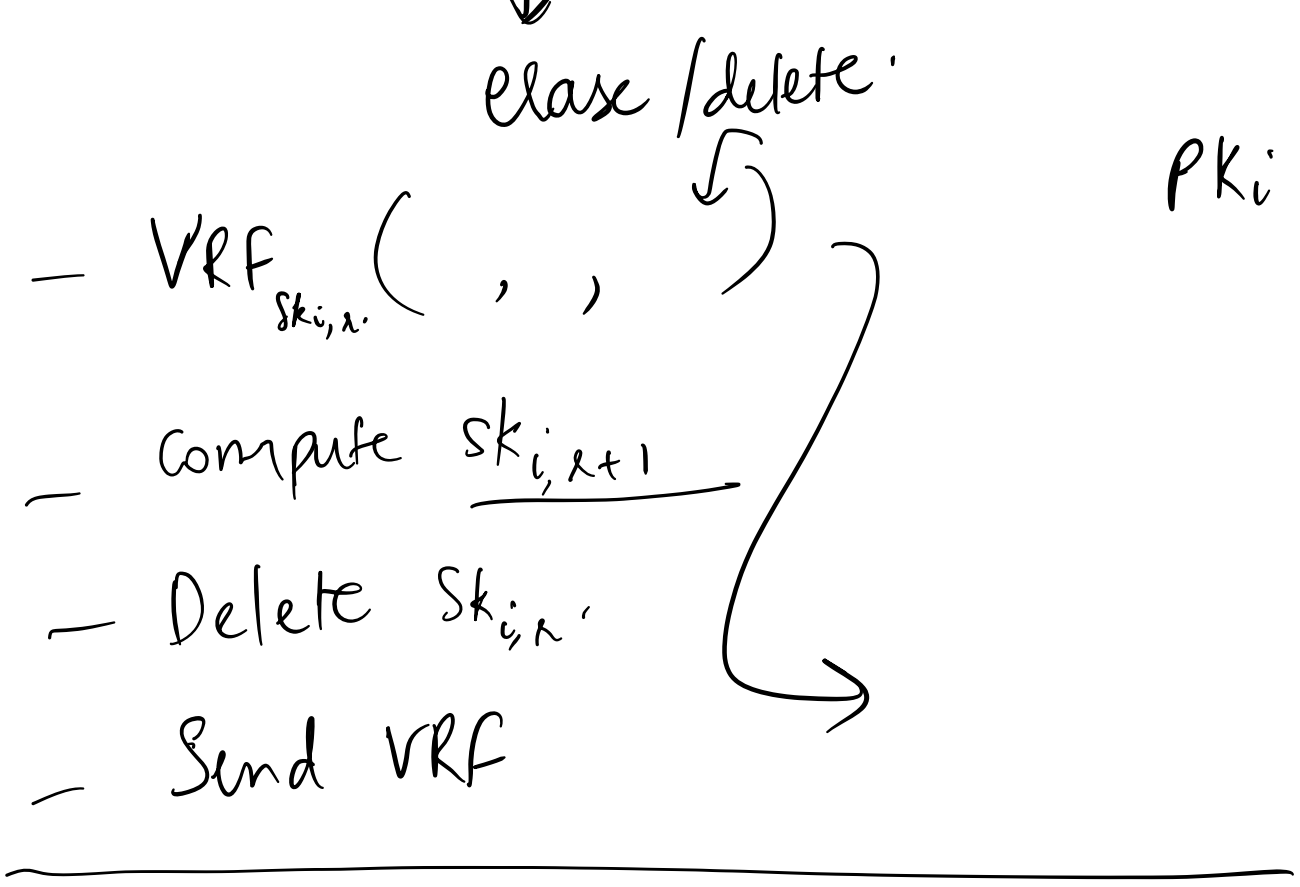
$$(1+\delta)k \le k \le (1-\delta)k$$

$$0.9k \le k \le 1.1k.$$

$k << n$

Key evolving signatures.

PK_i $\quad sk_{i,r} \Rightarrow sk_{i,r+1}$


erase / delete.
PKi
- $VRF_{sk_{i,x}}(\ ,\ )$
- Compute $sk_{i,x+1}$
- Delete $sk_{i,x}$.
- Send VRF