Implémenter une réparation et amélioration du consensus [BKL+23] : <https://eprint.iacr.org/2023/689.pdf>

En particulier la 1ère composante Jolteon, protocole partiellement synchrone :

expliquer protocole Jolteon et prouver sa safety (impossibilité d’avoir 2 blocs de chaînes concurrentes commited/finallized, preuve à refaire p.21-22-23 de <https://drive.google.com/file/d/11ov_j02DDNCRp0NdNQ_3Uje1lldV9bxc/view?usp=drive_link>

preuve actuelle suffisante pour nous M. Rambaud s’occupe de prouver l’amélioration (la preuve de l’amélioration dans un cas simplifié (consensus sur un bloc) page 1-2 dans 'brief announcement: malicious security comes for free in consensus with leaders')

2ème composante (uniquement pour les scénarios ou le réseau devient asynchrone) : 2-phase-vaba qui est vulnérable et qu’il faut remplacer par 2-PAC

implémentation en Rust ou possiblement en Go (plus flexible) :

nouvelle plateforme en Go : [DZ23] <https://github.com/fififish/waterbear>

par contre en construisant sur la base d’un consensus plus éloigné : [ZDZZ23] <https://eprint.iacr.org/2022/021.pdf>

### 

### AMÉLIORATION ET IMPLÉMENTATION DU PROTOCOLE DE CONSENSUS JOLTEON (CELUI DE L’EX-FACEBOOK DIEM) ”

Personnes intéressées:

groupe [nathan.rouille@telecom-paris.fr](mailto:nathan.rouille@telecom-paris.fr) [titouan.duhaze@telecom-paris.fr](mailto:titouan.duhaze@telecom-paris.fr) [adam.chgour@telecom-paris.fr](mailto:adam.chgour@telecom-paris.fr) [rayane.dakhlaoui@telecom-paris.fr](mailto:rayane.dakhlaoui@telecom-paris.fr) [aymane.hamdaoui@telecom-paris.fr](mailto:aymane.hamdaoui@telecom-paris.fr)

Objectifs du rendu Bibliographie: commencer par expliquer la preuve de safety de Jolteon <https://arxiv.org/abs/2106.10362>

Puis décrire l’idée de l’amélioration <https://dl.acm.org/doi/10.1145/3465084.3467953> (décrite sur une demi page, à votre disposition pour des documents plus détaillés)

Objectif du rendu “Implémentation”: implémenter cette amélioration dans l’implémentation pédagogique existante <https://github.com/asonnino/hotstuff> “Note to readers: This codebase is useful to get started with BFT consensus and as baseline when designing your own protocols”

(ou dans l’implémentation de production <https://github.com/diem/diem> , si vous voulez en faire votre métier). La modification “cryptographique” à faire consistera à remplacer les concaténation de signatures par des multi-signatures, par exemple celles de la librairie de Web3 <https://github.com/w3f/bls/tree/skalman-double-puclic-key-verify>,

2PAC, par exemple en partant de l’implémentation <https://github.com/danielxiangzl/Ditto/blob/main/consensus/src/fallback.rs> , en et en faisant les modifications nécessaires.

Vérifier s’il marche avec les scripts des auteurs pour lancer des simulations<https://github.com/danielxiangzl/Ditto>

Travail de recherche pour passer de 18 à 20: mesurer sa vitesse contre celle de Jolteon (i) avec un leader sur deux trop lent mais des temps de communication homogènes (ii) idem, avec un leader sur 5 trop lent mais des temps de communication quadruplés entre certains participants.

Tracer les graphes des latency vs throughput mesurées.