ONet Implementation of Gossip-based Signature Aggregation

Student:
Elias Manuel Poroma Wiri
SCIPER 294650

Responsible:
Prof. Bryan Ford
EPFL / DEDIS

Supervisor:
Gaylor Bosson
EPFL / DEDIS

EPFL, 10.01.2020
Decentralized cosigning protocols have the main purpose of collecting digital signatures of a message from many peers.

There are two existing implementations:
- BLS CoSi which uses trees.
- Gossip protocol.
Introduction

Project has 2 parts:

- Develop and compare alternative implementations of gossip-based aggregation. Be more efficient.
- Add an implementation inside Cothority's ONet library.
Background - Existing implementations

BLS CoSi

- Arranges participating nodes in a tree of depth 3.
- Rumors propagated following the tree structure.
- Earlier aggregation is done in intermediate nodes.
- Root does final aggregation of multi-signatures received from its children, who aggregated the signatures they received from their children.
Background - Existing implementations
Existing Gossip aggregation protocol

- Two variations:
  - Simple aggregation at the root after gossiping and collecting enough signatures.
  - During gossiping, aggregation is done using a binary tree.

- Rumor messages are push-messages only. Randomly selected recipients of rumor.
• Part 1 - Alternative Gossip-based aggregation implementations
• Part 2 - Hybrid protocol in ONet
Design and implementation

Mask gossip
Design and implementation

Mask gossip with early aggregation
Evaluation and results

Comparison of protocol duration ($n = 36$)

- Existing Gossip Aggregation
- Mask
- Mask Aggregation
Evaluation and results

Comparison of message count (n = 36)

- Existing Gossip Aggregation
- Mask
- Mask Aggregation
Evaluation and results

Comparison of data transferred ($n = 36$)

- Existing Gossip Aggregation
- Mask
- Mask Aggregation

Data sent per active node (kB)

Failing nodes
• Alternative Gossip-based aggregation implementations
• Part 2 - Hybrid protocol in ONet
Hybrid implementation in ONet

- Hybrid to get the best of both worlds, each propagation round has 2 parts:
  1. Using a $n$-ary tree of depth 2
  2. If signatures are missing after some time, send a gossip rumor among the nodes needed.

- Implementation of HybridRumor is done in the overlay layer of ONet.
Design and implementation

Hybrid implementation in ONet

- Evaluation is done with Couthority simulations.
- For cosigning, created a protocol that runs many rounds of HybridRumors until enough signatures are collected, then aggregates them.
Design and implementation

Hybrid implementation in ONet

Tree created for round

A
B
C
D
E

Tree part
A, B
A, B, C
A, B, C, D

Gossip part
A, B, C, D, E

A
B
C
D
E
Evaluation and results

Comparison of protocol duration ($n = 36$)

- Existing Gossip Aggregation
- ONet Hybrid Protocol

- time until signature (sec)

- failing nodes

0 1 2 3 4 5 6 7 8 9 10 11
Evaluation and results

Comparison of message count ($n = 36$)

- Existing Gossip Aggregation
- ONet Hybrid Protocol
Evaluation and results

Comparison of data transferred ($n = 36$)

- Existing Gossip Aggregation
- ONet Hybrid Protocol
Conclusions

- Mask gossip protocol had slightly better performance among the gossip-based aggregation implementations.

- Hybrid ONet implementation has a significant improvement in efficiency compared to existing gossip-based implementation.

- Future work: adapt to the properties of the network, use homomorphic subtraction of signatures to do earlier aggregation.
Questions?