Append-only Bulletin Board

Severin Hauser

PhD Workshop, Tarragona, April 25th, 2016

Content

- Introduction
- Past Work
- Current Work
- Summary and Outlook

Introduction

Work Overview

- Defining operations and wording
- Implementation
- UniVote2
- Understanding the problems behind append-only
- What are the trust assumptions
- Who to improve these assumptions

Vocabulary

- Properties - A Board can have some properties e.g. interlinked
- Message - Is posted to the bulletin board
- Attribute - Is added to a posted message to ensure a board property
- Post - A post represents the message and all it's attributes

Append-only

- No posted message *m* can be deleted
- No posted message *m* can be altered
- $\mathcal{P}_{\langle t \rangle} \subseteq \mathcal{P}_{\langle t+1 \rangle}$

Properties

- Prevent board flooding
- Give the user a receipt
- Create a hash chain over all messages.
- etc.

Past Work

Post

- Either the author or the board can add an attribute to m
  - list of author attributes α
  - list of board attributes β
- The post $p = (m, \alpha, \beta)$ is stored in $\mathcal{P}$
- For the author to gain full knowledge of the post, β must be returned.

$$\mathrm{Post}(m, \alpha) : \beta$$

Get

- Limit the result R by introducing query $Q \subseteq \mathcal{M} \times \mathcal{A} \times \mathcal{B}$
  - $R = \{(m, \alpha, \beta) \in \mathcal{P} : (m, \alpha, \beta) \in Q\} \subseteq \mathcal{P}$
- The board can add result attributes γ to R

$$\mathrm{Get}(Q) : R, \gamma$$

Properties

- Post properties
  - Adds an attribute to either α or β
- Get properties
  - Adds an attribute to γ
  - is added by the bulletin board
- Further properties
  - Adds additional operations to the board. Does not require attributes

Current Work

Trust assumptions

- The board does not delete published messages $\mathcal{P}_{\langle t \rangle} \subseteq \mathcal{P}_{\langle t+1 \rangle}$
- The board delivers always the complete set $\mathcal{P}$ on request.
- The board adds every valid message *m* to $\mathcal{P}$.
  $\mathit{valid}(m, \alpha) = \mathit{true} \rightarrow p \in \mathcal{P}$

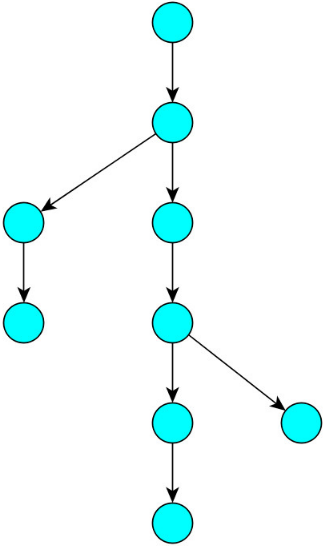
Robust PBB

- Assumption: At least t out of n are honest.
- If the post and get operations involve all n all other assumptions are true
- Has performance limits with some properties

Interlinked(hash-chain)

- Does not replace the assumptions but provides a degree of detection for misbehaviour
- This is true for the single and robust variant
- Its enough to detect a conflict

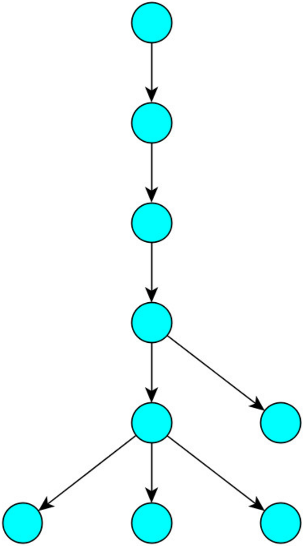
Interlinked cont.



Interlinked cont.

- Probability of conflicting hash values
  $1 - (2 * \sum_x depth(x)/n * (n-1))$
- Branches with size 1
- As late as possible

Interlinked cont.



Interlinked cont.

- Works best if views of $\mathcal{P}$ don't get shared
- View can be represented by the hash value of the last node
- Either use broadcast channels (multiple)
- For a single board something like an auditor-network might make sense

Auditor-network

- A network of n auditors with at least t honest
- The board need to send them every hash entry
- Elevates the assumptions for deletion and full view to t out of n as long as every operation is validated with the auditor-network

Summary and Outlook

Outlook

- Further work on the part around assumptions and interlinked
- Find differences in the broadcast channels(BitCoin, Twitter, GitHub)
- Is there a "robust" way for accepting valid messages without the board being robust?

Questions?

http://e-voting.bfh.ch

severin.hauser@bfh.ch

Sectioned

- Allows to separate unrelated messages into different sections
  - e.g. the data of various elections
- User attribute $s \in \mathcal{S}$ must be provided

Grouped

- Messages are organized into groups
- Messages in the same group are usually similar
- user attribute $g \in \mathcal{G}$ must be provided
- $\mathcal{G}$ is the same for every section s.

Typed

- Depends on Grouped
- Defines for g_i the set of correct messages $\mathcal{M}_i \subseteq \mathcal{M}$
- Does not add an attribute

Certified Posting

- With this property every user receives after a successful post a receipt from the board
- Board attribute $S_p = Sign_{sk_{BB}}(m, \alpha, \beta_I)$ is added by the bulletin board where
  - sk_{BB} is the secret key of the bulletin board
  - β_I is the sublist of all board attributes before S_p

Certified Reading

- This is a get property
- With this property the bulletin board commits to every result R
- Result attribute $S_Q = Sign_{sk_{BB}}(Q, R, \gamma_I)$ is added by the bulletin board
    - γ_I is the sublist of γ added before S_Q

Notifying

- This property belongs to further properties
- It allows an entity e to register for a Query Q on the bulletin board
- If a post full fills Q, e is notified.
- This property results in the following two operations:
    - Register(e, Q) : c
      Where Q represents the query for the messages the entity is interested in and c a return code, which can be used to unregister.
    - Unregister(c) : -
      By providing his/her return code c, one can unregister and will not receive any further notification.