Bern University of Applied Sciences

UniBoard

Properties and Operations

Severin Hauser

E-Voting Seminar, Bern (Switzerland), May 21th, 2014

Content

- Bulletin Board
- Basic Operations
- Properties
- Implementation
- Summary and Outlook

Bern University
of Applied Sciences

Bulletin Board

Bulletin Board Current State

- Allows to post messages
- Everyone can retrieve all posted messages from the bulletin board
- Append-only bulletin board $\rightarrow$ no message can be deleted

Roles

In general every user can incorporate these tow roles:

- author - posts a message on the board
- reader - reads messages that were retrieved from the board

Basic Operations

Simple Operations

- Simple bulletin board with no properties
  - $\mathrm{Post}(m) \rightarrow$ post a message $m \in \mathcal{M}$, where $\mathcal{M}$ is the set of possible messages the board can accept
  - $\mathrm{Get}() : R \rightarrow$ retrieve the current state of the board as result R
- Properties in general are achieved by adding attributes to either m or R.

Post

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

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

Post cont.

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

- $m \in \mathcal{M}$
- User attributes α
  - $\alpha = (\alpha_1, \ldots, \alpha_u) \in \mathcal{A}$
  - $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_u$
- Board attributes β
  - $\beta = (\beta_1, \ldots, \beta_v) \in \mathcal{B}$
  - $\mathcal{B} = \mathcal{B}_1 \times \cdots \times \mathcal{B}_v$

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

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

Notations

- Query Q might also be denoted:
  - for a single value $\alpha_i \in \mathcal{A}_i$ as $\langle \alpha_i \rangle$
  - for a subset $A_i \subseteq \mathcal{A}_i$ as $\langle A_i \rangle$
  - for multiple values or subsets as $\langle \alpha_i, \alpha_j \rangle = \langle \alpha_i \rangle \cap \langle \alpha_j \rangle$ or $\langle A_i, A_j \rangle = \langle A_i \rangle \cap \langle A_j \rangle$
- Result R might also be denoted as $\mathcal{P}_Q$
- A sublist of α is denoted as α_I where $I \subseteq \{1, \ldots, u\}$ same can be done for β and γ

Basic Operations

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

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

Properties

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

Sectioned

- Allows to separate unrelated messages into different sections
  - e.g. the data of each election in UniVote
- User attribute $s \in \mathcal{S}$ must be provided
- Reject m if $s \notin \mathcal{S}$

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.
- Reject m if $g \notin \mathcal{G}$

Typed

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

Access-Controlled

- Restricts the post operation to authorized users
- The bulletin board defines a set of authorized keys $\mathcal{K}$
  - static - $\mathcal{K}$ is publicly known and can not be changed
  - dynamic - $\mathcal{K} = K(\mathcal{P}, \alpha, t)$, where t is the current time
- The user attribute $S_m = Sign_{sk_U}(m, \alpha_I)$ must be provided, where
  - sk_U is the secret key of the user U
  - α_I the sublist of all attributes added before S_m
- The user attribute $vk \in \mathcal{K}$ must be provided, which is the public key to the secret key sk_U

Ordered

- This property ensures the total order of all published posts $\mathcal{P}_{\langle s \rangle}$ for section s
- Board attribute $n \in \mathbb{N}$ is added by the bulletin board
- When storing post p the order n is defined as $|\mathcal{P}_{\langle s \rangle}| + 1$

Chronological

- This property adds to every incoming message m a current timestamp t
- Board attribute $t \in \mathcal{T}$ is added by the bulletin board

Append-Only

- This property ensures that no message can be deleted from the board.
- Therefore for all times $\mathcal{P}_{\langle t \rangle} \subseteq \mathcal{P}_{\langle t+1 \rangle}$
- Possible Solution:
  - Create a hash chain H_s over $\mathcal{P}_{\langle s \rangle}$
  - Board attribute h_i is added by the bulletin board, where $h_i = Hash(h_{i-1}, m_i, \alpha, \beta_I)$

Certified Posting

- With this property every user receives after a successful post a receipt from the board
- Board attribute $S_p = Sign_{sk_{BB}}(m, \alpha, \beta_l)$ is added by the bulletin board where
  - sk_{BB} is the secret key of the bulletin board
  - β_l 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 that is returned at time t
- Result attribute t is added by the bulletin board
- 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.

Summarization for UniVote/UniCert

- Post properties
  - User attributes
    - Sectioned s
    - Grouped g
    - Access-controlled S_m and vk_U
  - Board attributes
    - Ordered n
    - Chronological t
    - Append-only h
    - Certified post S_p
- Get properties
  - Certified read t and S_Q

UniVote/UniCert Post

$\mathrm{Post}(m, (s, g, S_m, vk_U)) : (n, t, h, S_p)$

1. Check that section s exists.
2. Check that group g exists.
3. Check if message m is of the type of group g.
4. Check that S_m is a correct signature of $(m, (s, g))$ and belongs to vk_U.
5. Check that vk_U is authorized by checking $\mathcal{K}$.
6. Define the total order n of the message in s .
7. Add the current time t to the message.
8. Create the hash h for the message.
9. Create the signature for the post $S_p = Sign_{sk_{BB}}(m, (s, g, S_m, vk_U), (n, t, h))$
10. Send notifications for this message if necessary.
11. Save the post.
12. Return (n, t, h, S_p) to the user.

UniVote/UniCert Get

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

1. Search all messages $\mathcal{P}_Q$ satisfying Q.
2. Set timestamp t to the current time.
3. Create the signature S_Q.
4. return the result to the reader.

Implementation

Implementation Interface

```
1 public Attributes post(byte[] message, Attributes
      alpha, Attributes beta);
2 public ResultContainer get(Query query);
```

- Internal only
- No Exception thrown
- Error or Rejects are made known over the Attributes

Summary and Outlook

Outlook

- User and Roles - Naming
- Simplified notation for post
- Do the implementation

Questions?

http://e-voting.bfh.ch

https://github.com/bfh-evg/uniboard