A Rule-based Policy Language for Selective Trust Propagation in Social Networks

Stefano Braghin      Elena Ferrari      Alberto Trombetta

Dipartimento di Informatica e Comunicazione
Università degli Studi dell’Insubria
Varese (Italy)

ACM SIGMOD Workshop on Databases and Social Networks
June 12th 2011
Overview

1. Motivating Scenario

2. Rumor Propagation and Related Issues

3. Proposed Solution
   - Introduction to On-line Social Network Access Control
     - Virtual Relationships
   - Rule Language
   - Rule Management
   - Enforcement

4. Conclusions and Future Work
Motivating Scenario

- On-line users interact in order to exchange resources
- In order to do that users need to trust each other
- If Bob gives a book to Alice then they both agree the other peer is reliable for the transaction
- Such information may be useful to Alice’s friend Charlie to obtain the same book from Bob
- Equally, it may be useful to Edward, a contact of Bob, when Alice asks him another book.
Motivating Scenario

- On-line users interact in order to exchange resources
- In order to do that users need to trust each other
- If Bob gives a book to Alice then they both agree the other peer is reliable for the transaction
- Such information may be useful to Alice's friend Charlie to obtain the same book from Bob
- Equally, it may be useful to Edward, a contact of Bob, when Alice asks him another book.
Motivating Scenario

- On-line users interact in order to exchange resources
- In order to do that users need to trust each other
- If Bob gives a book to Alice then they both agree the other peer is reliable for the transaction
- Such information may be useful to Alice’s friend Charlie to obtain the same book from Bob
- Equally, it may be useful to Edward, a contact of Bob, when Alice asks him another book.

Alice
FriendOf
Charlie
Edward
Bob
FriendOf
Motivating Scenario

- On-line users interact in order to exchange resources
- In order to do that users need to trust each other
- If Bob gives a book to Alice then they both agree the other peer is reliable for the transaction
- Such information may be useful to Alice’s friend Charlie to obtain the same book from Bob
- Equally, it may be useful to Edward, a contact of Bob, when Alice asks him another book.
Motivating Scenario

- On-line users interact in order to exchange resources.
- In order to do that users need to trust each other.
- If Bob gives a book to Alice then they both agree the other peer is reliable for the transaction.
- Such information may be useful to Alice’s friend Charlie to obtain the same book from Bob.
- Equally, it may be useful to Edward, a contact of Bob, when Alice asks him another book.
# Rumor Propagation

## Advantages
- Enhances the collaboration among users of the network
- Facilitates the identification of reliable users
- Facilitates the identification of un-reliable users

## Issues
- Propagation of unwanted information
- Propagation of sensitive information
- Possibility to infer properties of the users

## Solution
- Allow the user to control how propagation is performed
Rumor propagation

**Advantages**
- Enhances the collaboration among users of the network
- Facilitates the identification of reliable users
- Facilitates the identification of unreliable users

**Issues**
- Propagation of unwanted information
- Propagation of sensitive information
- Possibility to infer properties of the users

**Solution**
- Allow the user to control how propagation is performed
Rumor propagation

Advantages

- Enhances the collaboration among users of the network
- Facilitates the identification of reliable users
- Facilitates the identification of un-reliable users

Issues

- Propagation of unwanted information
- Propagation of sensitive information
- Possibility to infer properties of the users

Solution

- Allow the user to control how propagation is performed
Proposed Solution

- Introduction of rumor propagation in an existing access control framework for On-line Social Networks
- Definition of a rule-based policy definition language
- Definition of rule enforcement mechanism
Introduction to On-line Social Network Access Control

Model definition

- OSN $SN$ is a directed labeled graph $(V_{SN}, E_{SN}, RT_{SN}, \phi_{SN})$:
  - nodes $V_{SN}$
  - supported relationship types $RT_{SN}$
  - edges $E_{SN} \subseteq V_{SN} \times V_{SN} \times RT_{SN}$
  - $\phi_{SN} : E_{SN} \rightarrow [0, 1]$, a function assigning to each edge $E_{SN}$ a trust level $t \in [0, 1]
Introduction to On-line Social Network Access Control

Model definition

OSN $SN$ is a directed labeled graph $(V_{SN}, E_{SN}, RT_{SN}, \phi_{SN})$:

- nodes $V_{SN}$
- supported relationship types $RT_{SN}$
- edges $E_{SN} \subseteq V_{SN} \times V_{SN} \times RT_{SN}$
- $\phi_{SN} : E_{SN} \rightarrow [0, 1]$, a function assigning to each edge $E_{SN}$ a trust level $t \in [0, 1]$
OSN $SN$ is a directed labeled graph $(V_{SN}, E_{SN}, RT_{SN}, \phi_{SN})$:

- nodes $V_{SN}$
- supported relationship types $RT_{SN}$
- edges $E_{SN} \subseteq V_{SN} \times V_{SN} \times RT_{SN}$
- $\phi_{SN} : E_{SN} \rightarrow [0, 1]$, a function assigning to each edge $E_{SN}$ a trust level $t \in [0, 1]$
OSN $SN$ is a directed labeled graph $(V_{SN}, E_{SN}, RT_{SN}, \phi_{SN})$:
- nodes $V_{SN}$
- supported relationship types $RT_{SN}$
- edges $E_{SN} \subseteq V_{SN} \times V_{SN} \times RT_{SN}$
- $\phi_{SN} : E_{SN} \rightarrow [0, 1]$, a function assigning to each edge $E_{SN}$ a trust level $t \in [0, 1]$
OSN $SN$ is a directed labeled graph $(V_{SN}, E_{SN}, RT_{SN}, \phi_{SN})$:
- nodes $V_{SN}$
- supported relationship types $RT_{SN}$
- edges $E_{SN} \subseteq V_{SN} \times V_{SN} \times RT_{SN}$
- $\phi_{SN} : E_{SN} \rightarrow [0, 1]$, a function assigning to each edge $E_{SN}$ a trust level $t \in [0, 1]$
Access control is performed expressing constraints

- on OSN users graph
  \( (rid, \{(A, colleagueOf, 2, \ast), (D, colleagueOf, 1, 0.2)\}) \)
- and on the resources owned by the users
  \( (rid, \{(rsc', \{(att_1, =, 5), (att_2, <, 3)\}), (rsc'', \emptyset)\}) \)

Note that constraints of both kinds can be used in the same access control rule.
Access control is performed expressing constraints

- on OSN users graph
  
  \[(\text{rid}, \{(A, \text{colleagueOf}, 2, \ast),
  
  (D, \text{colleagueOf}, 1, 0.2)\})\]

- and on the resources owned by the users
  
  \[(\text{rid}, \{(\text{rsc}', \{(\text{att}_1, =, 5),
  
  (\text{att}_2, <, 3)\}), (\text{rsc}'', \emptyset)\})\]
Access control is performed expressing constraints

- on OSN users graph
  \[(\text{rid}, \{(A, \text{colleagueOf}, 2, \ast),\})\]
  \[(D, \text{colleagueOf}, 1, 0.2))\}
- and on the resources owned by the users
  \[(\text{rid}, \{(rsc', \{(\text{att}_1, =, 5),\}),\})\]
  \[(\text{att}_2, <, 3)}), (rsc'', \emptyset)}))\]

Note that constraints of both kinds can be used in the same access control rule.
Access control is performed expressing constraints

- on OSN users graph
  \[ (\text{rid}, \{(A, \text{colleagueOf}, 2, \ast), (D, \text{colleagueOf}, 1, 0.2)\}) \]
  - and on the resources owned by the users
  \[ (\text{rid}, \{(\text{rsc}', \{(\text{att}_1, =, 5), (\text{att}_2, <, 3)\}), (\text{rsc}''', \emptyset)\}) \]

- Note that constraints of both kinds can be used in the same access control rule.
Introduction to On-line Social Network Access Control

Virtual relationships

- The access control procedure locally modifies the relationships between nodes.
- History of terminated interactions.
- For both interacting nodes for keeping track of:
  - who is willing to disclose a certain resource.
  - who is safe to disclose to a certain resource.
- The associated trust value is defined at runtime.
Introduction to On-line Social Network Access Control

Virtual relationships

- The access control procedure **locally** modifies the relationships between nodes
- History of terminated interactions
- For both interacting nodes for keeping track of
  - who is willing to disclose a certain resource
  - who is safe to disclose to a certain resource
- The associated trust value is defined at runtime
The access control procedure **locally** modifies the relationships between nodes.

- History of terminated interactions
  - For both interacting nodes for keeping track of
    - who is willing to disclose a certain resource
    - who is safe to disclose to a certain resource

- The associated trust value is defined at runtime
The access control procedure **locally** modifies the relationships between nodes

- History of terminated interactions
- For both interacting nodes for keeping track of
  - who is willing to disclose a certain resource
  - who is safe to disclose to a certain resource
- The associated trust value is defined at runtime
Rule-based Rumor Propagation

Rule language

- The proposed language inherits from the language used to define access control rules.

- Rules defined by conditions on:
  - Nodes identity, relationships and trust
  - Past interactions

- Rules specify the action to perform
Rule-based Rumor Propagation

Rule language

- The proposed language inherits from the language used to define access control rules.

- Rules defined by conditions on:
  - Nodes identity, relationships and trust
  - Past interactions

- Rules specify the action to perform

Propagate Conditions

- Defined as tuples of the form $(v, rt, t_{min})$
- Limited in the verification at the first level of neighborhood

Example

- $(A, *, *)$
- $(*, *, 0.3)$
- $(*, friendOf, 0.3)$
The proposed language inherits from the language used to define access control rules.

Rules defined by conditions on:
- Nodes identity, relationships and trust
- Past interactions

Rules specify the action to perform

History Conditions
- Defined as sets of the form \( \{r_1, \ldots, r_l\} \)
- Definable on (classes of) resources
- Take advantage of virtual relationships and logs

Example
- \( \{\text{ProjectXBudget, ProjectXDeliverable1}\} \)
- \( \{\text{ThesisChapter1, ThesisChapter2}\} \)
The proposed language inherits from the language used to define access control rules.

Rules defined by conditions on:
- Nodes identity, relationships and trust
- Past interactions

Rules specify the action to perform

**Propagation Rules**
- Defined as tuples of the form (action, CS, HC)
- Actions can both be allow or deny

**Example**
- \((allow, \{(*, colleagueOf, *)\}, \{rsc, rsc'\})\)
- \((allow, \{(*, colleagueOf, *), (*, friendOf, 0.9)\}, *)\)
- \((allow, \{(*, friendOf, *)\}, *)\)
- \((deny, \{(D, *, *)\}, *)\)
Rule-based Rumor Propagation

Rule management

- Propagation rules management simplified by a hierarchical organization
- Default set of rules
  \{\text{(allow}, \{\text{(*,*,*)}, \text{*}\}, \\
  \text{(deny}, \{\text{(*,colleagueOf,*\}, \text{*}\}\}
- Sets of rules for classes of resources
- Specific rules for single resources
- The rules at a lower level can both expand and refine the rules at upper levels
Rule-based Rumor Propagation

Rule management

- Propagation rules management simplified by a hierarchical organization
- Default set of rules
  \[
  \{(allow, \{(*, *, *)\}, *),
  (deny, \{(*, colleagueOf, *)\}, *)\}
  \]
- Sets of rules for classes of resources
- Specific rules for single resources
- The rules at a lower level can both expand and refine the rules at upper levels
Rule-based Rumor Propagation

Rule management

- Propagation rules management simplified by a hierarchical organization
- Default set of rules
  \{ (allow, \{(*, *, *)\}, *),
  (deny, \{(*, colleagueOf, *)\}, *) \}
- Sets of rules for classes of resources
- Specific rules for single resources
- The rules at a lower level can both expand and refine the rules at upper levels
Rule-based Rumor Propagation

Rule management

- Propagation rules management simplified by a hierarchical organization
- Default set of rules
  \[ (allow, \{(*, *, *)\}, *), (deny, \{(*, colleagueOf, *)\}, *) \]
- Sets of rules for classes of resources
- Specific rules for single resources

The rules at a lower level can both expand and refine the rules at upper levels.

Diagram:

- Default
  - Class\(_1\)
    - Res\(_1\)
    - Res\(_2\)
  - Class\(_2\)
    - Res\(_3\)
    - Res\(_4\)
  - Class\(_3\)
    - Res\(_5\)
    - Res\(_6\)
Propagation rules management simplified by a hierarchical organization

Default set of rules
\{(allow, \{(*, *, *)\}, *), (deny, \{(*, colleagueOf, *)\}, *)\}

Sets of rules for classes of resources

Specific rules for single resources

The rules at a lower level can both expand and refine the rules at upper levels
Rumor propagation is subject to preferences of both the users involved

- *Propagation Set*
- *Limitation Set*

Sets of rules applying to the current transaction are exchanged

The users entitled to receive the rumor are identified at runtime
Rule-based Rumor Propagation

Enforcement: an example

Example
Consider the following set of rules:

- **Propagation Set:**
  - \((allow, \{\ast, \ast, \ast\}, \ast)\)
  - \((deny, \{(C, \ast, \ast), \ast\})\)

- **Limitation Set:**
  - \((allow, \{(C, \ast, \ast)\}, \ast)\)
  - \((allow, \{(D, \ast, \ast)\}, \ast)\)
Example

Consider the following set of rules:

- **Propagation Set:**
  - \((allow, \{\star, \star, \star\}, \star)\)
  - \((deny, \{(C, \star, \star), \star\})\)

- **Limitation Set:**
  - \((allow, \{(C, \star, \star)\}, \star)\)
  - \((allow, \{(D, \star, \star)\}, \star)\)
Consider the following set of rules:

- **Propagation Set:**
  - `(allow, {{*, *, *}, *})`
  - `(deny, {{C, *, *}, *})`

- **Limitation Set:**
  - `(allow, {{C, *, *}, *})`
  - `(allow, {{D, *, *}, *})`
Conclusions and Future Work

Conclusions
- A rule-based rumor propagation control language
- Algorithms for the enforcement of the propagation rules

Future Work
- Extension of the rule definition language
- Extension of the enforcement beyond the first level of neighborhood
- Implementation of the enforcement algorithms in our framework
Conclusions and Future Work

Conclusions
- A rule-based rumor propagation control language
- Algorithms for the enforcement of the propagation rules

Future Work
- Extension of the rule definition language
- Extension of the enforcement beyond the first level of neighborhood
- Implementation of the enforcement algorithms in our framework
Questions?

stefano.braghin@uninsubria.it