Integrating Dirichlet Reputation into Usage Control

Li Yang and Alma Cemerlic
University of Tennessee at Chattanooga

Cyber Security and Information Intelligence Research Workshop
2009

Motivation of the Study

• We explore reliability of each peer to improve the quality of service and security.
• A approach to select reliable peer to share resources or services.
  Reputation System
• A way to handle uncertainty and mutability in attributes of a requester, requested objects, and contexts of a request during collaboration.
  Usage Control
Related Work

• Use behavior of a peer to estimate its reliability.
• Peers categorized as good or bad (Beta Reputation System)
  – Distinguishing between selfish and malicious behavior is not possible, such as CONFIDANT (Buchegger 2002a), CORE (Michiardi 2002a), SORI (Wang 2004b), SAFE (Rebahi 2005)
• Peers categorized into multiple groups


Adversary Models

• Adversaries are peers that misbehave in such ways that they degrade integrity and availability of the collaboration
• Misbehaviors include:
  – Attack or launch exploits to others by malicious peers when they are sharing files or applications with other peers
  – Minimize contribution to file or application sharing by selfish peers when they are supposed to share and receive services at the same time.
Dirichlet Reputation System

- “Self-organized system”
  - Decentralized, cooperation cannot be guaranteed

- Reputation systems – “soft security”
  - Stimulate ethical behavior and integrity of members in collaborative environments; recognize and sanction intolerable behavior, reward obedient members

- Each peer observes behavior and evaluates reputation of its requestors or providers.

- Peers exchange reputation information

- Reputation is integrated into usage control to make authorization decision.

Dirichlet Reputation Classification based on Behaviors

- Friendly Peers
  - Correctly share files with others; expected behavior

- Selfish Peers
  - Share files minimally; expect others to share their files;
  - Physical properties (battery, overload), attempt to save resources, random failure
  - Harm availability of the network

- Malicious Peers
  - Inject malware; hack other peers;
  - Harm integrity of packets

- If behavior changes, peer’s reputation changes correspondingly.
Dirichlet Reputation Evaluation

• **Total Reputation** is the combination of
  – first-hand and
  – second-hand reputation

• **First-hand reputation**
  – from direct observation of neighbors’ behavior

• **Second-hand reputation**
  – from neighbors’ reports

---

First-Hand Reputation (1)

• Each peer observes the behavior of its neighbors:
  – Sharing files correctly, receive files but contribute minimally, maliciously modified or inject malware.

• A peer’s behavior follows the *Dirichlet distribution* $\text{Dir}(\alpha)$, where $\alpha = (\alpha_1, \ldots, \alpha_n)$

• Conjugate prior to the Multinomial distribution
  – The probability that each independent trial result is exactly one of some fixed finite number $n$ of possible outcomes with probabilities $p_1, \ldots, p_n$

• Bayes Theorem
  – Incorporate new observation into prior knowledge and obtain posterior probability of node’s behavior.
First-Hand Reputation (2)

- Combine the Dirichlet distribution and Bayes Theorem

- Given the prior $P(X_i) = \text{Dir}(X_i | \alpha_1, \ldots, \alpha_\nu)$
  the posterior distribution is calculated as

  $$P(\alpha_i | D) = \frac{P(D | \alpha_i)P(\alpha_i)}{p(D)} = \text{Dir}(\alpha_i | \alpha_{i1} + N_{i1}, \ldots, \alpha_{i\nu} + N_{i\nu})$$

  - Starting with the initial state of the prior distribution, the parameters are updated when new data $D$ is available
  - $N$ represents instances of the new data

- First-hand reputation value at a time $t$ is equal to the expectation value of $\text{Dir}(\alpha)$

  $$x_i = \frac{\alpha_i}{\alpha_0}$$

First-Hand Reputation (Example)

<table>
<thead>
<tr>
<th>Window</th>
<th>Number of observed behaviors</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>40</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>30</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>40</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>35</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>40</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Total number of observed behaviors in windows 0-5</td>
<td>250</td>
<td>185</td>
<td>50</td>
<td>15</td>
</tr>
<tr>
<td>Expectation of $X_i$</td>
<td>$(185) / (250) = 0.74$</td>
<td>$(50) / (250) = 0.2$</td>
<td>$(15) / (250) = 0.06$</td>
<td></td>
</tr>
</tbody>
</table>

- Behaviors of a peer change from friendly to selfish and malicious.
Second-Hand Reputation

- Collaborative monitoring: exchange first-hand reputation with the neighbors
- Peers are required to periodically broadcast
- Deviation test – to detect false reports
  - Increase or decrease trust?
  - Incorporate the report into the total reputation value?

\[ |E_{ab}(\text{Dir}(\beta_1, \beta_2, \beta_3)) - E_{cb}(\text{Dir}(\alpha_1, \alpha_2, \alpha_3))| \leq d \]

Trust

- Indicates how trustworthy the neighbor’s reports are

- The same Bayesian approach as for the fine-grained reputation system
  - Only two possible instances of behavior: trustworthy and not trustworthy
  - Use the Beta distribution, conjugate prior to the Binomial distribution
  - \( T_{AB} \sim \text{Beta}(\gamma, \delta) \), where \( \gamma = \text{trustworthy}, \delta = \text{not trustworthy} \).

- Updated when the results of the deviation test are available
  Trust value is calculated as: \( \omega_{AB} = E(\text{Beta}(\gamma, \delta)) = \frac{\gamma}{\gamma + \delta} \)
Total Reputation Value

- Merge first-hand and second-hand reputation information

- Second-hand reputation is discounted by the factor $\omega$ (trust value) expressing the disbelief in the accuracy of the report

$$R_{AB} = F_{AB} + \omega_{AC}F_{CB}$$
Moving Windows Mechanism

- Two ways to update first-hand information
  - Based on all observations
  - Based on the most recent observations
    - Reduces computational complexity
    - Early detection of changes in behavior
    - Possibility of redemption over time for misbehaving peers

- To calculate first-hand information:
  - We divide historic information into equal size and considered only a limited number of the most recent intervals/behaviors

Moving Windows Mechanism -- Example

<table>
<thead>
<tr>
<th>Window</th>
<th>Number of observed packets</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>40</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

Reputation based on recent history better reflects changes in behavior

<table>
<thead>
<tr>
<th>Windows 0-5</th>
<th>Total number of observed packets in windows 3-5</th>
<th>Mode of ( X_i )</th>
<th>( (\text{220}) / (\text{250}) = 0.88 )</th>
<th>( (\text{0}) / (\text{250}) = 0 )</th>
<th>( (\text{30}) / (\text{250}) = 0.12 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Windows 3-5</th>
<th>Mode of ( X_i )</th>
<th>( (\text{120}) / (\text{150}) = 0.8 )</th>
<th>( (\text{0}) / (\text{150}) = 0 )</th>
<th>( (\text{30}) / (\text{150}) = 0.2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
More about Moving Window Mechanism

- When behaviors of a peer is becoming malicious or selfish, the size of observation window is reduced.
  - Fast detect of malicious or selfish behaviors
- When behaviors of a peer is becoming friendly from malicious or selfish, the size of observation window is enlarged.
  - Rebuilding reputation is slow and need efforts.

Usage Control Model

- Usage control model proposed by J. Park in 2004 is a generalization of access control to cover authorization, obligation, conditions, continuity (ongoing controls), and etc.
- Authorization handles decision on user accesses to target resources.
- Obligations are the mandatory requirements for a subject before or during a usage exercise.
- Conditions are subject, object, environmental or system requirements that have to be satisfied before granting of accesses.
- Subject and object attributes can be mutable.
Conclusion and Future Work

- We have proposed a framework to integrate reputation management into usage-based access control.
- Our framework is designed to solve behavior uncertainty and attributes mutability in a pervasive and collaborative environment.
- The authentication and authorization to a on-going request is checked constantly during the request.
- The granted request may be terminated if the reputation value is too low or access control rules are not met due to attributes mutability.
- We are working on federated identity management.