Detecting and Excluding Misbehaving Nodes in a P2P Network

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong
firstname.lastname@supelec.fr

Supélec, SSIR Group (EA 4039)

I²CS, June 2008
Schoelcher, Martinique
Main Line of Our Work

**Aim**

Guarantee Confidentiality, Integrity and Availability in P2P

**Specificities of P2P Networks**

*Dynamic and Collaborative* networks without *Central Authority*

**Approach**

1. Membership Control
2. Security Protocols tolerating a bounded number of attackers
Usage

Membership Control through Distributed Certification

1. Genuine members obtain a membership certificate [COPS ’08]
2. Misbehaving nodes are excluded

Exclusion

1. Detection of misbehaving nodes
2. Revocation of their certificates
Related Work

Admission Control to a Peer Group [Kim et al.]
- Admission based on a *Group Charter*
- But people don’t know each other...

Reputation Systems
- Reaction to some types of misbehaviors
- But lack of reactivity...

Proposed Detection and Exclusion
- Complementary reaction to some types of misbehaviors
- Attackers are immediately and globally excluded
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Use-case</td>
</tr>
<tr>
<td>2 Detecting Misbehaving Nodes</td>
</tr>
<tr>
<td>3 Excluding Misbehaving Nodes</td>
</tr>
<tr>
<td>4 Simulations</td>
</tr>
</tbody>
</table>
Use-case
Structured P2P Networks: Chord

DHT: key $\mapsto$ value
Sample Attack
Sample Attack
Sample Attack

dummy value

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong
Detecting and Excluding Misbehaving Nodes in P2P
Sample Attack
Sample Attack
Why Excluding these Attackers?

Our strategy

- Exclude liars
- Make lying an inefficient strategy
- Reduce the number of redundant requests
Detecting Misbehaving Nodes
Detection

Hypothesis

Most of the nodes are well-behaving

Principle

- Define Observable Behavior Specifications
- Compare nodes behaviors
- Generate misbehavior proofs
Timeframes

Legitimate update of value(k)

Time

value(k) = v

value(k) = a

value(k) = v

value(k) = v'
Timeframes

Legitimate update of value(k)

value(k) = v
value(k) = a
value(k) = v
value(k) = v'
Decorrelation

\[ P(\text{attacker}) = k \]

\[ P(\text{attacker}) = 1 \]
Precautions to Consider

Decorrelation

\[ P(\text{attacker}) = k \]
\[ P(\text{attacker}) = 1 \]
Three Messages showing an Attack

- **nodeId**
- **Operation**
- **Payload**
- **Timeframe**

**Discriminant**

- **Quorum**
  - 35
  - 43
  - 46

**K value**

- k

**Payload**

- value(k)

**Timeframe**

- t₁, t₂

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong

Detecting and Excluding Misbehaving Nodes in P2P
### Format of Observable Behavior Specifications

<table>
<thead>
<tr>
<th>Possible values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation</strong></td>
<td>Attacked operation</td>
</tr>
<tr>
<td><strong>Direction</strong></td>
<td>Direction of attack</td>
</tr>
<tr>
<td><strong>Discriminant</strong></td>
<td>Equal fields in compared messages</td>
</tr>
<tr>
<td><strong>Quorum</strong></td>
<td>Identical messages needed</td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td>Scope of the operation</td>
</tr>
</tbody>
</table>

- **Operation**: Get . . .
- **Direction**: Request | Response
- **Discriminant**: Field names
- **Quorum**: N
- **Scope**: Network | Replicas . . .
Example: OBS of the sample attack

<table>
<thead>
<tr>
<th>Operation</th>
<th>Direction</th>
<th>Discriminant</th>
<th>Quorum</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get</td>
<td>Response</td>
<td>key</td>
<td>$\frac{nbRep + 1}{2}$</td>
<td>Replica Set</td>
</tr>
</tbody>
</table>
Overview

Detection Architecture

Detection Architecture Overview

Class 1  Class 2  ..........  Class n
Checker
Attack ?
Classifier
Messages
Checker
Misbehavior proofs

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong
Detecting and Excluding Misbehaving Nodes in P2P
Detection Architecture

Classifier

Classifier creates classes of messages

1. Related to the same OBS
2. Having the same discriminant
3. During the same timeframe

If all the nodes are honest, all messages of a given class are identical

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong
Detecting and Excluding Misbehaving Nodes in P2P
Detection Architecture

**Checker**

Checker applies OBS to classes

1. Valid payload from the quorum
2. Proof against different payload(s)
3. Proof contains the minimal number of messages

The misbehavior proof is then used to exclude the attacker
Excluding Misbehaving Nodes
Excluding Misbehaving Nodes

1. Revoke the certificate of the attacker
2. Publish this revocation
Revocation

Revocation is done through Distributed Certification [AIMS 08]

1. Requires collaboration of \( t\% \) of the nodes, whatever the size of the network
2. Each node locally checks the validity of the proof

If \( t\% \) of the nodes validate the proof, the initiator node obtains the revocation to publish.
Revocation Publication

Publication

Notification of the revocation to all the nodes

1. Put in the DHT at \( h(\text{attacker}_{id}) \)
2. Directly notified to nodes connected to the attacker
   1. Nodes in its routing table
   2. Nodes having it in their routing table
   3. Nodes connected at application level
Simulations
Simulations

Percentage of honest nodes excluded
Simulations

Percentage of honest nodes excluded
Simulations

Percentage of Success of Certification Algorithm

Success with nbAsks = 5
Success with nbAsks = 1
Remaining attackers

5000 nodes, 15% of attackers
Detecting and Excluding Misbehaving Nodes

- Misbehaviors are detected using comparison to other nodes
- Attackers are excluded using revocation and publication
- Sound in the expected case

We are now looking further on this DHT use-case...
Detecting and Excluding Misbehaving Nodes in a P2P Network

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong
firstname.lastname@supelec.fr

Supélec, SSIR Group (EA 4039)

I²CS, June 2008
Schoelcher, Martinique