SMEF: An entropy-based security Framework for Cloud-oriented Service Mashup

Ruixuan Li, Li Nie, Xiaopu Ma,
Meng Dong and Wei Wang

School of Computer Science and Technology
Huazhong University of Science and Technology
What’s Service Mashup

- **Definition**
  Service mashup is a new technology that aggregates various services to form new applications to provide services according to user requirements.

- **Popular service mashup platform**
  - QEDWiKi (IBM)
  - Popfly (Microsoft)
  - Pipes (Yahoo)
  - FeedBurner (Google)
Service Mashup and Service Composition

- Service mashup has the following core features, compared with service composition
  - *End Consumer Centric*
    - Mashup is supposed to support programming for end consumer, not developer, without complex programming environment
  - *Light Weight*
  - *More Reusable*
- Service Mashup is mainly at “application” level instead of “interface” level
Service Mashup in Cloud Environment

- Service mashup will be an important way to aggregates services to form a new application in cloud environment.

- Different services have different security policies (may cause security strategy conflicts while mashuping services).
Related Work

- Most work focuses on how to better satisfy functional and QoS (Quality of Service) requirements.

- There is few research focusing on the security requirements in service mashup.

- No method for quantitative security evaluation of service mashup in cloud environment has been proposed.
Contributions

- Introduce entropy to assess security of service mashup
  - Security of single service
  - Security of mashup service chain

- A secure framework for cloud-oriented service mashup
  - A multi-objective selection method for service mashup
  - Simultaneously satisfy functional and nonfunctional (security) requirements
Outline

- Introduction
- Multi-hierarchy Security Entropy Modeling
- SMEF Framework for Cloud Service Mashup
- Evaluation
- Conclusion
Three-dimensional Factors

- Environment factors
  - Cyber attack, network communication, natural disaster caused by force majeure, et al.

- Service factors
  - Interoperability of services, services unavailability, credibility of service providers, et al.

- User factors
  - User privacy protection, user privilege, user credibility, et al.
Three-dimensional Factors

Environment

User

Service
Security Quantitation of Single Service

**Security degree**: A specified probability for each factor of every service to fulfill security needs. It quantifies possibilities of fulfillment corresponding to security demand.

- Construct security degree matrix
  - All security factors is divided into 3 levels according to the capability of security
  - Each level contains multiple factors

- Determine entropy weight of every factors

- Quantify the security of single service
Mashup Service Chain

*Mashup Service Chain*: a collection of services and their relationship for fulfilling some specific demand.

Taking into account the *interactions* of the mashup services, we quantify the entropy of mashup services.

- **Mutual entropy**
  \[
  H(X, Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} r(u_{ij}^x, u_{ij}^y) \log r(u_{ij}^x, u_{ij}^y)
  \]

- **Conditional entropy**
  \[
  H(Y / X) = -\sum_{i=1}^{n} \sum_{j=1}^{m} r(u_{ij}^x / u_{ij}^y) \log r(u_{ij}^x / u_{ij}^y)
  \]

  - *X,Y*: two different services
  - *u*: the value of security degree matrix in services
An Example of Mashup Service Chain
Mashup Structure Patterns

- Six different structures in a mashup service

- These six patterns can be divide into three categories:
Mashup Mapping Patterns

- One-to-one mapping
  \[ H(X \rightarrow Y) = H(X) + H(Y / X) \]

- Many-to-one mapping
  \[ H((X_1, \ldots, X_n) \rightarrow Y) = H(Y / X_1, \ldots, X_n) + H(X_1, \ldots, X_n) \]

- One-to-many mapping
  \[ H(Y \rightarrow (X_1, \ldots, X_n)) = \max(H(Y \rightarrow X_1), H(Y \rightarrow X_2), \ldots H(Y \rightarrow X_n)) \]

The mashup service chain is made up of the above three mapping patterns.
Mashup Services Chain
Outline

- Introduction
- Multi-hierarchy Security Entropy Modeling
- SMEF Framework for Cloud Service Mashup
- Evaluation
- Conclusion
SMEF Secure Framework

- The first stage: The functional mashup
  - Deal with the desired functionality requirements of the mashup services

- The second stage: The nonfunctional mashup
  - Involve QoS, constraints of security requirements
  - The entropies of single service and mashup service
SMEF Architecture

Nonfunctional mashup
- Optimal mashup service
- functional mashup
- Qos and entropy of security
- Multi-objective selection
- service mashup schemes

The Entropy of service chain
- The Calculation of Entropy of every mashup patterns
- The Calculation of Entropy of a set of mashup services chain

Cloud user
- Service mashup request

functional mashup
- Formalization description
- The services composition algorithm
- service mashup schemes only meeting function

The Entropy of each service
- The extraction and classification of security factors
- The Entropy set of single service
- Acquisition of safety
Algorithm: Nonfunctional-Satisfy(S)
Input: Service chain (S)
Output: True, false: S is satisfied or not
1: Rewrite criterion to normal form
2: For each (S)
3: For each (k ∈ S)
4: If (F(k) ∉ range) then
5: Return false;
6: End if
7: End for
8: If (Q_e(k) ≥ q_e )
9: Return false;
10: End if
11: End for
12: Return true
13: End
Outline

- Introduction
- Multi-hierarchy Security Entropy Modeling
- SMEF Framework for Cloud Service Mashup
- Evaluation
- Conclusion
Performance Metrics

- **Cost Time**
  - \(TNS\): cost time of finishing a mashup service without security entropy constraints
  - \(TS\): cost time of finishing a mashup service with security entropy constraints

- **ASR (Average success rate)**
  - \(SNS\): average success rate for mashup requests without security entropy constraints
  - \(SS\): average success rate for mashup requests considering security by using of SMEF and security entropy

- **FSR (False selection rate)**
  - The percentage of chosen services from all the preset services with low security degrees.
Simulation settings

- Service set are simulated data from China Web Service Cup (CWSC2011) Competition

- The security degrees of services are preset at initialization
  - The security degrees of one third of services follow a normal distribution with mean 0.2
  - The security degrees of the rest services obey the normal distribution with mean 0.5
Three Groups of Experiments

- We carry out the experiments with:
  - Different scales of atomic services (NA)
  - Different number of security factors of each service (NS)
  - Security degrees of preset services are adjusted dynamically (AdjSd)

- We have run each experiment for 100 times and take the average as the result
## Group NA Experiment

<table>
<thead>
<tr>
<th>NA</th>
<th>TNS (ms)</th>
<th>TS (ms)</th>
<th>SNS (%)</th>
<th>SS (%)</th>
<th>FRS (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>189.14</td>
<td>220.91</td>
<td>90</td>
<td>37</td>
<td>3.76</td>
</tr>
<tr>
<td>4000</td>
<td>363.67</td>
<td>404.73</td>
<td>93</td>
<td>34</td>
<td>4.07</td>
</tr>
<tr>
<td>6000</td>
<td>332.35</td>
<td>376.44</td>
<td>96</td>
<td>48</td>
<td>5.30</td>
</tr>
<tr>
<td>8000</td>
<td>634.23</td>
<td>672.12</td>
<td>97</td>
<td>56</td>
<td>4.98</td>
</tr>
<tr>
<td>10000</td>
<td>632.97</td>
<td>674.32</td>
<td>95</td>
<td>54</td>
<td>6.21</td>
</tr>
<tr>
<td>12000</td>
<td>823.32</td>
<td>875.23</td>
<td>96</td>
<td>56</td>
<td>6.32</td>
</tr>
</tbody>
</table>
### Group NS Experiment

<table>
<thead>
<tr>
<th>NS</th>
<th>TNS (ms)</th>
<th>TS (ms)</th>
<th>SNS (%)</th>
<th>SS (%)</th>
<th>FRS (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>232.14</td>
<td>300.15</td>
<td>92</td>
<td>46</td>
<td>9.03</td>
</tr>
<tr>
<td>6</td>
<td>232.14</td>
<td>333.32</td>
<td>92</td>
<td>43</td>
<td>10.07</td>
</tr>
<tr>
<td>9</td>
<td>232.14</td>
<td>341.42</td>
<td>92</td>
<td>34</td>
<td>5.30</td>
</tr>
<tr>
<td>12</td>
<td>232.14</td>
<td>321.53</td>
<td>92</td>
<td>32</td>
<td>8.98</td>
</tr>
<tr>
<td>15</td>
<td>232.14</td>
<td>348.29</td>
<td>92</td>
<td>38</td>
<td>6.21</td>
</tr>
<tr>
<td>18</td>
<td>232.14</td>
<td>339.53</td>
<td>92</td>
<td>37</td>
<td>6.32</td>
</tr>
<tr>
<td>21</td>
<td>232.14</td>
<td>342.63</td>
<td>92</td>
<td>35</td>
<td>7.21</td>
</tr>
<tr>
<td>24</td>
<td>232.14</td>
<td>347.34</td>
<td>92</td>
<td>31</td>
<td>5.32</td>
</tr>
</tbody>
</table>
Group AdjSd Experiment

- In the last set of experiment, we investigate frequency that one service will be selected by security service chains, when its security degree decreases or increases sharply.

- By the change of security degree it can be found that if a service increases its security degrees, the selected probability of this service will increase correspondingly.
Outline

- Introduction
- Multi-hierarchy Security Entropy Modeling
- SMEF Framework for Cloud Service Mashup
- Evaluation
- Conclusion
Conclusion

- A secure framework to choose a relatively optimal mashup service chain meeting both users’ functional and nonfunctional requirements.
- The Introduction of entropy to measure the security of single service and mashup service chain.
- A multi-objective selection method to aggregate multiple criteria as a single criterion.
Thanks for your attention

Contact information:

- Ruixuan Li
- Huazhong University of Science and Technology
- rxli@hust.edu.cn
- http://idc.hust.edu.cn/~rxli/