A trust evaluation method based on the distributed Cloud Trust Protocol (CTP) and opinion sharing

Abdelmageed Algamdi
Supervisors:
Alexei Lisitsa and Frans Coenen

Department of Computer Science
University of Liverpool,
Liverpool, UK

Jan 28-29, ICCAT 2017
Outline

- Cloud Computing
- Service Level Agreement - SLA
- Trust
- Reputation System
- Cloud Trust Protocol - CTP
- Motivation - Objective
- Proposed System
  - Problem Definition
  - Proposed Cloud Structure
  - Assessment System
    - Provider Side
    - Consumer Side
- Future Work
Cloud Computing

It allows users to remotely share the various resources over the internet. With its low cost, it provides customers with huge memory space and robust computation capability.

Service Level Agreement (SLA)

An SLA can be defined as a contract between the service user and the service provider that contains a description of the agreed-upon service.
Trust in our life

It is a belief that someone or something is reliable, good, honest, effective, etc or to have confidence in someone or something.

Trust in CS and Cloud Computing

- relies upon diverse attributes such as information security, compliance and data governance etc.
- Trust management monitors the quality of service (QoS) and the service level agreement between the user and the service provider and affect a reputation system. eg. Trust in Ubiquitous Computing (Ries S., 2009)
Reputation

- is a general belief about a person, such as his or her character. In general, reputation can be used as a source of trust. It is comprised of opinions (a collective picture) about an entity or person. 


- It calculates the trustworthiness of the service provider based on a self-assessment done by the provider.
- There are several contributions in order to design a reliable reputation system.
The Cloud Trust Protocol (CTP) is a protocol which was proposed by the late Ron Knode in 2010 and licensed for use in 2011 by the Cloud Security Alliance.

It provides a way for the user to request evidence or certificates from the cloud service provider regarding the operation of a specific service.

This information gives the user a whole picture about what he should expect while running the service.
Cloud Trust Protocol - CTP

- The main purpose of CTP is to generate an evidence-based information that everything is running based on the SLA agreement between the cloud user and the provider.
- These evidences are based on number of elements of transparencies.
- This is done through request/response technique over 24 Elements of Transparency (EoT).
- The CTP API uses the RESTful API for performing the request/response queries through HTTP methods such as GET, PUT, POST and DELETE.

Example

<table>
<thead>
<tr>
<th>Request</th>
<th>GET</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://cloudtrust.csc.com/ctp/%5BcustID">https://cloudtrust.csc.com/ctp/[custID</a>] /resources/2/</td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>HTTP/1.1 200</td>
</tr>
</tbody>
</table>
The CTP API uses the RESTful API for performing the request/response queries through HTTP methods such as GET, PUT, POST and DELETE.

Example

<table>
<thead>
<tr>
<th>Request</th>
<th>GET</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://cloudtrust.csc.com/ctp/%5BcustID">https://cloudtrust.csc.com/ctp/[custID</a>] /resources/2/</td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>HTTP/1.1 200</td>
</tr>
</tbody>
</table>

Example to the HTTP response codes to a given CTP request.

- **200**: response code with YES/OK answers.
- **400**: response code with NO answers.
- **204**: response acknowledgement that the user request arrived but the cloud provider doesn't want to respond.
- **401**: response code for unauthorized requests.
Our proposal main principle

- Can we utilise here collective opinions/trust?
  - Yes, reputation systems provide users of integrating opinions/trust.

  **How?**
  1: Use CTP to get information about the services.
  2: Provide infrastructure to support CTP/assessments requests.
  3: Give ability to provider to do self-assessment.
  4: Give the users the ability to reassess the service after using it.
Main Idea
Use the CTP protocol as a source of information about any service offered over the cloud in order to enhance the previous trust mechanisms over the cloud.

How?
By giving the user the ability to reassess the service and update the old trustworthiness value generated before from the provider self-assessment.
Proposed Cloud Structure
LTS/RqMg is used as a local trust server while calculating the trustworthiness for a specific provider and as a request manager in the case of applying the CTP protocol.
A general trust server (GTS) is used to collect and store the overall trust values for the service providers in assessment operations and to just route the requests/responses in the CTP protocol.
The CAIQ engine is used to store the CAIQ questionnaires and the assessed questionnaires from the users.
Different requests/responses

**CTP request/response:**

- The user asks for information relating to CTP. A CTP initiation request is sent with 00 leftmost flag. Once the initiation request is approved, CTP EoT can be requested also with 00 leftmost flags.

- The **LTS/RqMg** now is working as a **RqMg**. The EoT bits are asserted according to the requests required.

- The request is sent to the **GTS** which acts now as a router. It tells the request where should it go.

- Once the CTP is received at a cloud, the **RsMg** is responsible for the response also with 00 leftmost flag.

- The response is sent back from the **RsMg** to the **RqMg** via the **GTS** and then is delivered to the user.
CAIQ Assessment request/response:

- For those users who already used the CTP protocol to get trust information about a specific service provider, it is allowed to assess them.
- An assessment request is sent with 01 leftmost flag from the user to the LTS/RqMg which now works as a LTS.
- The request is forwarded from the LTS to the GTS which asks the CAIQ engine for the CAIQ questionnaire.
- The CAIQ questionnaire is sent back to the user via GTS and LTS servers.
- Once the user has finished the assessment, the trust value is calculated and stored at the LTS and a copy of it also will be sent to the GTS. So, now LTS contains entries for all the providers assessed from a member of the same organization with their trust values generated only from it while GTS contains entries for all the providers assessed from all organizations with the updated trust values.
Trust query request/response:

- The cloud consumer can ask for a list of all the providers offer a specific service. This is a request with 1d leftmost flag. This request has one of two destinations, either **LTS** or **GTS**.

- The **LTS** send a response directly if there at least an entry in its table with an accepted Boolean trust (true) and not timed out for the provider offer this service.

- If **LTS** has no direct answer or is not Boolean trusted or there was an entry but timed out, the request will be forwarded to the **GTS** which answers it.

- Every amount of time, all the entries exist in the **LTS**’s table—which are timed out– have to be updated from the **GTS** throw updating request.
We assume that the trustworthiness for any service is being represented by a scalar value. That value increases if the service is satisfied by the SLA agreement and hence satisfies the users demands and decreases if it violates the SLA agreement.

The case if the SLA agreement is not violated and the user feedback shows that he is unhappy, this mean that the user did the assessment in a wrong way and his opinion now shouldn’t affect the old trustworthiness value –untrusted agent. (Future Work)
Consensus Assessments Initiative Questionnaire (CAIQ) spreadsheet comprises 140 questions generated by the Cloud Security Alliance (CSA).

Yes-No questionnaire covering eleven control areas: compliance, data governance, facility security, human resources, information security, legal, operations management, risk management, release management, resiliency, and security architecture.
The Smals ICT for society group generated cloud security assessment models that can be used by clients (normal or experts).

It provides questionnaire which enables the cloud consumer to assess the security level of a cloud service offered by a Cloud Service Provider.

User questionnaire covers four control areas:
- Governance
- Identity and Access Management (IAM)
- IT Security
- Operational Security
Our proposed system is to propose a cloud structure that enables both the client and provider assessments.

Also, it specify the way the client is going to assess the service and how this assessment affects the original trust value generated from the self-assessment.
We will use the same CAIQ assessment used before from the providers to assess their own services and create their own initial trust value. **Already done before**
The Consumer questionnaire is divided into 4 attributes and each attribute contains more than one sub-attribute. Each sub-attribute consists of some Yes-No questions.
Calculating User Opinion

- For each sub attribute calculate $W_{sub} = (b_{x,sub}, d_{x,sub}, u_{x,sub}, a_{x,sub})$ based on the Yes/No answers including the Unknown.

\[
\begin{align*}
    b_{sub} &= \frac{p}{p + n + m} \\
    d_{sub} &= \frac{n}{p + n + m} \\
    u_{sub} &= \frac{m}{p + n + m} \\
    a_{sub} &= \frac{1}{2}
\end{align*}
\]

Where $p$ is the number of Yes answers, $n$ is the number of No answers and $m$ is the number of the unknown answers.
Calculating User Opinion

- For each attribute, calculate the product of all the opinions generated for all the sub attributes inside that attribute
  \[ W_{att} = \prod W_{sub} \]

- For sub-opinions with some uncertainty, the overall opinion will be collected by the consensus operator assuming that all the four attributes assess the service from different point of view
  \[ W_x = W_{x}^{att1} \oplus W_{x}^{att2} \oplus W_{x}^{att3} \oplus W_{x}^{att4} \]

- For sub-opinions with 0 or 1 uncertainty, the overall opinion will be generated by simple product operator between them
  \[ W_x = \prod_{i \in \{1,2,3,4\}} W_{x}^{att_i} \]
The users opinion $W_x = (b_x, d_x, u_x, a_x)$ where $b_x + d_x + u_x = 1$ can be visualized using Barycentric coordinates in order to know the aging factor $k$ of the users opinion on the initial trust value generated by the provider. eg. Subjective Logic (Audun Jasang, 2015)
**User opinion clustering:** classify the user opinion into one of six rating classes.
<table>
<thead>
<tr>
<th>Region</th>
<th>Belief</th>
<th>Disbelief</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Good, Certain</td>
<td>$b_x \geq 0.5$</td>
<td>$d_x &lt; 0.5$</td>
<td>$u_x &lt; 0.5$</td>
</tr>
<tr>
<td>Good, Certain</td>
<td>$0.25 &lt; b_x &lt; 0.5$</td>
<td>$d_x &lt; 0.25$</td>
<td>$u_x &lt; 0.5$</td>
</tr>
<tr>
<td>Very Bad, Certain</td>
<td>$b_x &lt; 0.5$</td>
<td>$d_x \geq 0.5$</td>
<td>$u_x &lt; 0.5$</td>
</tr>
<tr>
<td>Bad, Certain</td>
<td>$b_x &lt; 0.25$</td>
<td>$0.25 &lt; d_x &lt; 0.5$</td>
<td>$u_x &lt; 0.5$</td>
</tr>
<tr>
<td>Un-named, Certain</td>
<td>$0.25 \leq b_x &lt; 0.5$</td>
<td>$0.25 \leq d_x &lt; 0.5$</td>
<td>$u_x &lt; 0.5$</td>
</tr>
<tr>
<td>Very Uncertain</td>
<td>—</td>
<td>—</td>
<td>$u_x \geq 0.5$</td>
</tr>
</tbody>
</table>

**Table:** Opinion Classification
Determining the aging factor $k$: 
The value of $k$ is determined as follow and depends on the rating class for the consumer opinion:

- For very good and certain class $k = 1$.
- For good and certain class $k = \frac{1}{2}$.
- For very bad and certain class $k = -1$.
- For bad and certain class class $k = \frac{-1}{2}$.
- For unnamed and certain class $k = \frac{1}{4}$ if $P_x \geq 0.5$ and $K = \frac{-1}{4}$ if $P_x < 0.5$.
- For very uncertain class $k = 0$. 
Updating the old trustworthiness value (by aggregation and aging):
Let's define:

- $r_{y,t}$: is the initial rating value (only provider) generated from the provider self-assessment for service $y$.
- $R^x_{y,t}$: is the old rating value (Provider and user $x$) over time $t$ for service $y$.
- $R^x_{y,(t+1)}$: represents the overall (Provider and user $x$) new accumulated rating value after time period $t + 1$ for service $y$.
- $R_{y,(t+1)}$: represents the overall (Provider and all users) new accumulated rating value after time period $t + 1$ for service $y$.
- $\lambda$: aggregation constant.
Case 1: Aggregation with zero-aging

- For the first user assessment:
  \[ R^x_{y,(t+1)} = \lambda' + r_{y,t}, \text{ where } 0 \leq \lambda' \leq 1, \lambda' = (k_{t+1} - k_t) \times \lambda. \]

- For any user assessment except the first one:
  \[ R^x_{y,(t+1)} = \lambda' + R^x_{y,t}, \text{ where } 0 \leq \lambda' \leq 1, \lambda' = (k_{t+1} - k_t) \times \lambda. \]

The overall reputation (rating) generated from all the users \( x \in X \) – where \( X \) is the set of all users did the assessments – is simply generated from the average over all users’ ratings as follows:

\[
R_{y,(t+1)} = \frac{\sum_{x \in X} R^x_{y,(t+1)}}{|X|}
\]
Case 2: Aggregation with Λ-aging where Λ ≠ 0

- For the first user assessment, there is no need for doing any form of aging here in the first user assessment:
  \[ R_{y,(t+1)}^x = k_{t+1} \times \lambda + r_{y,t}, \text{ where } 0 \leq \lambda \leq 1. \]

- For any user assessment except the first one:
  \[ R_{y,(t+1)}^x = k_{t+1} \times \lambda + \Lambda \times R_{y,t}^x, \text{ where } 0 \leq \lambda \leq 1. \]
  - For decreasing the effect of the history we use \( \Lambda = 0.01 \).
  - For increasing the contribution of the history in the calculation of the current reputation value we use \( \Lambda = 0.99 \).
Case 2: Aggregation with Λ-aging where Λ \neq 0

The average rating $R_{y, (t+1)}^x$ generated by user $x$ towards the service $y$ at the current time $t + 1$ is calculated as follows:

$$R_{y, (t+1)}^x = \frac{R_{y, (t+1)}^x}{N}$$

where $N$ is the number of assessments for the user $x$ towards the service $y$.

The overall reputation (rating) generated from all the users $x \in X$—where $X$ is the set of all users did the assessments— is simply generated from the average over all users’ average ratings as follows:

$$R_{y, (t+1)} = \frac{\sum_{x \in X} R_{y, (t+1)}^x}{|X|}$$
Future Work

- Solving the problem of removing the untrusted agents before doing the assessments.
- Solving the problem of securing the approach against the malicious users attacks done by hackers.
Thank you for your attention