# A Framework for Ranking Cloud Security Services

Ahmed Taha*, Ruben Trapero§, Jesus Luna* and Neeraj Suri*
*Technische Universität Darmstadt, Germany §Atos Research & Innovation, Madrid, Spain
Email:{ataha,jluna,suri}@deeds.informatik.tu-darmstadt.de, ruben.trapero.external@atos.net

***Abstract*—Although the use of Cloud services is proliferating, the notion of Cloud security remains ambiguous. This typically arises from two causes, namely (a) the limited awareness about security details by the average Cloud customer which results in the customers being unable to clearly express their security requirements, or (b) the lack of interfaces/tools that can meaningfully capture the customer requirements. In general, the Cloud customers are only able to provide qualitative requirements due to their inability to express precise security requirements. Nevertheless, Cloud customers still need to assess and benchmark various security services provided by different providers in order to select the most suitable Cloud provider that can satisfy their "imprecise and uncertain" security requirements.**

**This paper proposes a methodology for enhancing the security aspects of Cloud services by quantitatively comparing the customer security requirements with the security offered by Cloud providers. The novelty of our approach is based on the usage of a fuzzy logic schema to manage the uncertainty of those qualitative requirements. We validate our framework by applying it to real-world data that leverages the standardized Cloud service level agreements structure proposed in the ISO/IEC 19086 standard.**

***Keywords*-Cloud security; security quantification; security service level agreements;**

## I. Introduction

In the Cloud market, the Cloud customers can access the sprectrum of desired computing resources (storage, servers, applications) on demand on a pay-as-you-go basis where the resources are provided as-a-service in a remotely accessible fashion. In such an environment, the provisioning of Cloud services depends on the stipulated Service Level Agreements (SLAs), i.e., the formal contracts established between the customer and the Cloud Service Provider (CSP)[1].

With security as a major driver, the different stakeholders in the Cloud community have advocated that defining security parameters in the SLAs (termed as secSLAs in this paper) can help in providing measurable security assurance across the CSPs and customers. Furthermore, the secSLAs can constitute useful means to express security using common semantics which allow to represent both the security level required by customers and offered by the CSPs.

With the growth of Cloud security services, multiple CSPs offer similar services at different prices and capabilities. In order to help the customers to find the best matching CSP that can satisfy their security requirements, a number of techniques have been proposed that can dynamically assess and validate the security claims of a CSP with respect to the customers security requirements [2]–[4]. However, most of these techniques require the customers to provide "detailed" values specifying their requirements and to submit weighting factors to model their priorities. This requires expert knowledge. Generally, the Cloud customers are not experts on security services and typically cannot detail the required security service or assign specific parameter values. For example, it is unlikely that a customer knows the details about which level of encryption can enhance security or how this level will impact other security service levels. Thus, if a customer was asked to provide detailed values specifying his/her requirements, the customer ends up either (a) accepting an inappropriate/inadequate service or (b) requiring an unachievable level of a security service which causes these requirements to be impossible to satisfy. To faciliate an easier adoption of secure Cloud services, it would be desirable to let customers express their requirements in a qualitative way (e.g., using approximate linguistic[2] descriptors or with natural language phrases). Qualitative requirements form a useful mechanism to engage customers, who are not security technically savvy, in the usage of secure Cloud services. However, the impreciseness and subjectivity of qualitative requirements can naturally also lead to inaccurate results when evaluating the security level of the Cloud services.

In this paper we aim to solve the problem of "accurately" evaluating the security level of a CSP when qualitative requirements are defined by the customers. We present a novel framework that allows the customers to (a) specify their requirements using natural language phrases and linguistic descriptors as well as precise values, thus enabling both novice and expert customers to provide their security requirements according to their expertise, and (b) assess, compare, and rank various security services provided by different CSPs. This is achieved through the following contributions by:

1) Employing membership functions to capture the customer's subjective requirements and utilizing a fuzzy logic system to derive precise security levels. Our ap-

[1]The SLA specifies how provisioning takes place as well as the respective rights and duties of both the Cloud customer and the CSP [1].

[2]Variables whose values are not numbers but words or sentences in a natural or artificial language [5].

proach also supports the blended submissions of different types of requirements.
2) Developing a secSLA assessment technique based on the fuzzy Analytic Hierarchy Process (AHP) for ranking the CSPs according to the customer requirements.
3) Validating the proposed framework by evaluating CSP secSLAs found on the public STAR (Security, Trust and Assurance Registry) [6] repository, which are complaint with the relevant ISO/IEC 19086 standard [7].
4) Developing a prototype control panel that implements the proposed model[3].

The rest of the paper is organized as follows. We describe the related work in Section II followed by Section III that presents the basic concepts about fuzzy logic and Cloud secSLAs. Section IV describes the proposed methodology, and the real-world use cases demonstrating the Cloud secSLAs evaluation are presented in Section V.

## II. Related Work

Multiple approaches exist to assess CSP functionality and security. For service-ranking approaches, reasearchers have investigated the use of Multi-criteria Decision Making (MCDM) methodologies to rank CSPs [3], [8]. Hamzeh et al. [9] developed a fuzzy MCDM approach that uses linguistic descriptors to model user preferences. However, all these studies focus on the performance of Cloud services rather than security properties. The authors in [10] and [11] evaluated the trust levels of CSPs according to the customer feedback, though the diverse requirements of different customers was not factored. In [12], a fuzzy inference system was used to evaluate the trust of the providers. Nevertheless, the customers were then required to "manually" tune the technically complex inference system according to their requirements.

Although multiple Cloud security frameworks have specified security aspects in secSLAs, few works exist that quantify the security services in secSLAs. Security requirements have been assessed by Casola et al. [13], who proposed a methodology to evaluate secSLAs for web services. Chaves et al. [14] explored security in SLAs applied on a monitoring architecture. In contrast to our research, the existing works do not propose techniques to assess secSLAs or empirically validate the proposed metrics. In [15] and [2], the authors present a framework to compare and rank CSPs. However, their approach depends on a manual static weight assessment for preference modeling, which is both tedious and does not account for the uncertainty associated with human assessment. Our proposed framework addresses such limitations by allowing customers to use linguistic descriptors to easily define their vague requirements and preferences.

## III. Terminology and Basic Concepts

This section introduces the concepts used in the paper.

[3]The implementation is at https://github.com/amtaha/fuzzyQHP.

### A. Security Service Level Agreements (secSLAs)

A Cloud security Service Level Agreement (secSLA) specifies the provided security services, and represents the binding commitment between a customer and a CSP. Basically each of these security services contains a list of Service Level Objectives (SLOs), which are the measurable elements of a secSLA that specify the service levels needed by the customers, and to be fulfilled by the CSP.

The unfulfillment of any SLO would result in the violation of the secSLA, and thus entail a possible penalty for the CSP. Based on the analysis presented in [2], Cloud secSLAs are typically represented as a hierarchy as shown in Figure 1. This hierarchy specifies the actual security services specified in a secSLA. Each of these services is composed of one or more SLOs. To formalize the concept of an secSLA we use the definition presented in [1].

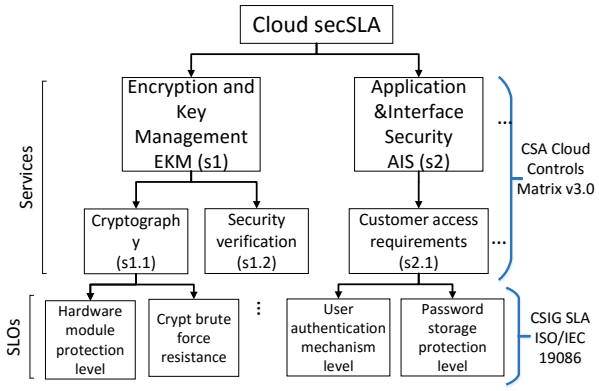


Figure 1. Cloud secSLA hierarchy

*Definition 1:* A secSLA consists of a set of services $S = s_1, ..., s_n$. Each service consists of a finite positive number $n$ of SLOs $k_i$; where $i = 1 \dots n$. Each SLO $k_i$ consists of $m$ different values. Each value indicates a different security level requested by the customer and offered by the CSP.

An example of an SLO, as shown in Figure 1, is "Crypt brute force resistance" SLO which is composed of $\{512bits, 1024bits, 2048bits\}$ values which are defined using security levels as $level_1$, $level_2$, $level_3$ respectively.

### B. Fuzzy Logic System

Fuzzy logic system [16] is used to solve reasoning problems in uncertain environments due to its ability to handle imprecise and fuzzy inputs. It maps a given input to an output using membership functions and linguistically specified rules such that:

- **Membership function ($\mu_F$):** defines to which degree the fuzzy element belongs to the corresponding fuzzy set. It maps specific values to membership degrees between $0-1$.

- **Rules:** are expressed as a collection of *if-then* statements which define the fuzzy model. The rule structure is: **if** *antecedent* **then** *consequent*, where antecedent and consequent are fuzzy terms. These rules help in quantifying values as well as linguistic variables, by using fuzzy membership functions (e.g.,"**If** $x_1$ is *warm* **then** $y_1$ is *quite low*" where $x_1$ may have a finite number of linguistic variables associated with it, ranging from extremely warm to extremely cold). Additionally we can combine multiple rules using AND or OR operators.

## IV. Security Assessment Methodology

There are three stages involved in the usage of the proposed system as shown in Figure 2. The first stage captures customers' descriptive requirements of different security SLOs. In this stage we receive the customers' requirements as well as the CSPs' secSLAs. Stage B tunes the membership functions, evaluates the rules and computes quantitative values for various CSPs based on their security levels measured according to the customers' security requirements. Using this quantitative assessment of CSPs, the CSPs are ranked (according to their secSLAs) for the best match to the customers' requirements in Stage C. We detail each of these stages in the subsequent sections.

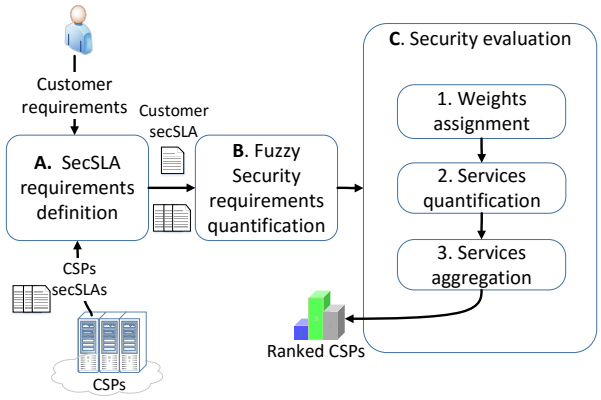


Figure 2. Proposed methodology stages.

### A. Stage A. SecSLAs Requirements Definition

Based on the secSLA hierarchical structure[4], customers have the ability to specify their security requirements (according to their expertise) at varied levels of the secSLA representation. The customers' secSLAs are based on the same secSLA template used by the CSPs to specify their security SLOs. Our methodology supports three types of requirements specification:

- **Linguistic terms**: Used by novice customers to specify qualitative requirements using terms such as (Not-Required (NR), Less-Required (LR), Required (R), Highly- Required (HR), and Do-not-know (Dk)) to reflect human requirements.
- **Natural language statements**: Since humans are accustomed to using descriptive phrases to make rough estimations, they can easily submit meaningful requirements in the form of natural language statements. To achieve this (i) such a language must be based upon information that customers find cognitively easy to reflect upon and express, and (ii) the information communicated in such language should be reasonably easy to interpret. Considering this list of requirements, it appears unavoidable that such a language would be based on linguistic terms or qualitative information. For example "I *highly require* a provider encrypting data in transit and at rest". These statements are then defined using *if-then* rules (detailed in Section IV-B).
- **Precise values**: Expert customers can specify their requirements at the SLO level by defining the required value for each SLO. These values can be qualitative or quantitative, which are then modeled as fuzzy numbers and represented using membership functions as explained in the next stage (Stage B).

Blended submissions of different types of requirements for the same secSLA is also supported in our system. Section V illustrates, using a real-world case study, how different types of requirements help customers to define their requirements.

### B. Stage B. Fuzzy Security Requirements Quantification

To assess and compare the security levels provided by different CSPs according to the different customers' expertise, a measurement model for different security SLOs should be defined. Hence, fuzzy numbers[5] are used to translate the vagueness and imprecision of customers' requirements according to their security expertise. The most commonly used shapes for fuzzy numbers are triangular, trapezoidal, piecewise linear and Gaussian. In this paper, the triangular fuzzy numbers (TFNs) [16] are used to capture the vagueness of the parameters.

A TFN is graphically shown in Figure 3, where the TFN $\tilde{M}$ is represented as $(l, m, u)$ in which the parameters $l$ and $u$ represents the two end points and denote the lowest possible value, and the upper possible value respectively. While $m$ denotes the most promising value that describes the fuzzy event (i.e., when $l = m = u$, the fuzzy number becomes a real number).

To formalize the usage of TFNs in order to model the

[4]A research from the European Commission [17] specified that SLAs are required by the CSPs to establish their credibility, attract or retain customers since they can be used as a mechanism for service differentiation in the Cloud market.

[5]A fuzzy number defines a fuzzy interval in the real number, in the sense that it does not refer to one single value but rather to a connected set of possible values, where each possible value has its own membership function between 0 and 1.

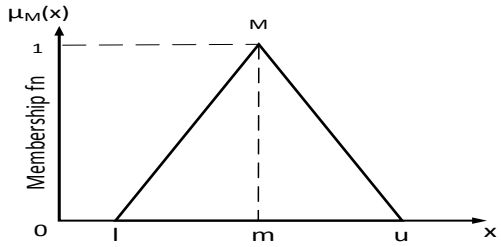


Figure 3. Triangular fuzzy number.

customers vague requirements, we extend Definition 1 by the following definition.

*Definition 2:* Each SLO $k_i$ in a secSLA consists of $m$ different values $\tilde{v}_i$. Each value implies a different security level offered by the CSP and required by the customer. Each value $\tilde{v}_i$ is specified using TFN $(l_i, m_i, u_i)$; such that $k_i = \{\tilde{v}_{i,1}, \tilde{v}_{i,2}, \ldots, \tilde{v}_{i,j}\}$. The total order of security levels is defined using an order relation "$\prec_i$"; such that $k_i = \tilde{v}_{i,1} \prec \tilde{v}_{i,2} \prec \ldots \prec \tilde{v}_{i,j}$.

In this paper, three cases for the TFN representation of the SLO values are considered. These cases are considered according to the type of customer requirements as specified at Stage A (cf., Section IV-A).

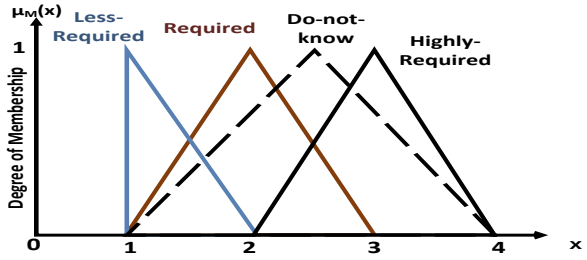


Figure 4. Linguistic terms for criterion importance.

**Linguistic terms:** are represented as TFNs and specified using their membership functions as depicted in Figure 4.

The proposed methodology allows the customers to: i) assign linguistic terms at varied levels of the hierarchical specification, and ii) individually adjust the linguistic terms according to their requirements. To further ease the task, especially for novice customers, the system can set default values for each linguistic requirement as shown in Figure 4 and Table I. In our model, we set the range of membership functions between 0 and 4 but as specified earlier this range is up to the user definition.

We represent *Not-Required (NR)* as $(0, 0, 0)$ as it is not required by the customer and *Do-not-know* which specifies customer uncertain requirements as a TFN that can have all possible ranges from 1 to 4, thus we define it as $(1, 2.5, 4)$, which means the most promising value is 2.5.

Table I
LINGUISTIC TERMS AND VALUES OF RATINGS.

| **Linguistic scale for importance** | **TFN** $(l, m, u)$ |
|---|---|
| Less-Required (LR) | $(1, 1, 2)$ |
| Required (R) | $(1, 2, 3)$ |
| Highly-Required (HR) | $(2, 3, 4)$ |
| Not-Required (NR) | $(0, 0, 0)$ |
| Do-not-know (Dk) | $(1, 2.5, 4)$ |

**Natural language statements:** are mapped into values representing the customer requirement. This can be done only if we have a consistent set of statements for a given language. In that case, every statement is mapped to the appropriate parameter that yields to a value function that represents the customer requirement. We define this set of statements using *if-then* rules, where the customer should specify the required *service name* and her/his *preferences* in her/his statement. Using fuzzy search algorithm[6] based on Levenshtein distance, we get the required service from the list of secSLA services and the needed preference. For example, "I highly require a provider encrypting data in transit and at rest" statement is converted to an *if-then* rule at the end of this stage, such that: **If** *data encryption in transit* $(s_1)$ is *highly required* AND *data encryption at rest* $(s_2)$ is *highly required* **then** $\tilde{v}_{s_1}$ is *HR* AND $\tilde{v}_{s_2}$ is *HR*.

Thus, at the end of this stage the customer requirements are assigned as linguistic terms and represented using TFNs. Multiple rules specifying different security controls can be combined using AND or OR operators. In addition, if a customer does not specify or does not care of other services, by default these services are set to *Less-Required*.

**Precise values** assigned as fixed values to each SLO $k_i$. These values are mapped to a progressive TFN values however $l = m = u$[7].

### C. Stage C. Security Evaluation Based on Fuzzy-AHP

The assessment approach is utilized so that, the CSPs are ranked (as per their secSLA's) for the best match to the customer requirements. The challenge is not only how to quantify different secSLA services, but also how to aggregate them in a meaningful metric. Multiple Criteria Decision Making (MCDM) [18] methods such as the Analytic Hierarchy Process (AHP) [19] are used to solve these issues. The advantages of AHP over other methods are its ability to (1) identify inconsistencies across requirements, and (2) handle composite qualitative and quantitative attributes [20]. AHP uses a pairwise comparisons for evaluating the alternatives. However according to [21], the AHP method is: (1) mainly

[6]Fuzzy search algorithm as well as basic notations used throughout the paper covering fuzzy and crisp sets, membership functions, operations of TFN, and a brief explanation of Chang's extent analysis on fuzzy-AHP are publicly available at https://github.com/amtaha/fuzzyQHP.

[7]Fixed values are mapped to TFN in order to allow our model to automatically rank the CSPs according to different customer specifications. We demonstrate that using a real-world case study in Section V.

used in nearly fixed decision applications and (2) it is not able to reflect the decision makers' uncertain preferences through fixed values.

In this paper, fuzzy-AHP is used to relieve the uncertainty and inability of the AHP in handling uncertain prefrences. It allows a more accurate description of the decision-making process, where fuzzy set theories are used to express the uncertain requirements and prefrences as fuzzy numbers. As an overview of our evaluation framework, the secSLA assessment and the ranking of CSPs are performed in the following progressive phases.

*1) Phase 1. Weights Assignment:* In order to compare two CSPs' security SLOs, the relative importance level of the customer requirements for each security SLO should be assigned as weights. We utilize the qualitative terms defined in Table I to specify the importance of each SLO. These qualitative terms are mapped to quantitative values and assigned as normalized numbers to satisfy the fuzzy-AHP requirements. The proposed framework allows the customers to assign qualitative weights at varied levels of the secSLA hierarchy structure.

*2) Phase 2. Services Quantification:* In order to assess each CSP secSLA, a quantification model for different SLOs should be defined. We use the AHP relative ranking model based on a pairwise relation of the services (a) provided by different CSPs, and (b) required by customers' such that:

$$CSP_{1,k_i}/CSP_{2,k_i} = \frac{\tilde{v}_{1,k_i}}{\tilde{v}_{2,k_i}} \tag{1}$$

where $CSP_{1,k_i}/CSP_{2,k_i}$ indicates the relative rank of $CSP_1$ over $CSP_2$, regarding $k_i$. such that:

$$\begin{aligned} CSP_{1,k_i}/CSP_{2,k_i} &= (1,1,1) && if && \tilde{v}_{1,k_i} \equiv \tilde{v}_{2,k_i} \\ &= \frac{(l_1,m_1,u_1)}{(l_2,m_2,u_2)} && if && \tilde{v}_{1,k_i} \not\equiv \tilde{v}_{2,k_i} \end{aligned}$$

The TFN division is defined as [22]:

$$\frac{(l_1,m_1,u_1)}{(l_2,m_2,u_2)} = (l_{12},m_{12},u_{12}) = (\frac{l_1}{u_2},\frac{m_1}{m_2},\frac{u_1}{l_2}) \tag{2}$$

Similarly, $CSP_{j,k_i}/CSC_{k_i}$ indicates the relative rank of $CSP_j$ over Cloud Service Customer $CSC$, which specifies if $CSP_j$ satisfies the customer requirements, with respect to $k_i$. This pairwise comparisons result in a one to one comparison matrix $\tilde{A}$ for each SLO. The comparison matrix is of size $(n+1)\times(n+1)$ considering $n$ CSPs and one CSC for each SLO such that:

$$\tilde{A}_{k_i} = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n \\ n+1 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & \dots & n & n+1 \end{array} \\ \begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1n} & \tilde{a}_{1u} \\ \tilde{a}_{21} & \tilde{a}_{22} & \dots & \tilde{a}_{2n} & \tilde{a}_{2u} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & \tilde{a}_{nn} & \tilde{a}_{nu} \\ \tilde{a}_{u1} & \tilde{a}_{u2} & \dots & \tilde{a}_{un} & \tilde{a}_{uu} \end{pmatrix} \end{array} \tag{3}$$

where $\tilde{a}_{ij} = CSP_i/CSP_j$. For example, $\tilde{a}_{12} = CSP_1/CSP_2$ regarding SLO $k_i$, which indicates the relative rank of $CSP_1$ over $CSP_2$. Note that, $\tilde{a}_{ij} = \frac{1}{\tilde{a}_{ji}} = (\frac{1}{u_{ji}},\frac{1}{m_{ji}},\frac{1}{l_{ji}})$ (cf., Equation 2).

Next, the respective scores for all the CSPs and the customer, for each SLO, are obtained by calculating the priority vector (PV) of the corresponding fuzzy comparison matrix $\tilde{A}_{k_i}$. There are several procedures to attain PV in fuzzy-AHP. The widely used Chang's extent analysis method [23] is the one utilized in this paper. The PV indicates the numerical ranking of all the CSPs by specifying an order of preference among them, as indicated by the ratios of the numerical values.

$$PV_{k_i} = \begin{array}{c} \begin{array}{ccccc} CSP_{1,k_i} & CSP_{2,k_i} & \dots & CSP_{n,k_i} & CSC_{k_i} \end{array} \\ \begin{pmatrix} N_{1,k_i} & N_{2,k_i} & \dots & N_{n,k_i} & N_{u,k_i} \end{pmatrix} \end{array} \tag{4}$$

Such that $N_{1,k_i}$ is a numerical value representing the relative rank of $CSP_1$ to other CSPs and the CSC regarding SLO $k_i$. Similarly, $N_{u,k_i}$ is the relative rank of the CSC required security level with respect to the security levels offered by the CSPs.

*3) Phase 3. Services Aggregation:* In this phase, a bottom up aggregation is performed in order to give an overall assessment of the security levels and a final ranking of the CSPs. To achieve that, the priority vector of each SLO is aggregated with their relative normalized weights assigned in Phase 1 using the weighted arithmetic means approach. This aggregation process is performed to all the SLOs in the hierarchy with their relative weights. This results in an overall ranking of CSPs according to the customer defined requirements and weights.

$$PV_{aggregated} = \begin{bmatrix} PV_{k_1} & \dots & PV_{k_n} \end{bmatrix} . \begin{bmatrix} W \end{bmatrix}^T \tag{5}$$

Where $W$ is the set of normalized weights of different SLOs such that $W = w_{k_1}, w_{k_2}, \dots, w_{k_n}$. Note that the weights are normalized to satisfy the fuzzy-AHP requirements. $PV_{k_1}$ is the PV calculated for SLO $k_1$. We demonstrate and validate the framework presented in this section using a real-world case study in Section V.

## V. Case Study: Selecting a CSP Based on its secSLA Data

This section shows an empirical validation of the proposed framework through a scenario that uses real world secSLA information provided by the STAR repository. The rationale for this decision is that (i) to the best of our knowledge there are not other publicly available Cloud secSLA repositories, (ii) most CSPs will not provide their secSLA information to non-customers, and (iii) major CSPs are still in the process

Table II
CASE STUDY: EXCERPT OF CSPS SECSLAS AND CUSTOMER REQUIREMENTS.

| Cloud secSLA based on STAR [6] | | | | Cloud Service Providers | | | Cloud Service Customer (CSC) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Services | | | SLO | $CSP_1$ | $CSP_2$ | $CSP_3$ | Case I | | Case II | | Case III | |
| category | category | category | | | | | req | prio | req | prio | req | prio |
| Root | Compliance CO | Audit planning CO1 | CO1.1 | $yes$ | $yes$ | $yes$ | $yes$ | LI | HR | HI | HR | HI |
| | | | CO1.2 | $level_3$ | $level_2$ | $level_4$ | $level_4$ | HI | | | | |
| | | Independent audits CO2 | CO2.1 | $no$ | $yes$ | $yes$ | $yes$ | HI | HR | HI | | |
| | | | CO2.2 | $yes$ | $yes$ | $yes$ | $yes$ | HI | | | | |
| | | | CO2.3 | $yes$ | $yes$ | $yes$ | $yes$ | HI | | | | |
| | | | CO2.4 | $yes$ | $yes$ | $yes$ | $yes$ | HI | | | | |
| | | Third-party audits CO3 | CO3.1 | $yes$ | $yes$ | $yes$ | $yes$ | MI | NR | NI | | |
| | | | CO3.2 | $yes$ | $yes$ | $yes$ | $yes$ | MI | | | | |
| | | | CO3.3 | $level_3$ | $level_2$ | $level_4$ | $level_4$ | MI | | | | |
| | Security architecture SA | Data security SA1 | SA1.1 | $no$ | $no$ | $yes$ | $no$ | NI | Dk | LI | NR | NI |
| | | | SA1.2 | $yes$ | $no$ | $yes$ | $yes$ | HI | | | | |
| | | Application security SA2 | SA2.1 | $yes$ | $yes$ | $yes$ | $yes$ | HI | | | | |
| | | | SA2.2 | $level_3$ | $level_2$ | $level_3$ | $level_3$ | HI | | | | |
| | Resiliency RS | Business testing RS1 | RS1.1 | $weekly$ | $weekly$ | $weekly$ | $weekly$ | LI | $weekly$ | EI | | |
| | | | RS1.2 | $yes$ | $yes$ | $yes$ | $no$ | NI | $yes$ | LI | | |

of restructuring their SLAs by leveraging the recently published ISO/IEC 19086[8]. This scenario demonstrates how a Cloud customer can apply the framework presented in this paper to compare side-by-side three different CSPs based on their advertised secSLAs (compliant with the ISO/IEC 19086 standard). Note that,

- The qualitative SLOs are specified as security levels such as $monthly$, $weekly$, and $daily$ are denoted as security levels $level_1, level_2, level_3$ and then modeled as TFN values $\tilde{1}, \tilde{2}, \tilde{3}$ (cf., Definition 2). Furthermore, $no$ and $yes$ are denoted as $\tilde{0}$ and $\tilde{3}$ respectively.
- In "Case I", all CSPs security SLOs are normalized to the customer requirements to eliminate masquerading [2].
- Qualitative terms assigned by customers to specify the importance of each SLO (cf., Section IV-C1) are transformed to quantitative values and assigned as normalized numbers to satisfy the fuzzy-AHP requirements such that: *HI, NI* indicate a relative value 1 and 0 respectively. *MI* and *LI* can be considered any intermediate values between 1 and 0. In this analysis *MI* and *LI* indicate a relative rank of *0.6* and *0.2* respectively.

Table II shows three sets of Cloud customer requirements used as baseline for comparing the selected CSPs. For validation purposes the Cloud Service Customer (CSC) requirements are being expressed at different levels of granularity, according to the customer expertise:

- In column "Case I", customer requirements and priorities are expressed at the lowest level (i.e., SLO level). This case represents a security expert customer.
- Column "Case II" shows customer specifying (i) linguistic terms at different levels of the secSLA hierarchy and (ii) detailed specification for other SLOs at the lowest level. Linguistic terms are specified as TFN as depicted in Figure 4. This case represents a semi-expert customer.
- Finally, in column "Case III", customer requirements are specified using natural language statements. This case represents a novice customer.

[8]The ISO/IEC 19086 standard has been published Q3/2016, and it is expected to be adopted by major CSPs (and probably certifiable) by Q2/2018

### A. Cloud Customer Case I Requirements: Expert Customer

The customer specifies his/her requirements at the lowest level of the secSLA (i.e., SLOs) and assigns different weights for all of these SLOs as shown in Table II. In this case, each SLO $k_i$ value is mapped to a progressive TFN value according to its order such that $k_i = \tilde{1} \prec \tilde{2} \prec \ldots \prec \tilde{j}$. These TFN values are represented as $k_i = (1,1,1) \prec (2,2,2) \prec \ldots \prec (j,j,j)$.

For the *Compliance* control of Cloud secSLA, there are three security controls (*CO1, CO2, and CO3*), which are further divided into SLOs (*CO1.1, CO1.2, CO2.1,...*). For *CO1.2* the providers and the customer can specify their values from $level_1$ to $level_4$. Using the data shown in Table II, Equation 1 is used to define the *CO1.2* pairwise relation such that:

$$CSP_1/CSP_2 = \frac{\tilde{3}}{\tilde{2}} = (\frac{3}{2}, \frac{3}{2}, \frac{3}{2}), CSC/CSP_3 = (1,1,1)$$

Thus, the comparison matrix of *CO1.2* $\tilde{A}_{CO1.2}$ as specified in Equation 3 is: $\tilde{A}_{CO1.2} =$

$$\begin{array}{c} \\ CSP_1 \\ CSP_2 \\ CSP_3 \\ CSC \end{array} \begin{array}{c} \begin{array}{cccc} CSP_1 & CSP_2 & CSP_3 & CSC \end{array} \\ \begin{pmatrix} (1,1,1) & (\frac{3}{2},\frac{3}{2},\frac{3}{2}) & (\frac{3}{4},\frac{3}{4},\frac{3}{4}) & (\frac{3}{4},\frac{3}{4},\frac{3}{4}) \\ (\frac{2}{3},\frac{2}{3},\frac{2}{3}) & (1,1,1) & (\frac{2}{4},\frac{2}{4},\frac{2}{4}) & (\frac{2}{4},\frac{2}{4},\frac{2}{4}) \\ (\frac{4}{3},\frac{4}{3},\frac{4}{3}) & (\frac{4}{2},\frac{4}{2},\frac{4}{2}) & (1,1,1) & (1,1,1) \\ (\frac{4}{3},\frac{4}{3},\frac{4}{3}) & (\frac{4}{2},\frac{4}{2},\frac{4}{2}) & (1,1,1) & (1,1,1) \end{pmatrix} \end{array}$$

Then, using Chang's extent analysis method, we get the relative ranking of the Cloud providers for *CO1.2*, which is given by the priority vector of $\tilde{A}_{CO1.2}$ ($PV_{CO1.2}$).

$$PV_{CO1.2} = \begin{array}{c} \begin{matrix} CSP_1 & CSP_2 & CSP_3 & CSC \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0.5 & 0.5 \end{pmatrix} \end{array}$$

This means that only $CSP_3$ equally satisfies $CSC$'s *CO1.2* requirement. However, $CSP_1$ and $CSP_2$ do not fulfill that requirement. Similarly, the priority vector of *CO1.1* is calculated using its comparison matrix $\tilde{A}_{CO1.1}$. The *CO1* priority vector is then premeditated by aggregating $PV_{CO1.1}$ and $PV_{CO1.2}$ with customer-defined normalized weights ($w_{CO1}$) using Equation 5. As specified earlier, in Case I the customer assigns *LI* and *HI* for *CO1.1* and *CO1.2* respectively such that:

$$PV_{CO1} = \begin{array}{c} \\ CSP_1 \\ CSP_2 \\ CSP_3 \\ CSC \end{array} \begin{array}{c} \begin{matrix} PV_{CO1.1} & PV_{CO1.2} \end{matrix} \\ \begin{pmatrix} 0.25 & 0 \\ 0.25 & 0 \\ 0.25 & 0.5 \\ 0.25 & 0.5 \end{pmatrix} \end{array} \cdot \begin{array}{c} w_{CO1} \\ \begin{pmatrix} 0.167 \\ 0.833 \end{pmatrix} \end{array}$$

Thus,

$$PV_{CO1} = \begin{array}{c} \begin{matrix} CSP_1 & CSP_2 & CSP_3 & CSC \end{matrix} \\ \begin{pmatrix} 0.042 & 0.042 & 0.458 & 0.458 \end{pmatrix} \end{array}$$

CO2, CO3, SA1, SA2, and RS1 priority vectors are calculated the same way. Finally, the priority vectors of CO, SA, and RS are aggregated to obtain the total secSLA priority vector, as shown in Figure 5.

$$PV_{total} = \begin{array}{c} \begin{matrix} CSP_1 & CSP_2 & CSP_3 & CSC \end{matrix} \\ \begin{pmatrix} 0.234 & 0.158 & 0.304 & 0.304 \end{pmatrix} \end{array}$$

Consequently, $CSP_3$ is the only provider who fulfills the customer's requirements, as shown in Figure 5. That was expected, as $CSP_1$ is not offering *CO2.1* and therefore is under-provisioning *CO1.2* and *CO3.3*. $CSP_2$ is not providing *SA1.2* and is not fulfilling customer requirements for *CO1.2*, *CO3.3*, and *SA2.2*. Only $CSP_3$ fulfills customer's requirements and, as a result, $CSP_3$ is the best matching provider according to the customer's requirements, followed by $CSP_1$ then $CSP_2$, as shown in Figure 5.

### *B. Cloud Customer Case II Requirements: Different Levels of Granularity*

The customer specifies his/her requirements using linguistic descriptors (depicted in Figure 4) at different levels of the secSLA hierarchy as shown in column "Case II" in Table II. Furthermore, the customer submit qualitative labels to specify the required level of importance.
Since *Audit planning* is assigned *HR*, the respective SLOs value of both *CO1.1* and *CO1.2* are set to $(2, 3, 4)$. For

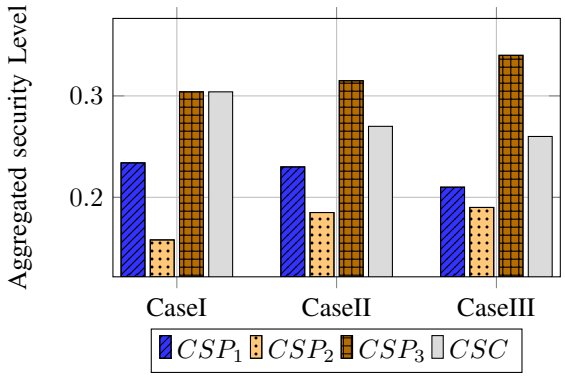


Figure 5. The total aggregated secSLA level with respect to customer requirements.

*CO1.2* the pairwise relation is defined using Equation 1 such that:

$$CSP_3/CSC = (\frac{4}{4}, \frac{4}{3}, \frac{4}{2}),\ CSC/CSP_1 = (\frac{2}{3}, \frac{3}{3}, \frac{4}{3})$$

Then after specifying $\tilde{A}_{CO1.2}$, we use Chang's extent analysis method to get the priority vector $PV_{CO1.2}$, so that:

$$PV_{CO1.2} = \begin{array}{c} \begin{matrix} CSP_1 & CSP_2 & CSP_3 & CSC \end{matrix} \\ \begin{pmatrix} 0.181 & 0 & 0.55 & 0.269 \end{pmatrix} \end{array}$$

$PV_{CO1}$ is then calculated by aggregating $PV_{CO1.1}$ and $PV_{CO1.2}$. Similarly the CO2 SLOs value are set to $(2, 3, 4)$ and the CO3 SLOs are denoted as *NR*. This means that CO3 will not affect the overall security level of the providers as it is not required by the customer. For SA, the customer specifies *Do-not-know*, which is assigned as $(1, 2.5, 4)$. Finally, $PV_{CO}$, $PV_{SA}$, and $PV_{RS}$ are aggregated to obtain the total secSLA priority vector, as shown in Figure 5.

$$PV_{total} = \begin{array}{c} \begin{matrix} CSP_1 & CSP_2 & CSP_3 & CSC \end{matrix} \\ \begin{pmatrix} 0.23 & 0.185 & 0.315 & 0.27 \end{pmatrix} \end{array}$$

Therefore, only $CSP_3$ satisfies the customer needs, whereas both $CSP_1$ and $CSP_2$ do not fulfill customer requirements, as shown in Figure 5.

### *C. Cloud Customer Case III Requirements: Natural Language Sentence*

The customer requires "a provider with only extremely high *Compliance* and does not require *Security Architecture* and *Resiliency*". Using "if-then" rule customer allocates *Highly-Required* for *Compliance*, and *Not-Required* for *Security architecture* and *Resilience*. Similarly, as shown in previous cases, the total priority vector is calculated:

$$PV_{total} = \begin{array}{c} \begin{matrix} CSP_1 & CSP_2 & CSP_3 & CSC \end{matrix} \\ \begin{pmatrix} 0.21 & 0.19 & 0.34 & 0.26 \end{pmatrix} \end{array}$$

Consequently, only $CSP_3$ is satisfying the customer requirements, followed by $CSP_1$ then $CSP_2$. Therefore, the presented framework can give accurate CSPs ranking even if the customer only specified requirements using natural language sentences.

**Note:** As part of our research, we developed a prototype for a control panel that implements the proposed model. The implementation is publicly available at https://github.com/amtaha/fuzzyQHP.

## VI. Conclusion

We propose a security evaluation framework that uses fuzzy logic scheme for CSP selection. The contributions of our framework are: 1) allowing customers, both novice and expert, to submit their requirements by modeling their vague requirements and uncertain preferences with linguistic descriptors and 2) enabling the assessment and benchmarking of CSPs according to the customers' fuzzy and subjective requirements. We employ membership functions to capture customers' uncertain requirements and use a fuzzy inference system to derive precise security levels for these requirements. Additionally, our system selects the CSP that best satisfies the customer requirements using a fuzzy Analytic Hierarchy Process (AHP) assessment technique. As a future work, we plan to develop selection policies for Cloud customers and brokers based on our CSPs evaluation to further automate Cloud deployment according to the customers' vague requirements.

## Acknowledgment

Research supported, in part by, EC H2020 ESCUDO-CLOUD GA #644579 and CIPSEC GA #700378

## References

[1] A. Taha, P. Metzler, R. Trapero, J. Luna, and N. Suri, "Identifying and utilizing dependencies across cloud security services," *Proc. of AsiaCCS*, pp. 329–340, 2016.

[2] J. Luna, A. Taha, R. Trapero, and N. Suri, "Quantitative reasoning about cloud security using service level agreements," *In Transactions on Cloud Computing*, no. 99, 2015.

[3] K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *In Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

[4] Z. Rehman, F. Hussain, and O. Hussain, "Towards multi-criteria cloud service selection," *Proc. of IMIS*, pp. 44–48, 2011.

[5] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *In Information sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[6] Cloud Security Alliance, "The Security, Trust & Assurance Registry (STAR)," *https://cloudsecurityalliance.org/star/.*

[7] "(Draft) Information Technology - Cloud Computing - Service Level Agreement (SLA) Framework and Terminology," International Organization for Standardization, Tech. Rep. ISO/IEC 19086, 2014.

[8] Z. Rehman, O. Hussain, and F. Hussain, "Iaas cloud selection using mcdm methods," *Proc. of ICEBE*, pp. 246–251, 2012.

[9] H. Alabool and A. Mahmood, "Trust-based service selection in public cloud computing using fuzzy modified vikor method," *In Australian Journal of Basic and Applied Sciences*, vol. 7, no. 9, pp. 211–220, 2013.

[10] T. Noor and Q. Sheng, "Trust as a service: a framework for trust management in cloud environments," *Proc. of WISE*, pp. 314–321, 2011.

[11] S. Habib, S. Ries, and M. Muhlhauser, "Towards a trust management system for cloud computing," *Proc. of TrustCom*, pp. 933–939, 2011.

[12] M. Supriya, L. Venkataramana, K. Sangeeta, and K. Patra, "Estimating trust value for cloud service providers using fuzzy logic," *International Journal of Computer Applications*, vol. 48, no. 19, pp. 28–34, 2012.

[13] V. Casola, A. Mazzeo, N. Mazzocca, and M. Rak, "A SLA evaluation methodology in Service Oriented Architectures," *In Quality of Protection*, pp. 119 – 130, 2006.

[14] S. Chaves, C. Westphall, and F. Lamin, "SLA perspective in security management for cloud computing," *Proc. of Networking and Services*, pp. 212–217, 2010.

[15] A. Taha, R. Trapero, J. Luna, and N. Suri, "AHP-Based Quantitative Approach for Assessing and Comparing Cloud Security," *Proc. of TrustCom*, pp. 284–291, 2014.

[16] J. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proc. of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

[17] "Cloud computing Service Level Agreements-Exploitation of Research Results," European Commission, Tech. Rep., 2013.

[18] M. Zeleny and J. Cochrane, "Multiple criteria decision making," *University of South Carolina Press*, 1973.

[19] T. Saaty, "How to make a decision: the analytic hierarchy process," *In European journal of operational research*, vol. 48, no. 1, pp. 9–26, 1990.

[20] R. Ramanathan, "A note on the use of the analytic hierarchy process for environmental impact assessment," *In Journal of environmental management*, vol. 63, no. 1, pp. 27–35, 2001.

[21] G. Kabir and M. Hasin, "Comparative analysis of ahp and fuzzy ahp models for multicriteria inventory classification," *In International Journal of Fuzzy Logic Systems*, vol. 1, no. 1, pp. 1–16, 2011.

[22] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of granular computing*. John Wiley & Sons, 2008.

[23] D. Chang, "Applications of the extent analysis method on fuzzy ahp," *In European journal of operational research*, vol. 95, no. 3, pp. 649–655, 1996.