Negotiating and Brokering Cloud Resources based on Security Level Agreements

Jesus Luna[1], Tsvetoslava Vateva-Gurova[1], Neeraj Suri[1], Massimiliano Rak[2] and Loredana Liccardo[2]
[1]*Department of Computer Science, Technische Universität Darmstadt, Darmstadt, Germany*
[2]*Dipartimento di Ingegneria dell'Informazione, Seconda Universita' di Napoli, Aversa (CE), Italy*
{*jluna, vateva, suri*}*@deeds.informatik.tu-darmstadt.de*, {*massimiliano.rak, loredana.liccardo*}*@unina2.it*

Keywords: Cloud Security, Security Level Agreements, Security Metrics, Security Negotiation, Resource Brokering

Abstract: Cloud users often motivate their choice of Cloud Service Provider (CSP) based on requirements related with the offered Service Level Agreements (SLA) and costs. Unfortunately, while security has started to play an important role in the decision of using the Cloud, it is quite uncommon for CSPs to specify the security levels associated with their services. This often results in users without the means (i.e., tools and semantics) to negotiate their security requirements with CSPs, in order to choose the one that best suits their needs. However, the recent industrial efforts on specification of Cloud security parameters in SLAs, also known as "Security Level Agreements" or SecLAs is a positive development. In this paper we propose a practical approach to enable the user-centric negotiation and brokering of Cloud resources, based on both the common semantic established by the use of SecLAs and, its quantitative evaluation. The contributed techniques and architecture are the result of jointly applying the security metrology-related techniques being developed by the EU FP7 project ABC4Trust and, the framework for SLA-based negotiation and Cloud resource brokering proposed by the EU FP7 mOSAIC project. The proposed negotiation approach is both feasible and well-suited for Cloud Federations, as demonstrated in this paper with a real-world case study. The presented scenario shows the negotiation of a user's security requirements with respect to a set of CSPs SecLAs, using both the information available in the Cloud Security Alliance's "Security, Trust & Assurance Registry" (CSA STAR) and the WS-Agreement standard.

1 INTRODUCTION

While the many economic and technological advantages of Cloud computing are apparent, the migration of key security relevant applications onto it has been limited, in part, due to the lack of accountable *security assurance* specification provided by the Cloud Service Provider. Furthermore, the typical Cloud user is not a security expert, though nevertheless has specific security requirements to fulfill (e.g., due to regulatory compliance) that are usually expressed at an informal level, thus making them difficult and expensive to align and negotiate with respect to the CSP's security offer. Unfortunately, at the state of practice (e.g., as discussed by the Cloud Security Alliance's SLA WG (Cloud Security Alliance, 2012)), many Cloud users find themselves without the means to match and further negotiate their security requirements with available CSPs. Contrary to Cloud resource negotiation based on non-security indicators (e.g., using performance metrics as presented in (Rak M., *et. al.*, 2011b)), the field of security-based negotiation presents several challenges mainly due to both the lack of security assurance/quantifiers, and the *semantic gap* among users and CSPs with respect to security.

Fortunately, security negotiation in Cloud computing has recently taken some initial and promising steps. Early academic works like (Kandukuri B.R., *et. al.*, 2009) and, the Cloud community (e.g., workgroups at the European Network and Information Security Agency (ENISA) (Dekker M. and Hogben G., 2011)) have identified that specifying security parameters in Service Level Agreements (termed as "Security Level Agreements" or SecLA over this paper) actually enables the establishment of a common semantic in order to model security among users and CSPs. However, despite the state of the art efforts aiming at building and representing Cloud SecLAs (e.g., the CSA's SLA and PLA working groups (Cloud Security Alliance, 2012)), there is still a gap on the techniques to *reason* about them. In particular we refer to the techniques aimed to quantitatively evaluate the security level provided by the SecLA, this being a core requirement to enable the proposed negotiation of Cloud resources based on security parameters (just as presented in our previous research (Luna J., *et.al.*, 2012b) and (Luna J., *et.al.*, 2012a)).

This paper proposes a novel methodology and architecture to systematically broker Cloud resources based on *(i)* a technique to quantitatively evaluate and rank SecLAs and, *(ii)* a set of building blocks to enable the user-centric negotiation of Cloud security parameters. Our joint research contributes with a practical approach that extends the Cloud SecLA evaluation technique contributed by Luna (Luna J., *et.al.*, 2012a), to enable the negotiation and brokerage of Cloud resources presented in (Rak M., *et. al.*, 2011b) using the well-known WS-Agreement protocol (Andrieux K., *et.al.*, 2007).

The overall vision of this paper is represented in Figure 1, where an iterative *Negotiation* process (Step 1) quantifies and ranks the user security requirements (i.e., represented as a *User SecLA*), with respect to one or more CSP SecLAs (Step 1b). Once an existing *CSP SecLA* offer matches the *User SecLA*, then an *Enforcement* stage takes place i.e., the broker acquires and delivers CSP resources to user (Step 2). This paper is focused on developing the details related with the underlying negotiation stage, whereas the resource brokering is out of scope. Notice that both the Negotiation (Step 1) and Evaluation (Step 1b) stages are not completely independent: security negotiation needs the quantitative evaluation of SecLA in order to rank the available CSP with respect to a user requirement, whereas the evaluation stage applies a user-defined negotiation criteria in order to classify the CSPs' security features.

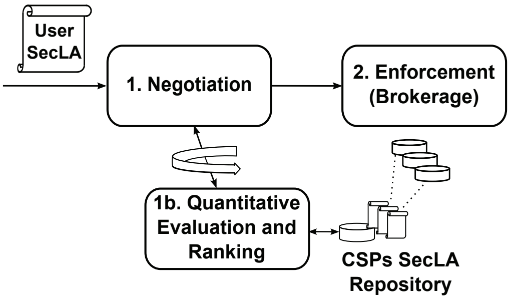
User SecLA
1. Negotiation
2. Enforcement (Brokerage)
1b. Quantitative Evaluation and Ranking
CSPs SecLA Repository

Figure 1: Overview of the proposed security negotiation and brokerage of Cloud resources.

This paper also proposes an architecture to implement the presented SecLA negotiation methodology in the context of a Cloud Federation, using the framework being developed by the EU FP7 mOSAIC project (mOSAIC, 2011). This framework enables the creation of distributed Cloud applications through a set of components based on the management and negotiation of Service Level Agreements.

Finally, to demonstrate the feasibility of our proposal we present the case study of a real system that implements the negotiation of a *User SecLA* with respect to a set of *CSPs SecLAs*, based on the information available in STAR (Cloud Security Alliance, 2011b) and the WS-Agreement standard (Andrieux K., *et.al.*, 2007).

The paper is organized as follows: Section 2 introduces the basic concepts behind the proposed security negotiation approach, Section 3 describes how Cloud SecLAs can be specified using WS-Agreement in order to enable its negotiation via the architecture presented in Section 4. A real case study that applies the proposed approach is discussed in Section 5, Section 6 analyzes related works and, finally Section 7 presents our conclusions.

2 Quantitatively evaluating and ranking Cloud SecLAs

Two base concepts driving our proposal are presented in this section. First, we discuss in further detail the notion of Cloud SecLAs (cf. Section 2.1). Second, in Section 2.2 are presented the basics of a technique to reason about SecLA, in particular to enable its quantitative evaluation as required by the negotiation process presented in this paper.

2.1 Cloud Security Level Agreements

The concept of SecLAs currently exists in varied dimensions and the Cloud is not an exception. The use of Cloud SecLAs has the potential to provide tangible benefits to CSPs especially associated with improved security administration and management practices, thus allowing for transparency to end users. The end users can also benefit from SecLAs by understanding the costs and benefits associated with this new service model. The importance of Cloud SecLAs has also been recognized in a recent study by ENISA (Dekker M. and Hogben G., 2011), showing that while SLAs are often used, and availability is often addressed in these SLAs, other security parameters (e.g., related with the confidentiality and integrity properties) are less well covered. As introduced by Bernsmed (Bernsmed K., *et.al.*, 2011), a Cloud SecLA usually models the CSP security at the *service level* and in practice, the content of these *Cloud SecLAs Templates* is designed by multi-disciplinary working groups (e.g., the Cloud Security Alliance's

SLA and PLA work groups (Cloud Security Alliance, 2012)). The result is an organized collection of security statements (also called "security provisions") in the form $\{$*security attribute, value*$\}$ (e.g., $\{$*Backup Frequency, Daily*$\}$ and $\{$*Encryption Key Size, 512 bits*$\}$), as also proposed in different industrial and academic works (Casola V., *et.al.*, 2006), (Casola et al., 2005), (Samani R., *et.al.*, 2011) and (Luna J., *et.al.*, 2011). In order to be manageable, these security provisions are usually organized into "hierarchical categories" derived from a taxonomy e.g., Savola (Savola R., *et.al.*, 2010) or the CSA's Consensus Assessment Initiative Questionnaire (CAIQ) (Cloud Security Alliance, 2011a). Cloud SecLAs are usually stored in publicly available – and trusted – repositories like e.g., the CSA's "Security, Trust & Assurance Registry" (Cloud Security Alliance, 2011b). Apart from the challenges related with the creation of SecLAs in real Cloud deployments, the current paucity of techniques to *quantitatively reason* about them has proven to be part of the obstacles in using SecLAs, just as mentioned by Almorsy (Almorsy M., *et.al.*, 2011) and (Luna J., *et.al.*, 2012b). In order to contribute towards bridging this gap, the next section presents the basics of a Cloud SecLA evaluation technique which will be used by the negotiation mechanism proposed later in this paper.

2.2 Quantitative SecLA evaluation at a glance

For the contributed negotiation process (cf., Step 1 in Figure 1) it is helpful to have a user-centric mechanism to quantitatively evaluate and objectively rank SecLAs with respect to a predefined user requirement (Step 1b). Our approach extends the notion of Cloud SecLA benchmarking proposed in (Luna J., *et.al.*, 2012a), through the use of quantitative rankings, as an enabler of the proposed negotiation process. The overall intent is to *(a)* systematically quantify the security level associated with each SecLA involved in the negotiation process (i.e., *User SecLA* and *CSPs SecLA*) and *(b)* use the data from *(a)* to allow the systematic elicitation of the CSP that is closer to the user's security requirements. For the purposes of this paper, only three basic concepts of the SecLA evaluation are presented and the interested readers are referred to (Luna J., *et.al.*, 2012a) for further details.

SecLAs have a twofold use: on one hand, the authors make the realistic assumption (cf., Section 2.1) that each CSP is associated with a *CSP SecLA*, on the other hand, they also advocate for the use of SecLAs to represent security requirements of Cloud users thus *establishing a common semantic with CSPs for reasoning about security*. User-defined requirements (termed as *User SecLA*) are a distinctive element of Cloud SecLA, where all the security provisions are *weighted* in order to represent their relative importance from user's perspective (e.g., for some users "Encryption Key Size" might be more important than "Backup Frequency"). Furthermore, for the sake of usability the technique proposed in (Luna J., *et.al.*, 2012a) also considers that either quantitative weights (e.g., from 0 to 1) or qualitative weights (e.g., low/medium/high) can be assigned at different levels of the *User SecLA*.

The second concept is a mapping process that allows representing any Cloud SecLA (usually these documents are informally formatted) as a data structure that can be systematically processed. These data structures (called "Quantitative Policy Trees" or QPT in (Luna J., *et.al.*, 2012a)), are an extended version of classical "AND-OR" trees used to integrate both security requirements and associated quantifiers needed by the quantitative evaluation process. For the purposes of this paper and due to space restrictions, only a high-level view of the process to map SecLAs to QPT is presented in Figure 2. The outcomes of the mapping process are *(i)* a *User QPT* populated with the weights and security provisions' values specified in the *User SecLA* and, *(ii)* one or more *CSP QPTs* mapped from its respective *CSP SecLAs* and also populated with the corresponding security provisions' values.

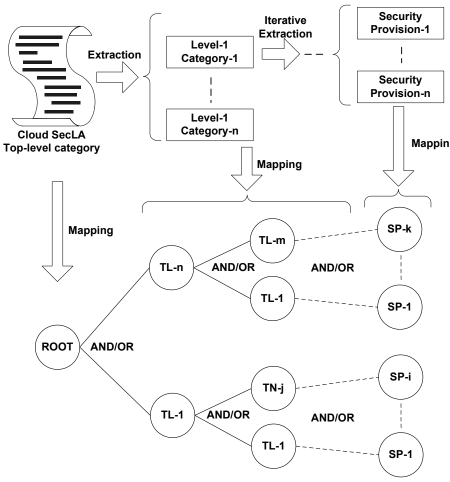

Figure 2: SecLA-to-QPT: mapping a Cloud SecLA into a QPT (Luna J., *et.al.*, 2012a).

The third and final concept required from (Luna J., *et.al.*, 2012a) is a set of rules to quantitatively aggregate and propagate the weights and security provision's values to the whole QPT (i.e., from leaf nodes up to the root node). Once the *User QPT* and the *CSP QPTs* have been populated with the aggregated values, it is possible to apply a ranking algorithm to determine how different CSPs under-/over-provision user's requirements. In (Luna J., *et.al.*, 2012a) are proposed two different ranking techniques: a quantitative ranking (e.g., a real number on the interval $\{0 \ldots 1\}$) that due to its nature is more suitable for automated systems than for humans and, a qualitative ranking that aims to be more "human-friendly" by using a set of qualitative labels (e.g., {*"Copper", "Silver", "Gold"*}) to represent the QPT evaluation's results.

In the rest of this paper, we will show that the previously presented notion of quantitative ranks can be extended to actually negotiate Cloud resources using a broker-based architecture.

3 Creating and specifying SecLAs using WS-Agreement

As introduced in Section 1, the goal of the proposed approach is to offer the systematic negotiation of security using Cloud SecLAs. In order to fulfill this goal our proposal *(i)* creates a set of *CSP SecLA* based on the security information derived from the the CSA STAR repository (Cloud Security Alliance, 2011b) and, *(ii)* represents both *CSP/User SecLA* using the WS-Agreement standard (Andrieux K., *et.al.*, 2007). In this section we discuss in further detail these two phases.

Listing 1: SDT element in WS-Agreement

```
<wsag:ServiceDescriptionTerm
    wsag:Name="
    CustomerAccessRequirements"
wsag:ServiceName="
    SecurityArchitecture">

     </wsag:ServiceDescriptionTerm>
```

First, given CSA STAR's broad adoption by major CSP our research proposes the creation of SecLA derived from the information stored there. Currently, STAR contains entries in the form of "Consensus Assessments Initiative Questionnaire" reports (CAIQ (Cloud Security Alliance, 2011a)), which provide industry-accepted ways to document what security controls exist in Cloud offerings. The current CAIQ report contains a set of 171 security parameters (all of these with a qualitative "YES/NO" answer) distributed in the following *controls*: Compliance (CO) – 14, Data Governance (DG) – 15, Facility Security (FS) – 9, Human Resources Security (HR) – 4, Information Security (IS) – 71, Legal (LG) – 2, Operations Management (OP) – 5, Risk Management (RI) – 12, Release Management (RM) – 5, Resilience (RS) – 11 and Security Architecture (SA) – 23. Given these CAIQ's properties, it is possible to create SecLAs with the features required by the evaluation and ranking methodology presented in Section 2.

Second, to allow the automated negotiation of Cloud SecLA (derived from the CAIQ as mentioned in the previous paragraph), we adopted the SLA-oriented language proposed by WS-Agreement which was created with the goal to standardize the terminology/protocol used when two parties are trying to establish an agreement. It mainly consists of a language for specifying the nature of the agreement and, a SOAP-based protocol for actually establishing the agreement between two participants. At state of art, the WS-Agreement language is widely used for SLA negotiation and has been adopted by projects like EU FP7 mOSAIC (mOSAIC, 2011).

The main component within the WS-Agreement standard is the *SLA specification core*, which consists of three elements: *Service Description Terms (SDT), Service Properties (SP) and, Guarantee Terms (GT).* A SDT is a fundamental element, providing a full or partial functional description of a service. One or more SDTs can be related to a service. A SP element defines properties/variables, associated with a service, and used for expressing guarantees on a service. Finally, a GT element defines an assurance on a service through an assertion (using the content of the SP element) expressed over the service described by the SDTs. In the rest of this section we present the process required to specify a SecLA using the WS-Agreement standard, however due to space restrictions our explanation will only show relevant excerpts of the resulting XML document.

Based on the CAIQ's structure (Cloud Security Alliance, 2011a), first we model each security control as a SDT. Then, the respective value of the controls along with the inputs required by the evaluation technique presented in Section 2 (i.e., the *User SecLA's* AND/OR relationships and weights) are modeled as GTs on the SDT. CAIQ's inherent hierarchical structure (i.e., sub-controls) is represented as security elements in WS-Agreement (cf., Listing 1).

Once the SDT has been specified, we have to focus on the SP element. First, we define a SP for each CAIQ sub-control, as required by its respective SDT (cf., Listing 2). Second, in the SP element we define the variables (e.g., weights) and related semantics used for expressing the security requirements through assertions in the GT element.

Listing 2: SP element in WS-Agreement

```
<wsag:ServiceProperties wsag:Name="
    CustomerAccessRequirements"
wsag:ServiceName="
    SecurityArchitecture">
 <wsag:VariableSet>
  <wsag:Variable wsag:Name="SA-01.1
       " wsag:Metric="boolean">
    <wsag:Location>$this/wsag:Terms
        /wsag:All/
        wsag:ServiceDescriptionTerm
  [@wsag:Name = '
      CustomerAccessRequirements']<
      /wsag:Location>
  </wsag:Variable>
  <wsag:Variable wsag:Name="SA-01.
      Q1" wsag:Metric="string">
    <wsag:Location>$this/wsag:Terms
        /wsag:All/
        wsag:ServiceDescriptionTerm
  [@wsag:Name = '
      CustomerAccessRequirements']<
      /wsag:Location>
  </wsag:Variable>
  <wsag:Variable wsag:Name="Weight"
       wsag:Metric="float">
    <wsag:Location>$this/wsag:Terms
        /wsag:All/
        wsag:ServiceDescriptionTerm
  [@wsag:Name = '
      CustomerAccessRequirements']<
      /wsag:Location>
  </wsag:Variable>
 </wsag:VariableSet>
</wsag:ServiceProperties>
```

For the GT description, the user can define a guarantee on a security service expressed as an assertion in the Service Level Objective (SLO). The data required to model this assertion are: the security level of the CAIQ control, the AND/OR relationships and the weights. A more detailed explanation of the GT element will be presented in Section 5.

Up to this point, we have fully developed the structure of an empty WS-Agreement template "compliant" with the Cloud SecLA evaluation technique summarized in Section 2. Both users and CSPs are now able to populate this template with their own security controls and values, so the final set of SecLA documents can be used for negotiation purposes just as presented in the following section.

4 ARCHITECTURAL MODEL

As outlined in Figure 1, our approach takes into consideration three different aspects (negotiation, evaluation and brokerage) which can be implemented as independent modules. In Rak (Rak M., *et. al.*, 2011b), a clear distinction was made among negotiation and brokerage of Cloud resources in order to propose an architecture to automatically perform these tasks using the framework developed by the EU FP7 mOSAIC project (mOSAIC, 2011). Despite the negotiation process presented by Rak (Rak M., *et. al.*, 2011b) is based on performance parameters (described in terms of measurable metrics), our research extrapolates its results to demonstrate that a similar architecture can be also used to negotiate quantified security parameters in the form of Cloud SecLAs. We propose the three-tier architecture shown in Figure 3, where the intermediate layer is comprised of the following modules: *(i)* the *Negotiation Module* which has the role of managing the interactions with End Users, *(ii)* the *SecLA Evaluation* which implements the technique presented in Section 2.2 and, *(ii)* the *Service Brokering* in charge of brokering the elicited Cloud services. In this section we will focus on the Negotiation and SecLA Evaluation modules, but interested readers are referred to Rak (Rak M., *et. al.*, 2011b) for a full description of the Service Brokering module.

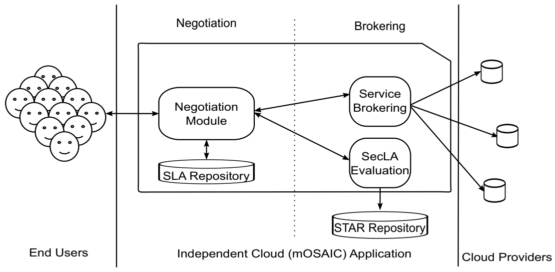

Figure 3: Proposed SecLA negotiation architecture.

4.1 The User-centric Negotiation Protocol

The Negotiation Module interacts with users in order to identify their requirements (i.e., create an *User SecLA*), and applies the results of the quantitative SecLA evaluation/ranking (cf., Section 2.2) in order to orchestrate the overall negotiation process. We assume that users are able to specify their security requirements in the form of a *User SecLA* (even if they are not security experts), using as a guidance the CSA CAIQ reports (Cloud Security Alliance, 2011a). As mentioned in Section 3, the STAR repository (Cloud Security Alliance, 2011b) can be used as a *trusted* source for creating *CSPs SecLAs*. The Cloud SecLA negotiation protocol shown in Figure 4, is the user-centric mechanism we propose to allow end users and CSPs arriving to an agreement on the requested/pro-

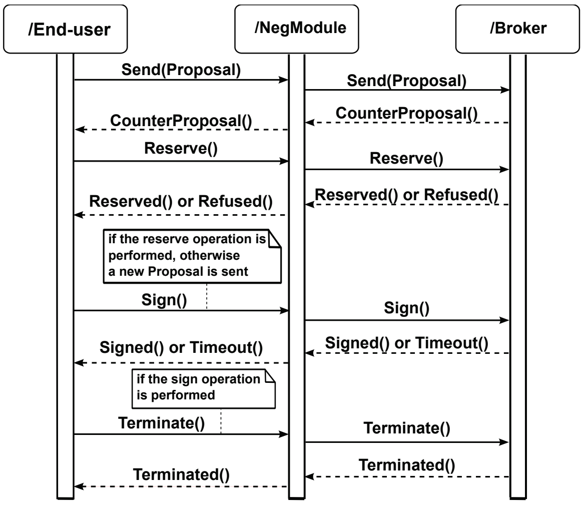

Figure 4: User-centric Cloud SecLA Negotiation Protocol

vided security levels.

The proposed protocol works as follows: once the user has created a *User SecLA* with her specific security requirements, the negotiation process starts by submitting it to the Negotiation Module. Upon reception, the Negotiation Module forwards the *User SecLA* to the SecLA Evaluation module, which implements the technique described in Section 2.2 to quantitatively evaluate and rank the user requirement with respect to a set of available *CSPs SecLAs* (previously fetched from a repository like STAR). As a result, the SecLA Evaluation module returns to the user this set of CSPs, but ordered with respect to the requested *User SecLA* (e.g., quantitatively ranked from best to worst). In this case the user is given the chance to make (either automatically or manually) an informed decision by choosing a CSP from this resulting set.

The user-selected *CSP SecLA* is then submitted to the Broker, so it can be either reserved or refused. If it is refused, then the user has to submit a new proposal (i.e., another CSP SecLA), and if it is reserved then a *timeout* is set for the *CSP SecLA*. This timeout is a pre-defined period of time, which will be allowed to elapse before a user finally agrees on the reserved CSP. Once the user decides to continue with the resource reservation, then an agreement will be signed. Finally, once the reservation ends then the signed agreement is terminated.

The negotiation protocol presented in this subsection can be deployed using both the SLA Framework developed in (Amato A., *et. al.*, 2012) and, the WS-Agreement standard. Although those implementation details are out of the scope of this paper, in the next section we show the feasibility of the proposed approach with a real case study.

5 Case Study: negotiating CAIQ-based security

In order to demonstrate how the proposed negotiation mechanism can be used with real-world information, in this section we present a case study that uses the CSP data stored in CSA STAR (Cloud Security Alliance, 2011b), a publicly available repository that documents the security controls provided by CSPs worldwide. We show that our user-centric negotiation mechanism can use STAR data *(i)* to establish a common semantic with respect to the security offered by the CSP and, *(ii)* to enable Cloud customers automatically choose the CSP that better fulfills their security requirements. As mentioned in Section 3, the STAR repository contains only *static* security controls (i.e., not updated in real-time by the CSP), however the proposed negotiation approach can be easily extended to manage real-time information e.g., generated by continuous security monitoring systems (cf., Section 7).

Listing 3: A Guarantee Term element in WS-Agreement

```
<wsag:GuaranteeTerm wsag:Name="
    UserRequirementOnCustomerAccess"
Obligated="Provider">
  <wsag:ServiceScope ServiceName="
      string">SecurityArchitecture</
      wsag:ServiceScope>

  <wsag:ServiceLevelObjective>
     <wsag:CustomServiceLevel>
      SA-01.1 EQ true, Weight EQ
          0.5) AND (SA-01.Q1 EQ SLA,
           Weight EQ 0.5)
     </wsag:CustomServiceLevel>
  </wsag:ServiceLevelObjective>

  <wsag:BusinessValueList>
      .....
  </wsag:BusinessValueList>
</wsag:GuaranteeTerm>
```

The main goal of this case study is to show the feasibility of our approach, so it has been simplified with respect to the information (i.e., amount of STAR's security controls) being used. Nevertheless, the base negotiation techniques can be applied to more complex case studies (as we will show in future research). In this section we also assume that an user only wants to specify in her *User SecLA* some specific requirements, mostly related to the security mechanisms implemented by the CSP. Using the CAIQ terminology (Cloud Security Alliance, 2011a), this user requirement translates to an *User SecLA* containing only the parameters under the Security Architecture (SA) control (cf., Figure 5).

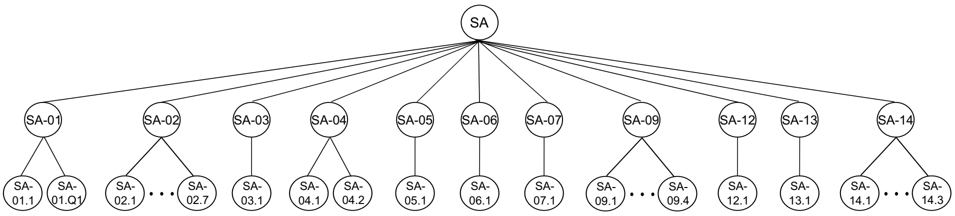

Figure 5: SecLA tree for the Security Architecture (SA) category of the CAIQ reports (Cloud Security Alliance, 2011a).

As mentioned in Section 3, the user expresses her security requirements as a WS-Agreement document that is, in terms of both a service and guarantee terms (cf. Section 3). More specifically, the user can define a GT over a particular service in the form of an assertion in the SLO element. This is the basis for applying the negotiation process described in Section 4. It should be noticed that all three the GT element, the CAIQ and the QPT (cf., Section 2) have the same hierarchical structure, therefore the user can specify her requirements with different levels of granularity. For example, she can define a minimum requirement at the "root" SA control, but also at each one of its "leaf" sub-controls (e.g., SA-01, SA-01.Q1, ...) just as shown in Listings 3 and 4. Furthermore, these excerpts of the WS-Agreement document also show that numeric weights can be assigned to individual controls, in order to represent its relative importance from a user perspective.

Listing 4: A Guarantee Term element in WS-Agreement

```
<wsag:GuaranteeTerm wsag:Name="
    UserRequirementOnSecurityArch"
    Obligated="Provider">
  <wsag:ServiceScope ServiceName="
      string">SecurityArchitecture</
      wsag:ServiceScope>

  <wsag:ServiceLevelObjective>
     <wsag:CustomServiceLevel>
      (CustomerAccessRequirements EQ
           0.4, Weight EQ 0.5)
     AND (UserIDCredentials EQ 0.4,
         Weight EQ 0.5)
     AND (DataSecurityIntegrity EQ
         0.4, Weight EQ 0.5)
     AND (ApplicationSecurity EQ
         0.4, Weight EQ 0.5)
     AND (DataIntegrity EQ 0.4,
         Weight EQ 0.5)
     AND (ProductionEnvironments EQ
         0.4, Weight EQ 0.5)
     AND (RemoteUserMultifactorAuth
         EQ 0.4, Weight EQ 0.5)
     AND (Segmentation EQ 0.4,
         Weight EQ 0.5)
     AND (ClockSynchronization EQ
         0.4, Weight EQ 0.5)
     AND (EquipmentIndentification
         EQ 0.4, Weight EQ 0.5)
     AND (
         AuditLoggingIntrusionDetection
          EQ 0.4, Weight EQ 0.5)
     </wsag:CustomServiceLevel>
  </wsag:ServiceLevelObjective>

  <wsag:BusinessValueList>
        .....
  </wsag:BusinessValueList>
</wsag:GuaranteeTerm>
```

Tables 1 and 2 show the quantitative results of the negotiation process, after applying the evaluation technique (cf., Section 2.2) to the following set of Cloud SecLAs:

- Three well-known CSPs (i.e., *CSP1* to *CSP3*) taken from the CSA STAR repository[1].
- Two different user requirements with the same relative weights per-SA control. The first one ($User_{min}$) using the minimum allowable security levels (i.e., LSL = 1), whereas the second one ($User_{max}$) using the maximum (i.e., LSL = 4). Notice that three SA-controls (i.e., SA-08, SA-10, SA-11) are not shown in the results, because they did not apply to the evaluated CSPs.

Obtained results in Tables 1 and 2, show that at the *SA Aggregated* level the contributed negotiation methodology allows ranking the available CSPs as $\{CSP_2, CSP_3, CSP_1\}$ for $User_{min}$, but as $\{CSP_1, CSP_3, CSP_2\}$ for $User_{max}$. As a rule of thumb, the more appropriate CSP (i.e., the one that best fulfills the user requirement) will be the one with the quantitative score closest to zero (i.e., the *User SecLA* baseline). Finally, the resulting set of ranked CSPs is returned to user, so she can either *(i) automatically* decide the one to use (i.e., the best ranked) or, *(ii) manually* apply additional criteria for the decision making process (e.g., CSP price). Future research will also focus on the evaluation of non-security related parameters, which might be taken into account to perform a more comprehensive negotiation process.

[1] Due to STAR's usage restrictions, it is not possible to disclose the real identity of the CSPs under evaluation.

Table 1: Quantitative evaluation of the $User_{min}$ SecLA requirement

	CSP_1	CSP_2	CSP_3
SA Aggregated	2.86	2.16	2.6
SA-01	1.50	3.00	2.50
SA-02	3.00	1.29	2.14
SA-03	3.00	3.00	3.00
SA-04	3.00	1.50	3.00
SA-05	3.00	3.00	3.00
SA-06	3.00	0	3.00
SA-07	3.00	3.00	0
SA-09	3.00	3.00	3.00
SA-12	3.00	3.00	3.00
SA-13	3.00	0	3.00
SA-14	3.00	3.00	3.00

Table 2: Quantitative evaluation of the $User_{max}$ SecLA requirement

	CSP_1	CSP_2	CSP_3
SA Aggregated	-0.14	-0.84	-0.4
SA-01	-1.50	0	-0.50
SA-02	0	-1.71	-0.86
SA-03	0	0	0
SA-04	0	-1.50	0
SA-05	0	0	0
SA-06	0	-3.00	0
SA-07	0	0	-3.00
SA-09	0	0	0
SA-12	0	0	0
SA-13	0	-3.00	0
SA-14	0	0	0

6 RELATED WORK

To the best of our knowledge, there are only two previous works related with the idea proposed in this paper for Cloud ecosystems. The first one was contributed by Max (Rak M., *et. al.*, 2011a), where authentication and authorization mechanisms are negotiated between users and CSPs via a SLA-based interface in the context of the EU FP7 mOSAIC project (mOSAIC, 2011). Our research improves over the ideas described in (Rak M., *et. al.*, 2011a), by contributing with a common semantic (the Cloud SecLA) and an evaluation technique to quantitatively match the user's security requirements with respect to a set of available CSP. In the second related work Hale (Hale M.L. and Gamble R., 2012) introduced SecAgreements, a framework for negotiating Cloud security risks via *(i)* a SLA-based matchmaking algorithm and, *(ii)* a set of extensions proposed for the WS-Agreement protocol (Andrieux K., *et.al.*, 2007). Despite the similarities with our research, on one hand SecAgreements' matchmaking algorithm is not user-centric and only can specify weights at the individual security provision-level, thus lacking of the usability offered by our evaluation approach (cf., Section 2.2). On the other hand, as future work we are planning to research if the risk-based approach proposed by SecAgreements (Hale M.L. and Gamble R., 2012) might be used to complement our own negotiation methodology.

7 CONCLUSIONS

In this paper we have introduced the foundations for negotiating and brokering Cloud resources based on the notion of Security Level Agreements. At the core of the negotiation stage is a user-centric technique for quantitatively evaluating and ranking SecLAs, being developed within the EU FP7 ABC4Trust project (ABC4Trust, 2011). Through the notion of Cloud SecLAs, our quantitative evaluation technique offers a common semantic to systematically match a *User SecLA* requirement with respect to the most appropriate CSP. Based on our experience within the EU FP7 mOSAIC project (mOSAIC, 2011), this paper also presented an architecture and protocol to implement the proposed Cloud negotiation mechanism. The feasibility of the proposed approach was demonstrated through a real-world case study that used the CSP information contained in the CSA STAR repository (Cloud Security Alliance, 2011b).

Despite STAR contains only static/declarative information about CSPs, our negotiation approach has the potential to use also "dynamic" security data (e.g, measured in real-time by network sensors), directly embedded into the WS-Agreement protocol. We have also shown that the contributed methodology is suitable for Cloud Federations, where the negotiation of security parameters is a critical factor taking into account the amount of available CSP.

Once the envisioned architecture is deployed using the mOSAIC framework (mOSAIC, 2011), future work will empirically analyze in detail the technical trade-offs (e.g., from the performance perspective) between SLA-based resource negotiation and, the SecLA-based mechanism presented in this paper. Finally, future activities will also research "advanced" negotiation features not considered so far e.g., re-negotiation and continuous monitoring.

Acknowledgment

Research supported in part by the Deutsche Forschungsgemeinschaft (German Research Foundation) Graduiertenkolleg 1362 - DFG GRK 1362, the EC FP7 project ABC4Trust (Grant Agreement no. 257782), the FP7-ICT-2009-5-256910 (mOSAIC) and Loewe TUD CASED.

REFERENCES

ABC4Trust (2011). ABC4Trust FP7. *Online: http://www.abc4trust.eu/.*

Almorsy M., *et.al.* (2011). Collaboration-Based Cloud Computing Security Management Framework. In *Proc. of IEEE Intl Conference on Cloud Computing*, pages 364–371.

Amato A., *et. al.* (2012). SLA Negotiation and Brokering for Sky Computing. In *Procs. of the Intl. Conference on Cloud Computing and Services Science*, pages 611–620. SciTePress.

Andrieux K., *et.al.* (2007). Web Services Agreement Specification (WS-Agreement). Technical Report TR-WSAgreement-2007, Open Grid Forum.

Bernsmed K., *et.al.* (2011). Security SLAs for Federated Cloud Services. In *Proc. of IEEE Availability, Reliability and Security*, pages 202–209.

Casola, V., Preziosi, R., Rak, M., and Troiano, L. (2005). A reference model for security level evaluation: Policy and fuzzy techniques. *J. UCS*, 11(1):150–174.

Casola V., *et.al.* (2006). A SLA evaluation methodology in Service Oriented Architectures. In *Quality of Protection*, volume 23 of *Springer Advances in Information Security*, pages 119 – 130.

Cloud Security Alliance (2011a). The Consensus Assessments Initiative Questionnaire. Online: https://cloudsecurityalliance.org/research/cai/.

Cloud Security Alliance (2011b). The Security, Trust & Assurance Registry (STAR). Online: https://cloudsecurityalliance.org/star/.

Cloud Security Alliance (2012). Security and Privacy Level Agreements working groups. Online: https://cloudsecurityalliance.org/research/pla/.

Dekker M. and Hogben G. (2011). Survey and analysis of security parameters in cloud SLAs across the European public sector. Technical Report TR-2011-12-19, European Network and Information Security Agency.

Hale M.L. and Gamble R. (2012). SecAgreement: Advancing Security Risk Calculations in Cloud Services. In *Proc. of the IEEE World Congress on Services*, pages 133 – 140.

Kandukuri B.R., *et. al.* (2009). Cloud Security Issues. In *Procs. of the IEEE Intl. Conference on Services Computing*, pages 517–520.

Luna J., *et.al.* (2011). A Security Metrics Framework for the Cloud. In *Proc. of Security and Cryptography*, pages 245–250.

Luna J., *et.al.* (2012a). Benchmarking Cloud Security Level Agreements Using Quantitative Policy Trees. In *Proc. of the ACM Cloud Computing Security Workshop*.

Luna J., *et.al.* (2012b). Quantitative Assessment of Cloud Security Level Agreements: A Case Study. In *Proc. of Security and Cryptography*.

mOSAIC (2011). mOSAIC FP7. *Online: http://www.mosaic-cloud.eu/.*

Rak M., *et. al.* (2011a). A SLA-based interface for security management in cloud and GRID integrations. In *Proc. of the IEEE Intl. Conf. on Information Assurance and Security*, pages 378 – 383.

Rak M., *et. al.* (2011b). User Centric Service Level Management in mOSAIC Application. In *Procs. of the Europar Workshop*. Springer.

Samani R., *et.al.* (2011). Common Assurance Maturity Model: Scoring Model. Online: http://common-assurance.com/.

Savola R., *et.al.* (2010). Towards Wider Cloud Service Applicability by Security, Privacy and Trust Measurements. In *Proc. of IEEE Application of Information and Communication Technologies*, pages 1–6.